PRIRUČNIK ZA BRZO RJEŠAVANJE PROBLEMA

# ADMINISTRIRANJE LINUX SUSTAVA

TOM ADELSTEIN i BILL LUBANOVIC

IT Expert
IT profesionalci za IT profesionalce
O'REILLY®

# Administriranje Linux sustava

# Administriranje Linux sustava

*Tom Adelstein i Bill Lubanovic*

*Prijevod:*
*Snježana Šlabek*
*Ivan Dražić*
*Bojana Bošnjak*

IT Expert
O'REILLY®

**Administriranje Linux sustava**
Tom Adelstein i Bill Lubanovic


| | |
|---|---|
| Nakladnik: | Dobar Plan, Zagreb |
| Za nakladnika: | Tomislav Kotnik |
| Urednik: | Aleksandar Dragosavljević |



**Copyright © 2007 Dobar Plan, Zagreb.** Autorizirani prijevod engleskog izdanja knjige *Linux System Administration* © O'Reilly Media, Inc. Ovaj prijevod je objavljen i prodaje se s dozvolom O'Reilly Media, Inc. koja je vlasnik svih prava za objavljivanje i prodaju.

Copyright Dobar Plan Publishing Company 2007. Authorized translation of the English edition of *Linux System Administration* © 2007 O'Reilly Media, Inc. This translation is published and sold by permission of O'Reilly Media, Inc., the owner of all rights to publish and sell the same.

Niti jedan dio ove knjige ne smije se reproducirati ili prenositi u bilo kojem obliku, ni na jedan način, elektronički ili mehanički, uključujući fotokopiranje, snimanje i ostale načine reproduciranja. Bez pismene dozvole nositelja autorskih prava zabranjeno je koristiti ovu knjigu za organizirano školovanje u javnim i privatnim obrazovnim organizacijama.

Iako je tijekom prijevoda i pripreme ove knjige za tisak uložen veliki trud kako bi se izbjegle pogreške, autor i izdavač ne preuzimaju odgovornost za pogreške ili propuste niti za štetu koja bi mogla nastati upotrebom informacija iz ove knjige.



CIP zapis dostupan u računalnom katalogu Nacionalne i sveučilišne knjižnice u Zagrebu pod brojem 642525.

ISBN: 978-953-7398-12-5

# Sadržaj

**Uvod** . . . . . **ix**

**1. Uvjeti koje mora ispunjavati administrator Linux sustava** . . . . . **1**

O ovoj knjizi 2
Kako vam mi možemo pomoći? 2
Gdje započeti? 3
Je li vam ova knjiga potrebna? 3
Kome ste vi potrebni? 4
Što administratori sustava moraju znati o Linuxu? 7
Što slijedi 7

**2. Postavljanje višenamjenskog Linux poslužitelja** . . . . . **8**

Uvjeti za postavljanje poslužitelja 9
Instalacija Debiana 10
Daljinsko prijavljivanje na sistem 12
Konfiguriranje mreže 13
Mijenjanje podrazumijevanih Debianovih paketa 15
Postavljanje kvota 16
Pružanje usluga imena domena 18
Dodavanje relacijske baze podataka: MySQL 20
Konfiguriranje sustava elektroničke pošte s Postfixom, POP3 i IMAP poslužiteljem 22
Pokretanje Apache poslužitelja 33
Dodavanje FTP servisa pomoću ProFTPD-a 34
Rezimiranje statistike poslužitelja pomoću Webalizera 35
Sinkroniziranje sistemskog sata 36
Instaliranje Perl modula potrebnih za SpamAssassin 36
Što slijedi 37

**3. Domain Name System (DNS) . . . . . 38**
Osnove DNS poslužitelja 38
BIND 40
Postavljanje DNS poslužitelja 41
Konfiguriranje pouzdanog DNS poslužitelja 44
BIND alati 62
Rješavanje problema s BIND-om 66
Što slijedi 71

**4. Početna okolina spremna za Internet . . . . . 73**
Instaliranje ISPConfiga 74
Postavljanje poslužitelja i korisnika s ISPConfigom 83
Zaštita Linux Web poslužitelja 96
Što slijedi 101

**5. Elektronička pošta . . . . . 102**
Ključni pojmovi vezani uz servis elektroničke pošte 103
Postfix, Sendmail i drugi agenti za prijenos pošte 103
Postfix SMTP poslužitelj elektroničke pošte na Debianu 105
Dodavanje provjere identiteta i šifriranja 111
Konfiguriranje POP3 i IMAP agenata za dostavu pošte 119
Konfiguriranje klijenta za elektroničku poštu 120
Što slijedi 121

**6. Administriranje Apachea . . . . . 122**
Statičke i dinamičke datoteke 122
Jednostavna LAMP konfiguracija 123
Instaliranje 124
Apacheove konfiguracijske datoteke 127
Datoteke dnevnika 140
SSL/TLS šifriranje 142
suEXEC podrška 143
Testiranje performansi 144
Instaliranje i administracija Drupala 145
Rješavanje problema 149
Dodatna literatura 153

**7. Grozdovi s raspoređivanjem opterećenja . . . . . 154**
Raspoređivanje opterećenja i visoka razina dostupnosti 154
Prilagođavanje potrebama bez servisa za raspoređivanje opterećenja i postizanje visoke razine dostupnosti 162
Dodatna literatura 162

**8. Servisi lokalne mreže. . . . . 163**
Distribuirani sustavi datoteka 164
Uvod u Sambu 164
Konfiguriranje mreže 165
DHCP 168
Servisi mrežnog prolaza 173
Servisi za ispis 181
Upravljanje korisnicima 186

**9. Virtualizacija u modernom poduzeću . . . . . 194**
Zašto je virtualizacija popularna 194
Sustavi visokih performansi 196
Instaliranje Xena na Fedoru 5 199
Instaliranje Wmwarea 204
Virtualizacija: samo prolazni hir? 210

**10. Skripte . . . . . 211**
bash počeci 212
Korisni elementi za bash skripte 218
Završna bitka skriptnih jezika 226
Dodatna literatura 235

**11. Izrada rezervnih kopija podataka . . . . . 236**
Kopiranje korisničkih podataka na poslužitelj pomoću alata rsync 237
tar arhive 242
Spremanje datoteka na optičke medije 245
Izrada rezervnih kopija i arhiviranje na vrpcu s Amandom 251
Izrada rezervnih kopija podataka iz MySQL baze 254

**Dodatak. Primjeri bash skripti . . . . . 257**

**Kazalo . . . . . 273**

# Uvod

U vrijeme kad smo Bill Lubanovic i ja dovršavali ovu knjigu, slučajno sam čuo razgovor dvojice suradnika u našem Cisco laboratoriju kako raspravljaju o Linuxu. Stariji od njih, stručnjak za mrežu, iznio je zanimljivo opažanje. Rekao je da se unatoč svem svojem znanju osjeća profesionalno nepotpuno jer nikada nije naučio Linux. Trenutak kasnije okrenuli su se prema meni i pogledali me u oči. Nasmiješio sam se i nastavio raditi.

Te isti večeri, naš direktor informacijske tehnologije rekao mi je za vrijeme konferencije da želi naučiti Apache, što me prilično iznenadilo, i kada sam ga upitao zašto, samo je odgovorio: „Jednostavno želim naučiti“ i to je bilo sve.

Kasnije na konferenciji direktor je tražio povratne informacije od grupe u vezi s rješenjem za upravljanje zakrpama, objašnjavajući i koristeći primjer programa *rsync*. Rekao je da želi nešto slično i polako je ulazio u detaljnu tehničku raspravu o cjelokupnosti upravljanja zakrpama. Iz radnog iskustva poznajem *rsync* ali nikada prije nisam čuo ni na kojem forumu tako detaljno akademsko objašnjenje bilo kojeg alata otvorenog izvornog koda.

U oba ova slučaja, a i u mnogim drugim, poželio sam da je ova knjigu spremna i da je mogu predati obučenim i vještim ljudima koji žele naučiti administraciju Linuxa. Možda ste i vi imali slična iskustva i poželjeli da imate pri ruci knjigu poput ove. Usuđujem se pretpostaviti da se razgovori poput ovih koje sam opisao odvijaju svakodnevno na mnogim mjestima.

Kada smo Andy Oram i ja počeli raspravljati o knjizi posvećenoj administriranju Linux sustava, imali smo pomalo različite zamisli o tome što želimo postići. Andy je govorio o knjizi u kojoj bi svako poglavlje korisnika provodilo kroz korake za izgradnju i instaliranje aplikacijskih poslužitelja bez uključivanja detaljnih objašnjenja. Predložio je da se objašnjenja pruže na jednom mjestu u poglavlju a tehnički koraci na drugom.

Kasnije sam predložio da svako poglavlje oblikujemo kao zaseban modul i dopustimo čitatelju da prouči module koje želi ili su mu potrebni. Kako se knjiga razvijala, osjećali smo da smo postigli taj cilj. Ovu knjigu ne morate čitati od korica do korica da biste postali administrator Linux sustava. Jednostavno započnite čitati poglavlje koje vas najviše zanima.

Kad sam započeo koristiti Linux, zajednica se sastojala uglavnom od programera i hobista. Ne sjećam se ni jedne diskusije o samostalnim ili komercijalnim aplikacijama. Prijavljivali smo se na Internet pokretanjem pozadinskog servisa. Nismo imali ni preglednike Weba poput današnjih. Većina ljudi koje sam poznavao sami su administrirali svoje sustave ili su to tek učili.

Sjećajući se vremena kada smo procjenjivali da na svijetu ima oko 30 tisuća korisnika Linuxa, zapanjen sam koliko korisnika ima danas a uopće ne znaju kako napisati konfiguracijsku datoteku. Čini se da su Linux forumi puni korisnika koji pitaju kako implementirati CUPS ili Sambu. Ljudi na listama slanja održavaju detaljne rasprave o tehničkim pojedinostima projekata kao što su Postfix, JBoss i Monit.

Mnogi korisnici još uvijek žele više naučiti o širokim mogućnostima Linuxa kao aplikacijske platforme. Ako koristite Linux i želite napredovati od naprednog korisnika do administratora, ova će vam knjiga pomoći pri tom prijelazu. Napisali smo ovu knjigu misleći na vas.

## Kako je knjiga organizirana

Poglavlje 1, *Uvjeti koje mora ispunjavati administrator Linux sustava*
: Obrazlaže ciljeve knjige i što ćete postići čitajući je.

Poglavlje 2, *Postavljanje višenamjenskog Linux poslužitelja*
: Pomaže vam započeti raditi s poslužiteljem gotovo spremnim za Internet.

Poglavlje 3, *Domain Name System (DNS)*
: Opisuje osnove postavljanja primarnog i sekundarnog DNS poslužitelja.

Poglavlje 4, *Početna okolina spremna za Internet*
: Koristi besplatnu aplikaciju ISPConfig za konfiguriranje sustava kako bi vam pomoglo započeti rad s mnoštvom servisa čije korištenje možete uvježbavati dok čitate knjigu.

Poglavlje 5, *Elektronička pošta*
: Opisuje postavljanje poslužitelja elektroničke pošte s Postfixom i SASL provjerom identiteta te POP i IMAP protokolima.

Poglavlje 6, *Administriranje Apachea*
: Pruža brz pregled popularne kombinacije Apachea, MySQL-a i PHP-a (koja je, zajedno s Linuxom, poznata kao LAMP poslužitelj), uključujući SSL provjeru identiteta.

Poglavlje 7, *Grozdovi s raspoređivanjem opterećenja*
: Proširuje konfiguraciju Apachea iz prethodnog poglavlja s IP virtualnim poslužiteljem i pomoćnim programom *ldirectord* kako bi se povećala dostupnost poslužitelja.

Poglavlje 8, *Servisi lokalne mreže*
: Opisuje kako raditi s korisnicima i konfigurirati uobičajene mrežne elemente kao što su DHCP i mrežni prolaz na lokalnim mrežama.

Poglavlje 9, *Virtualizacija u modernom poduzeću*
: Opisuje kako postaviti Xen i VMware na Linux domaćina te dodavati gostujuće operativne sustave.

Poglavlje 10, *Skripte*
: Opisuje neke osnovne tehnike za pisanje robusnih i moćnih *bash* skripti za školjku koje vam mogu uštedjeti mnogo vremena pri administriranju.

Poglavlje 11, *Izrada rezervnih kopija podataka*
: Opisuje mnoštvo tehnika za obavljanje ovog važnog posla, od osnovnih programa *rysnc* i *tar* do moćnog Amanda sustava.

Dodatak, *Primjeri bash skripti*
: Sadrži nekoliko skripti koje su nam bile korisne pri administriranju sustava a mogle bi vam služiti kao predložak kad pišete vlastite skripte.

## Pravila označavanja korištena u ovoj knjizi

U knjizi se koriste sljedeća pravila označavanja:

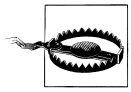
*Kurziv*
: Označava nove termine, URL adrese, naredbe i naredbene opcije, adrese elektroničke pošte, imena datoteka, nastavke imena datoteka i direktorije.

`Pismo konstantne širine`
: Označava sadržaj datoteka ili ispis rezultata izvedenih naredbi.

**`Podebljano pismo konstantne širine`**
: Ukazuje na naredbe ili drugi tekst koji bi korisnik trebao doslovce upisati. Koristi se i za isticanje ključnih dijelova koda ili datoteka.

*`Pismo konstantne širine u kurzivu`*
: Označava tekst koji korisnik treba zamijeniti svojim vrijednostima.

Ova sličica označava savjet, prijedlog ili napomenu.



Ova sličica označava upozorenje ili poziv na oprez.

## Upotreba primjera koda

Svrha knjige je da vam pomogne obaviti posao. Općenito, kôd iz knjige možete koristiti u svojim programima i dokumentaciji. Ne trebate nas kontaktirati za dopuštenje osim ako reproducirate značajan dio koda. Na primjer, ako napišete program koji koristi nekoliko odlomaka koda iz knjige, dopuštenje nije potrebno. Za prodaju ili distribuciju CD-ROM-a s primjerima iz knjiga koje objavljuje O'Reilly, dopuštenje je potrebno. Ako želite odgovoriti na pitanje citiranjem knjige i navođenjem primjera koda, dopuštenje nije potrebno. Uključivanje veće količine primjera koda iz knjige u dokumentaciju vašeg proizvoda zahtijeva dopuštenje.

Cijenimo, ali ne zahtijevamo, da nas navodite uz referencu. Navođenje reference obično uključuje naslov, ime autora, izdavača i ISBN. Na primjer: „Administriranje Linux sustava, Tom Adelstein i Bill Lubanovic. Copyright 2007 O'Reilly Media, Inc., 978-0-596-00952-6."

Ako smatrate da upotreba primjera koda izlazi van okvira poštene primjene ili danog dopuštenja, slobodno nam pišite na adresu info@knjizara.hr.

## Zahvale

Knjige kao što je *Administriranje Linux sustava* mogu nastati samo zajedničkim trudom mnogih ljudi. Na žalost, nemoguće ih je sve ovdje nabrojati.

Prvo, zahvaljujemo Andyu Oramu, čiji su napori pri uređivanju, pisanju i organiziranju ove knjigu bili zadivljujući. Osim što je radio kao urednik cijele knjige, Andy je sadržaju i materijalno pridonio. Bio je rukovoditelj projekta i pri tom iskazao strpljenje i disciplinu.

Knjizi su značajno doprinijeli Falko Timme, Phil Howard i Herschel Cohen. Falko je uložio vrijeme i stručno znanje u poglavljima 2 i 4. Phil je napisao veći dio poglavlja 11 i oblikovao okvir za poglavlje 10 te popratni dodatak sa skriptama. Herschel je napisao odlomke u nekoliko poglavlja, uključujući 8. i 10., te je svoje stručno znanje uložio u izradu poglavlja 6. Sva trojica su pregledala i ostale dijelove knjige.

Zahvalnost dugujemo i našim tehničkim stručnjacima Markusu Amersdorferu, Keithu Burgessu, Robertu Dayu, Ammaru Ibrahimu i Yamanu Saqqi, koji su neizmjerno mnogo sati proveli pregledavajući knjigu, testirajući kod i dajući nam prijedloge.

Posebne zahvale upućujemo Yvonne Adelstein i Mary Lubanovic, našim suprugama, koje su pokazale neizmjerno strpljenje. Ovu knjigu ne bismo mogli napisati bez vaše pune podrške.

POGLAVLJE 1

# Uvjeti koje mora ispunjavati administrator Linux sustava

Mi volimo Linux. Od svih Unix i njemu sličnih operativnih sustava koje smo koristili, a neki su već i zaboravljeni, Linux je naš omiljeni. Radi se o izvrsnoj poslužiteljskoj platformi, dobrom operativnom sustavu za osobna računala te središtu mnogih inovacija u današnjem računalnom svijetu.

Od svih operativnih sustava, Linux vjerojatno ima najširi opseg primjene, od sićušnih sustava veličine telefonskih utičnica, preko mobilnih telefona, do skupina superračunala veličine stambene zgrade. Proširio se na telekomunikacije, ugrađene sustave, satelite, medicinsku opremu, vojne sustave, računalnu grafiku i na kraju, što nije i najmanje važno – na kućna računala.

Linux je u razmjerno kratkom vremenu napredovao od hobija finskog hakera do vrhunskog višerazinskog sustava koji podržavaju velike tvrtke poput IBM-a i Oraclea. Broj korisnika porastao je s oko 30 tisuća u 1995. godini na današnje stotine milijuna. Za vrijeme procvata Interneta u 1990-ima, mnoge je administratore Unix sustava iznenadilo što Linux na računalima iskazuje bolje performanse od skupocjenih Unix radnih stanica i poslužitelja. Mnogi administratori Windows i Novell sustava uvidjeli su da Linux može obrađivati DNS, elekroničku poštu i sustave datoteka pouzdanije i uz manje stručnog osoblja nego njihove tadašnje platforme. Rast Interneta, a posebice Weba, ubrzao je ulazak Linux poslužitelja u sve širu upotrebu i izazvao potrebu za osobljem koje će njima upravljati. Rast Interneta i posebice Weba ubrzao je ulazak Linux poslužitelja u sve širu upotrebu i izazvao potrebu za osobama koje će njima upravljati.

Ova je knjiga namijenjena administratorima Linux sustava. Međutim, ako ste veteran Unixa, sistemski inženjer s Microsoftovim certifikatom ili administrator središnjih računala, istraživat ćete novo područje pa su vam potrebne neke smjernice. Neka će vam područja biti poznata, a neka ne. Knjiga opisuje mnoge teme koje su tek nedavno postale uobičajene, primjerice grozdove s raspoređivanjem opterećenja i virtualizaciju.

Uspjeh Interneta i softvera otvorenog izvornog koda mijenjaju današnji način poslovanja. Google, Amazon, eBay i drugi izgradili su ogromne poslužiteljske farme s prosječnim hardverom i relativno malo administratora u usporedbi s tradicionalnim središnjim računalima i računalnim instalacijama.

Znanje potrebno za razvoj i održavanje takvih distribuiranih sustava i aplikacija ne dobiva se u školama već se stiče iskustvom, ponekad učeći na vlastitim pogreškama.

Za vrijeme pisanja ove knjige neprestano smo testirali najnovije distribucije i alate a isto ćemo nastaviti i nakon njenog objavljivanja. Pozivamo čitatelje da posjete stranicu za testiranje koju smo postavili za potrebe ove knjige, *http://www.centralsoft.org*, na kojoj ćemo objavljivati dopune primjera iz knjige, ukazivati na korisne alate koje smo otkrili i davati druge savjete.

## O ovoj knjizi

Knjige o administriranju sustava obično su bile vrlo predvidljive. Opisivale su kako raditi s korisnicima, sustavima datoteka, uređajima, procesima, pisačima, mrežama i tako dalje. Nisu savjetovale što biste trebali učiniti kad se pojave novi problemi. Ako je vaša Web lokacija postala popularna, brzo ste morali naučiti sve o posredničkim poslužiteljima, različitim razinama smještanja u privremenu memoriju, raspoređivanju opterećenja, distribuiranoj provjeri identiteta i drugim zamršenim zadaćama. Ako ste dodali bazu podataka, ubrzo ste je morali prilagoditi i naučiti kako izbjeći napade ubacivanjem SQL koda. Funkcioniranje lokacije iznenada bi postalo stvar od najveće važnosti, pa vam je bila potrebna mogućnost za izrađivanje rezervnih kopija na sustavima koji neprekidno rade.

Ako ste imali takvih neugodnih iskustava, možda ste se umorili od učenja na teži način, susrećući se skoro svakodnevno s novim izazovima uz malo izvora pomoći. Tehnička dokumentacija, bilo za komercijalni ili softver otvorenog izvornog koda, rijetko ide ukorak s tehnologijom i čini se da raskorak postaje sve veći. Na primjer, poslužitelji imena postaju vrlo važni za upravljanje računalima, korisnicima i resursima. Izvorni standardizirani protokoli temelj su mnogim popularnim proizvodima ali dobra dokumentacija za projekte zajednice iznenađujuće je rijetka.

## Kako vam mi možemo pomoći?

Korisnici Linuxa naviknuti su na rješavanje problema. Tipični napredni Linux korisnik može postaviti mali poslužitelj, uvesti namjensku Internet vezu sa statičkom IP adresom u svoj dom, registrirati ime domene i konfigurirati poslužitelj na Internetu. Ako pripadate ovoj kategoriji, možete jednostavno proučiti druge teme u knjizi i proširiti svoje poslovne mogućnosti.

Ali, neki će smatrati da knjiga pruža previše informacija koje bi trebalo odjedanput usvojiti. Ako ste među njima, jednostavno započnite s nekim od poglavlja i napredujte korak po korak. Kao što stara poslovica kaže, Rim nije izgrađen u jednom danu.

Možda imate certifikate za neke druge operativne sustave, a ne za Linux. Dok primijenjujete zakrpe i ispravke, šef bi od vas mogao zatražiti da postavite Apache poslužitelj, izvedete vlastite DNS potrage ili Exchange zamijenite Zimbrom.

Bilo da samo želite naučiti ili ste u situaciji da *morate* naučiti, vjerojatno će vam trebati pomoć da se razvijete u naprednog Linux korisnika. Upravo zato smo mi ovdje: da bismo vam pomogli istražiti područje Linux sustava bez poteškoća koje su iskusili naši prethodnici.

## Gdje započeti?

Ova knjiga opisuje korake koje trebate slijediti kako biste mogli postavljati samostalne funkcionalne poslužitelje. Ako trebate konfigurirati poslužitelj elektroničke pošte, postaviti Web poslužitelj i sustav za vođenje mrežnog dnevnika ili postaviti mrežni prolaz za svoju lokalnu mrežu, možete odmah prijeći na središnji dio knjige. Knjigu *Administriranje Linux sustava* ne morate čitati od korica do korica.

Odmah započinjemo s radom i već vam u poglavlju 2 pružamo vodič koji će vam korak po korak pokazati kako postaviti Linux poslužitelj. Možete krenuti smjerom koji vam odgovara, bez obzira je li to izrada sofisticiranog grozda za Web servise, konsolidacija poslužitelja kroz virtualizaciju upotrebom Xena ili VMwarea ili postavljanje poslužitelja za lokalne mreže.

Upotreba modernog operativnog sustava nevjerojatno je jeftina. Možete postaviti sofisticirani sustav za eksperimentiranje na zastarjelom hardveru koji bi mnoge tvrtke otpisale. Mi smo započeli s rabljenim računalom s Intelovim procesorom dvije generacije starijim od aktualnih modela, dodali smo dva starija tvrda diska, memoriju i instalirali jednostavnu besplatnu inačicu Linuxa.

## Je li vam ova knjiga potrebna?

Tehničke knjige su donekle izgubile popularnost otkad se Internet sadržajno obogatio. Da bi danas napisao uspješnu knjigu, autor čitatelju mora pružiti uistinu vrijedne informacije. Zanimljiva priča o jednoj od prvih Web lokacija za elektroničko poslovanje mogla bi nam pomoći da objasnimo kakvu bi korist knjiga trebala pružiti. U najranijim danima Weba jedna je slastičarska tvrtka objavila svoj oglas. Prema priči, prošlo je nekoliko mjeseci a tvrtka nije zaprimila ni jednu narudžbu. Direktor tvrtke tada je povukao neobičan potez i objavio njihov tajni recept za kolač od sira. Već za nekoliko sati počeo je primati telefonske pozive i kupci su počeli naveliko naručivati taj kolač. Kad su vidjeli recept i shvatili koliko bi truda morali uložiti u izradu takvog kolača, prepoznali su prednost njegove kupnje od proizvođača.

Mnoge informacije u ovoj knjizi bile su raspršene diljem Interneta, na listama slanja, forumima i po diskusijskim skupinama a druge su pronađene u knjigama, časopisima i prikupljene iz iskustava kolega. Tijekom pripreme knjige riješili smo mnoge probleme za koje rješenja nisu uopće bila dokumentirana pa vam prenosimo ono što smo naučili.

Mnoge izvrsne Web lokacije projekata nemaju odgovarajuću dokumentaciju. Programeri se trude izraditi izvrstan besplatni softver ali iz mnogih razloga kôd nije popraćen dokumentacijom: zbog nedostatka vremena, resursa i zanimanja, jezičnih zapreka i tako dalje.

Zajedno s našim čitateljima, urednicima i kritičarima, nadamo se da smo u jednom dijelu računalnog svijeta bar malo smanjili entropiju.

# Kome ste vi potrebni?

Prije nekoliko godina, većina administratora Linux sustava rekla bi da nisu odabrali svoje zvanje već je Linux odabrao njih. U prijašnjim je vremenima Linux bio poput mladog Unixa. Većina administratora Linux sustava stekla je osnovno znanje na radnoj stanici i vrlo malim mrežama. Linux je od Unixa naslijedio neke poslužitelje (BIND, Sendmail, Apache), no malo uredskog softvera i samo nekoliko aplikacija. Danas administracija Linux sustava uključuje tisuće paketa i interoperabilnost s drugim operativnim sustavima.

Kome su potrebni Linux administratori? Na primjer, NASA-inom Centru za računalne znanosti (NCCS) u Goddard Space Flight Centeru. Njihovi računalni grozdovi visokih performansi temeljeni na Linuxu namijenjeni su značajnom povećanju propusne moći koju će koristiti različite aplikacije, od onih za proučavanje meteoroloških prilika i klimatskih promjena do simulacija astrofizičkih fenomena. Linux nadopunjuje NCCS arhitekturu projektiranu za obradu do 40 trilijuna operacija sa pomičnim zarezom u sekundi (TFLOPS).

S Linuxom radi više svjetskih vrhunskih superračunala nego s bilo kojim drugim operativnim sustavom. Ustvari, trenutno se Linux izvodi na nevjerojatnih 75 % od 500 vrhunskih superračunala na svijetu.* Prema informacijama rukovoditelja odjela kalifornijskog instituta Lawrence Livermore National Laboratory, Linux pokreće 10 njihovih velikih sustava od kojih se svi nalaze na popisu 500 najboljih. Ti sustavi uključuju BlueGene/L, najmoćije superračunalo na svijetu, i Thunder koje je trenutno na devetnaestom mjestu (*http://www.top500.opg/list/2006/11/100*).

## Što se od vas očekuje?

Linux administratori vrlo su traženi. Kako bismo vam pružili uvid u sve što se od njih očekuje, pregledali smo neke od desetaka tisuća oglasa na stranicama američke agencije za zapošljavanje u kojima se traže administratori Linux sustava. Evo kratkog prikaza odgovornosti vezanih uz taj posao:

- Administriranje i upravljanje velikim Linux okruženjem s naglaskom na nadzor izvođenja, podešavanje i upravljanje.
- Nadgledanje projektiranja baza podataka, njihovo administriranje i dokumentiranje.

* Pogledajte *http://www.top500.org/stats/28/osfam*.

- Rješavanje problema s mrežom, podrška korisnicima servisa i proaktivno nadziranje važnih sustava.
- Odabiranje i preporučavanje tehničkih rješenja za organizaciju, obučavanje i usmjeravanje mlađih administratora.
- Pružanje svakodnevne tehničke podrške i savjeta za hardver i okruženje operativnog sustava koji podržava kolektivnu platformu, administriranje infrastrukture Linux poslužitelja da bi se održala stabilnost te maksimiziranje učinkovitosti računalnog okruženja.
- Instaliranje, konfiguriranje i rješavanje problema s hardverom, perifernim uređajima i opremom potrebnom za postizanje ciljeva integriranih sustava, pružanje podrške za problematične komponente.
- Pružanje djelotvorne podrške prve i druge razine za Linux okruženje u tvrtki, a koje se sastoji od 300 i više poslužitelja, uključujući i Linux blade.
- Upravljanje svim aspektima integriranja okoline, uključujući sigurnost, nadziranje (kapaciteta i izvedbe), promjenu kontrole i upravljanje softverom.
- Suradnja s drugim unutarnjim grupama za podršku kao što su grupe za razvoj aplikacija, inženjering, administratori baza podataka, Web usluge, pohrana, operacije i naredbeni centri.
- Administriranje infrastrukturalnih servisa, kao što su DNS, NIS, LDAP, FTP, SMPT, Postfix/Sendmail, NFS i Samba, te aplikacijskih poslužitelja i poslužitelja baza podataka, s naglaskom na automatizaciju i nadziranje.

Linux je danas standardna kompanijska platforma i osobe koje znaju s njim raditi vrlo su tražene. Ako želite naučiti Linux radi poboljšanja svoje financijske situacije, postoji mnogo dokaza o porastu potražnje u tvrtkama za osobama s poznavanjem administracije Linuxa.

## Analiza poslovnih odgovornosti

Ako upitate upravitelje različitih informatičkih sustava koja je uloga administratora sustava, dobit ćete različite odgovore. Tromost tržišta iznenadila je današnju generaciju upravitelja koji nisu upućeni u Linux. Oni ne znaju što bi stručnjaci za Linux trebali znati, a stručnjaci za Linux rijetko razumiju te upravitelje.

Mnogi upravitelji sustava koji razumiju Unix pokušavaju ograničiti administratore Linuxa na standarde za Unix. To rijetko kada funkcionira. Iako administratori Unixa smatraju da mogu lako prijeći na Linux, ubrzo otkrivaju razlike u potrebnom znanju. Administratori Linuxa lakše će prijeći na Unix nego obrnuto. Jedno od objašnjenja je to što administratori Linuxa imaju puno bolje razumijevanje svojih sustava zbog same prirode softvera otvorenog izvornog koda.

Poslovi vezani uz administriranje sustava često uključuju Internet. Velik broj transakcija vezan je uz elektroničku poštu i rad s Web stranicama, kao i telekomunikacije i mobilnost. Nekoć je elektronička pošta činila 70% cjelokupnog prometa na Internetu. Danas, širokopojasne aplikacije poput Voice over IP (VoIP) aplikacija i drugi oblici komuniciranja, uključujući slanje instant poruka, povećavaju promet na račun poruka elektroničke pošte. No, bez obzira kakvi se protokoli i mediji koriste, Internet i dalje ostaje glavno područje na koje je Linux usmjeren.

Nastavimo analizirati poslovne odgovornosti administratora spomenute u prethodnom odjeljku. Zadnji spomenuti skup obaveza (administriranje infrastrukturalnih servisa) može vam pružiti uvid u znanje kojim mora ovladati administrator Linux sustava. Poslodavci žele administratore koji znaju raditi s infrastrukturalnim servisima koji uključuju i internetske tehnologije. Od komponenti Linuxa koje je potrebno poznavati, većina poslova bit će vezana uz DNS, LDAP, FTP, SMTP i Postfix/Sendmail. Većinu tih komponenti opisat ćemo u poglavljima od 2 do 6.

Ostale poslovne obaveze većinom pripadaju kategoriji potreba poslovanja unutar tvrtke. To uključuje podršku korisnicima servisa, tehničku podršku i pružanje savjeta putem telefona za hardver i okruženja operativnog sustava. Većina administratora Linux sustava trebala bi znati kako pružiti te usluge ali upute za stjecanje tog znanja izlaze iz okvira ove knjige jer nisu sasvim tehničke prirode.

Ostale odgovornosti pripadaju kategoriji fleksibilnih odgovornosti. U prošlosti se od prosječnog administratora sustava nije očekivalo da svoj rad usklađuje s drugim unutarnjim grupama kao što su odjel za razvoj aplikacija, inženjering, administracija baza podataka ili Web servisi. Međutim, danas administrator sustava nije samo tehničar koji poznaje tajne sustava već je mjerodavan član osoblja tvrtke.

Obično veće odgovornosti i specijalizacije slijede nakon ovladavanja osnovama. Te ćemo teme samo površno spomenuti u ovoj knjizi jer smatramo da se one ne uklapaju sasvim u njen sadržaj. Druge knjige izdavača O´Reilly te radno iskustvo pomoći će vam da ovladate tim vrijednim sposobnostima. Za sada ćemo vam pomoći u područjima u kojima je administriranje sustava doživjelo najveći napredak i gdje nedostaje dokumentacija.

Za razliku od ostalih područja računalne znanosti i tehnologije, malo škola nudi tečajeve administriranja Linuxa, a još manje njih nudi stjecanje službeno priznate kvalifikacije za taj posao. Ako želite naučiti administrirati Linux sustav, materijale i tečajeve morat ćete potražiti izvan školskih ustanova. Ali većina postojećeg materijala koji možete pronaći neće sadržavati ono što stručnjaci za Linux smatraju najpresudnijim temama.

Većina je administratora Linuxa samouka a učili su kad se za to pojavila potreba. Tad je nastupio trenutak kad su se ti samouki administratori zaposlili. Potrebe su se počele pojavljivati sve većom brzinom, prisiljavajući ih da uče sve više, sve dok nisu naučili sve što administrator sustava mora znati raditi. To je jedno od područja u kojem vam ova knjiga može pomoći da brže i uspješnije steknete stručnost u velikom broju poslova.

## Što administratori sustava moraju znati o Linuxu?

Jedna od najvažnijih stvari koju bi stručnjak za informacijsku tehnologiju morao znati jeste da Linux nije Unix. Pored toga što Linux može izvoditi većinu Unix programa, on ima i opsežniju primjenu u javnim i privatnim mrežama. Linux administratori mogu konfigurirati distribucije odabiranjem između velikog broja komponenti koje obavljaju slične poslove. Na primjer, u skoro svim Unix distribucijama Sendmail je jedini agent za prijenos pošte. Ali kod Linuxa možete odabirati između mnoštva sličnih agenata za prijenos pošte, ovisno o tome želite li aplikaciju za radnu grupu tvrtke, veliki sustav elektroničke pošte s podrškom za imenik ili jednostavnu Web aplikaciju za obrađivanje obrazaca za kontakt.

Daljnja potvrda Linuxove fleksibilnosti je to što je Linux prvi operativni sustav koji je IBM primijenio na svim svojim hardverskim platformama, od xSeries Intel klase poslužitelja, preko pSeries and iSeries do S/390 and zSeries središnjih računala.

Ako želite Linux administratora i koristite velike IBM-ove sustave, vaš će kandidat morati poznavati arhitekturu središnjeg računala i biti upoznat s pojmovima kao što su „DASD“ za pohranu na tvrdom disku, „IPL“ za pokretanje sustava, „katalog“ za imenik i „popis naredbi“ za skriptu za školjku. No, nemojte podcjenjivati administratore Linuxa. Jednom smo pohađali dvodnevni seminar s grupom Linux administratora koji su drugi dan nakon predavanja započeli instalirati Linux na IBM zSeries računalu.

Ako se poznavaoci Linuxa mogu ičim pohvaliti, to je da brzo uče, brzo se prilagođavaju i imaju široko znanje kakvo nećete pronaći kod drugih tehničara. Mogu naučiti koristiti Microsoft kutije za manje vremena nego što je potrebno Microsoft sistemskom inženjeru da nauči obaviti najjednostavniji zadatak na Linuxu.

## Što slijedi

Znamo da ne volite spor način učenja i zamršene uvode (zapravo smo iznenađeni što ste ovo poglavlje čitali do ovog mjesta) zato želimo započeti što je prije moguće. Želimo postaviti funkcionalan poslužitelj koji će izvoditi mnoge Linuxove zadaće koje možete naučiti i koristiti. Zato ćemo u sljedećem poglavlju započeti s poslužiteljem spremnim za Internet. Bit će vam potrebni internetski alati poput Web poslužitelja i sustava elektroničke pošte bez obzira na to kako koristite svoj poslužitelj (čak i ako samo poslužuje lokalnu mrežu) a ti će vam alati biti korisni već od samog početka.

Ostatak knjige opširnije opisuje iste teme te uvodi druge koje možda nećete svakodnevno susretati. *Administriranje Linux sustava* je kombinacija priručnika i vodiča za poučavanje pa možete učiti i uz doručak. Glavne teme obično objašnjavamo na samom početku poglavlja te nastavljamo sa sažetim koracima i primjenama tih tema. Ako samo želite slijediti upute korak po korak, učinite tako. Kasnije možete bolje proučiti ono što ste izveli. Vjerujemo da će vas naš pristup odvesti u pravom smjeru.

Prema naprijed i prema vrhu.

POGLAVLJE 2

# Postavljanje višenamjenskog Linux poslužitelja

Postoji stvarna razlika između čitanja o nečemu i prakticiranja. Zato u školama postoje laboratoriji za tako puno predmeta. Ako planirate naučiti Linuxovu sistemsku administraciju, potreban vam je poslužitelj. Tako je prvi vaš zadatak u ovoj knjizi izgradnja temeljnog okružja za postavljanje poslužitelja. Kada ćete imati dobro postavljeni poslužitelj, imat ćete dobre temelje za daljnje učenje i prakticiranje Linuxa.

Operativni sustav Linux podsjeća na kostur automobila, koji može imati bezbroj funkcija ovisno o izboru šasije i ostalih obilježja. Kako dodajete funkcije poput elektroničke pošte ili baze podataka, sustav dobiva drugačiji karakter. Treba li vam Web poslužitelj, razvojna platforma, mrežni prolaz ili datotečni ili ispisni poslužitelj? Kakav god poslužitelj vam je potreban, morate imati temeljno znanje koje će vam pružiti ovo poglavlje.

Započet ćemo sa poslužiteljem koji možete pronaći na Internetu i koji udomljava Web lokacije. Možete se pitati zašto. Zato što Internet poslužitelj možete prilagoditi da radi mnoge dodatne zadatke, poput provjere identiteta korisnika, pružanja usluga za ispis i pohranu datoteka, upravljanja lokalnom elektroničkom poštom i omogućavanja daljinskog pristupa. Sustav možete staviti na javni Web poslužitelj, spojiti ga i započeti s pružanjem Web usluga. Možete ga čak držati u vlastitom domu, ako vam je davatelj Internet usluga dodijelio statičku IP adresu.

Postavljanje poslužitelja na Internet može promijeniti vaš pogled na računarstvo. Postavljanje mreže širokog područja (WAN) razlikuje se od upotrebe Linuxa na stolnom računalu, kao datotečnog i ispisnog poslužitelja ili vatrozida.

Administratori početnici se prilikom konfiguriranja poslužitelja mogu osjećati pomalo zbunjeni nepoznatim terminima i konceptima. Nećete imati uobičajeno X Window grafičko sučelje i morat ćete upisivati naredbe umjesto da pritišćete ikone. Radit ćete u konzolskom modu u okruženju odzivnika.

U okviru naše strategije da vas naučimo administrirati sustav, u sljedećem poglavlju ćemo vam pokazati kako da na vaš sustav postavite Web alat. Pružatelji usluga koriste ovaj alat temeljen na Web sučelju za upravljanje Linux poslužiteljima koje iznajmljuju zakupcima za smještaj Web stranica i druge namjene. Zbog toga ćete ponešto obavljati i izvan crno-bijelog tekstualnog okruženja.

Slijedeći instrukcije u ovome poglavlju postavit ćete sustav koji udomljava Web lokaciju i koji ćete kasnije moći prilagoditi za druge svrhe. Vaš će sustav sadržavati:

- Web poslužitelj (Apache 2.0.x)
- Poslužitelj elektroničke pošte (Postfix)
- DNS poslužitelj (BIND 9)
- FTP poslužitelj (ProFTPD)
- Agente za slanje pošte (POP3/POP3s/IMAP/IMAPs)
- Webalizer za izradu statističkih analiza

Iako postoji mnogo načina za postavljanje daljinskog Web poslužitelja, slijedeći ovdje priložene instrukcije stvorit ćete dobre temelje za shvaćanje Linuxa. Kada svladate ovakvu konfiguraciju, moći ćete sami konfigurirati poslužitelje ovisno o potrebama.

Dok budete postavljali poslužitelj vrlo ćete se vjerojatno susresti s vama manje bliskim naredbama i konceptima. Tražit ćemo da unesete podatke za koje ćete smatrati da nemaju smisla. Iako ćemo se truditi da vam što bolje objasnimo postupak postavljanja, možda nećete biti zadovoljni informacijama iz ovog poglavlja.

Svima je teško usvojiti kompleksne informacije s prvim čitanjem. Tako, kad vam se čini da zadavanje nekih naredbi nema puno smisla, to će vam omogućiti da sakupite informacije čiji ćete smisao prepoznati nešto kasnije. Svaku ćemo temu detaljnije objasniti u poglavljima koja slijede i tako ćemo vam pomoći da što bolje ovladate temom.

Svijet Linuxa i poslužitelja vas čeka. Počnimo!

## Uvjeti za postavljanje poslužitelja

Kako biste konfigurirali Web poslužitelj možete koristiti bilo koju distribuciju Linuxa. U ovoj vježbi koristit ćemo Debian. Za ovu distribuciju smo se opredijelili zato što želimo imati stabilnu instalaciju Linuxa. Vodeće komercijalne distribucije – Red Hat Enterprise Linux i Novell SUSE Linux Enterprise Server – se napalaćuju, no Debian možete dobiti besplatno. Također, Red Hat i SUSE koriste vlasničke alate za

administriranje što otežava upoznavanje Linuxu. O klasičnoj upotrebi Linuxa više ćete naučiti koristeći Debian nego SUSE ili Ret Hat.

Kako biste postavili Linux Internet poslužitelj, treba vam veza s Internetom i statička IP adresa. Ako ne možete dobiti statičku IP adresu, možete postaviti sistem s adresom koju vam je dodijelio davatelj Internet usluga i konfigurirati ju statički. Saznajte koliko adresa traje jer će vam ta informacija trebati u slučaju da morate promijeniti IP adresu za vrijeme rada sistema.

Trebat će vam računalo, u najmanju ruku Pentium III s minimalno 256 MB RAM-a i 10 GB prostora na tvrdom disku. Sustav će bolje raditi ako imate noviji procesor i više memorije.

Ovo se poglavlje temelji na stabilnoj Debian distribuciji. Posebno vam preporučujemo korištenje CD-a s Netinstall jezgrom. CD možete skinuti s Debianove Web lokacije (http://www.debian.org).

## Instalacija Debiana

Pretpostavljamo da znate izvesti mrežnu instalaciju Linuxa. Trebat će vam nekoliko smjernica da postavite osnove.

Nakon što ste podignuli sustav s Debian CD-ROM diska vidjet ćete zaslon za prijavljivanje. Upišite `linux26` kako biste dobili najnoviju inačicu jezgre 2.6 umjesto stare inačice 2.4.

Instalacija će vas sama voditi dalje. Kada se pojavi zaslon „Configure the Network“, Debian će vam prvo predložiti konfiguraciju mreže s DHCP-om. Ako imate na raspolaganju DHCP, možete preko njega. No, ako nemate, Debian će vas odvesti do zaslona koji će vam omogućiti da mrežu konfigurirate ručno. Trebat ćete upisati ime poslužiteljskog računala, ime domene, mrežnog prolaza, IP adresu, mrežnu masku i poslužitelj imena. Ako imate registriranu domenu i statičku IP adresu, spremni ste za daljnji rad. Ako nemate registriranu domenu, trebat će vam u svakom slučaju.

Možete doći do imena domene iz puno različitih izvora i to već od 3 dolara. Pretražite Internet koristeći ključne riječi „domain registration“. Vidjet ćete koliko će vam se otvoriti mogućnosti. Mnogi prodavači nude tu uslugu za malo novaca, a neki čak i besplatno. Kako biste dobili ime domene potrebna su vam dva registrirana DNS poslužitelja. Ako nemate računalo koje bi služilo kao drugi DNS poslužitelj, može vam poslužiti DNS poslužitelj organizacije kod koje ste registrirali domenu. Svaka registrirana domena zahtijeva primarni DNS poslužitelj te rezervni ili sekundarni DNS poslužitelj.

Kada ste konfigurirali mrežno okruženje možete nastaviti s instalacijskim zadacima kako biste kompletirali bazični sistem. Debianove instalacijske skripte vodit će vas kroz sljedeće sekcije.

Ubrzo će vam se pojaviti zaslon za particioniranje tvrdog diska. Za potrebe ove knjige napravite samo jednu veliku particiju s točkom montiranja / (samo kosa crta) i particiju za razmjenu. Odaberite opciju za smještanje svih datoteka na jednu particiju. Na kraju odaberite opciju za dovršavanje particioniranja i zapišite rezultat na disk.

Osnovna Debian instalacija koju koristimo sastoji se od dva koraka. Prvi instalira GNU/Linux infrastrukturu koja omogućava pokretanje s tvrdog diska i pristup odzivniku. Također, u ovom se koraku datoteke s CD-a kopiraju na tvrdi disk.

Kada se završi prvi korak bit ćete zamoljeni izvaditi CD-ROM disk koji ste koristili za instalaciju. Od te točke nadalje instalacija će se odvijati koristeći datoteke kopirane na tvrdi disk.

Nastavljamo dalje kroz još nekoliko preostalih instalacijskih zaslona. Zadnji od njih će tražiti da ponovno pokrenemo sustav kako bismo inicijalizirali jezgru i završili instalaciju.

Nakon ponovnog pokretanja računala Debian će tražiti da dodamo neprivilegiranog korisnika za potrebe instalacije. To omogućuje da se prijavite i koristite naredbu *su* da biste dobili *root* dopuštenja. Iz sigurnosnih razloga sistemski su administratori ustanovili standardnu praksu da se nikada ne prijavljuju na sustav kao *root* korisnik, osim kada treba oporaviti sustav od kvara.

Prvi korisnički račun nazovite *Administrator* i dodijelite mu korisnički identifikator *admin*. Nemojte koristiti istu lozinku za račune *admin* i *root*. Korisnički račun *admin* koristit ćemo i u ostalim poglavljima.

Kada dođete do Debianovog zaslona za odabir softvera, pomaknite kursor u polje koje se nalazi odmah do polja „mail server“, pritisnite razmak i pustite da sustav instalira podrazumijevane pakete sve dok ne dođete do opcije kada ćete vidjeti *libc* klijent.

Trebate instalirati *libc* klijent s regularnom Unix podrškom za poštanski sandučić umjesto s *maildir* podrškom. Unix poštanski sandučić čuva svu poštu u jednoj datoteci, dok *maildir* čuva svaku poruku u posebnoj datoteci. Unix poštanske sandučiće lakše je konfigurirati i koristiti pa ćemo započeti s njima.

Debian će također tražiti da konfigurirate Exim kao agent za prijenos pošte (engl. *Mail Transfer Agent*, *MTA*), no nemojte to učiniti. Zamijenit ćemo Exim s Postfixom malo kasnije u ovome poglavlju. U međuvremenu, kada se pojavi zaslon s natpisom „Configuring Exim v4“ odaberite opciju „No“. Odgovorite potvrdno kada vas program upita „Really leave the mail system unconfigured“.

Na kraju, u zadnjem zaslonu koji se pojavljuje prilikom konfiguracije Eximа, unesite korisničko ime *admin* kao primaoca elektroničke pošte za račune *root* i *postmaster*.

### Agenti za prijenos pošte: Sendmail i alternative

Debianov podrazumijevani instalacijski proces koristi Exim, dok ostale Linux distribucije općenito podrazumijevano koriste Sendmail. Sendmail je godinam bio *de facto* standard i koristile su ga sve rane distribucije. Skoro svi procesi u Linuxu koji imaju veze s elektroničkomm poštom koriste konfiguracijske datoteke Sendmaila i većina aplikacija slobodnog softvera očekuje da Sendmail bude dio operativnog sustava.

Moguće je zavarati Linux tako da misli da koristi Sendmail, iako je on zamijenjen nekim drugim agentom. Kada instalirate Red Hat, na primjer, Sendmail se instalira podrazumijevano. Međutim, Red Hat i Fedora dolaze s programom koji omogućuje korisniku da se prebaci na Postfix, a mi ćemo to učiniti ručno.

Projektni menadžeri Debiana izabrali su Exim kao podrazumijevani agent za prijenos pošte zato što ga je autor licencirao pod licencom General Public License (GPL). Poput Postfixa, Exim je također zamjena za Sendmail.

Zbog mnogih razloga koje ćemo opisati kasnije u poglavlju, danas se uobičajeno koristi Postfix. Nećete oštetiti sustav ako Exim zamijenite Postfixom. U stvari, Postfix ćete čak preuzeti s Debianovog repozitorija.

## Daljinsko prijavljivanje na sistem

Kada ste završili s instalacijom, trebate se prijaviti na poslužitelj preko udaljene konzole s radne površine. Preporučujemo vam da daljnju administraciju radite s drugog sistema (može i s laptopa) zato što pouzdan poslužitelj normalno radi u *headless* režimu – a to znači bez monitora i tipkovnice. Počnite se privikavati da tako administrirate vaše poslužitelje, kao da ste pored njega. Na udaljenom stroju trebate samo SSH klijent koji imaju gotovo sve Linux distribucije a koji se može preuzeti i za ostale operativne sustave.

Sljedeći ispis je tipičan ispis s kojim ćete se susresti kada se preko SSH-a spojite na vaš novi Linux poslužitelj:

```
$ssh admin@server1.centralsoft.org
The authenticity of host 'server1.centralsoft.org (70.253.158.42)' can't
be established.
RSA key fingerprint is 9f:26:c7:cc:f2:f6:da:74:af:fe:15:16:97:4d:b3:e6.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'server1.centralsoft.org,70.253.158.42' (RSA)
to the list of known hosts.
Password: unesite lozinku administratora
Linux server1 2.6.8-2-386 #1 Thu May 19 17:40:50 JST 2005 i686 GNU/Linux

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.
```

```
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.

Last login: Sun Dec 25 19:07:38 2005 from 70.255.197.162
admin@server1:~$
```

U ovom trenutku uspostavili ste daljinsku vezu i možete izvoditi zadatke jednako kao i da sjedite izravno za računalom s priključenim monitorom i tipkovnicom. Ako želite, možete maknuti tipkovnicu, miš i monitor spojen na poslužitelj.

## Konfiguriranje mreže

Ako ste za vrijeme instalacije Debiana koristili DHCP, sad biste trebali konfigurirati poslužitelj sa statičkom adresom, tako da možete izvoditi testiranje koje će vam trebati nešto kasnije u poglavlju. Ako ste imali javnu IP adresu i konfigurirali je kao statičku, možete prijeći na sljedeći odjeljak.

Ako ste instalirali Debian s IP adresom dobijenom od usmjerivača ili davatelja Internet usluga, sada trebate ponovno konfigurirati mrežu. To je vrlo dobra lekcija za proučavanje Linuxove mrežne konfiguracije.

Kako biste izmijenili postavke tako da se koristi statička IP adrese, trebate biti *root* i urediti datoteku */etc/network/interfaces* kako bi odgovarala vašim potrebama. Kao primjer navodimo IP adresu 70.153.258.42.

Naša konfiguracijska datoteka izgleda ovako:

```
# /etc/network/interfaces -- configuration file for ifup(8), ifdown(8)
# The loopback interface
auto lo
iface lo inet loopback
# The first network card - this entry was created during the Debian
# installation
# (network, broadcast, and gateway are optional)
# The primary network interface
iface eth0 inet dhcp
```

Kako bismo dodali IP adresu 70.153.258.42 sučelju *eth0*, moramo izmijeniti datoteku da izgleda poput ove (morat ćete nabaviti neke informacije od davatelja Internet usluga):

```
# /etc/network/interfaces -- configuration file for ifup(8), ifdown(8)
# The loopback interface
auto lo
iface lo inet loopback
# The first network card - this entry was created during the Debian
# installation
# (network, broadcast, and gateway are optional)
auto eth0
iface eth0 inet static
        address 70.153.258.42
        netmask 255.255.255.248
        network 70.153.258.0
```

```
broadcast 70.153.258.47
gateway 70.153.258.46
```

Nakon uređivanja datoteke */etc/network/interfaces* ponovno pokrenite mrežu unosom:

```
# /etc/init.d/networking restart
```

Poslije toga morate urediti */etc/resolv.conf* i dodati poslužitelje imena kako biste razriješili Internetska imena računala u prikladne IP adrese. Još nismo konfigurirali vlastiti poslužitelj imena i to nas čeka malo kasnije u ovom poglavlju. U ovom trenutku jednostavno ćemo postaviti minimalni DNS poslužitelj. Drugi poslužitelji imena trebaju zadavati IP adrese DNS poslužitelja vašeg davatelja Internet usluga. Naša datoteka *resolv.conf* izgleda ovako:

```
search server
nameserver 70.153.258.42
nameserver 70.253.158.45
nameserver 151.164.1.8
```

Budite sigurni da koristite DNS poslužitelj koji radi s vašom domenom. U protivnom vaš DNS poslužitelj neće pokazivati da je ovlašten za vašu domenu.

Sada uredite */etc/hosts* i dodajte vaše IP adrese:

```
127.0.0.1       localhost.localdomain       localhost       server1
70.153.258.42   server1.centralsoft.org     server1
```

Zanemarite IPv6 informacije u datoteci */etc/hosts*. U osmom poglavlju pokazat ćemo vam kako postaviti IPv6.

Kako biste postavili ime računala, unesite ove naredbe:

```
# echo server1.centralsoft.org > /etc/hostname
# /bin/hostname -F /etc/hostname
```

Trebat ćete koristiti iste naredbe bez obzira kako ste postavili svoju mrežu za vrijeme instalacije, zamjenjujući svoje ime domene sa *server1.centralsoft.org*.

Sada provjerite jeste li ispravno konfigurirali ime računala tako da pokrenete naredbu *hostname*:

```
~$ hostname
server1
~$ hostname -f
server1.centralsoft.org
```

Ako dobijete takav rezultat, spremni ste za sljedeći odjeljak. Ako ne dobijete, pogledajte datoteku */etc/hostname*. Možda otkrijete da izgleda ovako:

```
#less /etc/hostname
server1
```

Ups. Trebalo bi pisati *server1.centralsoft.org*. Možete to sada izmijeniti.

## Mijenjanje podrazumijevanih Debianovih paketa

Započeli smo s paketima koji su standardni dio Debian distribucije. Kako je ranije napomenuto, moramo nešto izmijeniti da bismo mogli koristiti Postfix. Mogli biste pomisliti da kritiziramo dobar rad Debianovog tima, no nije to baš tako.

Debianov tim je odlučio da podrazumijevano instalira servise primjerene upotrebi u lokalnoj mreži, poput sustava datoteka Network File System (NFS). No, mi ćemo povezati poslužitelj na Internet pa nam ne trebaju NFS i neke druge usluge, ali nam trebaju druge, poput OpenSSL-a pa ćemo ih dodati.

Kako bismo dobili datoteke potrebne za ovo poglavlje, izvršite sljedeće naredbe:

```
# apt-get install wget bzip2 rdate fetchmail libdb3++-dev \
unzip zip ncftp xlispstat libarchive-zip-perl \
zlib1g-dev libpopt-dev nmap openssl lynx fileutils
```

Tada ćete vidjeti kako Debian preuzima datoteke u konzolu. Postupak preuzimanja će se uskoro završiti i program će pitati želite li nastaviti:

```
0 upgraded, 42 newly installed, 0 to remove and 0 not upgraded.
Need to get 12.2MB of archives.
After unpacking 35.8MB of additional disk space will be used.
Do you want to continue? [Y/n]
```

Unosom `Y` završit će se instalacija dodatnih datoteka.

Nadalje, poželjet ćete ukloniti usluge koje nećete koristiti. Izvršite sljedeće naredbe i vidjet ćete ovakav ispis:

```
# apt-get remove lpr nfs-common portmap pidentd pcmcia-cs \
pppoe pppoeconf ppp pppconfig
Reading Package Lists... Done
Building Dependency Tree... Done
Package pcmcia-cs is not installed, so not removed
The following packages will be REMOVED:
  lpr nfs-common pidentd portmap ppp pppconfig pppoe pppoeconf
0 upgraded, 0 newly installed, 8 to remove and 0 not upgraded.
Need to get 0B of archives.
After unpacking 3598kB disk space will be freed.
Do you want to continue? [Y/n] Y
(Reading database ... 22425 files and directories currently installed.)
Removing lpr ...
Stopping printer spooler: lpd .
Removing nfs-common ...
Stopping NFS common utilities: statd.
Removing pidentd ...
Removing portmap ...
Stopping portmap daemon: portmap.
Removing pppoeconf ...
Removing pppoe ...
```

```
Removing pppconfig ...
Removing ppp ...
Stopping all PPP connections...done.
```

Dvaput provjerite naredbe koje upisujete. Ako pogriješite, Debian će vam reći da ne može pronaći zadanu datoteku. U takvom slučaju jednostavno ponovno unesite *apt-get* zadajući samo ime paketa.

Kako ste napravili promjene u bazi podataka s paketima, trebate promijeniti i skripte koje započinju prilikom pokretanja računala. Upotrijebite sljedeće naredbe kako biste modificirali početne skripte:

```
# update-rc.d -f exim remove
Removing any system startup links for /etc/init.d/exim ...
# update-inetd --remove daytime
# update-inetd --remove telnet
# update-inetd --remove time
# update-inetd --remove finger
# update-inetd --remove talk
# update-inetd --remove ntalk
# update-inetd --remove ftp
# update-inetd --remove discard
```

Sada trebate ponovno pokrenuti *inetd* – poslužiteljski proces za standardne Internetske servise. *inetd* se općenito pokreće zajedno s računalom, no budući da ste promijenili servise sistema, zato ga i trebate ponovno pokrenuti kako bi sistem registrirao promjenu konfiguracije. Naredba *inetd* prihvaća argument koji pokazuje na konfiguracijsku datoteku s popisom usluga koje pruža. No, ako nije naveden niti jedan argument, *inetd* će konfiguraciju pročitati iz datoteke */etc/inetd.conf*, što nam odgovara. Naredbe *update-inetd* spremaju naše promjene u toj datoteci.

Da biste ponovno pokrenuli *inetd* s podrazumijevanom konfiguracijskom datotekom unesite:

```
# /etc/init.d/inetd reload
```

Na konzoli će se pojaviti sljedeća poruka:

```
Reloading internet superserver: inetd
```

# Postavljanje kvota

Apacheov Web poslužitelj daje Linuxu mogućnost *virtualnog udomljavanja*, što znači da poslužitelj može udomiti nekoliko Web lokacija s imenima domena koja se razlikuju od imena poslužitelja. U konfiguracijskoj datoteci Web poslužitelja možete definirati različite domene koristeći klauzule virtualnog udomljavanja. Na primjer, iako je u ovoj knjizi korišteno ime domene *centralsoft.org*, mogli smo dati naziv *mothersmagic.com, wildbills.info* ili bilo koji drugi naziv domene koji smo registrirali i koji će koristiti istu IP adresu.

Ovaj koncept razradili smo detaljno u šestom poglavlju. Za sada, razmišljajte o IP adresi kao o telefonskom broju u kući u kojoj stanuje više osoba. Kada preglednik pristupi ulazu 80, može pristupiti bilo kojoj domeni koju ste postavili.

Linux pruža način za upravljanje prostorom na disku koji će koristiti višestruke domene – *kvote*. Unix je izvorno zadavao kvote za korisničke račune kako ne bi zauzimali previše mjesta na poslužitelju. Na primjer, ako imate 50 korisnika koji dijele prostor na poslužiteljevom disku, bez sustava kvota, jedan bi korisnik mogao zauzeti sav slobodan prostor na disku tako da ostali korisnici ne mogu više ništa spremiti.

Kvote tjeraju korisnike da ostanu unutar zadanih okvira i onemogućuju im da neograničeno koriste prostor na disku. Sistem bilježi kvote za korisnike i za datotečne sustave. Ako imate više datotečnih sustava u koje korisnici mogu spremati datoteke, postavite kvote za svaki datotečni sustav posebno.

Možete koristiti isti sistem kvota kako biste ograničili prostor dodijeljen domeni koju udomljujete. Postoje različiti alati s kojima možete administrirati i automatizirati održavanje kvota. U ovoj fazi postavljanja poslužitelja dodat ćete sustav kvota koji možete koristiti kasnije.

Prvo instalirajte *quota* pakete upotrebom *apt-get*:

```
# apt-get install quota quotatool
```

Pojavit će se sljedeći upit:

```
Enable this option if you want the warnquota utility to be run daily to alert users
when they are over quota.
Send daily reminders to users over quota?
<Yes>                       <No>
```

Odaberite <No>.

Debian će instalirati i konfigurirati dva paketa a vi ćete morati urediti */etc/fstab* kako biste omogućili kvote na svakom datotečnom sustavu na kojem ih želite imati. Zato što naš sustav ima samo jednu particiju za sve korisničke datoteke, možete samo dodati opcije *usrquota* i *grpquota* particiji s točkom montiranja /:

```
# /etc/fstab: static filesystem information.
#
# <filesystem>  <mount point>   <type>  <options>       <dump>  <pass>
proc            /proc           proc    defaults        0       0
/dev/sda1       /               ext3    defaults,errors=remount-
ro,usrquota,grpquota 0       1
/dev/sda5       none            swap    sw              0       0
/dev/hdc        /media/cdrom0   iso9660 ro,user,noauto  0       0
/dev/fd0        /media/floppy0  auto    rw,user,noauto  0       0
```

Sada zadajte sljedeće naredbe kako biste dodali datoteke u korijenski direktorij:

```
# touch /quota.user /quota.group
# chmod 600 /quota.*
# mount -o remount /
# quotacheck -avugm
```

Linuxova jezgra obično podrazumijevano ima podršku za kvote. Jezgra čita postavke kvota iz */etc/fstab* i provjerava *quota.user* i *quota.group* kako bi utvrdila imaju li korisnici i/ili grupe ograničenja u korištenju prostora na disku.

Sada ćete ovo vidjeti na vašoj konzoli:

```
quotacheck : Scanning / dev/ hda1 [/] done
```

Također ćete vidjeti i poruku poput ove:

```
quotacheck: Checked 1912 directories and 28410 files
```

Sad možete zadati sljedeću naredbu:

```
# quotaon -avug
```

Vidjet ćete sljedeće poruke:

```
/dev/hda1 [/]: group quotas turned on
/dev/hda1 [/]: user quotas turned on
```

Pitate li se što ste upravo učinili? Ovaj niz je uključio primjenu kvota na sustavu. Ako mislite da trebate znati nešto više o tome, provjerite što piše o kvotama na stranicama s uputama (manpages). Poslužitelj je sada postavljen tako da koristi kvote.

# Pružanje usluga imena domena

U trećem poglavlju naučit ćete kako upravljati imenima domena za vaš poslužitelj i virtualne domene smještene na računalu. Za sada ćemo postaviti minimalnu konfiguraciju BIND-a, sveprisutnog DNS poslužitelja.

Debian u svojem repozitoriju nudi stabilnu verziju BIND-a. Instalirat ćemo i postaviti BIND i čuvati ga u *chroot* okruženju, što znači da neće moći vidjeti datoteke izvan svojeg stabla direktorija niti ćete im moći pristupati. To je vrlo važna sigurnosna tehnika. Pojam *chroot* se odnosi na domišljatost promjene korijenskog datotečnog sustava koji proces vidi, tako da mu je veći dio sustava efektivno nepristupačan.

Također ćemo BIND konfigurirati tako da se izvodi kao neki drugi korisnik, a ne *root*. Na takav način, ako netko želi pristupiti BIND-u, neće moći dobiti privilegije *root* korisnika ili kontrolirati druge procese.

Kako biste instalirali BIND na svoj Debian poslužitelj, zadajte ovu naredbu:

```
# apt-get install bind9
```

Debian preuzima i konfigurira datoteku kao internetski servis. Na konzoli će se pojaviti sljedeća poruka:

```
Setting up bind9 (9.2.4-1)
Adding group `bind' (104)
Done.
Adding system user `bind'
Adding new user `bind' (104) with group `bind'.
Not creating home directory.
Starting domain name service: named.
```

Sličan ispis dobit ćete i ako s pomoću alata *apt-get* instalirate ili uklonite neke druge usluge.

Kako bi se BIND nalazio u sigurnom okruženju, potrebno je izraditi direktorij u kojem će se servis moći izvršavati tako da nije izložen drugim procesima. Osim toga, izvršavat će se kao neprivilegirani korisnik i samo će korisnik *root* moći pristupiti tom direktoriju.

Najprije zaustavite servis zadavanjem sljedeće naredbe

```
# /etc/init.d/bind9 stop
```

Zatim uredite datoteku /etc/default/bind9 tako da se servis izvršava kao neprivilegirani korisnik *bind*, čiji je korijenski datotečni sustav /var/lib/named. Promijenite kod:

```
OPTS="-u bind"
```

tako da bude:

```
OPTIONS="-bind –t /var/lib/named"
```

Kako biste stvorili kompletnu okolinu potrebnu za pokretanje BIND-a, izradite potrebne direktorije u /var/lib:

```
# mkdir -p /var/lib/named/etc
# mkdir /var/lib/named/dev
# mkdir -p /var/lib/named/var/cache/bind
# mkdir -p /var/lib/named/var/run/bind/run
```

Zatim premjestite direktorij *config* s */etc* na */var/lib/named/etc*:

```
# mv /etc/bind/var/lib/named/etc
```

Nadalje, izradite simbolički link za novi *config* direktorij sa stare lokacije, kako biste izbjegli probleme kada u budućnosti budete nadograđivali BIND:

```
# ln -s/var/lib/named/etc/bind/etc/bind
```

Pripremite uređaje *null* i *random* za upotrebu s BIND-om i uredite dopuštenja za direktorije:

```
# mknod /var/lib/named/dev/null c 1 3
# mknod /var/lib/named/dev/random c 1 8
```

Zatim promijenite dopuštenja i vlasništvo nad datotekama:

```
# chmod 666 /var/lib/named/dev/null /var/lib/named/dev/random
# chown -R bind:bind /var/lib/named/var/*
# chown -R bind:bind /var/lib/named/etc/bind
```

Također ćete morati izmijeniti skriptu */etc/init.d/sysklogd* tako da možete vidjeti poruke u sistemskim dnevnicima. Promijenite red:

```
SYSLOGD=""
```

tako da glasi:

```
SYSLOGD="-a /var/lib/named/dev/log"
```

Sada ponovno pokrenite proces bilježenja poruka u dnevnik s ovom naredbom:

```
# /etc/init.d/sysklogd restart
```

Pojavit će se sljedeća poruka:

```
Restarting system log daemon: syslogd.
```

Napokon pokrenite BIND:

```
# /etc/init.d/bind9 start
```

Provjerite */var/log/syslog* u slučaju kakve pogreške. Kroz datoteku se možete kretati upotrebom:

```
# less /var/log/syslog
```

Možete biti sigurni da ste uspješno pokrenuli BIND ako vidite poruku:

```
Starting domain name service: named.
```

Provjerimo sada funkcionira li *named* bez problema. Izvršite ovu naredbu i trebali biste dobiti rezultat koji slijedi:

```
server1:/home/admin# rndc status

number of zones: 6
debug level: 0
xfers running: 0
xfers deferred: 0
soa queries in progress: 0
query logging is OFF
server is up and running
server1:/home/admin#
```

Ako DNS ne radi ispravno, vidjet ćete poruku poput ove:

```
server1:~# rndc status
rndc: neither /etc/bind/rndc.conf nor /etc/bind/rndc.key was found
server1:~#
```

Srećom, naš DNS poslužitelj radi ispravno.

Do sada nismo postavili datoteke primarne zone ni konfigurirali sistemski DNS za bilo što drugo osim za poslužitelj privremene memorije koji popunjava svoj spremnik svaki put kada netko zatraži Web stranicu. U trećem ćemo poglavlju objasniti kako konfigurirati primarni i sekundarni DNS poslužitelj.

Iako mnogi ne pridaje tome veliku važnost, dobro poznavanje DNS sustava je krucijalno zato što se mnogi drugi servisi oslanjaju na njega. Vidjet ćete da je DNS kritična komponenta svakog internetskog servisa koji vaš sustav pruža.

## Dodavanje relacijske baze podataka: MySQL

Web lokacije i aplikacije Web servisa koriste relacijske baze podataka za ugrađivanje objekata na Web stranice. To omogućava brzu obradu zahtjeva za Web stranicama. Preglednik Weba može poslati trideset zahtjeva odjednom opterećujući tako CPU, memoriju i tvrdi disk.

Relacijske baze podataka u kombinaciji s Web poslužiteljem mogu uspješno i vrlo brzo generirati kompleksne Web stranice.

U ovoj knjizi nismo obuhvatili kompleksnu temu o administriranju baza podataka. Međutim, administratori Linux sustava često se nađu u situaciji kada razvojni tim očekuje od njih da instaliraju bazu podataka koja će se koristiti pri razvoju. Zbog toga ćemo vam pokazati kako konfigurirati Linux poslužitelj s jednom od popularnih baza podataka otvorenog izvornog koda: MySQL. Da biste učinkovito koristili bazu, trebate znati kako:

1. Instalirati i pokrenuti MySQL.
2. Izraditi korisnički račun *root*.
3. Izraditi korisnički račun regularnog MySQL korisnika koji će program koristiti za pristup bazi.
4. Praviti rezervne kopije podataka te restaurirati podatke nakon gubitka.

Da biste instalirali poslužitelj baze podataka, uobičajeni klijentetski program koji možete koristiti za administriranje poslužitelja i biblioteku potrebnu za rad ovih programa, zadajte ovu naredbu:

```
# apt-get install mysql-server mysql-client libmysqlclient12-dev
```

Debian će preuzeti MySQL iz svojeg repozitorija i započeti proces instaliranja. Vidjet ćete sljedeće poruke:

```
Install Hints
MySQL will only install if you have a NON-NUMERIC hostname that is
resolvable via the /etc/hosts file. E.g. if the "hostname" command
returns "myhostname" then there must be a line like "10.0.0.1
myhostname".
A new mysql user "debian-sys-maint" will be created. This mysql account
is used in the start/stop and cron scripts. Don't delete.
Please remember to set a PASSWORD for the MySQL root user! If you use a
/root/.my.cnf, always write the "user" and the "password" lines in
there, never only the password!
See /usr/share/doc/mysql-server/README.Debian for more information.
<Ok>
```

Administrativno, MySQL se može usporediti s Linuxom: i jedan i drugi imaju *root* korisnika koji kontrolira sve što se zbiva i može dodjeljivati i ukidati privilegije ostalim korisnicima. MySQL *root* korisnik nema nikakve veze s Linuxovim *root* korisnikom – samo je ime isto. Unosom sljedeće naredbe izradit ćete MySQL *root* korisnika:

```
# mysqladmin -u root password ´lozinka´
```

Odaberite lozinku koju neće biti lako pogoditi. Kada god u budućnosti poželite administrirati MySQL, morat ćete unijeti sljedeću naredbu i upisati lozinku:

```
# mysql -u root -p
Enter password:
```

Provjerite jesu li poslužitelj i klijent pokrenuti te možete li pristupiti poslužitelju. Trebali biste na konzoli vidjeti ispis sličan ovome:

```
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 14 to server version: 4.0.24_Debian-10-log
Type 'help;' or '\h' for help. Type '\c' to clear the buffer.
mysql>
```

Upišite `/q` ili `quit;` za izlazak.

Pošto je MySQL pokrenut, možete zadati *netstat -tap* i vidjet ćete red poput ovog:

```
tcp  0  0 localhost.localdo:mysql *:*  LISTEN  2449/mysqld
```

MySQL dostupan je na lokalnom računalu (127.0.0.1) na ulazu 3306. Ako ne vidite ovaj red uredite */etc/mysql/my.cnf* (konfiguracijska datoteka u kojoj i klijent i poslužitelj traže operacijske parametre) i dodajte znak # kako biste zakomentirali `skip-networking`:

```
#skip-networking
```

Ako želite da MySQL osluškuje na svim raspoloživim IP adresama, uredite */etc/mysql/my.cnf* i zakomentirajte red `bind-address = 127.0.0.1`:

```
#bind-address  = 127.0.0.1.
```

Ako morate urediti /etc/mysql/my.cnf, ponovno pokrenite MySQL zadavanjem ove naredbe:

```
# /etc/init.d/mysql restart
```

Ovaj odlomak ne obuhvaća sve funkcije koje će razvojni tim tražiti od vas. MySQL je sada postavljen na poslužitelj i to je dovoljno da možete krenuti sa sljedećim koracima. U poglavljima 6 i 11 više ćemo se pozabaviti MySQL-om.

## Konfiguriranje sustava elektroničke pošte s Postfixom, POP3 i IMAP poslužiteljem

U ovome dijelu dodat ćemo sustav za prijenos elektroničke pošte i agente za dostavu te implementirati strog sustav kontrole njihovog okruženja. Pokazat ćemo vam kako da autorizirate dobronamjerne korisnike sustava elektroničke pošte i spriječite neovlašteni pristup.

Više od 25 godina Sendmail je služio kao primarni agent za prijenos elektroničkih poruka na Internetu. Mnoge aplikacije napisane za Linux očekuju da će na poslužitelju biti instaliran Sendmail. Budući da je projektiran prije nego što je Internet postao dostupan javnosti, Sendmail ima mnoge sigurnosne probleme navedene na Web stranici Common Vulnerabilities and Exposure (CVE) na adresi *http://cve.mitre.org*.

Srećom, pojavili su se drugi agenti za prijenos poruka kako bi zauzeli mjesto Sendmaila. Glavni problem tih agenata je u očekivanju glavnih aplikacija da Sendmail i dalje bude instaliran na Linux poslužitelju. Agenti poput Postfixa i Exima moraju biti u mogućnosti predstaviti se aplikacijama kao da su Sendmailu. Njih zovemo *„igrači s klupe“* i mogu raditi u Sendmail režimu.

Preferiramo Postfix kao zamjenu za Sendmail. Postfix je brži od Sendmaila, sigurniji je, ima modularnu arhitekturu i brojne mogućnosti potrebne korisnicima koji šalju i primaju mnogo poruka. Postfix koristi internetski protokol Standard Mail Transport Protocol (SMPT) i ima najmanji broj stavki na popisu poznatih sigurnosnih problem. Zbog svih tih razlogat ćemo kao agent za prijenos poruka koristiti Postfix radije nego Sendmail.

Sigurnost elektroničke pošte podrazumijeva držanje svih neautoriziranih korisnika podalje od poslužitelja (tako da ga ne mogu koristiti za masovno slanje neželjenih poruka), osiguravanje da se nitko ne može predstaviti kao registrirani korisnik te čuvanje sadržaja svake poruke od neovlaštenog čitanja ili izmjene u prijenosu.

Slaba sigurnost elektroničke pošte može omogućiti varalicama da obmanjuju korisnike. Da bismo unaprijedili sustav provjere identiteta korisnika, instalirat ćemo Postfix s protokolom Transport Layer Security (TLS) poznatijim pod nazivom Secure Sockets Layer (SSL). To sprječava slanje nezaštićenih tekstualnih lozinki od klijenta elektroničke pošte do poslužitelja.

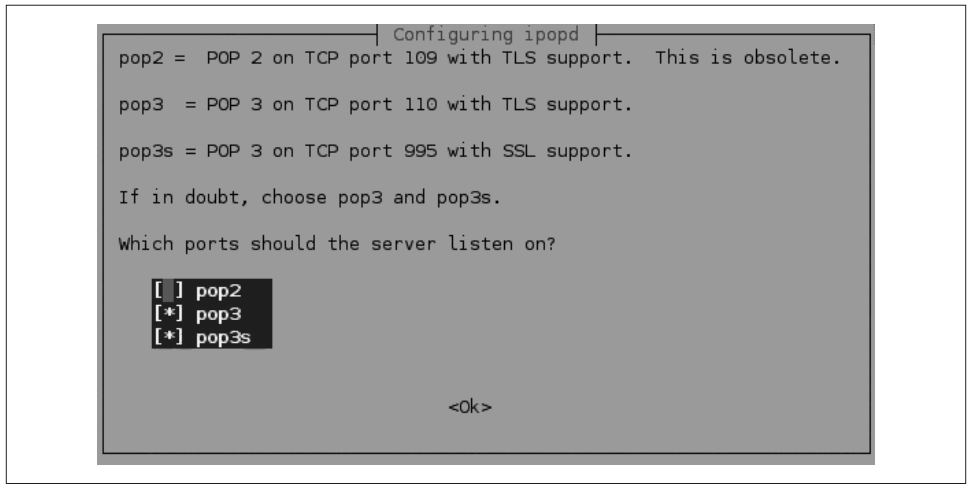
Također tražit ćemo od korisnika da se predstave ili prijave na naš poštanski poslužitelj. Zbog toga ćemo instalirati Simple Authentication and Security Layer (SASL) sloj. To će izraditi proširenje (ESMTP) koje omogućava SMTP klijentu da se identificira poslužitelju.

Za instaliranje paketa potrebnih Postfixu i drugih komponenata sustava elektroničke pošte unesite:

```
# apt-get install postfix postfix-tls libsasl2 sasl2-bin \
libsasl2-modules ipopd-ssl uw-imapd-ssl
```

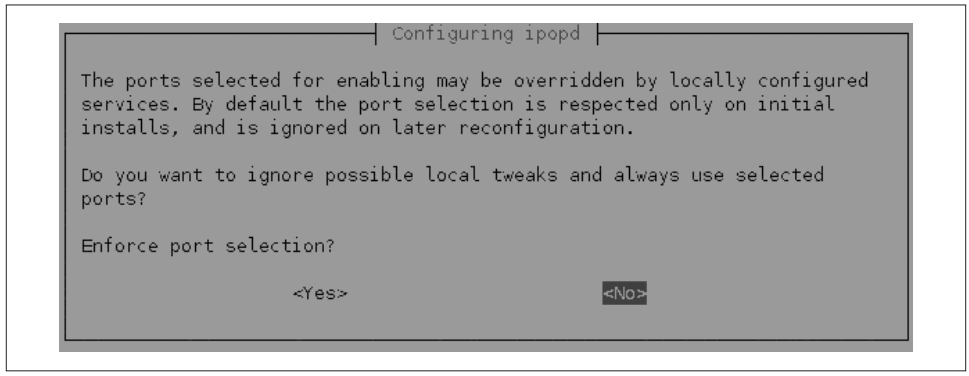
Dok Debian bude instalirao pakete, preko cijelog zaslona (mogućnost temeljena na *ncurses*) će se prikazati nekoliko okvira s pitanjima.

Kada se otvori okvir „Configuring ipopd“, prikazan na slici 2-1, odaberite `pop3` i `pop3s`.



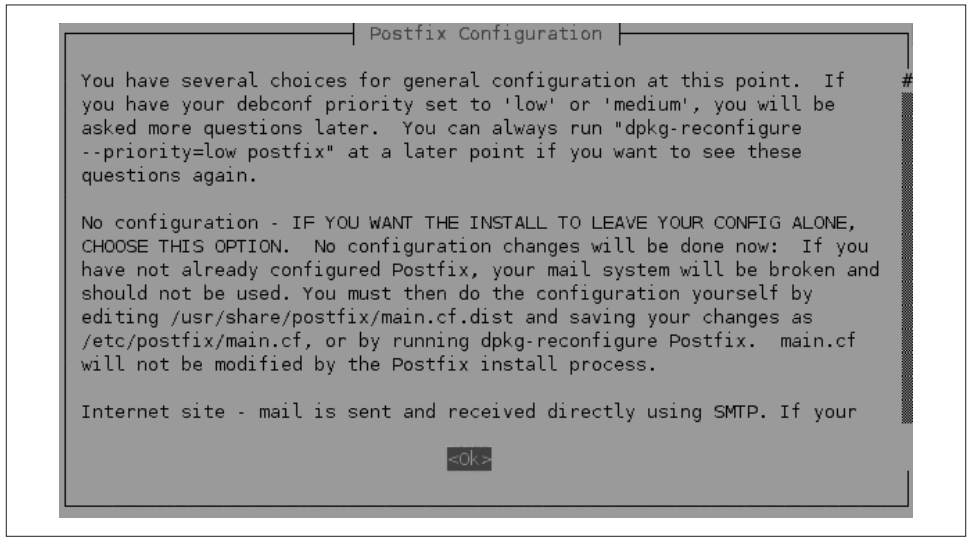

*Slika 2-1. Debianov okvir za konfiguriranje sustava elektroničke pošte*

U nastavku će se prikazati okvir kao na slici 2-2. U njemu trebate odabrati <No> da biste kasnije, ako se pokaže potreba, mogli preusmjeriti ulaze. Možete prihvatiti podrazumijevane ulaze zato što koristimo TLS i pozadinski servis (engl. *deamon*) SASL.

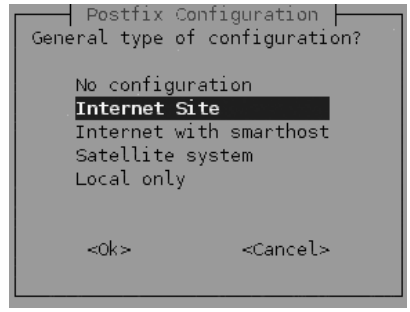



*Slika 2-2. Prihvatite podrazumijevane ulaze za elektroničku poštu*

Okvir na slici 2-3 je informativan. Program za instaliranje Debiana govori vam koje sve opcije imate na raspolaganju za konfiguriranje elektroničke pošte. Stisnite OK kako biste došlo do okvira sa slike 2-4 u kojem možete izabrati opcije. Za potrebe izlaganja u ovoj knjizi odabrali smo Internet Site, zato što ćemo za sav promet – unutar lokalne mreže i prema Internetu – koristiti SMPT. U tom će slučaju Debian ponuditi konfiguracijsku datoteku koja najbolje odgovara našim potrebama. Kasnije možemo dodavati značajke u ovu podrazumijevanu konfiguraciju.



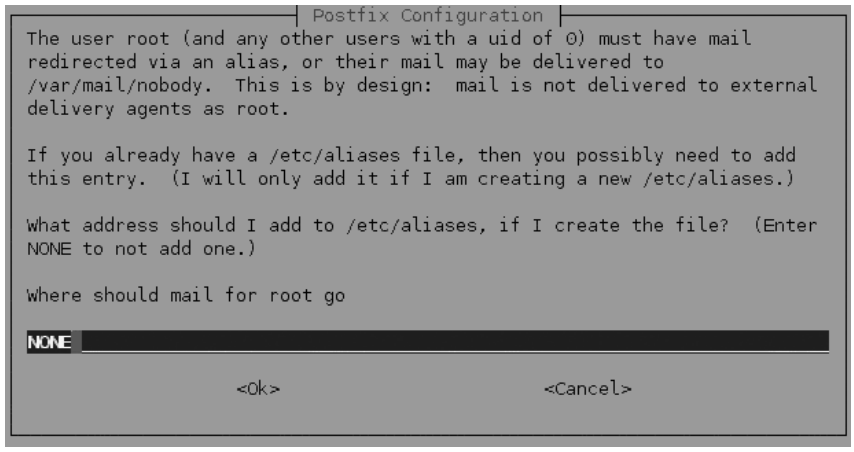

*Slika 2-3. Konfiguracijske opcije Postfixa*




*Slika 2-4. Odabir opcije Internet Site s konfiguracijskog izbornika*

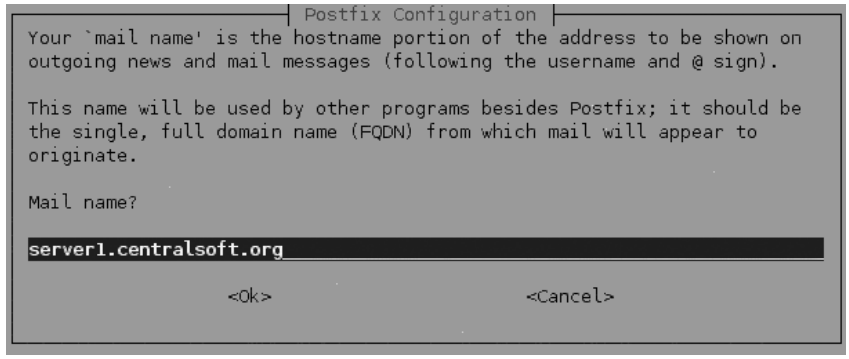
Kada ste postavili Postfix da može slati poštu, radit će kao standardni agent za prijenos poruka. Nemojte izabrati opciju sa slike 2-4 za upotrebu drugog poštanskog poslužitelja kao *smarthosta*. Drugim riječima, vaš će sistem biti poštanski autoritet za vašu domenu. Ako ste u prošlosti koristili drugi poslužitelj (poput popularnog portala za elektroničku poštu ili poslužitelja koji održava davatelj Internet usluga) za slanje i primanje pošte, vaš će poslužitelj sada preuzeti te poslove.

Nadalje, u okviru sa slike 2-5 odgovorite `NONE`. Postfix će tada izraditi vlastitu datoteku sa aliasima.



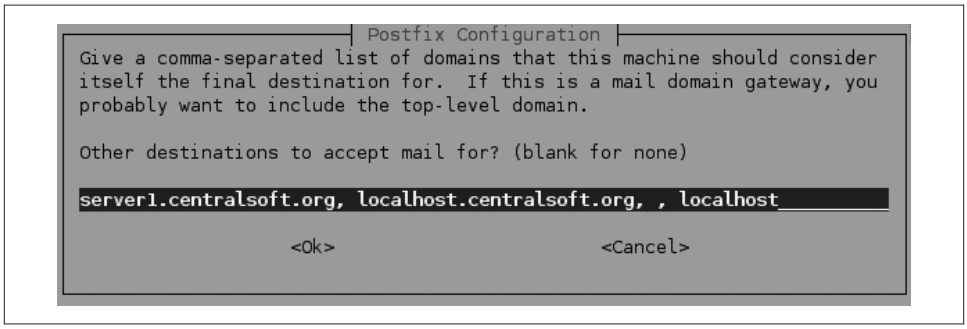

*Slika 2-5. Mogućnost upotrebe postojećeg alias računa*

Na slikama 2-6 i 2-7 Postfixov konfigurator želi znati za koga će primati i prosljeđivati poštu. Glavno ime domene je također i „ime za primanje pošte“. Postfix će to ime koristiti kako bi provjerio poštu upućenu poslužitelju. Kada dođete do okvira prikazanih na slikama 2-6 i 2-7, oni će imati podrazumijevane vrijednosti u plavim poljima za tekst. Možete prihvatiti sliku 2-6 kako je prikazana.




*Slika 2-6. Provjera zadanog potpuno kvalificiranog imena domene za Postfix*

Ime domene koje koristimo u ovoj knjizi je *centralsoft.org*. Vi ga zamijeniti s imenom stvarne domene koju ćete koristiti.



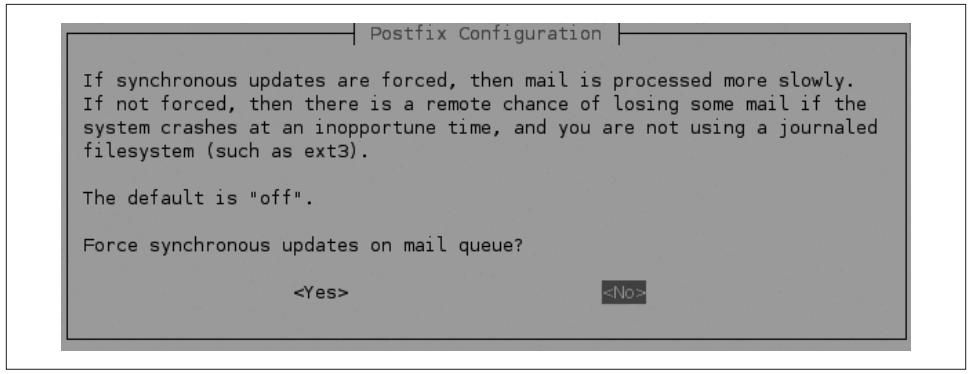

*Slika 2-7. Popis internih domena koje će Postfix opsluživati*

Na slici 2-7 primijetit ćete da nakon imena *localhost.centralsoft.org* slijede dva zareza. Obrišite drugi zarez.

Na slici 2-8 Postfix konfigurator pita za *sinkronizirano ažuriranje*. Za sad odgovorite <No> i krenite dalje. U većem dijelu petog poglavlja bavit ćemo se administriranjem poštanskih poslužitelja.

Nakon što je Debian završio s instaliranjem i kada vidite da se konzola vratila sistemskom odzivniku, morat ćete početi sa slaganjem različitih komponenti sustava za rad s elektroničkom poštom. To znači da ćete upisivati unose u Postfix konfiguracijske datoteke te generirati certifikate i ključeve za šifriranje.

O tome smo vas upozorili na početku poglavlja. Neke naredbe koje ćete koristiti neće vam imati previše smisla. Ne brinite o tome, shvatit ćete u kojem se smjeru krećemo pogledate li odjeljke s početka ovog poglavlja.




*Slika 2-8. Odbijte sinkronizirano ažuriranje*

Naredba *postconf* nalazi se u direktoriju */usr/sbin*. Koristit ćete ju za ispis vrijednosti Postfixovih parametara u Postfixovu konfiguracijsku datoteku *main.cf*.

Kako ste instalirali Postfix i Debian ga je postavio kao servis, trebate nadalje reći Postfixu kako da izvodi provjeru identiteta korisnika. Koristite sljedeće naredbe:

```
# postconf -e 'smtpd_sasl_local_domain ='
# postconf -e 'smtpd_sasl_auth_enable = yes'
# postconf -e 'smtpd_sasl_security_options = noanonymous'
# postconf -e 'broken_sasl_auth_clients = yes'
# postconf -e 'smtpd_recipient_restrictions = \
permit_sasl_authenticated,permit_mynetworks,reject_unauth_destination'
# postconf -e 'inet_interfaces = all'
```

Sljedeće naredbe upisuju tekstualne unose u datoteku *smtpd.conf*:

```
# echo 'pwcheck_method: saslauthd' >> /etc/postfix/sasl/smtpd.conf
# echo 'mech_list: plain login' >> /etc/postfix/sasl/smtpd.conf
```

Sada izradite direktorij za SSL certifikate te generirajte i certifikate i ključeve za šifriranje:

```
# mkdir /etc/postfix/ssl
# cd /etc/postfix/ssl/
# openssl genrsa -des3 -rand /etc/hosts -out smtpd.key 1024
293 semi-random bytes loaded
Generating RSA private key, 1024 bit long modulus
.....................................++++++
.....................................++++++
e is 65537 (0x10001)
Enter pass phrase for smtpd.key:
Verifying - Enter pass phrase for smtpd.key:
```

Tada upišite ovu naredbu kako biste promijenili dopuštenja za datoteku koja sadrži RSA ključ OpenSSL:

```
# chmod 600 smtpd.key
```

Generirajte još jedan ključ i certifikat te zamijenite postojeće ključeve s novim:

```
# openssl req -new -key smtpd.key -out smtpd.csr
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
Country Name (2 letter code) [AU]:
State or Province Name (full name) [Some-State]:
Locality Name (eg, city) []:
Organization Name (eg, company) [Internet Widgits Pty Ltd]: centralsoft.org
Organizational Unit Name (eg, section) []: web
Common Name (eg, YOUR name) []:
Email Address []:
Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:
An optional company name []:cso
# openssl x509 -req -days 3650 -in smtpd.csr -signkey smtpd.key -out \
smtpd.crt
Signature ok
subject=/C=US/ST=Texas/L=Dallas/O=centralsoft.org/OU=web/CN=Tom_Adelstein/
emailAddress=tom.adelstein@centralsoft.org
Getting Private key
Enter pass phrase for smtpd.key:
# openssl rsa -in smtpd.key -out smtpd.key.unencrypted
Enter pass phrase for smtpd.key:
writing RSA key
# mv -f smtpd.key.unencrypted smtpd.key
# openssl req -new -x509 -extensions v3_ca -keyout cakey.pem -out \
cacert.pem -days 3650
Generating a 1024 bit RSA private key
.....................++++++
.........................++++++
writing new private key to 'cakey.pem'
Enter PEM pass phrase:
Verifying - Enter PEM pass phrase:
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank
Country Name (2 letter code) [AU]:
State or Province Name (full name) [Some-State]:
Locality Name (eg, city) []:
Organization Name (eg, company) [Internet Widgits Pty Ltd]:
Organizational Unit Name (eg, section) []:
Common Name (eg, YOUR name) []:
Email Address []:
```

Postoje rasprave na temu treba li ili ne dati ispravne informacije kod izrade samogenerirajućih certifikata. Preporučujemo da unesete informacije koje odgovaraju vašim uvjetima.

Sada je potrebno Postfix obavijestiti o ključevima i certifikatima koristeći sljedeće *postconf* naredbe:

```
# postconf -e 'smtpd_tls_auth_only = no'
# postconf -e 'smtp_use_tls = yes'
# postconf -e 'smtpd_use_tls = yes'
# postconf -e 'smtp_tls_note_starttls_offer = yes'
# postconf -e 'smtpd_tls_key_file = /etc/postfix/ssl/smtpd.key'
# postconf -e 'smtpd_tls_cert_file = /etc/postfix/ssl/smtpd.crt'
# postconf -e 'smtpd_tls_CAfile = /etc/postfix/ssl/cacert.pem'
# postconf -e 'smtpd_tls_loglevel = 1'
# postconf -e 'smtpd_tls_received_header = yes'
# postconf -e 'smtpd_tls_session_cache_timeout = 3600s'
# postconf -e 'tls_random_source = dev:/dev/urandom'
```

Datoteka */etc/postfix/main.cf* sada bi trebala izgledati ovako:

```
# See /usr/share/postfix/main.cf.dist for a commented, more complete
# version
smtpd_banner = $myhostname ESMTP $mail_name (Debian/GNU)
biff = no
# Appending .domain is the MUA's job
append_dot_mydomain = no
# Uncomment the next line to generate "delayed mail" warnings
#delay_warning_time = 4h
myhostname = server1.example.com
alias_maps = hash:/etc/aliases
alias_database = hash:/etc/aliases
myorigin = /etc/mailname
mydestination = server1.example.com, localhost.example.com, localhost
relayhost =
mynetworks = 127.0.0.0/8
mailbox_command = procmail -a "$EXTENSION"
mailbox_size_limit = 0
recipient_delimiter = +
inet_interfaces = all
smtpd_sasl_local_domain =
smtpd_sasl_auth_enable = yes
smtpd_sasl_security_options = noanonymous
broken_sasl_auth_clients = yes
smtpd_recipient_restrictions =
permit_sasl_authenticated,permit_mynetworks,reject_unauth_destination
smtpd_tls_auth_only = no
smtp_use_tls = yes
smtpd_use_tls = yes
smtp_tls_note_starttls_offer = yes
smtpd_tls_key_file = /etc/postfix/ssl/smtpd.key
smtpd_tls_cert_file = /etc/postfix/ssl/smtpd.crt
smtpd_tls_CAfile = /etc/postfix/ssl/cacert.pem
```

```
smtpd_tls_loglevel = 1
smtpd_tls_received_header = yes
smtpd_tls_session_cache_timeout = 3600s
tls_random_source = dev:/dev/urandom
```

Ako vaša datoteka izgleda ovako, možete koristiti sljedeće naredbe za uvođenje promjena:

```
# /etc/init.d/postfix restart
Stopping mail transport agent: Postfix.
Starting mail transport agent: Postfix.
```

Provjeru identiteta provodit će *saslauthd*, SASL pozadinski servis, no morat ćete izmijeniti neke stvari kako bi sve ispravno radilo. Kako se Postfix izvodi preusmjeren u */var/spool/postfix*, zadajte sljedeće naredbe

```
# mkdir -p /var/spool/postfix/var/run/saslauthd
# rm -fr /var/run/saslauthd
```

Sada morate urediti datoteku */etc/default/saslauthd* kako biste aktivirali *saslauthd*. Obrišite znak # ispred `START=yes` i dodajte red `PARAMS="-m/var/spool/postfix/var/run/saslauthd"`, tako da datoteka izgleda ovako:

```
# This needs to be uncommented before saslauthd will be run automatically
START=yes
PARAMS="-m /var/spool/postfix/var/run/saslauthd"
# You must specify the authentication mechanisms you wish to use.
# This defaults to "pam" for PAM support, but may also include
# "shadow" or "sasldb", like this:
# MECHANISMS="pam shadow"
MECHANISMS="pam"
```

Na kraju uredite datoteku */etc/init.d/saslauthd*. Promijenit red:

```
dir=´dpkg-stateoverride --list $PWDIR´
```

u

```
#dir=´dpkg-stateoverride --list $PWDIR´
```

Tada izmijenite varijable `PWDIR` i `PIDFILE` i dodajte na početak varijablu `dir` :

```
PWDIR="/var/spool/postfix/var/run/${NAME}"
PIDFILE="${PWDIR}/saslauthd.pid"
dir="root sasl 755 ${PWDIR}"
```

*/etc/init.d/saslauthd* bi sada trebala izgledati ovako:

```
#!/bin/sh -e
NAME=saslauthd
DAEMON="/usr/sbin/${NAME}"
DESC="SASL Authentication Daemon"
DEFAULTS=/etc/default/saslauthd
PWDIR="/var/spool/postfix/var/run/${NAME}"
PIDFILE="${PWDIR}/saslauthd.pid"
dir="root sasl 755 ${PWDIR}"
createdir() {
# $1 = user
```

```
# $2 = group
# $3 = permissions (octal)
# $4 = path to directory
        [ -d "$4" ] || mkdir -p "$4"
        chown -c -h "$1:$2" "$4"
        chmod -c "$3" "$4"
}
test -f "${DAEMON}" || exit 0
# Source defaults file; edit that file to configure this script.
if [ -e "${DEFAULTS}" ]; then
    . "${DEFAULTS}"
fi
# If we're not to start the daemon, simply exit
if [ "${START}" != "yes" ]; then
    exit 0
fi
# If we have no mechanisms defined
if [ "x${MECHANISMS}" = "x" ]; then
    echo "You need to configure ${DEFAULTS} with mechanisms to be used"
    exit 0
fi
# Add our mechanisms with the necessary flag
PARAMS="${PARAMS} -a ${MECHANISMS}"
START="--start --quiet --pidfile ${PIDFILE} --startas ${DAEMON} --name
    ${NAME} -- ${PARAMS}"
# Consider our options
case "${1}" in
    start)
        echo -n "Starting ${DESC}: "
        #dir=`dpkg-statoverride --list $PWDIR`
        test -z "$dir" || createdir $dir
        if start-stop-daemon ${START} >/dev/null 2>&1 ; then
                echo "${NAME}."
        else
                if start-stop-daemon --test ${START} >/dev/null 2>&1; then
                        echo "(failed)."
                        exit 1
                else
                        echo "${DAEMON} already running."
                        exit 0
                fi
        fi
        ;;
  stop)
        echo -n "Stopping ${DESC}: "
        if start-stop-daemon --stop --quiet --pidfile "${PIDFILE}" \
                --startas ${DAEMON} --retry 10 --name ${NAME} \
                >/dev/null 2>&1 ; then
                        echo "${NAME}."
        else
                if start-stop-daemon --test ${START} >/dev/null 2>&1; then
                        echo "(not running)."
                        exit 0
                else
```

```
                        echo "(failed)."
                        exit 1
                fi
        fi
        ;;
  restart|force-reload)
          $0 stop
        exec $0 start
        ;;
  *)
        echo "Usage: /etc/init.d/${NAME} {start|stop|restart|force-reload}" >&2
        exit 1
        ;;
esac
exit 0
```

Sada pokrenite *saslauthd:*

```
# /etc/init.d/saslauthd start
Starting SASL Authentication Daemon: changed ownership of
`/var/spool/postfix/var/run/saslauthd' to root:sasl
saslauthd.
```

Da biste provjerili rade li ispravno SMPT-AUTH i TLS zadajte sljedeću naredbu:

```
# telnet localhost 25
Trying 127.0.0.1...
Connected to localhost.localdomain.
Escape character is '^]'.
220 server1.centralsoft.org ESMTP Postfix (Debian/GNU)
```

Ovo će uspostaviti vezu s Postfixom. Sada upišite:

```
# echo localhost
```

Ako vidite redove:

```
server1:/etc/postfix# telnet localhost 25
Trying 127.0.0.1...
Connected to localhost.localdomain.
Escape character is '^]'.
220 server1.centralsoft.org ESMTP Postfix (Debian/GNU)
ehlo localhost
250-server1.centralsoft.org
250-PIPELINING
250-SIZE 10240000
250-VRFY
250-ETRN
250-STARTTLS
250-AUTH LOGIN PLAIN
250-AUTH=LOGIN PLAIN
250 8BITMIME
```

Vaša bi konfiguracija trebala raditi i završili ste s ovim dijelom konfiguriranja sustava za razmjenu pošte. Možete upisati **`quit`** i prijeći na sljedeću lekciju.

# Pokretanje Apache poslužitelja

Kako je spomenuto ranije u ovome poglavlju, u početnu konfiguraciju uključujemo Web poslužitelj zato što je vrlo važno da naučite osnove administriranja poslužitelja i zato što poslužitelj može biti izuzetno koristan i za druge alate. Na kraju ovog poglavlja iskoristit ćemo ga za prikazivanje statističkih podataka koje je generirao Webalizer.

Netcraft je u studenome 2006. godine objavio izvješće u kojem tvrdi da 60 posto Web lokacija koristi Apache. To je mnogo više od svih ostalih Web poslužitelja ukupno.

Apache je dobro integriran u većini Linux distribucija. U ovom ćemo odjeljku slijediti dobro poznatu shemu te instalirati i konfigurirati Apache koristeći ove naredbe:

```
# apt-get install apache2 apache2-doc
Setting up ssl-cert (1.0-11) ...
Setting up apache2-utils (2.0.54-5) ...
Setting up apache2-common (2.0.54-5) ...
Setting Apache2 to Listen on port 80. If this is not desired, please edit
/etc/apache2/ports.conf as desired. Note that the Port directive no longer
works.
Module userdir installed; run /etc/init.d/apache2 force-reload to enable.
Setting up apache2-mpm-worker (2.0.54-5) ...
Starting web server: Apache2.
Setting up apache2 (2.0.54-5) ...
Setting up apache2-doc (2.0.54-5) ...
```

Kada Debian završi instalaciju *apache httpd* poslužitelja, zadajte:

```
# apt-get install libapache2-mod-php4 libapache2-mod-perl2 \
php4 php4-cli php4-common php4-curl php4-dev php4-domxml \
php4-gd php4-imap php4-ldap php4-mcal php4-mhash php4-mysql \
php4-odbc php4-pear php4-xslt curl libwww-perl imagemagick
```

Ta naredba uzima i konfigurira 48 datoteka, tako da će njeno izvršavanje potrajati. Kada se završi, možete prijeći na sljedeći korak.

Promijenite direktivu `DirectoryIndex` u datoteci */etc/apache2/apache2.conf* s:

```
DirectoryIndex index.html index.cgi index.pl index.php index.xhtml
```

u:

```
DirectoryIndex index.html index.htm index.shtml index.cgi index.php
index.php3 index.pl index.xhtml
```

Nadalje, dodajte znak # kao što je prikazano da biste zakomentirali sljedeće redove u */etc/mime.types* datoteci:

```
#application/x-httpd-php                        phtml pht php
#application/x-httpd-php-source                 phps
#application/x-httpd-php3                       php3
#application/x-httpd-php3-preprocessed          php3p
#application/x-httpd-php4                       php4
```

Također je potrebno zakomentirati dva reda u */etc/apache2/mods-enabled/php4.conf:*

```
<IfModule mod_php4.c>
#AddType application/x-httpd-php .php .phtml .php3
#AddType application/x-httpd-php-source .phps
</IfModule>
```

Zatim provjerite jesu li sljedeća dva reda koda prisutna u datoteci */etc/apache2/ports.conf*. Ako je potrebno, dodajte ih:

```
Listen 80
Listen 443
```

Sada je potrebno osposobiti neke Apacheove module (SSL, *rewrite* i *suexec*) tako što ćete ih simbolički povezati s datotekama u poddirektoriju *mods-enabled*:

```
# cd /etc/apache2/mods-enabled
# ln -s /etc/apache2/mods-available/ssl.conf ssl.conf
# ln -s /etc/apache2/mods-available/ssl.load ssl.load
# ln -s /etc/apache2/mods-available/rewrite.load rewrite.load
# ln -s /etc/apache2/mods-available/suexec.load suexec.load
# ln -s /etc/apache2/mods-available/include.load include.load
```

Kao što ste vidjeli prilikom instaliranja drugih procesa ranije u ovome poglavlju, instaliranje odgovarajućih modula pomoću *apt-get* automatski pokreće Apache na sustavu. Budući da ste izmijenili konfiguraciju, trebate ponovno pokrenuti Apache da bi izmjene nastupile, bez potrebe za ponovnim pokretanjem poslužitelja. Unesite ovu naredbu:

```
# /etc/init.d/apache2 restart
```

Vaš će se Web poslužitelj ponovno pokrenuti te uključiti nove module i ostale izmjene konfiguracije.

## Dodavanje FTP servisa pomoću ProFTPD-a

Zajedno s *httpd* poslužiteljem za prikaz Web stranica, poželjet ćete implementirati i File Transfer Protocol (FTP) poslužitelj. Za tu namjenu koristit ćemo alat slobodnog izvora koda ProFTPD jer je popularan, siguran i može se konfigurirati na različite načine.

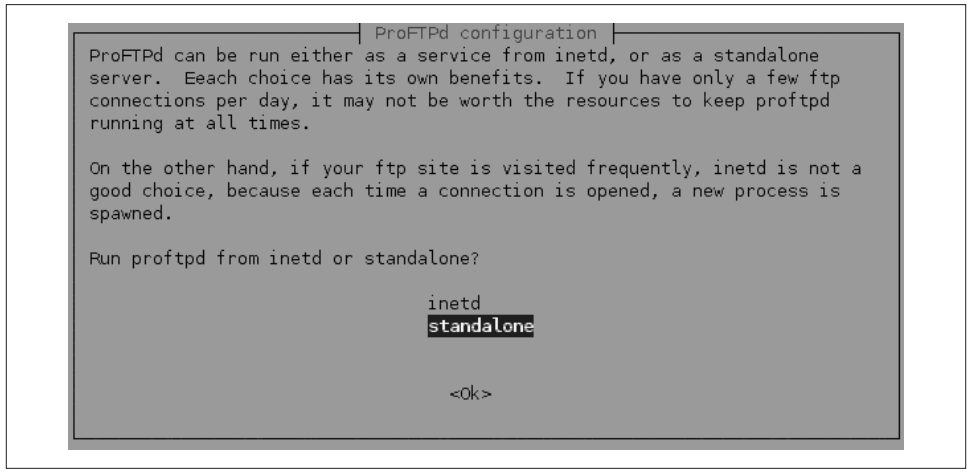
FTP poslužitelj koristi jednu, glavnu, konfiguracijsku datoteku, s direktivama i skupinama direktiva koje će razumjeti svaki administrator koji je ikada koristio Apache. ProFTPD ima po jednu konfiguracijsku datoteku *.ftpaccess* za svaki direktorij, slične Apacheovim *.htaccess* datotekama koje prisiljavaju korisnike da unesu svoj korisnički ID i lozinku kako bi pristupili određenim direktorijima.

ProFTPD omogućuje konfiguriranje više virtualnih FTP poslužitelja i anonimnih FTP servisa. Ne izvršava vanjske programe i izvršava se kao neprivilegirani korisnik.

Instalirajte ProFTPD koristeći ovu naredbu:

```
# apt-get install proftpd
```

Slika 2-9 prikazuje okvir koji ćete vidjeti kada Debian preuzme i započne instalirati ProFTPD. ProFTPD se može izvršavati samostalno ili kao *inetd* servis. Iz sigurnosnih ćemo razloga ProFTPD pokrenuti samostalno.




*Slika 2-9. Debianov konfiguracijski okvir za ProFTPd*

Nadalje, dodajte sljedeće redove u datoteku */etc/proftpd*.conf:

```
DefaultRoot ~
IdentLookups off
ServerIdent on "FTP Server ready."
```

Kao što smo radili i s ostalim procesima, ponovno pokrenite ProFTPd zadavanjem sljedeće naredbe:

```
# /etc/init.d/proftpd restart
```

# Rezimiranje statistike poslužitelja pomoću Webalizera

Webalizer generira statistička izvješća na temelju dnevničkih zapisa. Možete ga koristiti sa standardnim Web preglednikom, a daje detaljna, prilagodljiva izvješća o upotrebi u HTML formatu.

Debian u svojem repozitoriju uključuje Webalizer pa ga možete instalirati koristeći ovu naredbu:

```
# apt-get install webalizer
```

Za vrijeme instalacije morat ćete potvrditi direktorij instalacije *(/var/www/webalizer)*, zadati ime koje će se koristiti u naslovima statističkih izvješća (možete navesti ime domene, na primjer) te lokaciju dnevničke datoteke na Web poslužitelju (na našem sustavu je to */var/log/apache/access.log.1*):

```
Which directory should webalizer put the output in?
/var/www/webalizer
```

```
Enter the title of the reports webalizer will generate.
Usage Statistics forserver1.centralsoft.org
What is the filename of the rotated webserver log?
/var/log/apache/access.log.1
```

## Sinkroniziranje sistemskog sata

Satovi ugrađeni u računalo vrlo često kasne ili idu naprijed. Zato je osnovni zadatak povezati računalo s Network Time Protocol (NTP) poslužiteljem koji će održavati sat u računalu tako da ima otklon od samo nekoliko sekundi.

Kako biste sinkronizirali sistemsko vrijeme s NTP poslužiteljem, dodajte sljedeće naredbe u */var/spool/cron/crontabs/root:*

```
# update time with NTP server
0 3,9,15,21 * * * /usr/sbin/rdate 128.2.136.71 | logger -t NTP
```

Ako ta datoteka ne postoji, možete ju izraditi na ovaj način:

```
# touch /var/spool/cron/crontabs/root
```

IP adresa 128.2.136.71 pripada javnom vremenskom poslužitelju sveučilišta Carnegie Mellon.

Pomijenite dopuštenja za *crontab* zadavanjem:

```
# chmod 600 /var/spool/cron/crontabs/root
```

i ponovno pokrenite servis *cron* zadavanjem:

```
# /etc/init.d/cron restart
```

## Instaliranje Perl modula potrebnih za SpamAssassin

Mnogi alati oslanjaju se na Perl programski jeziku ili nude Perl sučelje koje možete upotrijebiti da biste ih prilagodili (iako i ostali jezici dobivaju sljedbenike u svijetu otvorenog izvornog koda i Unixa). SpamAssassin, vrlo važan alat za administratore poštanskih sustava (pa čak i za korisnike) koji ćemo koristiti u ovoj knjizi i koji se oslanja na Perl. Kao sistemski administrator, čak i ako ne želite programirati u Perlu, trebate znati preuzeti Perl module s vrlo popularnog i pouzdanog repozitorija Comprehensive Perl Archive Network (CPAN).

Da biste dobili osjećaj za instaliranje Perl modula, dodat ćemo ih sad nekoliko koristeći Perl CPAN školjku. Ta okolina služi za pretraživanje arhive i instaliranje modula s nje.

Prijavite se kao *root* i zadajte sljedeće naredbe kako biste pokrenuli Perl CPAN školjku:

```
server1:/home/admin# perl -MCPAN -e shell
/etc/perl/CPAN/Config.pm initialized.
```

Na sva pitanja odgovorite pritiskom na tipku Return da biste prihvatili podrazumijevane opcije. Tada zadajte sljedeće naredbe da biste instalirali module koje ćemo koristiti u sljedećem poglavlju:

```
> install HTML::Parser
> install DB_File
> install Net::DNS
```

Na upit `enable tests?` odgovorite `no`.

Ako modul već postoji na vašem sustavu, pojavit će se poruka poput `HTML::Parser is up to date`. Kada je modul uspješno instaliran, pojavit će se `/usr/bin/make install – OK`.

Kada završite, jednostavno upišite `q` kako bi napustili Perl i brzo se vratili do odzivnika sustava.

# Što slijedi

Kako ste završili s poslovima vezanim uz postavljanje poslužitelja, poželjet ćete ga koristiti u njegovom radnom režimu. Trebat ćete instalirati vlastite DNS usluge i obavijestiti registar gdje ste postavili svoju domenu (to je tema sljedećeg poglavlja). Kada završite s konfiguracijom DNS-a, možete instalirati Web aplikaciju (mi ćemo koristiti ISPConfig) i započeti istraživati kako rade Web aplikacije.

POGLAVLJE 3

# Domain Name System (DNS)

U ovom ćete poglavlju naučiti kako postaviti Domain Name System (DNS) pomoću BIND-a. Kada završite s ovim poglavljem, trebali biste znati instalirati, konfigurirati i održavati DNS poslužitelj te rješavati probleme na poslužitelju za bilo koju registriranu domenu. Započet ćemo s uvodom u DNS, koji možete preskočiti ako želite odmah prijeći na dio koji vas korak-po-korak vodi kroz proces instalairanja i konfiguriranja. Ako naiđete na probleme, vjerojatno ćete se vratiti i pogledati prethodna poglavlja.

## Osnove DNS poslužitelja

Ako budete malo istraživali članke o DNS sustavu na Internetu, vjerojatno ćete naići na tvrdnju da je DNS najveća svjetska baza podataka. Usporedba s drugim bazama podataka poput Oraclea ili MySQL-a može zavarati. U stvari, DNS je najveći distribuirani digitalni imenik na svijetu. Poput mrežnog telefonskog imenika, DNS poslužitelji pridružuju IP adrese mnogobrojnih poslužitelja spojenih na Internet, kako onih malih koji upravljaju malim Web lokacijama, tako i gigantskih poslužiteljskih farmi poput Googlea i Amazona, s njihovim imenima.

Poput javnih biblioteka u kojima su telefonski imenici kategorizirani po državama, i DNS grupira domene u kategorije. Glavna kolekcija kategorija nalazi se u *korijenskim* imenicima. Ta je kolekcija podijeljena na domene najviše razine, na sličan način kao što je telefonski imenik podijeljen na države. Umjesto da traži telefonski broj s pozivnim brojem npr. New Yorka, DNS traži imena domena koja završavaju sufiksima poput *.edu, .org, .com, .net, .mil., .de, .fr* itd. Domene unutar svake domene najviše razine na kraju će vas dovesti do adrese koju možete koristiti za komunikaciju s poslužiteljem.

DNS sustav (izvorno definiran u RFC dokumentu 882 iz 1983. godine i kasnije revidiran u RFC dokumentima 1034 i 1035) donio je mnoštvo ideja kako upravljati preslikavanjem uobičajenih imena u IP adrese. Sistem hijerarhijski distribuira podatke i imena računala u *prostor imena domena*. Domene su slične granama stabla, a svaka grana ima podgrane. Programi pod nazivom *poslužitelji imena* (engl. *nameservers*) pružaju informacije

o svojim dijelovima stabla, dok programi koje nazivamo *razriješivači* (engl. *resolvers*) traže od poslužitelja imena informacije o domeni na zahtjev klijentskih programa.

Sheme hijerarhijskog imenovanja, poput DNS-a, sprječavaju dupliciranje podataka. Svaka je domena jedinstvena i unutar domene možete imati koliko god želite poslužitelja – jednostavno dodajte njihova imena ispred imena domene. Na primjer u okviru domene *centralsoft.org* može postojati bilo koji broj računala s imenima poput *server1.centralsoft.org*, *ldap.centralsoft.org*, *mail.centralsoft.org* itd.

## Prednosti lokalne DNS administracije

Manje organizacije obično prepuštaju pružateljima Internet usluga da upravljaju potrebnim DNS podacima. No, postavljanje vlastitog poslužitelja ima svoje prednosti. To vam daje potpunu kontrolu nad raspoređivanjem javno dostupnih usluga (npr. Web servisa i elektroničke pošte) na računala. Uključivanje DNS-a u vašu infrastrukturu pruža vam i mogućnost proširivanja: po potrebi možete dodavati poslužitelje i čak raspoređivati opterećenje između njih. To postaje bitno ako imate više aktivnih domena ili interni servis provjere identiteta. Također ćete imati veću kontrolu nad ažuriranjem imena. Ukratko, u današnjem poslovnom okruženju vrijedno je imati kontrolu nad svojim DNS-om, umjesto da to prepustite nekom drugom.

Mnoge kompanije učinile su svoje središnje poslovne aplikacije dostupnim preko Weba. Umjesto da zamijene postojeće poslovne aplikacije, one ih žele staviti na raspolaganje kroz nova i elegantna Web sučelja. To rade dodavanjem Web sučelja, dok u pozadini stoje Web aplikacije koje povezuju različite sustave. IT odjeli za te pozadinske poslove koriste aplikacijske poslužitelje kao što su JBoss (sada u vlasništvu tvrtke Red Hat), IBM-ov WebSphere, WebLogic tvrtke BEA te brojne druge proizvode za implementiranje sučelja. U svakom slučaju, DNS postaje integralni dio cjelokupnog sustava koji omogućava poslovanje preko Weba jer se u njima intenzivno koriste.

DNS poslužitelji imaju istaknuto mjesto u novim Web uslugama i „izvršnom Internetu“, kada osobe koriste aplikacije poput onih koje nude Google, Yahoo! i druge tvrtke. Brzo i pouzdano pronalaženje IP adresa od kritične je važnosti za uspjeh takvih aplikacija na Internetu, ali i unutar poduzeća. Možete smatrati da je konfiguriranje i održavanje DNS poslužitelja jedna od najvrjednijih administratorskih vještina koje možete imati.

Što onda trebate raditi kao sistemski administrator kako biste vodili vlastiti javni DNS poslužitelj? Morate pružiti adrese dva ili više takvih poslužitelja vaše registru domene (neophodne su najmanje dvije adrese jer se tako osigurava da bar jedan poslužitelj bude operativan kada korisnik zatraži adresu poslužitelja kojem želi pristupiti). Također morate upravljati imenima domena sistema – Web poslužitelja, poslužitelja elektroničke pošte itd. – za koje želite da budu javno dostupni.

Kad počnete učiti o DNS sustavu, otkrit ćete da uopće nije intuitivan. Ponekad stručni izrazi djeluju poput stranog jezika. Neće vam se činiti da ima previše smisla sve dok ga ne počnete koristiti malo više vremena. Pokazat ćemo vam kako u vrlo kratkom vremenu izgraditi DNS poslužitelj. Zatim ćemo detaljnije obraditi neke osnovne koncepte i pojmove prije nego što se posvetimo konfiguracijskim datotekama.

# BIND

Većina DNS poslužitelja temelji se na Berkley Internet Name Daemonu, ili BIND-u. BIND je standard na svakoj inačici Unixa i Linuxa. Kako će se administratori gotovo sigurno susresti s njim, u ovom poglavlju ćemo ga detaljno obraditi.

Najpopularnija alternativa BIND-u je paket *djbdns*. Radi dobro, koristi se na mnogim velikim poslužiteljima imena i jednostavnije se konfigurira (što je diskutabilno). Za više detalja pogledajte *http://cr.yp.to/djbdns*.

Nećemo mnogo govoriti o povijesti BIND-a jer bismo vas tako samo uspavali. No, moramo naglasiti jednu važnu povijesnu činjenicu. Neki još uvijek koriste zastarjelu inačicu BIND-a: inačicu 4. U ovome ćemo poglavlju koristiti inačicu 9.

Ako radite na sustavu s DNS konfiguracijskim datotekama koje imaju drugačiju sintaksu od sintakse pokazane u ovome poglavlju, to je vrlo vjerojatno zbog toga što vaš sustav koristi BIND 4. Kako smo ranije napomenuli, u poslovnim okruženjima postojeći sustavi se rijetko mijenjaju i vjerojatno se mora dogoditi elementarna nepogoda da bi IT odjel nadogradio BIND sa inačice 8 na inačicu 9. Ali zbog sigurnosnih propusta koje ima BIND 4, svakako biste ga trebali nadograditi na noviju inačicu. (Usput da kažemo, nakon inačice 4 objavljena je inačica 8 BIND-a kako bi numeriranje bilo usklađeno s inačicama Sendmaila. Stoga, nemojte nikome dopustiti da vam proda BIND 5, 6 ili 7).

## Komponente BIND-a

BIND ima tri komponente. Prva je servis koji pokreće dio DNS-a vezan uz odgovaranje na zahtjeve. Ta se komponenta još zove *named*. Ona se javlja na telefon kada zazvoni.

Druga komponenta u BIND paketu je biblioteka *razrješivača* (engl. *resolver)*. To je dio koji Web preglednik, program za rad s elektroničkom poštom i ostale aplikacije konzultiraju kada pokušavaju u internetskoj džungli pronaći poslužitelj po njegovom imenu.

Neki stručnjaci doživljavaju razrješivač kao klijent unutar BIND-a. Za razliku od poslužitelja, klijent nije samostalan program; u stvari to je biblioteka povezana sa svakim Web preglednikom, klijentom elektroničke pošte itd. Kôd razrješivača šalje upite DNS poslužiteljima kako bi preveo imena u IP adrese.

Taj dio BIND-a koristi svoj mali imenik koji se zove *resolv.conf* i nalazi se na svakom računalnom sustavu. Vaš je zadatak konfigurirati *resolv.conf*. Evo kako izgleda *resolv.conf* na računalima na *centralsoft.org* domeni.

```
search centralsoft.org
nameserver 70.253.158.42
nameserver 70.253.158.45
```

Kao što možete vidjeti, konfiguracijska datoteka BIND razrješivača je jednostavna. Prvi red koda traži poslužitelj na lokalnoj domeni. Sljedeći red daje adrese poslužitelja imena za koja znaju administratori, a na koje se razrješivač može osloniti ako inicijalna potraga ne da rezultat.

Treća komponenta BIND-a pruža alate poput naredbe *dig* za testiranje DNS-a. Otvorite konzolu, upišite *dig yahoo.com* (ili navedite neku drugu domenu koju poznajete) i pogledajte što se događa. Kasnije ćemo detaljnije razmotriti *dig* i ostale korisne alate.

# Postavljanje DNS poslužitelja

Kako bismo izgradili poslužitelj, koristit ćemo svježu instalaciju posljednje stabilne inačice Debiana i konfigurirati ga s minimalnim brojem paketa.

Ako nemate disk za mrežnu instalaciju koji smo koristili u drugom poglavlju, preuzmite ga s *http://www.us.debian.org/CD/netinst*. Izvedite mrežnu instalaciju i navedite potpuno kvalificirano ime domene. Zatim konfigurirajte Debian kako smo predložili.

Kada nabavite aktualno izdanje Debiana GNU/Linuxa, možda ćete primijetiti razliku između njega i inačice na temelju koje smo pisali ovaj priručnik. Razvojni timovi često ažuriraju svoje distribucije, a i s novim inačicama, zakrpama i jezgrama, mijenjaju se i instalacijske procedure. Ako otkrijete razlike u instalacijskim procedurama koje smo opisali, obratite pažnju na bitne točke koje smo objasnili i ne biste trebali imati probleme s najnovijim izdanjima.

Nakon početnih faza instalacije Debiana vidjet ćete upit o izboru željenog tipa instalacije. Zaslon će izgledati ovako:

```
( ) Desktop Environment
( ) Web Server
( ) Print Server
( ) DNS Server
( ) File Server
( ) Mail Server
( ) SQL database
( ) manual package selection
```

Nemojte izabrati niti jednu opciju; samo pritisnite tipku Tab. Pritisnite označeni OK gumb i Debianov program za instaliranje će započeti s preuzimanjem i instaliranjem paketa.

Za vrijeme preuzimanja pojavit će se još jedan zaslon. On će vas upitati želite li konfigurirati Exim (*Exim-config*). Izaberite „no-configuration“. Zatim će vas upitati „Really leave the mail system unconfigured?“. Odgovorite sa „Yes“.

Kada završite s minimalnom Debianovom instalacijom trebali biste ukloniti neke nepotrebne programe koji se koriste u lokalnim mrežama, no ne pripadaju internetskom poslužitelju elektroničke pošte. Možete ih izbrisati koristeći Debianov pomoćni program *apt-get*:

```
# apt-get remove lpr nfs-common portmap pidentd pcmcia-cs pppoe \
pppoeconf ppp pppconfig
```

Ako ste odlučili umjesto Debiana koristiti SUSE ili Fedora Linux, te pakete možete izbrisati pomoću metode koju preferirate.

Hajdemo sada izrezati neke servisne skripte i ponovno pokrenuti *inetd*:

```
# update-inetd --remove daytime
# update-inetd --remove telnet
# update-inetd --remove time
# update-inetd --remove finger
# update-inetd --remove talk
# update-inetd --remove ntalk
# update-inetd --remove ftp
# update-inetd --remove discard
# /etc/init.d/inetd reload
```

Kako biste instalirali BIND na vaš Debian poslužitelj, zadajte sljedeću naredbu:

```
# apt-get install bind9
```

Debian će preuzeti datoteku i konfigurirati je kao Internet servis. Na konzoli ćete vidjeti sljedeće poruke:

```
Setting up bind9 (9.2.4-1)
Adding group `bind' (104)
Done.
Adding system user `bind'
Adding new user `bind' (104) with group `bind'.
Not creating home directory.
Starting domain name service: named.
```

## Sigurnosna primjena okoline chroot

Mnogi administratori preporučuju da se BIND-a pokreće kao poseban korisnik (nikako ne *root*) u izoliranom direktoriju – *chroot okruženju*. Ovaj pristup štiti vas od mogućnosti da se otkrije sigurnosna pukotina unutar BIND-a, koja bi potencijalno omogućila napad na servis *named* i prodor u sustav. Čak i ako je servis *named* već napadnut i zloupotrijebljen, chroot okruženje ograničava štetu koja bi mogla biti učinjena DNS servisu.

Da biste stavili BIND u *chroot* okolinu, morate prvo izraditi direktorij gdje se servis može odvijati odvojeno od ostalih procesa. Izvršavat će se kao neprivilegirani korisnik i uz to će samo *root* korisnik moći pristupiti direktoriju. Taj će direktorij sadržavati sve datoteke koje su potrebne BIND-u i izgledat će mu kao cjelokupni datotečni sustav nakon što zadate naredbu *chroot*.

Najprije zaustavite servis koristeći ovu naredbu:

```
# /etc/init.d/bind9  stop
```

Nadalje uredite datoteku */etc/default/bind9* tako da se pozadinski servis izvršava kao neprivilegirani korisnik *bind*, preusmjeren na */var/lib/named*. Promijenite red:

```
OPTS="-u bind"
```

tako da glasi:

```
OPTIONS="u-bind -t /var/lib/named"
```

Kako biste pružili kompletnu okolinu za pokretanje BIND-a, izradite neophodne direktorije pod */var/lib*:

```
# mkdir -p /var/lib/named/etc
# mkdir /var/lib/named/dev
# mkdir -p /var/lib/named/var/cache/bind
# mkdir -p /var/lib/named/var/run/bind/run
```

Zatim premjestite direktorij *config* iz */etc* u */var/lib/named/etc*:

```
# mv /etc/bind/var/lib/named/etc
```

Sada izradite simbolički link za novi *config* direktorij sa stare lokacije da biste izbjegli probleme kada u budućnosti budete nadograđivali BIND:

```
# ln -s /var/lib/named/etc/bind /etc/bind
```

Pripremite uređaje *null* i *random* za upotrebu s BIND-om i uredite dopuštenja za direktorije:

```
# mknod /var/lib/named/dev/null c 1 3
# mknod /var/lib/named/dev/random c 1 8
```

Zatim promijenite dopuštenja i vlasništvo na datotekama:

```
# chmod 666 /var/lib/named/dev/null /var/lib/named/dev/random
# chown -R bind:bind /var/lib/named/var/*
# chown -R bind:bind /var/lib/named/etc/bind
```

Također ćete morati promijeniti i početnu skriptu */etc/init.d/sysklogd* tako da i dalje možete vidjeti poruke u sistemskom dnevniku događaja. Izmijenite red:

```
SYSLOGD=""
```

tako da glasi:

```
SYSLOGD="-a/var/lib/named/dev/log"
```

Sada ponovno pokrenite sustav za bilježenje u dnevnik pomoću ove naredbe:

```
# /etc/init.d/sysklogd restart
```

Vidjet ćete sljedeću poruku:

```
Restarting system log daemon: syslogd.
```

Napokon pokrenite BIND:

```
# /etc/init.d/bind9 start
```

Provjerite jesu li u */var/log/syslog* zabilježene bilo kakve pogreške. Možete se kretati kroz datoteku zadavanjem:

```
# less /var/log/syslog
```

Ako datoteka *syslog* pokaže sljedeće, znat ćete da je BIND uspješno pokrenut:

```
Starting domain name services: named.
```

Nažalost *named* se pokrenuo, no ne može učitati inicijalne datoteke s podacima što ga čini nefunkcionalnim. Provjerite funkcionira li *named* zadavanjem:

```
# rndc status
number of zones: 6
debug level: 0
xfers running: 0
xfers deferred: 0
soa queries in progress: 0
query logging is OFF
server is up and running
server1:/home/admin#
```

Ako DNS ne radi ispravno, umjesto toga vidjet ćete nešto ovakvo:

```
# rndc status
rndc: neither /etc/bind/rndc.conf nor /etc/bind/rndc.key was found
```

Ako vam se pojavi ova pogreška, pogledajte odjeljak „Neuspjelo povezivanje preko rndc-a“ pri kraju ovog poglavlja.

## Konfiguriranje pouzdanog DNS poslužitelja

Ako želite pronaći telefonski broj od Jane Doe u digitalnom telefonskom imeniku, to možete učiniti jer telefonska kompanija objavljuje takve informacije. No, ako želite pronaći *janedoe.com,* sistemski administrator mora uzeti ime domene i IP adresu i učiniti ih sastavnim dijelom distribuiranog DNS imenika. Administratori to rade stvarajući ispise koje stručnjaci za DNS zovu *zonske datoteke*.

*Zona* sadrži informacije za domenu ili, ako nastavimo s prijašnjom analogijom telefonskog imenika, za kućanstvo. Zamislite da u vašoj kući živi petnaestero djece i nazove netko tko treba jedno od njih. Svako dijete ima svoj mobilni telefon, no ne znate napamet sve brojeve. Ali, imate vlastiti popis, imenik u koji možete pogledati kako biste pronašli telefonski broj djeteta.

Slično tome, možete imati 15 poslužitelja smještenih u jednom podatkovnom središtu, ili 15 Web lokacija udomljenih na vašem poslužitelju. Radi bolje ilustracije recimo da administrirate poslužitelj koji udomljuje pet različitih Web lokacija, od kojih svaka ima drugačije ime domene. Pretpostavimo da se jedna zove *centralsoft.org*, a ostale *linhelp.com, supportcall.org, jdshelp.net* i *linuxconf.net*. Svi vlasnici Web lokacija zamolili su vas da upravljate njihovim DNS zapisima. BIND-ova svestranost omogućava da upravljate s nekoliko DNS poslužitelja odjednom te da neovisno upravljate s više domena na jednom poslužitelju.

Svaka Web lokacija je u drugoj domeni, tako da morate pisati zonsku datoteku za svaku od njih. U bazi podataka registratora domene vaš će DNS poslužitelj biti naveden kao *poslužitelj imena* za navedena imena domena. Drugim riječima, *server1.centralsoft.org* će biti naveden kao „osoba“ koju treba kontaktirati da bi se saznao „broj telefona djeteta koje živi u kući“ (*linhelp.com, supportcall*.org itd.).

Datoteka koja odgovara popisu brojeva mobilnih telefona djece jeste */etc/named.conf*. To znači da je */etc/named.conf* imenik zonskih datoteka jer vam pruža informacije o lokacijama svih zona na vašem sustavu.

## Vaša odgovornost u DNS sustavu

Kako je prethodno navedeno, DNS distribuira imenike. Kada platite i registrirate domenu, jedno od pitanja na koja morate odgovoriti vezano je uz poslužitelje imena. Morate dati imena i adrese dva poslužitelja imena koji su registrirani u DNS sustavu.

Sada imati približnu sliku o tome što radi administrator sustava. Sve poslužitelje imena pod vašom domenom morate konfigurirati sukladno specifikacijama koje je postavila organizacija Internet Engineering Task Force (IETF). Ako ne slijedite zadane protokole, vaš sustav neće postati dio univerzalnog sustava imenika.

Nadamo se da vam je prethodno izlaganje predočilo DNS. Sada je vrijeme za detaljniji pogled na funkcioniranje vašeg imenika.

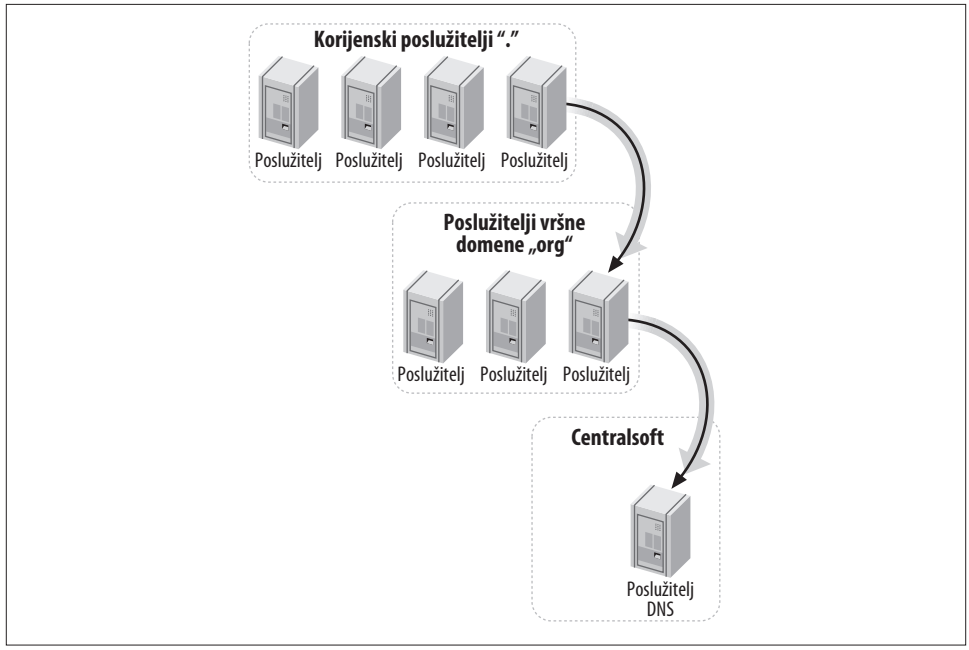
## Distribuirana metoda razrješavanja imena domene

Pogledajmo još jednom strukturu DNS imenika. Imenik ima tri razine. Prva skupina poslužitelja zovu se *korijenski* poslužitelji jer predstavljaju početnu točku za upite. Druga se skupina sastoji od poslužitelja za *vršne domene*. Vršne domene uključuju *.com, .net, .org, .mil, .gov, .edu* itd. te domene država poput *.de* (U imenima domena se ne razlikuju velika i mala slova: .com i .COM predstavljaju isto).

Slika 3-1 ilustrira strukturu DNS sustava. Na vrhu slike možete vidjeti korijenske poslužitelje. Oni sadrže samo imena i IP adrese poslužitelja sljedeće razine i odgovorni su isključivo za preusmjeravanje zahtjeva prema određenom poslužitelju vršne domene.

Na sredini slike možete vidjeti poslužitelje imena za domenu .org. Ti poslužitelji sadrže imena i IP adrese svih registriranih DNS poslužitelja sa sufiksom .org. Ako registrirate domenu sa sufiksom .org, njezina IP adresa će se nalaziti na svakom poslužitelju imena za domenu .org. Vi morate osigurati informacije o preostalim poddomenama, uključujući i poslužitelje unutar vaše domene.

Na dnu slike 3-1 možete vidjeti primarni poslužitelj imena koji se zove *server1.centralsoft.org.* On je odgovoran za mnoge domene, kako ćete vidjeti kasnije. Za sada, dovoljno je da znate da *server1.centralsoft.org* predstavlja dio DNS sustava kojim ćete naučiti upravljati.




*Slika 3-1. Distribuirana struktura DNS imenika*

## Pronalaženje domene

Kao što smo spomenuli ranije, osim što osigurava pozadinski servis za upisivanje DNS zapisa u distribuirani imenik, BIND pruža i mehanizam za čitanje iz imenika. Kada računalo treba pronaći adresu Web lokacije, šalje upit zadanim DNS poslužiteljima (koji su obično smješteni na lokalnoj mreži ili kod pružatelja Internet usluga preko kojeg ste spojeni na Internet).

Recimo da vaš preglednik želi pronaći www.google.com. BIND-ov „klijent" izvršava naredbu koja u biti pita DNS poslužitelj zna li za adresu Web lokacije. Ako DNS ne zna za tu adresu, on dalje pita korijenski poslužitelj.

Korijenski poslužitelj odgovara „Ne znam, ali znam gdje možeš pronaći odgovor. Započni s poslužiteljem imena za vršnu domenu .com." i daje IP adresu poslužitelja koji zna sve domene (a ima ih puno!) registrirane pod *.com*.

Na zahtjev vašeg preglednika, razrješivač na DNS poslužitelju tada pita *.com* poslužitelj za adresu. Poslužitelj *.com* odgovara „Nemam tu informaciju, no znam za ime poslužitelja koji zna. Njegova adresa je 64.233.167.99 i ime mu je *ns1.google.com*.".

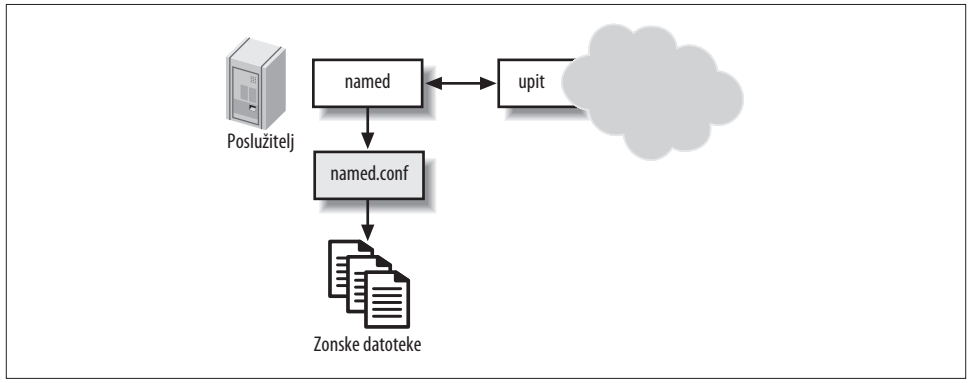
DNS poslužitelj odlazi do adrese, čita informacije iz imenika koji pruža ns1.google.com i vraća se kako bi vašem pregledniku javio adresu od www.google.com. DNS poslužitelj zatim sprema tu informaciju u svoju privremenu memoriju tako da ju neće opet morati tražiti od Googlea.

U biti *resolv.conf* kontrolira upite o adresama imena domena koje preglednici i ostali klijenti upućuju, dok komponenta *named* odgovara na upite i osigurava da informacije budu ažurirane na svim poslužiteljima.

## Odgovaranje na upite

Slika 3-2 ilustrira proces koji se koristi za odgovaranje na upite. Pogledajmo kako radi.

U gornjem lijevom uglu slike nacrtan je poslužitelj (u našem primjeru taj se poslužitelj zove *server1.centralsoft.org* i on obavlja istu funkciju kao i *ns1.google.com*). Pretpostavimo da se na poslužitelju izvršavaju Linux i BIND. Poslužitelj na višoj razini usmjerava razrješivače prema sustavu (u slučaju *server1.centralsoft.org* zahtjeve šalje poslužitelj imena vršne domene *.org*).




*Slika 3-2. Odgovaranje na upite*

Servis *named* osluškuje UDP ulaz 53 i očekuje zahtjeve za imenima u domeni. Kada zaprimi zahtjev, komponenta *named* se konzultira sa svojom konfiguracijskom datotekom */etc/named.conf*. Ako poslužitelj posjeduje informacije o domeni iz upita, pretražuje odgovarajuću *zonsku datoteku*. Ako *zonska datoteka* posjeduje traženu informaciju, poslužitelj ju prenosi sustavu koji ju je tražio.

Neki korisnici konfiguracijske datoteke tretiraju kao *datoteke pravila* (engl. *rule file*). To ima smisla zato što ispravno funkcioniranje DNS sustava zahtijeva čvrsto pridržavanje pravila i protokola. Međutim, *zonske datoteke* funkcioniraju kao dio DNS imenika. Njihova je primarna zadaća da pružaju informacije, a ne da postavljaju pravila.

## Primarni i sekundarni DNS poslužitelj

Kako smo već ranije spomenuli, trebate navesti imena najmanje dva DNS poslužitelja kada registrirate domenu. Ako baš želite, možete izraditi kopiju svih informacija s prvog poslužitelja i smjestiti ih na drugi poslužitelj. Neki administratori ili organizacije to rade, no mnogo uobičajenija i opravdanija praksa je postaviti jedan poslužitelj

kao *primarni* ili *master* poslužitelj (na kojem ćete sve promjene unositi ručno) i drugi kao *sekundarni* ili *slave* poslužitelj. BIND zatim dopušta sekundarnom poslužitelju da kontaktira primarni i automatski reproducira imenik – to se još naziva *prijenos zone* (engl. *zone transfer*).

Sekundarni poslužitelji su pouzdani jednako kao i primarni poslužitelji. Sekundarni poslužitelji mogu odgovarati na upite i pružati informacije za sve zone za koje su odgovorni. Razlika je u tome što, kada unosite izmjene, trebate ih učiniti samo na primarnom poslužitelju. Sekundarni će poslužitelj te izmjene dobiti od primarnog.

Primarni poslužitelj neće odmah sekundarnome poslati novu konfiguraciju. Umjesto toga sekundarni poslužitelj u regularnim intervalima proziva primarni da mu pošalje izmjene, ako su izvedene. Sekundarni poslužitelj zna da mora prozvati starijeg brata zato što je označen izrazom *slave* u datoteci *named.conf,* kao što je ovdje prikazano:

```
zone "centralsoft.org" {
     type slave;
     file "sec.centralsoft.org";
     masters { 70.253.158.42; };
};
```

Nećemo sada razmatrati kompletnu sintaksu i ulogu ovog unosa. Ono što je važno naglasiti je red `type slave;` u kojem je poslužitelj definiran kao sekundarni i red `masters` koji navodi primarne poslužitelje. U ovome primjeru primarni poslužitelj se nalazi na IP adresi 70.253.158.42. Ta se adresa slaže s našim prethodnim unosom u *resolv.conf* datoteci (pogledaj odjeljak „Komponente BIND-a“). Datoteka *resolv.conf* pomaže klijentu da se spoji s DNS poslužiteljem, dok prethodni unos u datoteci *named.conf* pomaže sekundarnom DNS poslužitelju da pronađe primarni poslužitelj.

### Problemi s vatrozidom

Ako na primarnom poslužitelju imate postavljen vatrozid, odblokirajte ulaz 53. On se koristi za primanje i slanje upita. Ako se s druge strane vatrozida nalazi sekundarni poslužitelj, morate i tamo odblokirati TCP ulaz 53. Sekundarni poslužitelji koriste i TCP i UDP za prijenos zona koji je potreban kako bi poslužitelji uvijek bili ažurni.

Označavanje sekundarnog poslužitelja sa `slave` daje mu instrukcije da periodički provjerava kod primarnog poslužitelja jesu li se u međuvremenu dogodile kakve izmjene u glavnom imeniku domene. Datoteka *named.conf* na svakom poslužitelju zadaje kako će izvoditi prozivanje i prijenos zona. Vrijednost *refresh* govori sekundarnom poslužitelju koliko često treba provjeravati stanje na primarnom poslužitelju. *Serial number* je vrijednost koju morate povećati na primarnom poslužitelju, svaki put kada promijenite informaciju koju nudi. Sekundarni poslužitelj uspoređuje primarnu vrijednost sa vlastitom kako bi utvrdio treba li izvesti prijenos zona.

Primarna konfiguracijska datoteka također zadaje vrijednost *retry* koju sekundarni poslužitelj koristi umjesto vrijednosti *refresh* ako ne može pristupiti primarnom poslužitelju. To se može dogoditi ako se primarni poslužitelj blokira. U tome slučaju sekundarni poslužitelj privremeno preuzima ulogu primarnog.

Sekundarni se poslužitelj ne može zauvijek pretvarati da je primarni. Na kraju, njegove informacije mogu postati toliko zastarjele da je bolje da uopće ne odgovara na upite. Stoga konfiguracijska datoteka zadaje i vrijeme *expiry*. Ako to vrijeme prođe bez uspješnog ažuriranja, sekundarni će poslužitelj nastaviti pokušavati kontaktirati primarni poslužitelj, no odbijat će odgovarati na upite.

Postoji još jedna vrijednost koju morate poznavati prije nego što se upustite u dublje proučavanje konfiguracijskih datoteka: *minimum time to live* (TTL). Kada udaljeni DNS poslužitelj primi odgovor na upit, on ga sprema u privremenu memoriju i ponovno koristi u periodu zadanom u vrijednosti TTL. Spremanje u privremenu memoriju je ključno u funkcioniranju DNS-a. Zahvaljujući tome, ako netko provede jedan sat pregledavajući različite Web stranice na vašoj lokaciji (što može uključivati višestruke upite), poslužitelj blizu korisnika trebat će samo jednom pitati za ime domene. Nakon toga će na svaki upit odgovarati iz privremene memorije. Kako bi se izbjegla mogućnost da pohranjene informacije postanu zastarjele, vrijednost TTL osigurava da poslužitelj na kraju odbaci spremljene vrijednosti i preuzme nove.

Sve te vrijednosti vidjet ćete u zonskoj datoteci, a ne u *named.conf* datoteci. Datoteka *named.conf* pokazuje na lokaciju vaše zonske datoteke.

## DNS poslužitelji samo sa privremenom memorijom

Osim primarnih i sekundarnih DNS poslužitelja, postoje i poslužitelji koji samo spremaju u privremenu memoriju (engl. *caching only servers*). Administratori ih koriste kako bi smanjili opterećenje glavnog poslužitelja. Poslužitelji sa privremenom memorijom nemaju ovlasti. Oni samo ubrzavaju DNS sustav tako što spremaju imena domena koja su preuzeli od nadređenih poslužitelja i nude ih klijentima..

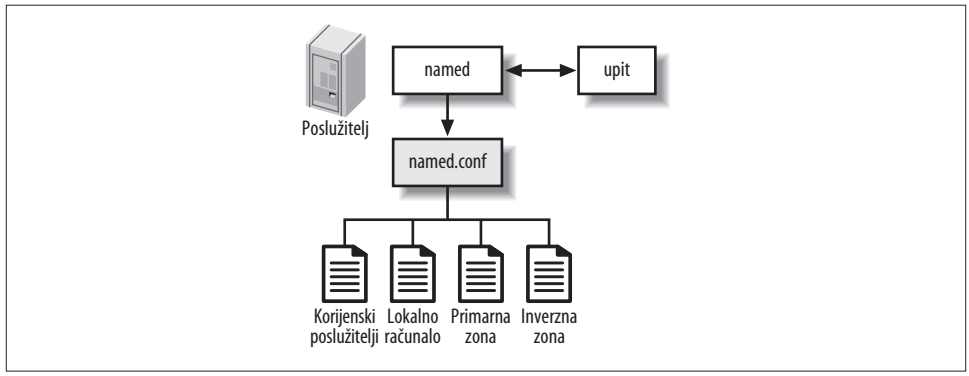
Poslužitelj koji ste postavili da udomljuje domene obično je zauzet odgovaranjem na upite drugih DNS poslužitelja na Internetu. Obavljanje samo tog posla troši resurse, tako da administratori često koriste poslužitelje sa privremenom memorijom za lokalnu pohranu informacija i njihovu isporuku. Kod pružatelja Internet usluga ovakvi poslužitelji se koriste za posluživanje klijenata. Drugi poslužitelj će se koristiti za pružanje imena domena za Web lokacije koje su kod njega udomljene.

Prilikom instalacije BIND-a podrazumijevano se postavlja i poslužitelj sa priručnom memorijom. Kada zadate upit, poslužitelj ga sprema u privremenu memoriju. Kada sljedeći put zatražite podatke za istu Web lokaciju nećete trebati prolaziti kroz čitav proces traženja: informacije o IP adresi dobit ćete iz privremene memorije.

## Uređivanje konfiguracijskih datoteka

Do sada smo istražili sustav Domain Name System i objasnili dijelove koje trebate održavati. Sada je potrebno detaljnije razmotriti konfiguracijske datoteke, tako da ih možete pisati, mijenjati i popravljati u slučaju potrebe.

Kada instalirate BIND instalirat će se i konfiguracijske datoteke. Nećete ih morati sami pisati. Slika 3-3 prikazuje osnovne datoteke. Započet ćemo s datotekom *named.conf* koja koordinira čitav sustav na svakom BIND poslužitelju i pokazuje na ostale.




*Slika 3-3. BIND konfiguracijske datoteke*

## named.conf

Podsjetite se da smo u dijelu „Odgovaranje na upite“ govorili o tome da, kada komponenta *named* zaprimi zahtjev, konzultira svoj mali imenik – konfiguracijsku datoteku *named.conf*. To usmjerava komponentu *named* prema zonskoj datoteci za traženu domenu.

Pogledajmo jednostavanu *named.conf* datoteku. Ako ju još uvijek ne možete razumjeti, barem ćete se upoznati s njezinim izgledom. Uskoro ćemo ju podijeliti na komponente.

Podsjetite se da se ta datoteka podrazumijevano instalirana na Linux poslužitelj. Ovisno o distribuciji. *Named.conf* može se nalaziti u različitim direktorijima (u BIND-u 9 pod Debianom smještena je u */etc/bind/named.conf*) i neznatno se razlikovati. Ta datoteka ponekad može biti puna komentara. Ovdje dajemo primjer datoteke *named.conf*. Komentari se nalaze nakon dvostruke kose crte:

```
options {
     pid-file "/var/run/bind/run/named.pid";
     directory "/etc/bind";
     // query-source address * port 53; };
```

```
//
// a master nameserver config
//
zone "." {
     type hint;
     file "db.root";
};
zone "0.0.127.in-addr.arpa" {
     type master;
     file "db.local";
};
zone "158.253.70.in-addr.arpa" {
     type master;
     file "pri.158.253.70.in-addr.arpa";
};
zone "centralsoft.org" {
     type master;
     file "pri.centralsoft.org";
};
```

### Elementarna sigurnost u prijenosu podataka

U našoj trenutnoj konfiguraciji svakom poslužitelju imena dopušteno je da prenese *centralsoft.org* zonu s našeg primarnog poslužitelja. Kako želimo da samo sekundarni poslužitelj imena (70.253.158.45) može prenositi zone, moramo dodati sljedeći red koda u zonu *centralsoft.org* koja se nalazi u datoteci *named.conf* na našem primarnom poslužitelju imena *server1.centralsoft.org*:

```
allow-transfer {70.253.158.45; }
```

Zona treba izgledati ovako:

```
zone "centralsoft.org" {
     type master;
     file "pri.centralsoft.org";
     allow-transfer { 70.253.158.45; };
};
```

Primjer datoteke *named.conf* oslanja se na četiri druge kofiguracijske datoteke. Treći red koda navodi mapu */etc/bind* u kojoj se nalaze konfiguracijske datoteke.

Iskaz `options` sastoji se od dva reda koda. Prvi pokazuje lokaciju datoteke *named.pid* koja sadrži identifikator procesa *named* koji se izvršava. Možda vam to izgleda kao suvišna informacija, no potrebna je nekim pomoćnim programima koji moraju prekinuti ili ponovno pokrenuti *named*. Još je značajniji drugi red iskaza koji zadaje direktorij u kojem je pohranjena komponena *named* i datoteke važne za njeno izvršavanje.

Sljedeći iskazi zone, čije smo primjere vidjeli ranije, identificiraju lokacije više datoteka koje sadrže konfiguracijske informacije. Ukratko, *named.conf* treba pokazivati na sljedeće datoteke u iskazima zone:

*Datoteka korijenskih poslužitelja (za* `zone "."`*)*
: Ova datoteka sadrži imena i adrese korijenskih poslužitelja na Internetu. *named* mora poznavati adrese tih poslužitelja tako da može zadati upit kada niti jedna komponenta tražene domene nije smiještena u privremenoj memoriji *named*.

*Datoteka za lokalno računalo (za* `zone "0.0.127.in-addr.arpa"`*)*
: Ova datoteka predstavlja vaš sustav (IP adresa 127.0.0.1). Glavni smisao izrade lokalnih zonskih datoteka za svaki aspekt vašeg lokalnog računala je reduciranje prometa i dopuštanje softveru da radi na isti način, bez obzira pristupa li lokalnom ili udaljenom računalu.

*Inverzna zonska datoteka (za* `zone "158.253.70.in-addr.arpa"`*)*
: Ova datoteka preslikava IP adrese na imena računala. To je „slika u ogledalu“ primarne zonske datoteke. Inverznu zonsku datoteku možete prepoznati po nastavku *in-addr.arpa* i koristi `PTR` zapise koje ćemo detaljnije objasniti kasnije.

*Primarna zonska datoteka (za* `zone "centralsoft.org"`*)*
: Ova datoteka, ponekad se naziva baza podataka domene, sadrži najveći dio informacija potrebnih za razrješavanje upita o domeni koju administrirate. Ona ne dolazi unaprijed konfigurirana nakon što instalirate BIND. Obično morate sami napisati ovu datoteku od početka ili na temelju jedne od datoteka koje dolaze uz BIND.

  Primarna zonska datoteka preslikava imena u IP adrese i pruža informacije o uslugama koje vaše računalo pruža na Internetu (uključujući Web i FTP poslužitelj, poslužitelj elektroničke pošte, poslužitelje imena itd.).

Podrazumijevana konfiguracijska datoteka sadrži prva dva iskaza zone (za korijenske poslužitelje i lokalno računalo – one se instaliraju zajedno s BIND-om i ne trebate ih mijenjati). Morat ćete sami dodati unose za inverznu i primarnu zonsku datoteku.

Zonske datoteke koriste nekoliko tipova zapisa, uključujući:

- `SOA` (Start of Authority)
- `NS` (Name Server)
- `MX` (Mail eXchanger, koji identificira poštanski poslužitelj u domeni)
- `A` (Preslikavanje imena računala u adresu)
- `CNAME` (Kanoničko ime koje definira alias za ime računala iz zapisa `A`)
- `PTR` (Pokazivač koji preslikava adrese u imena)

Ne morate odmah pamtiti ili razumijeti ove tipove zapisa. Imate ćete mnoge prilike da ih koristite kako budemo sve dublje ulazili u temu.

Slijedi detaljan pregled primarne zonske datoteke.

## Primarna zonska datoteka

Primarna zonska datoteka sadrži većinu konfiguracijskih informacija potrebnih DNS-u. Format datoteke nije standardiziran, ali su njezini elementi zadani specifikacijom RFC 1035.

Ako koristite skupinu datoteka koje pruža Debianova instalacija, trebali biste primarnu zonsku datoteku nazvati po svojoj domeni. Mi smo je nazvali *pri.centralsoft.org* po domeni *centralsoft.org* (Prefiks *pri* govori da se radi o primarnoj zonskoj datoteci). Objasnit ćemo svaki dio datoteke. Ona je u cijelosti prikazana u odjeljku „Sažetak".

Prvi red pruža informacije potrebne za sinkroniziranje sekundarnog ili slave poslužitelja:

```
@ IN SOA server1.centralsoft.org. root.localhost. (
                        2006012103; serial-no
                        28800; refresh, seconds
                        7200; retry, seconds
                        604800; expiry, seconds
                        86400 ); minimum-TTL, seconds
;
```

To je `SOA` zapis. `SOA` je akronim za Start of Authority i razdvaja informacije za glavne poslužitelje (primarne i sekundarne) od informacija za pomoćne poslužitelje sa privremenom memorijom. Kada ste napisali svoj dio distribuiranog DNS imenika, sustav vam je prepustio odgovornost za njega. Zbog toga vaša zonska datoteka mora govoriti gdje počinju vaše ovlasti.

Točka-zarez ne označava kraj reda koda, već početak komentara. Prema tome, ako ne želite uključiti komentar „serial-no", mogli biste red:

```
2006012103; serial-no
```

napisati:

```
2006012103
```

Pogledajmo prvi red, onaj koji počinje sa znakom @. Slijeva nadesno polja su:

*Ime*
: Korijensko ime zone. Znak @ predstavlja skraćena referenca tekuće zone u datoteci */etc/named.conf*. Drugim riječima, to je isto kao da smo koristili `server1.centralsoft.org` u našem primjeru. Znak @ u DNS žargonu označava *izvor*.

*Klasa*
: DNS klasa. Postoji određeni broj klasa, no velika većina lokacija koristi IN (Internet) klasu. Ostale klase postoje za ne-internetske protokole i funkcije.

*Tip*
: Tip DNS zapisa. U ovome slučaju to je SOA zapis.

*Poslužitelj imena*
: Puno ime primarnog poslužitelja imena. Jedan detalj, koji se lako može propustiti, vrlo je važan: ime mora završavati točkom (.) koja označava korijen DNS hijerarhije, kako bi se istaknulo da je putanja puno ime domene.

*Adresa elektroničke pošte*
: Adresa elektroničke pošte osobe odgovorne za domenu. Ovdje postoji još jedna važna konvencija specifična za DNS sustav: ne smije se koristiti znak @ koji se pojavljuje u svakoj adresi elektroničke pošte na Internetu. Kao što smo vidjeli, @ u ovoj datoteci ima drugo značenje. Zato ga zamjenjujemo s točkom. Želimo navesti korisnika *root* lokalnog sustava, odnosno njegovu adresu *root@localhost* ali u neuobičajenom formatu `root.localhost`. Adresa elektroničke pošte također mora završiti točkom.

  Sljedeći redovi u SOA zapisu sadrže polja koja će koristiti sekundarni poslužitelj:

*Serijski broj*
: Serijski broj tekuće konfiguracije. Taj broj se povećava svaki put kada se promijeni konfiguracija DNS-a tako da sekundarni poslužitelj znati da mora ažurirati svoje informacije. Taj broj je obično u datumskom formatu GGGGMMDD, sa dvoznamenkastim brojem dodanim na kraj (to omogućava višestruke izmjene tijekom jednog dana). Tako je svaki serijski broj veći od prethodnog i dokumentira datum promjene. Svaki sekundarni poslužitelj periodično provjerava serijski broj na primarnom kako bi provjerio da li se nešto promijenilo. Ako je tekući broj na primarnom poslužitelju veći od onoga na sekundarnom, sekundarni prenosi zonu. 2006012103 je početni serijski broj u našem primjeru zonske datoteke.

*Period ažuriranja*
: Interval u kojem će sekundarni DNS poslužitelj provjeravati na primarnom da bi utvrdio treba li se prenijeti zonu. Vrijednost se zadaje u sekundama. U našem primjeru datoteke zadana je vrijednost 28800 (28,800 sekundi odnosno 8 sati).

*Period ponovnog pokušaja*
: Zadaje koliko često bi se sekundarni poslužitelj trebao povezivati sa primarnim u slučaju neuspjelog povezivanja. Interval u našem primjeru iznosi 7200 (7,200 sekundi odnosno 2 sata).

*Rok trajanja*
: Period tijekom kojeg bi sekundarni poslužitelj trebao pokušavati kontaktirati primarni poslužitelj prije nego što podaci koje sadrži zastare. Ako podaci zastare, a sekundarni poslužitelj nije u mogućnosti kontaktirati primarni kako bi preuzeo svježe informacije, budući upiti bit će preusmjereni prema korijenskim poslužiteljima. Zadano vrijeme je također i period u kojem bi sekundarni poslužitelj trebao nastaviti odgovarati na upite, čak i ako ne može ažurirati zonsku datoteku, tj. period toleriranja zastarjelih informacija. U našem slučaju koristimo 604800 (604,000 sekundi = 7 dana).

*Minimalno vrijeme života*
: Podrazumijevano vrijeme života za ovu domenu u sekundama. Svaki zapis koji nema zadano vrijeme života koristi podrazumijevanu vrijednost od 864000 koja odgovara jednom danu.

To su sva polja SOA zapis. Nakon njih slijedi popis imena različitih računala:

```
NS server1.centralsoft.org.;
NS server2.centralsoft.org.;
```

Ovi `NS` zapisi navode poslužitelje imena za domenu (one koje ste naveli kod registriranja domene). Točka-zarez nije potrebna, ali je možete staviti na kraj reda ako želite dodati komentar.

Slijedi `MX` zapis koji identificira poštanski poslužitelj za domenu:

```
MX 10 server1.centralsoft.org.
```

Koristili smo samo jedan poštanski poslužitelj u našem slučaju, ali mnoge radne okoline pružaju ih nekoliko (kako bi se rasporedilo opterećenje te kako bi postojao rezervni poslužitelj u slučaju kada glavni otkaže). Drugo polje ovog zapisa (`10` u našem primjeru) može se koristiti za zadavanje redoslijeda po kojem bi se trebali isprobavati poslužitelji. Ono se koristi za zadavanje prioriteta poslužitelja.

Nakon `MX` zapisa u našem primjeru primarne zonske datoteke slijedi nekoliko `A` zapisa:

```
centralsoft.org.    A 70.253.158.42
www                 A 70.253.158.42
server1             A 70.253.158.42
server2             A 70.253.158.45
```

`A` zapis pridružuje ime IP adresi. Kako jednom računalu može biti pridruženo više imena, možete imati više `A` zapisa koji pokazuju na jednu IP adresu. Međutim, svako ime računala može imati najviše jedan `A` zapis. Naša datoteka ima četiri `A` zapisa dodjeljujući tri imena jednoj adresi te jedno ime drugoj adresi.

### Poboljšanja i napredne mogućnosti

Ako postavite datoteku sa sadržajem kao u prethodnom odjeljku i zamijenite imena računala i IP adrese sa stvarnim imenima i IP adresama koje koristite, dobit ćete primarnu zonsku datoteku koju ćete moći koristiti bez problema u svom radnom okruženju. (Naravno, trebat ćete i druge datoteke, no to ćemo kasnije objasniti). Međutim, trebali biste znati i za neke druge korisne stvari koje možete raditi sa primarnom zonskom datotekom.

**MX zapis** Kao što ste vidjeli, tipični `MX` zapis izgleda ovako:

```
MX 10 server1.centralsoft.org.
```

Taj zapis govori da elektronička pošta adresirana na domenu `centralsoft.org` treba biti isporučena na `server1.centralsoft.org` (poštanski poslužitelj za domenu), koja ima prioritet `10`.

Prioriteti dolaze do izražaja u složenim konfiguracijama, kada postoji više poštanskih poslužitelja. Manji brojevi označavaju veći prioritet pa je 1 najveći prioritet. Sustav prioriteta radi na sljedeći način: udaljeni poštanski poslužitelj prvo pokušava kontaktirati poslužitelj s vašeg popisa koji ima najveći prioritet. Ako taj poslužitelj ne odgovara, pokušava kontaktirati prvi sljedeći poslužitelj s manjim prioritetom i tako dalje prema kraju popisu. Recimo da ste na popisu naveli više poštanskih poslužitelja:

```
MX 10 server1.centralsoft.org.
MX 20 mail.someotherdomain.com.
```

Ako je pošta poslana na *centralsoft.org* izvorni agent za prijenos pošte prvo pokušava uspostaviti kontakt sa poslužiteljem *server1.centralsoft.org* jer ima najviši prioritet (10). Ako ne može pristupiti poslužitelje *server1.centralsoft.org*, izvorni agent će kontaktirati sljedeći poslužitelj, *mail.someotherdomain.com*, koji ima prioritet 20.

DNS specifikacija ne definira kako postupati u situaciji kada je navedeno više poštanskih poslužitelja sa istim prioritetom. Mnogi nasumice odaberu jedan kako bi implementirali jednostavno raspoređivanje opterećenja na više poslužitelja.

Do sada smo definirali MX zapise samo za elektroničku poštu adresiranu na *user@centralsoft.org*. Što ako želite preusmjeriti elektroničku poštu u druge odjele kompanije ili vladine agencije? To možete učiniti dodajući poddomene MX zapisima.

Tako, dodavanje *accounting.centralsoft.org* bi zahtijevao još jedan MX zapis:

```
accounting.centralsoft.org.  MX 10 server1.centralsoft.org.
```

Zamijetite točku nakon accounting.sentralsoft.org.. Ako ne biste naveli točku izvor zone bio bi spojen s imenom. Na primjer, ako biste napisali:

```
accounting.centralsoft.org    MX 10 server1.centralsoft.org.
```

bez točke na kraju, taj zapis bi se pretvorio u *accounting.centralsoft.org.centralsoft.org* što je neispravno.

**A zapisi.** NS i MX zapisi koriste imena računala poput *centralsoft.org, server1.centralsoft.org i server2.centralsoft.org*, no primarna zonska datoteka mora sadržavati i IP adrese na koje se ta imena preslikavaju. A zapisi izvode to preslikavanje. Mnogi ih smatraju najvažnijim DNS zapisima zato što ih možete koristiti za definiranje adresa računala poput *www.centralsoft.org*, gdje je *www* računalo.

Sljedeći primjer A zapisa iz primarne zonske datoteke pokazuje da *centralsoft.org* ima IP adresu 70.253.158.42:

```
centralsoft.org.    A 70.253.158.42
```

(Ne zaboravite točku na kraju imena računala.)

U preglednik vjerojatno upisujete *www.centralsoft.org* umjesto samo *centralsoft.org*. Tehnički, *www.centralsoft.org* potpuno je drugačije od *centralsoft.org*, no većina

posjetitelja očekuje da će doći na istu Web lokaciju bez obzira jesu li upisali *www.* na početku ili nisu. Zato smo i izradili ovaj zapis:

```
www       A 70.253.158.42
```

Nakon www ne dolazi točka, tako da BIND dodaje početak zone. Efekt je isti kako i kada biste zadali:

```
www.centralsoft.org.   A 70.253.158.42
```

Navedite IP adrese za *server1.centralsoft.org* i *server2.centralsoft.org*:

```
server1                A 70.253.158.42
server2                A 70.253.158.45
```

Zapis za *server2.centralsoft.org* pokazuje na drugačiju IP adresu, koja ima smisla zato što se radi o našem sekundarnom poslužitelju imena koji se mora nalaziti na drugom sustavu u slučaju da primarni poslužitelj imena otkaže.

### Problem samopodizanja i spojni zapisi

Možete se pitati kako se *server1.centralsoft.org* i *server2.centralsoft.org* mogu koristiti za traženje zapisa za *centralsoft.org* ako se nalaze u zoni koja se mora pretraživati. To je klasičan problem početnog pokretanja sustava: ne možete koristiti istu tehniku za pokretanje potrage i za izvođenje pretraživanja nakon što je ona pokrenuta.

Rješenje uključuje upotrebu *spojnih zapisa* (engl. *glue records*). Kada poslužitelji vršne domene *.org* usmjere korisnike na poslužitelje imena za *centralsoft.org*, normalno daju ime, a ne IP adresu računala (npr. *server1.centralsoft.org* umjesto 70.253.158.42). Ali, na poslužitelju imena vršne domene postoje spojni zapisi za glavne poslužitelje imena u zoni koja se pretražuje koji preslikavaju ime na IP adresu (u našem slučaju server1.centralsoft.org se preslikava na 70.253.158.42) a poslužitelj vršne domene isporučuje IP adresu umjesto imena poslužitelja imena. To znači da ga ne morate pronaći prije nego što budete mogli pitati gdje je.

**CNAME zapisi.** `CNAME` je kratica od *canonical name* (kanoničko ime). `CNAME` zapise možete smatrati aliasima `A` zapisa. Na primjer:

```
ftp     CNAME www
```

znači da je *ftp.centralsoft.org* alternativno ime za *www.centralsoft.org*, tako *ftp.centralsoft.org* pokazuje na isto računalo kao i *www.centralsoft.org*. Možete se naći u situaciji, posebice kada ppreuzimate Linux instalacijske pakete, kada repozitorij izgleda poput *http://ftp.mirrors.kernel.org*. U takvim slučajevima gotovo je sigurno da se `CNAME` zapis koristio za dodavanje dijela *ftp* imenu računala koje ima drugačije ime u svojem `A` zapisu.

`CNAME` zapis mora uvijek pokazivati na `A` zapis, a ne na drugi `CNAME` zapis. Nadalje, ne smijete koristiti `CNAME` imena računala u `MX` ili `SOA` zapisima. Na primjer, ovo nije dopušteno:

```
MX 10 ftp
```

Upotreba CNAME zapisa ima svoje prednosti i nedostatke. Mnogi specijalisti za DNS zapise preporučuju da se ne koriste. No, možete vidjeti da CNAME zapisi mogu biti korisni. Na primjer, ako vaš DNS imenik sadrži mnogo A zapisa koji pokazuju na istu IP adresu i preselite se kod druge tvrtke za udomljavanje poslužitelja koja vam je dodijelila drugu IP adresu, morat ćete ažurirati svaki A zapis. No, ako imate samo jedan A zapis, a sva ostala imena se nalaze u CNAME zapisu, morat ćete ažurirati samo jedan A zapis. Zbog toga vjerujemo da CNAME zapisi još uvijek imaju rezervirano mjesto u DNS panteonu.

**TXT i SPF zapisi.** TXT zapisi dopuštaju da dodajete tekst u zonu. Administratori ih prvenstveno korsite kako bi ugradili SPF (Sender Policy Framework) zapise, koji kontroliraju da li poštanski poslužitelji trebaju prihvatiti elektroničku poštu adresiranu s njihove domene. Veći pružatelji usluga elektroničke pošte, kao što su Yahoo! i Hotmail danas se uvelike oslanjaju na SPF zapise kako bi spriječili pošiljatelje neželjenih poruka da navedu pošiljatelja poruke kao da je s njihove domene. Ako elektronička poruka dolazi s računala koje nije na popisu u SPF zapisu, agent za prijenos pošte ju može klasificirati kao neželjenu poruku.

Čarobnjaka za izradu SPF zapisa možete pronaći na adresi *http://www.openspf.org/wizard.html?mydomain=&x=26&y=8*. Upotrijebili smo ovog čarobnjaka za izradu dva SPF zapisa za *centralsoft.org* koje smo zatim umetnuli u TXT zapise i dodali ih u našu zonsku datoteku:

```
centralsoft.org.                TXT "v=spf1 a mx ~all"
server1.centralsoft.org.        TXT "v=spf1 a -all"
```

### Sažetak

Pogledajmo sada našu zonsku datoteku *pri.centralsoft.org*. Primijetite da smo, pored ranije objašnjenih dijelova, dodali i CNAME i TXT zapise:

```
@ IN SOA server1.centralsoft.org. root.localhost. (
                        2006012103; serial-no
                        28800; refresh, seconds
                        7200; retry, seconds
                        604800; expiry, seconds
                        86400 ); minimum-TTL, seconds
;
                    NS server1.centralsoft.org.;
                    NS server2.centralsoft.org.;
;
                    MX 10 server1.centralsoft.org.
;
centralsoft.org.    A 70.253.158.42
www                 A 70.253.158.42
server1             A 70.253.158.42
server2             A 70.253.158.45
ftp                 CNAME www
centralsoft.org.                TXT "v=spf1 a mx ~all"
server1.centralsoft.org.        TXT "v=spf1 a -all"
```

## Inverzna zonska datoteka

Kada smo kompletirali primarnu zonsku datoteku, programi mogu pretraživati *centralsoft.org* domenu i sve njezine poddomene u DNS-u. No, još uvijek nam je potrebna inverzna zonska datoteka.

Inverzna zonska datoteka preslikava IP adrese u imena računala. Ona je poput slike u ogledalu primarne zonske datoteke. Umjesto imena računala, ona prvo ispisuje IP adrese.

Zašto bi netko koristio inverznu zonsku datoteku? U prošlosti, mnoge organizacije nisu dopuštale korištenje svojih usluga ako nisu mogle pingati vaše ime domene. Danas, mnogi poslužitelji na Internetu koriste obrnuto traženje za provjeru izvora elektroničke pošte kako bi zaustavili neželjene poruke. To je i svrha `SPF` zapisa o kojima smo raspravljali ranije.

Sustav koji ovdje opisujemo ima problem s odašiljanjem pošte s kojim ćemo se detaljnije pozabaviti u petom poglavlju. DNS zadaje koji je agent za prijenos pošte odgovoran za poštu s domene kojoj pripada pošiljateljeva adresa elektroničke pošte. Mnogi pošiljatelji neželjene pošte pokušavaju slati poštu koristeći različite agente, no agent koji prima poštu može pogledati odakle pošta dolazi, prepoznati nepravilnost i odbiti takvu elektroničku poruku.

Ako želimo da elektroničke poruke koje dolaze sa *centralsoft.org* ne budu klasificirane kao neželjene, moramo izraditi inverznu zonsku datoteku. Prvo, da bismo pokazali na ovu datoteku morat ćemo dodati unos u datoteku *named.conf*:

```
zone "158.253.70.in-addr.arpa" {
     type master;
     file "pri.158.253.70.in-addr.arpa";
};
```

Brojevi mogi izgledati čudno, no oni slijede jednostavnu shemu. *centralsoft.org* je u mreži broj 70.253.158, pa obrnemo elemente 70.253.158 i dobijemo 158.253.70 te to koristimo u iskazu `zone` u *named.conf*. Domena *in-addr.arpa* je domena najviše razine koju koriste sve inverzne potrage.

Našu inverznu zonsku datoteku nazvat ćemo *pri.158.253.70.in-addr.arpa* i smjestit ćemo datoteku u isti direktorij u kojem se nalazi naša primarna zonska datoteka *pri.centralsoft.org*.

Početak datoteke *pri.158.253.70.in-addr.arpa* izgleda isto kao i početak datoteke *pri.centralsoft.org*:

```
@ IN SOA server1.centralsoft.org. root.localhost. (
                          2006012103; serial-no
                          28800; refresh, seconds
                          7200; retry, seconds
                          604800; expiry, seconds
                          86400 ); minimum-TTL, seconds
;
                    NS server1.centralsoft.org.;
                    NS server2.centralsoft.org.;
```

No, u ovu datoteku ne dodajemo niti jedan `A`, `MX` ili `CNAME` zapis. Umjesto toga dodajemo `PTR` zapise.

### PTR zapisi

PTR je kratica za Pointer, a PTR zapis upravo to i predstavlja – pokazivač na ime domene. Izradimo jedan počevši s IP adresom od *centralsoft.org*, 70.253.158.42. Datoteka *named.conf* već je zadala, u iskazu zone koje smo pokazali u prethodnom odjeljku, da ta datoteka definira računala u domeni 70.253.158. Stoga, sve što PTR zapis mora zadati je završni dio IP adrese, 42:

```
42   PTR   centralsoft.org.
```

Izradite točno jedan PTR zapis za svaku IP adresu u vašoj domeni. U našem primjeru koristimo još samo IP adresu 70.253.158.45 (za *server2.centralsof.org*) pa dodajemo:

```
45   PTR   server2.centralsoft.org.
```

I to je sve. Naša inverzna zonska datoteka sada izgleda ovako:

```
@ IN SOA server1.centralsoft.org. root.localhost. (
                          2006012103; serial-no
                          28800; refresh, seconds
                          7200; retry, seconds
                          604800; expiry, seconds
                          86400 ); minimum-TTL, seconds

;
                    NS server1.centralsoft.org.;
                    NS server2.centralsoft.org.;

42                  PTR    centralsoft.org.
45                  PTR    server2.centralsoft.org.
```

## Testiranje

Kada ste uredili sve konfiguracijske i zonske datoteke, morate promjene prijaviti BIND-u. Ponovno pokrenite *named* na sljedeći način:

```
# /etc/init.d/bind9 stop
# /etc/init.d/bind9 start
```

Ako dođe do bilo kakve pogreške ili se BIND ne ponaša kako očekujete, pogledajte odjeljak o rješavanju problema malo kasnije u ovom poglavlju.

Ubuduće, ako je jedina izmjena koju radite ažuriranje zonske datoteke s novim DNS unosima za korespondirajuću domenu, dovoljno je reći BIND-u da samo ponovno učita informacije o toj zoni, umjesto da ga ponovno pokrećete:

```
# rndc reload centralsoft.org
```

O naredbi *rndc* detaljnije ćemo diskutirati malo kasnije u knjizi.

Sada možemo isprobati našu konfiguraciju tako što ćemo potražiti domenu koristeći alat *dig*. Potražit ćemo IP adresu od *centralsoft.org*:

```
# dig centralsoft.org

; <<>> DiG 9.2.1 <<>> centralsoft.org
;; global options:  printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 48489
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 0

;; QUESTION SECTION:
;centralsoft.org.               IN      A

;; ANSWER SECTION:
centralsoft.org.        86400   IN      A       70.253.158.42

;; Query time: 198 msec
;; SERVER: 81.169.163.104#53(81.169.163.104)
;; WHEN: Sat Mar 11 18:55:21 2006
;; MSG SIZE  rcvd: 49
```

Kao što vidite, ovaj upit vraća IP adresu 70.253.158.42

Sada možemo izvesti inverznu pretragu:

```
# dig -x 70.253.158.42

; <<>> DiG 9.2.1 <<>> -x 70.253.158.42
;; global options:  printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 4096
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 0

;; QUESTION SECTION:
;42.158.253.70.in-addr.arpa.    IN      PTR

;; ANSWER SECTION:
42.158.253.70.in-addr.arpa. 5304 IN     PTR     centralsoft.org.

;; Query time: 2 msec
;; SERVER: 81.169.163.104#53(81.169.163.104)
;; WHEN: Sat Mar 11 18:57:54 2006
;; MSG SIZE  rcvd: 98
```

Regularna i inverzna pretraga međusobno se slažu. Uspješno smo konfigurirali primarni poslužitelj.

## Konfiguriranje sekundarnog poslužitelja imena

Konfigurirajmo sada sekundarni poslužitelj imena, *server2.centralsoft.org*. Ponašat će se kao rezervna kopija u slučaju da se primarni poslužitelj (*server1.centralsoft.org*) blokira, tako da korisnici i u takvom slučaju mogu pretraživati *centralsoft.org* i njezine poddomene.

Datoteka *named.conf* za sekundarni poslužitelj sliči onoj za primarni, uz nekoliko malih razlika:

```
options {
     pid-file "/var/run/bind/run/named.pid";
     directory "/etc/bind";
     // query-source address * port 53;
};

zone "." {
     type hint;
     file "db.root";
};

zone "0.0.127.in-addr.arpa" {
     type master;
     file "db.local";
};

zone "centralsoft.org" {
     type slave;
     file "sec.centralsoft.org";
     masters { 70.253.158.42; };
};
```

Najbitnija razlika je ona o kojoj smo diskutirali ranije u ovom poglavlju. Red `type slave`; u završnom iskazu `zone` govori da se radi o sekundarnoj zoni. U redu `file` navodimo ime datoteke u kojoj treba biti pohranjena sekundarna zona, dok u redu `masters` zadajemo IP adresu primarnog poslužitelja imena.

To je sve što trebamo učiniti da bismo konfigurirali sekundarni poslužitelj imena.

Ponovno pokrenite *named* na *server2.centralsoft.org* i nedugo nakon toga ćete pronaći datoteku */etc/bind/sec.centralsoft.org* na sekundarnom poslužitelju imena. Što se dogodilo? Sekundarni poslužitelj imena kontaktirao je primarni i dogodio se prijenos zone.

Kada god ažurirate zonu na primarnom poslužitelju imena, povećajte i serijski broj. U protivnom, ažurirana zona se neće prenijeti na sekundarni poslužitelj imena.

# BIND alati

Kao što smo spomenuli u ovom poglavlju, BIND se sastoji od tri dijela: servisa *named*, biblioteke *razrješivača* i još nekih alata.

Jedan alat ste već koristili – *dig*. Administratori ga koriste za testiranje poslužitelja imena. *dig* izvodi DNS pretraživanje i prikazuje dobiveni odgovor i statističke podatke o izvedenom upitu.

Većina DNS administratora koristi *dig* za rješavanje problema s DNS-om jer je to fleksibilan i jednostavan alat s jasnim rezultatima. Ostali alati manje su funkcionalni. No, alternativni alat za koji biste trebali znati je *nslookup.* Također ćemo razmotriti i *rndc,* koristan administratorski alat koji je uključen u BIND.

## nslookup

*nslookup* funkcionira slično kao *dig*, ali ga Linux korisnici potcjenjuju. Njegova upotreba nije jednostavna, no morate biti upoznati s njim zato što se u Microsoft Windowsima još uvijek koristi kao primarni alat za pretraživanje poslužitelja imena.

*nslookup* postavlja upite poslužiteljima imena u dva režima: interaktivnom i neinteraktivnom. Interaktivni režim omogućava postavljanje upita poslužitelju imena o različitim računalima i domenama, ili ispis popisa računala u domeni.

Neinteraktivni režim jednostavno ispisuje ime i traženu informaciju za računalo ili domenu. Na primjer, možete zadati sljedeću pretragu kako biste dobili informacije o Googleovom poslužitelju:

```
# nslookup ns1.google.com
Server:         68.94.156.1
Address:        68.94.156.1#53

Non-authoritative answer:
Name:   ns1.google.com
Address: 216.239.32.10
```

U interaktivnom režimu *nslookup* pruža odzivnik u kojem se mogu zadavati naredbe.

Na primjer:

```
# nslookup
>
```

U odzivniku možete zadavati male pretrage, poput ove za IP adresu:

```
> 70.253.158.42
Server:         172.30.1.2
Address:        172.30.1.2#53

Non-authoritative answer:
42.158.253.70.in-addr.arpa      name = adsl-70-253-158-42.dsl.rcsntx.swbell.net.

Authoritative answers can be found from:
158.253.70.in-addr.arpa nameserver = ns1.swbell.net.
158.253.70.in-addr.arpa nameserver = ns2.swbell.net.
>
```

Možete zadati nekoliko naredbi, uključujući *lserver* (koja koristi lokalni poslužitelj za izvođenje pretrage), *server* (koja koristi drugi poslužitelj) i *host.* Naredba *lserver* daje rezultat poput ovog:

```
> lserver google.com
Default server: google.com
```

```
Address: 64.233.167.99#53
Default server: google.com
Address: 64.233.187.99#53
Default server: google.com
Address: 72.14.207.99#53
```

Podnaredba *host* pruža jednostavan pomoćni alat za izvođenje pretrage. Kada nisu zadani argumenti i opcije, *host* ispisuje kratak popis svojih argumenata i opcija. Ljudi to prvenstveno koriste za pretvaranje imena u IP adrese i obrnuto. Evo primjera:

```
> host centralsoft.org
centralsoft.org has address 70.253.158.42
```

Kada *host* izvodite u detaljnom režimu, sa zadanom opcijom `-v`, dobit ćete rezultat sličan rezultatu naredbe *dig*:

```
> host -v centralsoft.org
Trying "centralsoft.org"
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 43756
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 1, ADDITIONAL: 0

;; QUESTION SECTION:
;centralsoft.org.               IN      A

;; ANSWER SECTION:
centralsoft.org.        86400   IN      A       70.253.158.42

;; AUTHORITY SECTION:
centralsoft.org.        29437   IN      NS      server1.centralsoft.org.

Received 71 bytes from 68.94.156.1#53 in 30 ms
```

Ove informacije došle su s IP adrese 68.94.156.1, ulaza broj 53, što je poslužitelj imena zadan u datoteci *resolv.conf* na računalu na kojem je zadano pretraživanje.

Možete ponovno koristiti *host* kako biste otkrili ime toga poslužitelja:

```
> host 68.94.156.1
1.156.94.68.in-addr.arpa domain name pointer dnsr1.sbcglobal.net.
```

Upišite `exit` kako biste završili interaktivnu *nslookup* sesiju.

*named* u nekim situacijama možete koristiti i za rješavanje problema. Na primjer, da biste saznali broj inačice vaše BIND implementacije, zadajte sljedeću naredbu:

```
# named -v
named 8.4.6-REL-NOESW Tue Feb  1 10:10:48 UTC 2005
        buildd@rockhopper:/build/buildd/bind-8.4.6/src/bin/named
```

## rndc

BIND pruža alat *rndc* u okviru svoje instalacije. *rndc* omogućava da administrirate *named* iz odzivnika. On šalje naredbe zadane u odzivniku servisu *named*. *rndc* se koristi i u inicijalizacijskim skriptama BIND-a 9.

Kako biste spriječili neautorizirane korisnike da pristupaju vašem poslužitelju imena, trebate koristiti tajni ključ za odobravanje pristupa samo autoriziranim korisnicima. Kako bi *rndc* slao naredbe bilo kojem servisu *named*, pa čak i onom na lokalnom stroju, obje strane moraju dijeliti isti ključ. Taj je ključ pohranjenu u datoteci */etc/bind/rndc.key*. Iz nje će ga čitati obje naredbe, *named* i *rndc*. Ključ *rndc.key* treba biti izrađen tijekom instaliranja BIND-a.

Naredba *rndc* ima sljedeću sintaksu:

```
# rndc rndc_opcije naredba opcije_naredbe
```

Slijedi nekoliko vrlo korisnih *rndc_opcija* koje možete koristiti (pročitajte upute za *rndc* da vidite kompletan popis):

*-k datoteka_ključa*
: Koristi zadanu `datoteku ključa` umjesto podrazumijevane */etc/bind/rndc.key* datoteke.

*-s poslužitelj*
: Šalje naredbu zadanom `poslužitelju` umjesto lokalnom.

*-V*
: Uključuje opširno ispisivanje za tu naredbu.

Evo nekoliko uobičajenih naredbi koje *rndc* može poslati servisu *named* (za kompletan popis upišite samo *rndc*):

*halt*
: Zaustavlja poslužitelj imena.

*querylog*
: Uključuje ili isključuje bilježenje u dnevnik svih upita od strane klijenata na dotičnom poslužitelju imena. Ova naredba djeluje poput prekidača: uključuje bilježenje ako je ono trenutno isključeno ili ga isključuje ako je uključeno.

*reload [zona]*
: Ponovno učitava zonsku datoteku, no čuva sve odgovore spremljene u privremenu memoriju. To omogućava da mijenjate zonske datoteke i da se to odražava na poslužiteljima imena bez gubitka svih pohranjenih odgovora. Ako je promijenjena samo jedna zona, možete reći servisu *named* da ponovno učita samo tu zonu.

*retransfer zona*
: Forsira ponovni prijenos specifične `zone` bez provjeravanja serijskog broja.

*stats*
: Ispisuje tekuću statistiku servisa *named* u datoteku *named.stats*.

*status*
: Pokazuje trenutni status poslužitelja imena.

*stop*
: Zaustavlja poslužitelj i sprema bilo kakvu dinamičku promjenu prije izlaza.

# Rješavanje problema s BIND-om

Do sada biste već trebali imati funkcionalno poznavanje DNS-a. Također, trebali biste znati kako konfigurirati datoteke te kako u njima pronaći sintaksne pogreške, poput slovnih. U ovom odjeljku objasnit ćemo neke osnovne, uobičajene probleme koji se mogu javiti kod korištenja BIND-a i DNS-a. Ovo nije detaljna rasprava, ali bi vam trebala pomoći da pokrenete DNS sustav na Linuxu u slučaju da domena ima problema s otkrivanjem imena računala ili prijenosom zona.

Domain Name System dizajniran je prilično robusno, no neke čudne pogreške se ipak povremeno mogu dogoditi. No, ako strogo pratite upute i primjere za izradu zonskih datoteka opisanih ranije u ovom poglavlju, možete izbjeći suptilne probleme koje ne možemo obuhvatiti u ovoj knjizi.

## Ne možete se povezati koristeći rndc

Za početak, pogledajmo indikaciju normalnog rada DNS poslužitelja. Ranije smo govorili o upotrebi *rndc* naredbe *status* za prikaz trenutnog statusa DNS poslužitelja. Pokušajmo se na poslužitelj prijaviti kao *root* i zadati naredbu:

```
server1:~# rndc status
number of zones: 6
debug level: 0
xfers running: 0
xfers deferred: 0
soa queries in progress: 0
query logging is OFF
server is up and running
server1:~#
```

Naredba *rndc* oslanja se na datoteku ključa */etc/bind/rndc.key* kako bi servisu *named* zadavala naredbe. Problemi s tom datotekom mogu spriječiti *rndc* da pošalje naredbe. Slijedi primjer onoga što bismo trebali vidjeti ako nedostaje datoteka s ključem:

```
server1:~# rndc status
rndc: neither /etc/bind/rndc.conf nor /etc/bind/rndc.key was found
server1:~#
```

S ovom naredbom možemo definitivno utvrditi da li datoteka stvarno nedostaje:

```
server1:~# ls -l /etc/bind/rndc.key
ls: /etc/bind/rndc.key: No such file or directory
```

Problem možemo popraviti izradom datoteke na isti način kako je to učinila BIND inicijalizacija:

```
server1:~# rndc-confgen -a
server1:~# ls -l /etc/bind/rndc.key
-rw-------  1 root bind 77 Jul 19 22:38 /etc/bind/rndc.key
server1:~#
```

Kako *named* sada nema novi ključ, moramo ga ponovno pokrenuti. Za to koristimo sistemsku naredbu *killall* koja uzima čitavu putanju programa *named*. Kako bismo zaustavili *named* što bezbolnije, izvršavamo dvije *killall* naredbe s nekoliko sekundi pauze i zatim ponovno pokrećemo *named*:

```
server1:~# killall -TERM /usr/sbin/named
server1:~# killall -KILL /usr/sbin/named
/usr/sbin/named: no process killed
server1:~# /etc/init.d/bind9 start
Starting domain name service: named.
server1:~# rndc status
number of zones: 6
debug level: 0
xfers running: 0
xfers deferred: 0
soa queries in progress: 0
query logging is OFF
server is up and running
server1:~#
```

## named je pokrenut, ali ne razrješava imena

Pogledajmo sada situaciju kada *named* ne radi ispravno. Neispravno smještene BIND datoteke često uzrokuju probleme, posebno kada su BIND datoteke smještene u izolirani direktorij.

Ako se *named* uspješno pokrene, ali ne učitava niti jednu zonsku datoteku, možda nije smješten u izolirani direktorij. Morat ćete pogledati u datoteku */var/log/syslog* kako biste provjerili. Evo primjera iz dnevnika:

```
starting BIND 9.2.4 -u bind -t /var/lib/named
using 1 CPU
loading configuration from '/etc/bind/named.conf'
listening on IPv4 interface lo, 127.0.0.1#53
listening on IPv4 interface eth0, 70.253.158.42#53
command channel listening on 127.0.0.1#953
command channel listening on ::1#953
running
```

Dnevnik pokazuje da je BIND pokrenut, ali nema zapisa koji govori da su zonske datoteke učitane. Kako se *named* izvodi unutar *chroot* okoline u */var/lib/named*, on će sve datoteke tražiti u odnosu na taj direktorij. Zbog toga on čita datoteku */var/lib/named/etc/bind/named.conf* kako bi dobio popis zona koje treba učitati. Svaka od tih zonskih datoteka mora biti smještena u odnosu na direktorij */var/lib/named*.

Sljedeća uobičajena pogreška je nemogućnost spajanja preko *rndc-a* kada se poslužitelj imena ponovno pokreće:

```
# /etc/init.d/ bind9 reload
Stopping named: rndc: connect failed: connection refused
[OK]
Starting named: [OK]
#
```

Ova vrsta pogreške također se događa i kao rezultat pokretanja BIND-a u chroot okolini, kada jedna ili više datoteka nedostaje u izoliranom direktoriju. Možete provjeriti neke bitne datoteke kako biste bili sigurni da su na ispravnim lokacijama:

```
# ls -l /var/lib/named/etc/bind/named.conf
-rw-r--r-- 1 root bind 1611 2006-09-07 12:21 /var/lib/named/etc/bind/named.conf
# ls /var/lib/named/etc/bind/
db.0      db.local     named.conf.local            pri.centralsoft.org
db.127    db.root      named.conf.options          pri.opensourcetoday.org
db.255    named.conf   pri.156.18.67.in-addr.arpa  rndc.key
db.empty  named.conf~  pri.156.18.67.in-addr.arpa~ zones.rfc1918
#
...
```

Ako te datoteke na postoje, chroot okolina nije propisno ili kompletno postavljena. Vratite su na odjeljak „Sigurnosna primjena okoline chroot“ blizu početka ovog poglavlja i pažljivo slijedite instrukcije kako biste bili sigurni da je svaka datoteka na svojemu mjestu.

Kada ste riješili problem, trebate ponovno pokrenuti *named* kako biste omogućili alatu *rndc* da dođe do pokrenutog poslužitelja. Koristite niz *killall* naredbi kao što je opisano u prethodnom dijelu knjige:

```
server1:~# killall -TERM /usr/sbin/named
server1:~# killall -KILL /usr/sbin/named
/usr/sbin/named: no process killed
server1:~# /etc/init.d/bind9 start
Starting domain name service: named.
server1:~#
```

Zatim provjerite vašu datoteku */var/log/syslog* da biste provjerili jesu li zonske datoteke učitane. Trebali biste vidjeti nešto poput ovog:

```
starting BIND 9.2.4 -u bind -t /var/lib/named
using 1 CPU
loading configuration from '/etc/bind/named.conf'
listening on IPv4 interface lo, 127.0.0.1#53
listening on IPv4 interface eth0, 70.253.158.42#53
command channel listening on 127.0.0.1#953
command channel listening on ::1#953
zone 0.0.127.in-addr.arpa/IN: loaded serial 1
zone 158.253.70.in-addr.arpa/IN: loaded serial 2006070401
zone centralsoft.org/IN: loaded serial 2006070502
zone supportcall.org/IN: loaded serial 2006062704
running
```

## Računala nisu prepoznata

Sljedeće što trebate provjeri jeste da li se na DNS upite ispravno odgovara. Prvo, trebate biti sigurni da datoteka */etc/resolv*.conf popisuje imena poslužitelja s ispravnim adresama. Većina programa koristi adrese iz te datoteke kako bi odredili kojem poslužitelju imena poslati upit i kojim redoslijedom:

```
server1:~# cat /etc/resolv.conf
search centralsoft.org
nameserver 70.253.158.42
nameserver 70.253.158.45
server1:~#
```

Naredba *host* radi jednostavnu pretragu DNS-a koristeći poslužitelje popisane u datoteci */etc/resolv.conf*. Ona kao argument uzima računalo koje želite pronaći. Možete zadati i drugi opcijski argument koji zadaje pretragu na specifičnom poslužitelju imena. Slijede dva primjera naredbe *host* i njezinih rezultata:

```
server1:~# host www.centralsoft.org
www.centralsoft.org has address 70.253.158.42
server1:~# host www.centralsoft.org server2.centralsoft.org
Using domain server:
Name: server1.centralsoft.org
Address: 70.253.158.45#53
Aliases:

www.centralsoft.org has address 70.253.158.42
server1:~#
```

Alternativa naredbi *host* je naredba *dig*, koja je složenija, no daje detaljnije odgovore. Ona ima i dodatne opcije koje omogućavaju da zadajete vrlo specifične upite.

Rezultat naredbe *dig* je oblikovan u sintaksi zonske datoteke. To je vrlo zgodno zato što, kada svladate način na koji su oblikovani zapisi u zonskoj datoteci, lako ćete razumjeti detalje u rezultatu naredbe *dig*. Naredba *dig* pruža i neke dodatne informacije o rezultatima upita u obliku zonskih komentara koji započinju sa dvotočkom.

Pogledajmo rezultat naredbe *dig*. Mnogi redovi rezultata naredbe *dig* vrlo su dugačke i ne mogu stati na stranicu ove knjige. U sljedećem ispisu lome se u sljedeći red. Vrlo vjerojatno ćete vidjeti sličan rezultat kad zadate ovu naredbe u vašoj školjci:

```
server1:~# dig www.centralsoft.org a

; <<>> DiG 9.2.4 <<>> www.centralsoft.org a
;; global options:  printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 1633
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 2, ADDITIONAL: 2

;; QUESTION SECTION:
;www.centralsoft.org.           IN      A

;; ANSWER SECTION:
www.centralsoft.org.    86400   IN      A       70.253.158.42

;; AUTHORITY SECTION:
centralsoft.org.        86400   IN      NS      server1.centralsoft.org.
centralsoft.org.        86400   IN      NS      server2.centralsoft.org.
```

```
;; ADDITIONAL SECTION:
server1.centralsoft.org. 86400  IN      A       70.253.158.42
server2.centralsoft.org. 86400  IN      A       70.253.158.45

;; Query time: 1 msec
;; SERVER: 70.253.158.42#53(70.253.158.42)
;; WHEN: Mon Jul 17 23:30:51 2006
;; MSG SIZE  rcvd: 129

server1:~#
```

Prvi dio daje različite statusne kodove i zastavice. Obratite pozornost na vrijednost `status` u četvrtom redu. U ovome primjeru vrijednost je `NOERROR`. Bilo koja druga vrijednost vjerojatnu upućuje na nekakav problem.

Aktualni podaci o zoni pojavljuju se u četiri sekcije:

`QUESTION`
: Pruža detalje o samom upitu. Prikazana je kao komentar zato što to nije nešto što bi se trebalo nalaziti u zonskoj datoteci.

`ANSWER`
: Sadrži odgovor na postavljeni upit. Pokazat će specifičan zapis koji je zatražen, ako je to moguće, ili sve zapise ako je zadan poseban tip upita `any`.

`AUTHORITY`
: Identificira službene poslužitelje imena za zonu za koju dolazi odgovor.

`ADDITIONAL`
: Ova sekcija daje adrese nekih ili svih imena iz prijašnjih sekcija, kako ne biste morali zadavati dodatne upite za dobijanje tih informacija.

Pogledajmo sada što biste dobili da je postojala neka pogreška. Prethodni primjer koristio je ispravno ime računala za Web poslužitelj. Ovaj put ćemo zatražiti ime FTP poslužitelja koji nismo konfigurirali u našoj zonskoj datoteci:

```
server1:~# dig ftp.centralsoft.org a

; <<>> DiG 9.2.4 <<>> ftp.centralsoft.org a
;; global options:  printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NXDOMAIN, id: 6531
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 0, AUTHORITY: 1, ADDITIONAL: 0

;; QUESTION SECTION:
;ftp.centralsoft.org.           IN      A

;; AUTHORITY SECTION:
centralsoft.org.        86400   IN      SOA     server1.centralsoft.org. admin.
centralsoft.org. 2006070502 28800 7200 604800 86400
```

```
;; Query time: 1 msec
;; SERVER: 70.253.158.42#53(70.253.158.42)
;; WHEN: Mon Jul 17 23:30:59 2006
;; MSG SIZE  rcvd: 87

server1:~#
```

Primijetite da je `status` ovog upita `NXDOMAIN`, što u biti znači „nema takvog imena domene“. Ako ste propustili ili krivo napisali ime računala u zonskoj datoteci, dobit ćete ovakvu pogrešku.

Druga vrsta pogreške koja se može javiti kada koristite *dig* jeste kada je ime domene delegirano vašem poslužitelju imena, no ta domena nije konfigurirana na poslužitelju ili se zbog nečeg drugog ne može učitati. Ta vrsta pogreške daje `status SERVFAIL`. Ako vidite takvu pogrešku, trebate domenu dodati u datoteku *named.conf* i osigurati da za nju postoji odgovarajuća zonska datoteka. Ako se pogreška pojavljuje i nakon što to učinite, pogledajte da li se u datoteci */var/log/syslog* pojavljuje zapis koji objašnjava zašto zona nije učitana. Pokazat ćemo problem kada je ime domene registrirano, ali se ne koristi:

```
server1:~# dig linhelp.org a

; <<>> DiG 9.2.4 <<>> linhelp.org a
;; global options:  printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: SERVFAIL, id: 29949
;; flags: qr rd ra; QUERY: 1, ANSWER: 0, AUTHORITY: 0, ADDITIONAL: 0

;; QUESTION SECTION:
;linhelp.org.           IN      A

;; Query time: 2 msec
;; SERVER: 70.253.158.42#53(70.253.158.42)
;; WHEN: Mon Jul 17 23:47:14 2006
;; MSG SIZE  rcvd: 37

server1:~#
```

# Što slijedi

Do sada biste trebali dobro poznavati osnove DNS sustava i alata BIND. Admistratori u malim i srednjim tvrtkama mogu otkriti da su im informacije iz ovog poglavlja sve što im je potrebno. No administratori većih sustava mogu se susresti s ozbiljnijim potrebama ili problemima nego što su ovi koje možemo opisati u jednom poglavlju.

Postoji nekoliko knjiga koje nude mnogo više informacija za administratore velikih sustava. među njima su *DNS and BIND* autora Criketa Liua i Paula Albitza (u izdanju O'Reilly), *DNS & BIND Cookbook* Criketa Lua (također u izdanju O'Reilly), *Pro DNS and BIND* autora Rona Aitchisona (u izdanju Apress) te *DNS in Action: A Detailed and Practical Guide to DNS Implementation, Configuration and Administration* autora L. Dostaleka i A. Kableova (u izdanju Packt).

Kada imate funkcionalne poslužitelje imena koji odgovaraju na upite i imaju rezervne kopije podataka na sekundarnom poslužitelju, možete prijeći na instaliranje Web servisa. Nove aplikacije će iskoristiti servise koje ste postavili u drugom poglavlju. Kada aplikacija ISPConfig bude postavljena i funkcionalna, imat ćete radni primjer potpuno operativne Web lokacije. Zatim možemo započeti s metodama administriranja kompletnog skupa Linuxovih usluga na Internetu.

POGLAVLJE 4

# Početna okolina spremna za Internet

Jedno od Linuxovih značajnijih obilježja je njegova fleksibilnost. Tvrtke kao što je Cisco skrivaju Linux iza vrlo jednostavnih sučelja kako bi učinile svoje Linksys usmjerivače i druge proizvode jednostavnim za upotrebu. Mi to možemo učiniti, također.

ISPConfig (*http://ispconfig.org*), jednostavni Linux projekt pod licencom besplatnog softvera (BSD), omogućava da postavimo višenamjenski internetski poslužitelj preuzimanjem i pokretanjem samo jedne aplikacije. Kada ju instaliramo, imat ćemo alat koji olakšava konfiguriranju i administriranje poslužitelja. On će nam pružati poslužitelj imena domena, koristit ćemo ga za prijenos datoteka i elektroničke pošte te dodavanje korisnika, administratora i svih drugih koji će mu pristupati zbog različitih administracijskih zadaća. Ah, jesmo li spomenuli da svu tu administraciju možemo obavljati iz grafičkog korisničkog sučelja?

ISPConfig smo prvenstveno odabrali zato što dopušta razvijanje moćnih poslužiteljskih aplikacija na Linuxu bez žrtvovanja snage ili fleksibilnosti. Nadalje:

- ISPConfig koristi standardne pozadinske servise koji dolaze s Linux distribucijom. Za posluživanje Web lokacija koristit ćemo Apache, za elektroničku poštu Postfix, za FTP ProFTPD, BIND za DNS i MySQL kao pozadinsku bazu podataka.
- Instaliranje ISPConfiga automatski konfigurira različite poslužiteljske komponente.
- Paketi uključeni u ISPConfig rade s većinom raspoloživih Linux distribucija.
- Mogu se koristiti i standardni paketi iz distribucije.
- Tehnička podrška za svaku skupinu komponenti dostupna je na Internetu.
- ISPConfig razvojni tim pruža mrežnu podršku za kompletnu aplikaciju.

Kako budete napredovali kroz ovo poglavlje bit će vam jasnije što sve treba učiniti da bi se uspostavile različite usluge koje će poslužitelj pružati. Također ćete naučiti odlučiti dopuštaju li vam potrebe da koristite paletu vizualnih administrativnih alata umjesto alata koji se koriste iz odzivnika.

ISPConfig sam po sebi ne pruža sučelje odzivnika. Umjesto toga omogućava da upravljate poslužiteljima kroz Web sučelje – panel – opisano kasnije u ovom poglavlju. Na početku ćete morati malo raditi s odzivnikom, primjerice kod postavljanja ISPConfiga tako da kasnije može sam instalirati sve ostalo, no kasnije ćemo se u ovom poglavlju fokusirati na vizualno sučelje.

ISPConfigovo Web sučelje pojednostavljuje obavljanje mnogih Linux administracijskih zadaća, no važno je znati i kako se koriste standardni alati za odzivnik da bi se došlo do istih rezultata. Te ćemo teme obraditi u sljedećim poglavljima. Nećete biti vezani za ISPConfig; ako odlučite da želite raditi bez njega, imat ćete dovoljno znanja da tako i postupite.

# Instaliranje ISPConfiga

ISPConfig razvija tvrtka Projektfarm GmbH. Aplikaciju su razvili Till Brehm i Falko Timme i ona se izvorno prodavala kao vlasnički sustav reklamiran na *http://42go.de*. Danas ga možete preuzeti s *http://sourceforge.net/projects/ispconfig*.

Projekt konfigurira sljedeće usluge:

- *httpd* (virtualna računala, temeljena na domeni i IP adresama)
- FTP
- BIND
- POP3 automatizirano odgovaranje
- MySQL klijentske baze podataka
- Webalizer statistike
- Kvote na tvrdom disku
- Kvote za elektroničku poštu
- Ograničenja prometa
- IP adrese
- SSL
- SSI
- Školjku
- Mailscanner (antivirusna zaštita)
- Vatrozid

## Zahtjevi

U vrijeme pisanja ove knjige sistemski zahtjevi su bili:

*Operacijski sustav*
: Linux (jezgra 2.4 ili novija s *glibc6* bibliotekom). Podržane su sljedeće distribucije:

  - CentOS 4.1, 4.2, 4.3 i 4.4
  - Debian inačica 3.0 ili novija
  - Fedora Core 1 do 6
  - Mandrake Linux Version 8.1 ili noviji
  - Red Hat Linux 7.3 ili noviji
  - SUSE Linux 7.2 ili noviji
  - Ubuntu 5.04 do 6.10

*Linux paketi*
: Postoji popis specifičnih komponenata Linux distribucije koje moraju biti instalirane na vaš sustav prije nego što budete mogli instalirati ISPConfig. To uključuje:

  - Apache Web poslužitelj 1.3.12 ili noviji, ili 2.0.40 ili noviji
  - BIND 8 ili 9
  - *iptables ili ipchains*
  - MySQL bazu podataka
  - OpenSSL i *mod_ssl* za izradu SSL virtualnih računala
  - PHP 4.0.5 ili noviji kao modul Apache poslužitelja
  - POP3/IMAP servis koji podržava tradicionalni Unixov format poštanskog sandučića (npr. *gnu-pop3d, qpopper, ipop3d, popa3d* ili *vm-pop3d*) ili format *maildir* (npr. Courier-Imap, Dovecot)
  - Procmail
  - ProFTP kao samostalna aplikacija ili *vsftpd* kao *inetd/xinetd* samostalna aplikacija
  - *quota* paket
  - Sendmail ili Postfix

Važno je znati da ovi poslužitelji i paketi moraju već biti instalirani na vaš sustav kao što je opisano u drugom poglavlju ove knjige, prije nego započnete s instaliranjem ISPConfiga. Te usluge nisu dio paketa ISPConfig, ali on zahtijeva da budu instalirane na vašem sustavu. Prednost ovog pristupa je u tome što možete koristiti podrazumijevane pakete vaše distribucije i kasnije ih ažurirati kao što biste najnormalnije ažurirali pakete na vašem sustavu upotrebom alata koji dolazi uz vašu distribuciju. Ne trebate prevoditi ove usluge iz izvornog koda sa specifičnim opcijama za upotrebu s ISPConfigom – sasvim dobro će funkcionirati podrazumijevani paketi.

ISPConfig postavlja dva direktorija koji sadrže datoteke i poddirektorije i čine panel: */root/ispconfig* i */home/admispconfig*. Možete deinstalirati ISPConfig i vratitit se na standardni, tekstualno orijentirani, poslužitelj pokretanjem */root/ispconfig/uninstall*. Neki čitatelji će možda to odabrati nakon što malo dulje budu čitali ovu knjigu.

### Specijalni ISPConfig pozadinski servisi

Osim što omogućava upravljanje aplikacijama koje ste instalirali na sustav, ISPConfig održava nekoliko svojih inačica aplikacija za vlastitu upotrebu. Njihov izvorni kôd možete pronaći u direktoriju *insatall_ispconfig/compile_aps* paketa. Ovi dodatni servisi postoje da biste mogli nastaviti upravljati ISPConfigom čak i ako regularni servisi (poput javnog Apache Web poslužitelja) zakažu.

ISPConfig dopušta i javnim i internim poslužiteljima da se izvode i pritom koriste nestandardne ulaze za interne poslužitelje. Na primjer, ISPConfigov interni Apache poslužitelj osluškuje ulaz 81 umjesto ulaza 80, koji tipično koriste distribucijski Web poslužitelji koji udomljavaju javne Web lokacije.

## Početak

Poput mnogih Linux i Unix paketa, ISPConfig ima skup datoteka kombiniranih s alatom *tar,* koji se često zove *tarball*. Kada pritisnete vezu Download na stranici *http://sourceforge.net/projects/ispconfig,* odvest će vas na jedno od lokacijskih SourceForge ogledala. Tipična lokacija koja sadrži ISPConfig je *http://superb-west.dl.sourceforge.net/sourceforge/ispconfig/ISPConfig-2.2.6.tar.gz*.

Možete jednostavno pritisnuti vezu Download, no budući da je datoteka prilično velika, može biti korisno da kopirate URL u naredbu *wget* u prozoru terminala. Prednost upotrebe naredbe *wget* je u tome što omogućava oporavak prijenosa ako ga nešto prekine. Ako zadate naredbu s opcijom *-c*, moći ćete nastaviti preuzimanje podataka umjesto da krećete od početka: ako se preuzimanje prekine, ponovno zadajte naredbu *wget* kao i ranije. Preuzimanje će se nastaviti gdje je bilo prekinuto.

U ovom poglavlju pretpostavljamo da započinjete u direktoriju */root* na vašem sustavu. Možete preuzeti ISPConfig arhivu zadajući ovu naredbu (u jednom redu, zamjenjujući URL s URL-om najnovije inačice na lokaciji SourceForge):

```
# wget -c http://superb-west.dl.sourceforge.net/sourceforge/ispconfig/ISPConfig-
2.2.6.tar.gz
```

Na terminalu će se ispisati poruke slične ovima:

```
--16:20:48--  http://superb-west.dl.sourceforge.net/sourceforge/ispconfig/ISPConfig-
2.2.1.tar.gz
          => `ISPConfig-2.2.1.tar.gz'
```

```
Resolving superb-west.dl.sourceforge.net... 209.160.59.253
Connecting to superb-west.dl.sourceforge.net|209.160.59.253|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 26,633,490 (25M) [application/x-gzip]
24% [========>                            ] 6,533,049    252.80K/s    ETA 01:32
```

Raspakirajte *ISPConfig-archive* koristeći naredbu:

```
# tar xvfz ISPConfig*.tar.gz
```

Stvara se poddirektorij *install_ispconfig*. Promijenite ga u */root/install_ispconfig*. Otvorite datoteku *dist.txt* i pogledajte odgovaraju li zadane vrijednosti vašem Linux poslužitelju.

Za Debian 3.1. vrijednosti u *dist.txt* izgledaju poput ovih:

```
dist_init_scripts=/etc/init.d ##       # debian31
dist_runlevel=/etc ##                  # debian31
```

Datoteka sadrži 19 dodatnih vrijednosti za Debian koje nismo ovdje popisali. Sve dok ne steknete određenu količinu znanja o administriranju Linuxa i korištenju ISPConfiga, držite se podrazumijevanih vrijednosti. Trebale bi raditi sve dok koristite jednu od podržanih distribucija popisanih ranije u ovom poglavlju. Administratorski znalci mogu mijenjati vrijednosti, sve dok je očuvan format datoteke.

Sada započnite s instaliranjem. Pokrenite instalaciju koristeći naredbu *./setup* iz *root* odzivnika. Instalacijske skripte započet će s prevođenjem Apachea i PHP-a 5 koji će se izvoditi na ulazu 81. Prvo ćete biti zamoljeni da odaberete jezik:

```
server2:~/install_ispconfig # ./setup
SuSE 10.0
Neuinstallation eines ISPConfig-Systems. / Installation of a new ISPConfig system. /
Installation d'ISPConfig sur un nouveau systeme.
Whlen Sie Ihre Sprache (deutsch/englisch/spanisch/franzsisch/italienisch/
niederlndisch/polnisch/
schwedisch): / Please choose your language (German/English/Spanish/French/Italian/
Dutch/Polish/Swedish): / Merci de choisir votre langue (Allemand/Anglais/Espagnol/
Francais/Italien/Nerlandais/Polonais/Sudois):
1) de
2) en
3) es
4) fr
5) it
6) nl
7) pl
8) se
Ihre Wahl: / Your Choice: / Votre Choix:
```

Vidjet ćete zaslon s upozorenjem:

```
With the system installation, some system files are replaced where adjustments were
made. This can lead to loss of entries in httpd.conf, named.conf as well as in the
Sendmail configuration.
Do you want to continue with the installation? [y/n] y
```

Sustav će prikazati tekst licence. Pročitajte ga i prihvatite:

```
Do you accept the license? [y/n] y
```

Instalacijski program nastavit će vam postavljati pitanja vezana uz sistemske postavke (npr. koji agent za prijenos pošte, FTP poslužitelj, Web poslužitelj i dnevnik koristiti i tako dalje). Budući da ste te pakete već instalirali na susta, trebali biste znati odgovoriti na sva pitanja.

Tijekom rane faze instalacije skripta će vas upitati u kojem režimu želite pokrenuti instalaciju. Odaberite režim Expert:

```
1) standard
2) expert
Your choice: 2
```

U tom režimu moći ćete zadati dodatne opcije kojima ISPConfig dodjeljuje podrazumijevane vrijednosti u standardnom režimu.

```
When prompted for a default directory, you can choose any directory you like, but
make sure it is on a partition with enough disk space for the web sites you plan to
host. Furthermore, if you want to configure quotas with ISPConfig, make sure you
enabled quotas for that partition as described in Chapter 2. If you want to enable
suExec for web sites that are allowed to run Perl/CGI scripts, the directory should
be within suExec's document root. On Debian and Fedora/Red Hat, suExec's default
document root is/var/www, while on SUSE it's/srv/www. If you're enabling suExec,
the document root is a good choice for the directory in which to put ISPConfig:
########## WEB SERVER ##########
Checking for program httpd...
/usr/sbin/httpd
OK
Checking the syntax of the httpd.conf...
Syntax OK
The syntax is ok!
Web-Root: /home/www
Is this correct? [y/n]n
Web-Root:/var/www
```

suExec je sigurnosno poboljšanje na Web poslužitelju koje zahtijeva da CGI skripte posjeduju i pokreću određeni korisnici.

Instalacija sada započinje s prevođenjem Apache poslužitelja koji će prikazivati Web sučelje ISPConfiga na ulazu 81. Kada je izgradnja Apachea za ISPConfig kompletirana, vidjet ćete preveden prilagođeni SSL certifikat. Instalacijski program će zatražiti nekoliko vrijednosti. Možete prihvatiti podrazumijevane vrijednosti ili zadati vlastite. Zaslon će izgledati poput ovoga:

```
SSL Certificate Generation Utility (mkcert.sh)
Copyright (c) 1998-2000 Ralf S. Engelschall, All Rights Reserved.
Generating custom certificate signed by own CA [CUSTOM]
```

---

```
STEP 0: Decide the signature algorithm used for certificates
The generated X.509 certificates can contain either
RSA or DSA based ingredients. Select the one you want to use.
Signature Algorithm ((R)SA or (D)SA) [R]:

_____________________________________________________________
STEP 1: Generating RSA private key for CA (1024 bit) [ca.key]
1698765 semi-random bytes loaded
Generating RSA private key, 1024 bit long modulus
............++++++
.........++++++
e is 65537 (0x10001)

_____________________________________________________________
STEP 2: Generating X.509 certificate signing request for CA [ca.csr]
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
1. Country Name             (2 letter code)  [XY]:
2. State or Province Name   (full name)      [Snake Desert]:
3. Locality Name            (e.g, city)      [Snake Town]:
4. Organization Name        (e.g, company)   [Snake Oil, Ltd]:
5. Organizational Unit Name (e.g, section)   [Certificate Authority]:
6. Common Name              (eg, CA name)    [Snake Oil CA]:
7. Email Address            (e.g, name@FQDN) [ca@snakeoil.dom]:
8. Certificate Validity     (days)           [365]:

_____________________________________________________________
STEP 3: Generating X.509 certificate for CA signed by itself [ca.crt]
Certificate Version (1 or 3) [3]:
Signature ok
subject=/C=XY/ST=Snake Desert/L=Snake Town/O=Snake Oil, Ltd/OU=Certificate Authority/
CN=Snake Oil CA/emailAddress=ca@snakeoil.dom
Getting Private key
Verify: matching certificate & key modulus
Verify: matching certificate signature
../conf/ssl.crt/ca.crt: /C=XY/ST=Snake Desert/L=Snake Town/O=Snake Oil, Ltd/
OU=Certificate Authority/CN=Snake Oil CA/emailAddress=ca@snakeoil.dom
error 18 at 0 depth lookup:self signed certificate
OK

_____________________________________________________________
STEP 4: Generating RSA private key for SERVER (1024 bit) [server.key]
1698765 semi-random bytes loaded
Generating RSA private key, 1024 bit long modulus
..........................++++++
.............++++++
e is 65537 (0x10001)

_____________________________________________________________
STEP 5: Generating X.509 certificate signing request for SERVER [server.csr]
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
```

```
If you enter '.', the field will be left blank.
1. Country Name             (2 letter code) [XY]:
2. State or Province Name   (full name)     [Snake Desert]:
3. Locality Name            (eg, city)      [Snake Town]:
4. Organization Name        (eg, company)   [Snake Oil, Ltd]:
5. Organizational Unit Name (eg, section)   [Webserver Team]:
6. Common Name              (eg, FQDN)      [www.snakeoil.dom]:
7. Email Address            (eg, name@fqdn) [www@snakeoil.dom]:
8. Certificate Validity     (days)          [365]:
_____________________________________________________________
STEP 6: Generating X.509 certificate signed by own CA [server.crt]
Certificate Version (1 or 3) [3]:
Signature ok
subject=/C=XY/ST=Snake Desert/L=Snake Town/O=Snake Oil, Ltd/OU=Webserver Team/CN=www.
snakeoil.dom/emailAddress=www@snakeoil.dom
Getting CA Private Key
Verify: matching certificate signature
../conf/ssl.crt/server.crt: OK
```

U koracima 7 i 8 procesa izrade certifikata bit ćete upitani želite li sada šifrirati odgovarajuće ključeve:

```
_____________________________________________________________
STEP 7: Enrypting RSA private key of CA with a pass phrase for security [ca.key]
The contents of the ca.key file (the generated private key) has to be
kept secret. So we strongly recommend you to encrypt the server.key file
with a Triple-DES cipher and a Pass Phrase.
Encrypt the private key now? [Y/n]:n
writing RSA key
Enter PEM pass phrase:
Verifying - Enter PEM pass phrase:
Fine, you're using an encrypted private key.
_____________________________________________________________
STEP 8: Enrypting RSA private key of SERVER with a pass phrase for security [server.
key]
The contents of the server.key file (the generated private key) has to be
kept secret. So we strongly recommend you to encrypt the server.key file
with a Triple-DES cipher and a Pass Phrase.
Encrypt the private key now? (Y/n):n
What email address or URL should be used in the suspected-spam report text for users
who want more information on your filter installation?
(In particular, ISPs should change this to a local Postmaster contact)
default text: [the administrator of that system]
```

Na pitanja odgovorite s n. U protivnom će vas program pitati za lozinku kada god budete pokretali ISPConfig sustav. To ujedno znači da se neće moći ponovno pokrenuti bez intervencije čovjeka.

Ako prevođenje ne uspije, instaliranje će se prekinuti i sve prevedene datoteke će biti obrisane. Poruka o pogrešci koju ćete dobiti trebala bi upućivati na uzrok neuspjeha. U većini slučajeva nedostaju glavne datoteke za paket.

Koji god da je razlog, pogledajte postavke poslužitelja i riješite problem. Ako direktorij *install_ispconfig* nije obrisan unatoč pogrešci, obrišite ga ručno.

Zatim opet otpakirajte arhivu ISPConfig, prijeđite u novi direktorij *install_ispconfig* i pokrenite *./setup*. Ne možete ponovno instalirati ISPConfig iz istog *install_ispconfig* direktorija nakon što se dogodila bilo kakva pogreška.

Slično tome, ako bilo koji od zahtijevanih paketa nije prisutan, instalacijska rutina će biti zaustavljena. Instalirajte pakete koji nedostaju, obrišite direktorij *install_ispconfig*, raspakirajte ponovno ISPConfig i počnite ispočetka.

Instalacijska skripta provjerava sintaksu postojeće Apache konfiguracijske datoteke. Bilo koja pogreška uzrokovat će prekidanje ISPConfig instalacije.

Ako su zadovoljeni svi uvjeti, trebat ćete navesti vrijednosti za vrijeme instalacije. To uključuje:

```
Please enter your MySQL server: localhost
Please enter your MySQL user: root
Please enter your MySQL password: vaša MySQL lozinka
Please enter a name for the ISPConfig database: ispconfigdb
Please enter the IP address of the ISPConfig web: 192.168.0.1
Please enter the host name: www
Please enter the domain: xyz.de
```

Nadalje, konfiguracijski program će vas pitati koji protokol želite koristiti. Odaberite stavku 2, HTTP:

```
Please select the protocol (http or https (SSL encryption)) to use to access the
ISPConfig system:
1) HTTPS
2) HTTP
Your Choice: 2
```

Vidjet ćete da sustav izvodi završne skripte i ponovno pokreće neke servise:

```
Connected successfully to MySQL server
ls: /etc/apache2/vhosts.d/*.conf: No such file or directory
Restarting some services...
which: no apachectl in (/sbin:/usr/sbin:/usr/local/sbin:/root/bin:/usr/local/bin:/
bin:/usr/bin:/usr/X11R6/bin:/usr/local/libexec)
Shutting down mail service (Postfix)                                    done
Starting mail service (Postfix)                                         done
Shutting down mail service (Postfix)                                    done
Starting mail service (Postfix)                                         done
Shutting down ProFTPD Server:                                           done
Starting ProFTPD Server:  - warning: "ProFTPD" address/port (70.253.158.45:21)
alreadyinuseby"ProFTPDDefaultInstallation"
done
Shutting down ProFTPD Server:                                           done
Starting ProFTPD Server:  - warning: "ProFTPD" address/port (70.253.158.45:21)
already in use by "ProFTPD Default Installation"
                                                                        done
Starting ISPConfig system...
/root/ispconfig/httpd/bin/apachectl startssl: httpd started
ISPConfig system is now up and running!
```

Programeri su završili instalacijsku skriptu s ovom porukom:

```
Congratulations! Your ISPConfig system is now installed.If you had to install quota,
please take the steps described in the installation manual. Otherwise your system is
now available without reboot.
```

Sada možete u preglednik unijeti IP adresu ili ime poslužitelja iza kojeg ćete dodati `:81` kako biste pristupili ISPConfigovom ekranu za prijavljivanje.

## Struktura direktorija ISPConfig

Kako je već napomenuto, glavni direktorij koji postavlja ISPConfig zove se *ispconfig* i smješten je u direktoriju u kojem ste ga izgrađivali sustav (u ovoj knjizi to je direktorij */root*). Vidjet ćete još jedan direktorij u */home* koji se zove *admispconfig*. Svaki direktorij sadrži datoteke potrebne za neovisno izvođenje ISPConfiga.

Pogledajmo prvo ovaj direktorij */root/ispconfig*:

```
-rwxr-xr-x   1 root root 33660 2006-04-26 12:28 cronolog
-rwxr-xr-x   1 root root  9673 2006-04-26 12:28 cronosplit
drwxr-xr-x  12 root root  4096 2006-04-26 09:55 httpd
drwxr-xr-x  12 root root  4096 2006-04-26 12:28 isp
-rw-r--r--   1 root root     8 2006-04-26 13:54 .old_path_httpd_root
drwxr-xr-x   6 root root  4096 2006-04-26 09:50 openssl
drwxr-xr-x   6 root root  4096 2006-04-26 10:00 php
drwxr-xr-x   4 root root  4096 2006-04-26 12:28 scripts
drwxr-xr-x   4 root root  4096 2006-04-26 12:28 standard_cgis
drwxr-xr-x   2 root root  4096 2006-04-26 12:28 sv
-rwx------   1 root root  9389 2006-04-26 12:28 uninstall
```

On sadrži ISPConfigove konfiguracijske datoteke za Apache, PHP i OpenSSL te predložke za sve vrste konfiguracijskih datoteka (za Apache, Postfix, Sendmail, BIND, *procmail* upute itd.). ISPConfig koristi te predloške za pisanje konfiguracijskih datoteka za servise koje konfigurira.

Također ćete vidjeti mnogo PHP klasa koje pružaju funkcije za pisanje konfiguracijskih datoteka sustava. Ukratko */root/ispconfig* je „strojarnica“ ISPConfiga.

U direktoriju */home/admispconfig* vidjet ćete skup drugih direktorija:

```
-rwxr-xr-x  1 admispconfig admispconfig   24 2006-04-26 12:28 .forward
drwxr-xr-x  8 admispconfig admispconfig 4096 2006-04-26 13:53 ispconfig
drwxr-xr-x  2 admispconfig admispconfig 4096 2006-04-26 12:28 mailstats
-rwxr-xr-x  1 admispconfig admispconfig  176 2006-04-26 12:28 .procmailrc
```

Oni sadrže ISPConfig Web sučelje te alate kao što su SpamAssassin (*http://spamassassin.org*) i ClamAV (*http://clamav.elektrapro.com)*. Možete ih konfigurirati kroz ISPConfig kao zaštitu protiv neželjenih poruka i virusa.

# Postavljanje poslužitelja i korisnika s ISPConfigom

Postavljanje Web lokacije jedan je od prvih koraka koje morate učiniti kako biste imali potpuni internetski poslužitelj. U ovom dijelu provest ćemo vas kroz sve bitne korake tog postupka.

Ako se pitate zašto vas jednostavno ne zamolimo da Web lokaciju ISPConfig i pročitate upute, uzmite u obzir sljedeće: razvojni tim ISPConfiga piše dokumentaciju za pružatelje internetskih usluga koji se bave udomljavanjem Web lokacija korisnika. Ako ćete i vi koristiti ISPConfig za tu svrhu, preporučujemo da pročitate upute na stranici *http://ispconfig.org*. U protivnom, pretpostavljamo da će vaš poslužitelj imati jednog administriratora koji će sam upravljati servisima sigurnih Web stranica, pošte i FTP-a.

ISPConfig zahtijeva da postavite *klijenta* koji će imati jednu ili više internetskih domena. U našem primjeru, postavit ćemo jednostavnog klijenta (jednog od autora ove knjige) koji će održavati četiri domene:

- *centralsoft.org*
- *linuxnewswire.org*
- *opensourcetoday.org*
- *tadelstein.com*

Kada pogledate u direktorij */var/www* vidjet ćete kako je ISPConfig postavio domene:

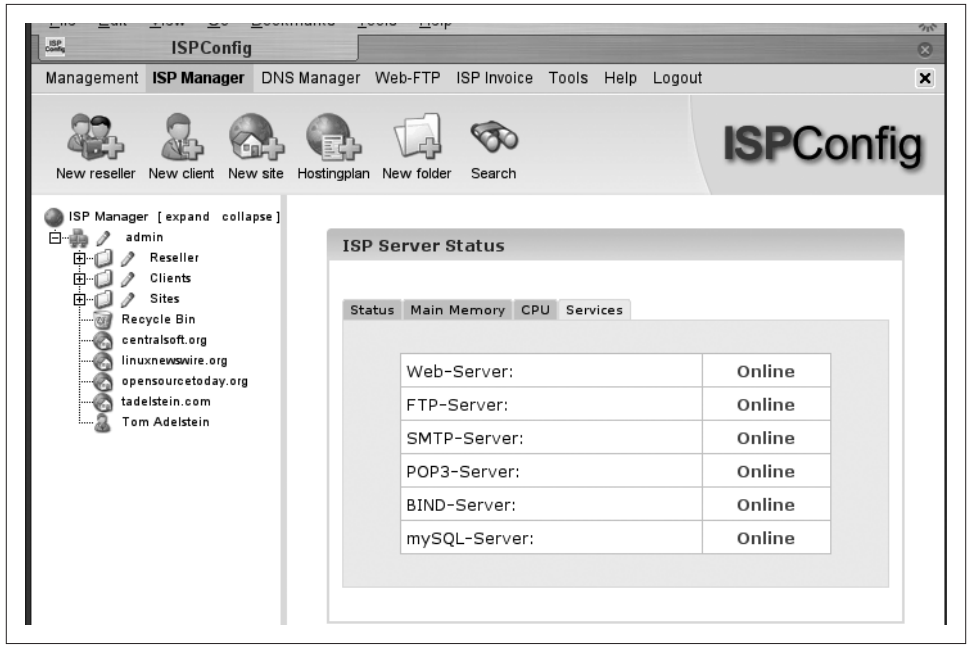
```
$ ls -a
apache2-default  sharedip  web2  web4  webalizer    www.opensourcetoday.org
localhost        web1      web3    www.centralsoft.org  www.linuxnewswire.org
www.tadelstein.com
```

Usporedite ovaj ispis s popisom Web lokacija na slici 4-1. Svaka Web lokacija sadrži direktorij. Direktoriji *www* čija imena odražavaju domene (npr. *www.opensourcetoday.org* ) su simboličke veze prema onome što sustav prepoznaje kao *web1*, *web2* itd.

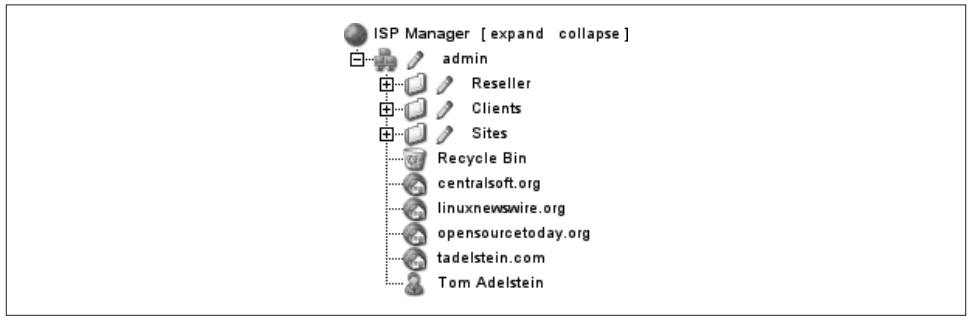
Na slici 4-2 možete bolje vidjeti popis domena. Primijetite da se na slici 4-2 domena pojavljuje za svaki direktorij u ispisu.

## Dodavanje klijenata i Web lokacija

Da biste konfigurirali klijenta i domene, morate se prvo prijaviti na ISPConfigovo sučelje. U Web preglednik unesite IP adresu poslužitelja i nakon nje broj ulaza za ISPConfig – u našem slučaju :81: *http://70.253.158.45:81* (koristite *https://* ako ste odabrali HTTPS kao ISPConfigov protokol prilikom instalacije). Na zaslonu za prijavljivanje na sustav (slika 4-3) unesite korisničko ime *admin* i lozinku *admin*.




*Slika 4-1. Sučelje ISP Manager*



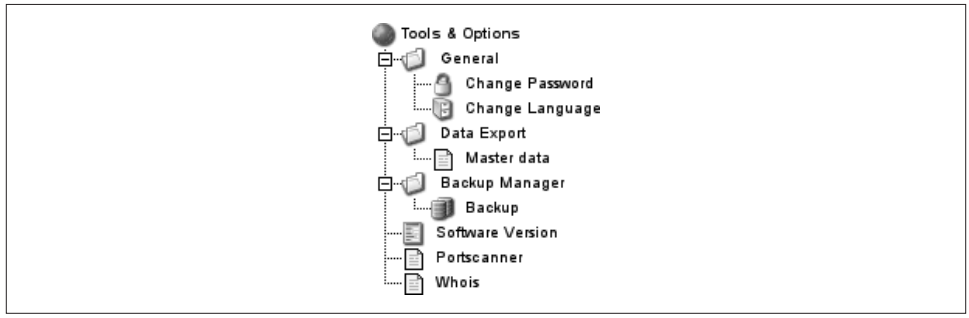

*Slika 4-2. Popis domena u ISP Manageru*

Nadalje, odmah promijenite lozinku kako biste ju znali samo vi. Da biste to učinili, odaberite Tools s alatne vrpce i pritisnite na simbol za lozinku (slika 4-4).

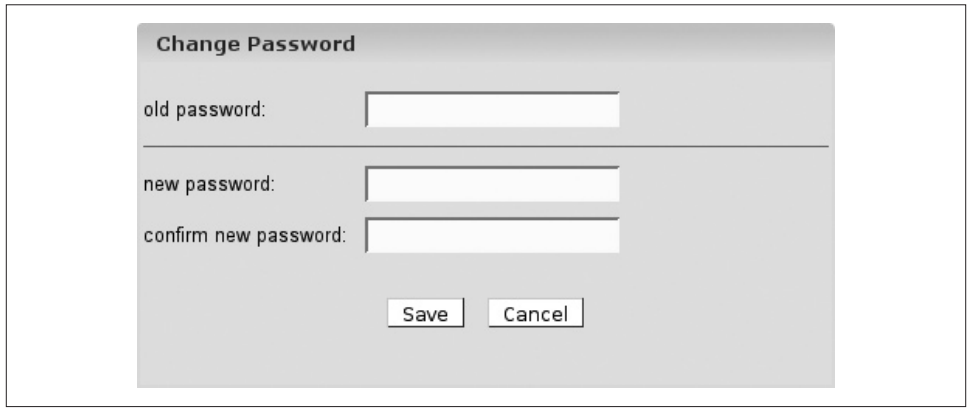
Pojavit će se dijalog za promjenu lozinke, prikazan na slici 4-5, koji trebate ispuniti.

Odjavite se i ponovno prijavite na sustav s novom lozinkom.

*Slika 4-3. ISPConfigov zaslon za prijavljivanje na sustav*

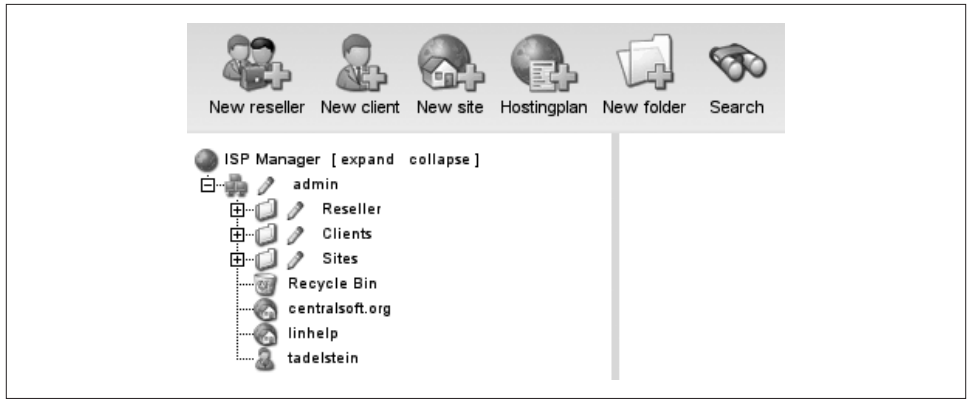



*Slika 4-4. Alatna vrpca*



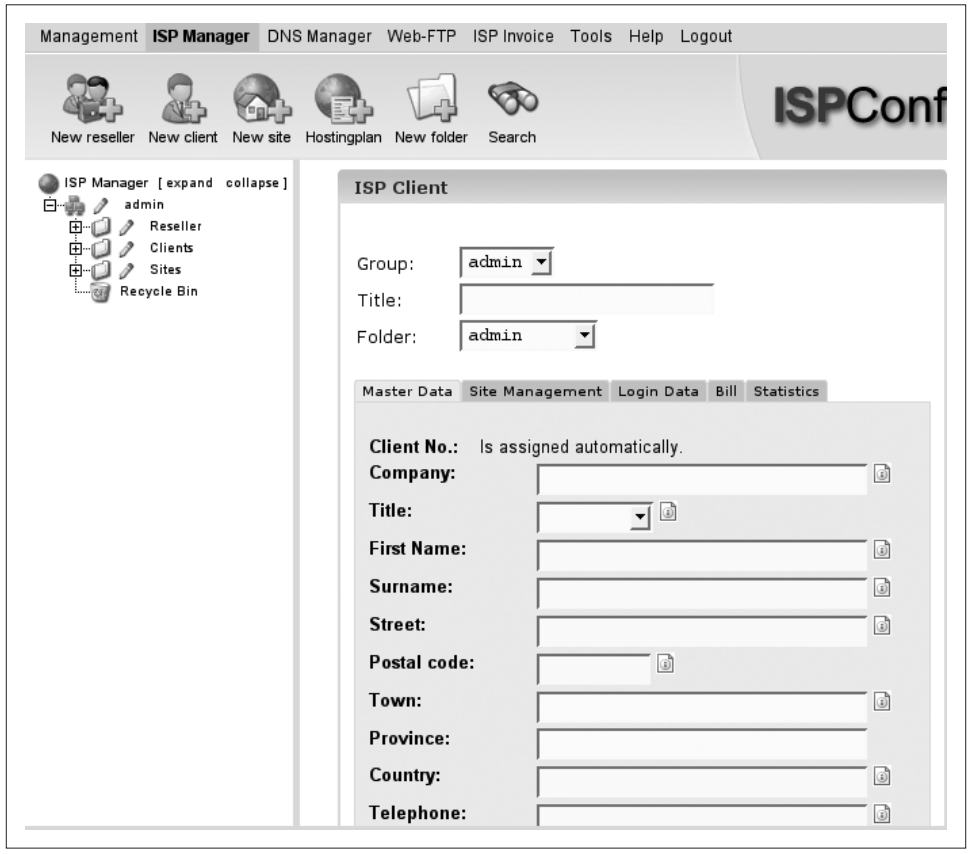

*Slika 4-5. ISPConfig dijalog za promjenu lozinke*

Prije neko što budete mogli postaviti Web lokaciju, morat ćete definirati vlasnika lokacije. Osaberite ISP Manager na vrhu alatne vrpce. Vidjet ćete navigacijski izbornik poput onog na slici 4-6.



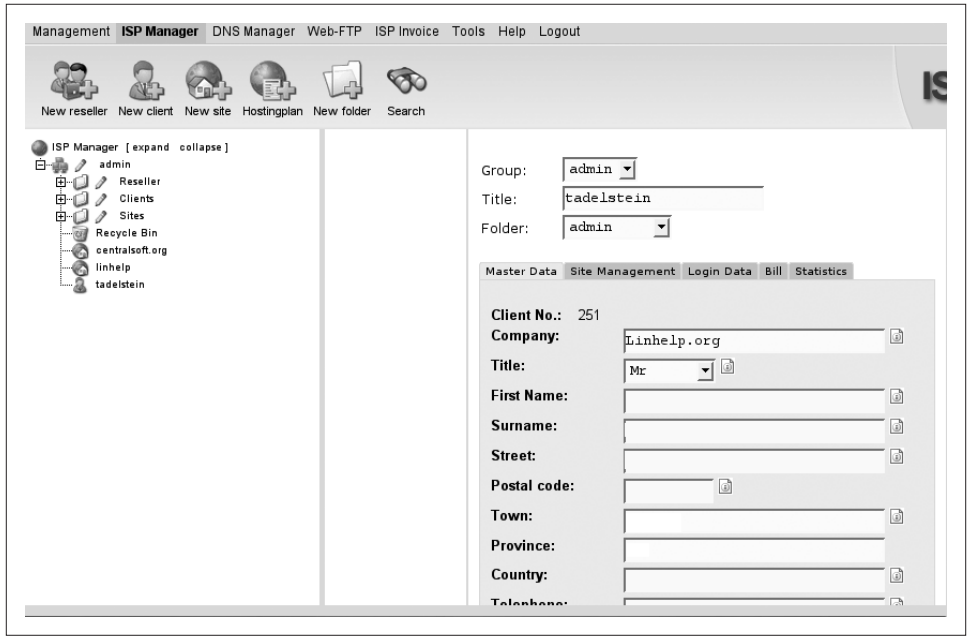

*Slika 4-6. Izbornik ISP Manager s dodanim klijentom i domenom*

Pogledajmo sada kako smo dodali klijenta *tadelstein* i Web lokaciju *linhelp*. Odaberite New Client s izbornika ISPConfig Manager. Vidjet ćete dijalog poput onog na slici 4-7.




*Slika 4-7. Obrazac za unos informacija o klijentu*

Unesite relevantne informacije o klijentu. Slika 4-8 prikazuje kako smo mi ispunili formu. Primijetite da smo koristili *Linhelp.org* kao ime tvrtke.




*Slika 4-8. Ispunjen obrazac za administrativnog klijenta*

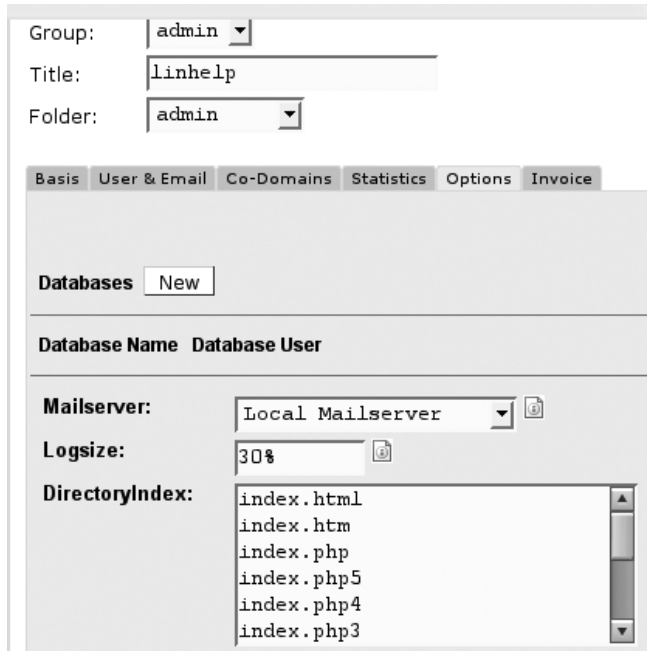
Na lijevoj strani navigacijskog izbornika vidjet ćete novu ikonu koja predstavlja osobu, popraćenu imenom klijenta. Sada možete postaviti Web lokaciju. Jednostavno pritisnite New site na alatnoj vrpci i vidjet ćete dijalog prikazan na slici 4-9.

Zadajte ime i IP adresu Web lokacije te izradite DNS zapis. Zamijetite i kartice na obrascu, duž područja u kojem ste unijeli ime lokacije:

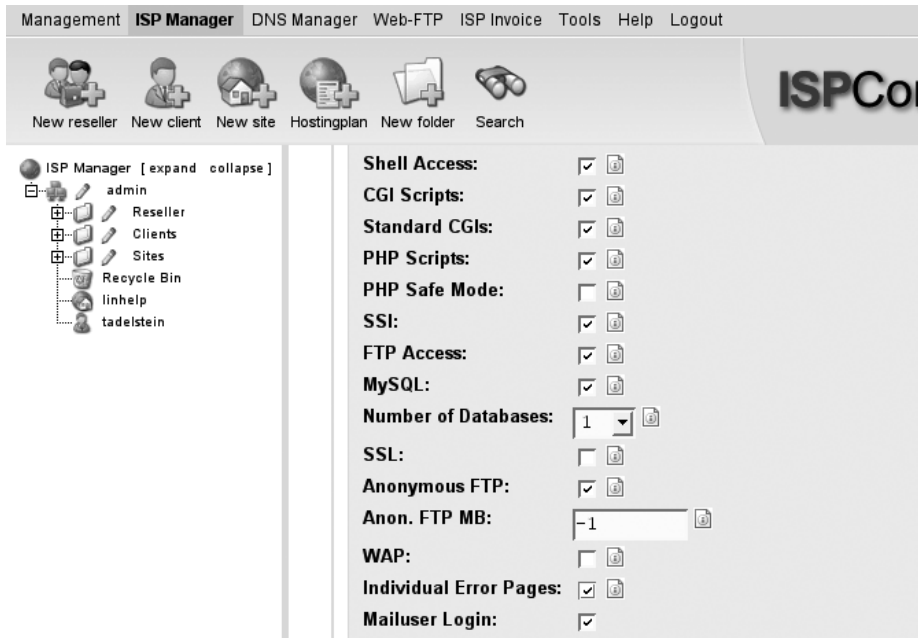
- Basis
- User & Email
- Co-Domains
- Statistics
- Options
- Invoice

Svaka kartica pruža različite konfiguracijske i upravljačke funkcije.

Slika 4-9 ne prikazuje sve opcije na kartici Basis. Također ćete pronaći nekoliko drugih mogućnosti koje možete dodijeliti administratoru lokacije. Za našu lokaciju pružit ćemo pristup školjci, bazi podataka, FTP i opcije prijavljivanja na sustav, kao što je prikazano na slici 4-10.



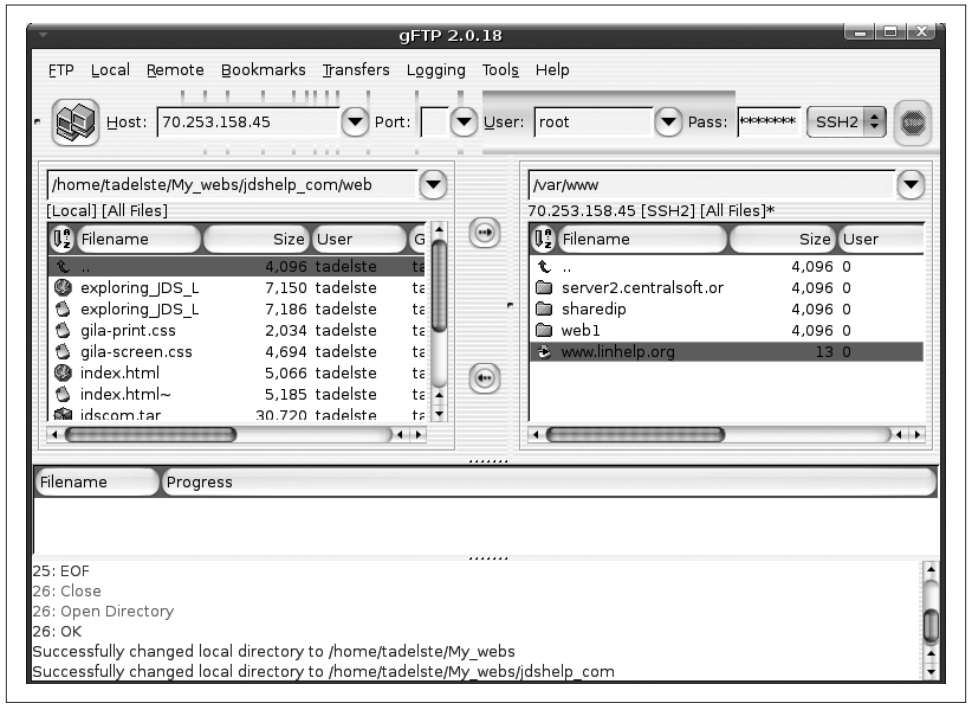

*Slika 4-9. Obrazac korišten za postavljanje Web lokacije linhelp.org*




*Slika 4-10. Opcije Web lokacije*

Zamijetite da je na slici 4-10 u polju Anon. FTP MB zadana podrazumijevana vrijednost -1. To omogućava lokaciji da pruži neograničeni prostor za FTP prijenos na disku. Možda želite osigurati takav pristup ako zrcalite lokaciju za preuzimanje dattoeka. U suprotnom, može vam više odgovarati da postavite ograničenja tako da nitko ne može spremiti previše podataka i zauzeti prostor na disku koji koriste i druge usluge.

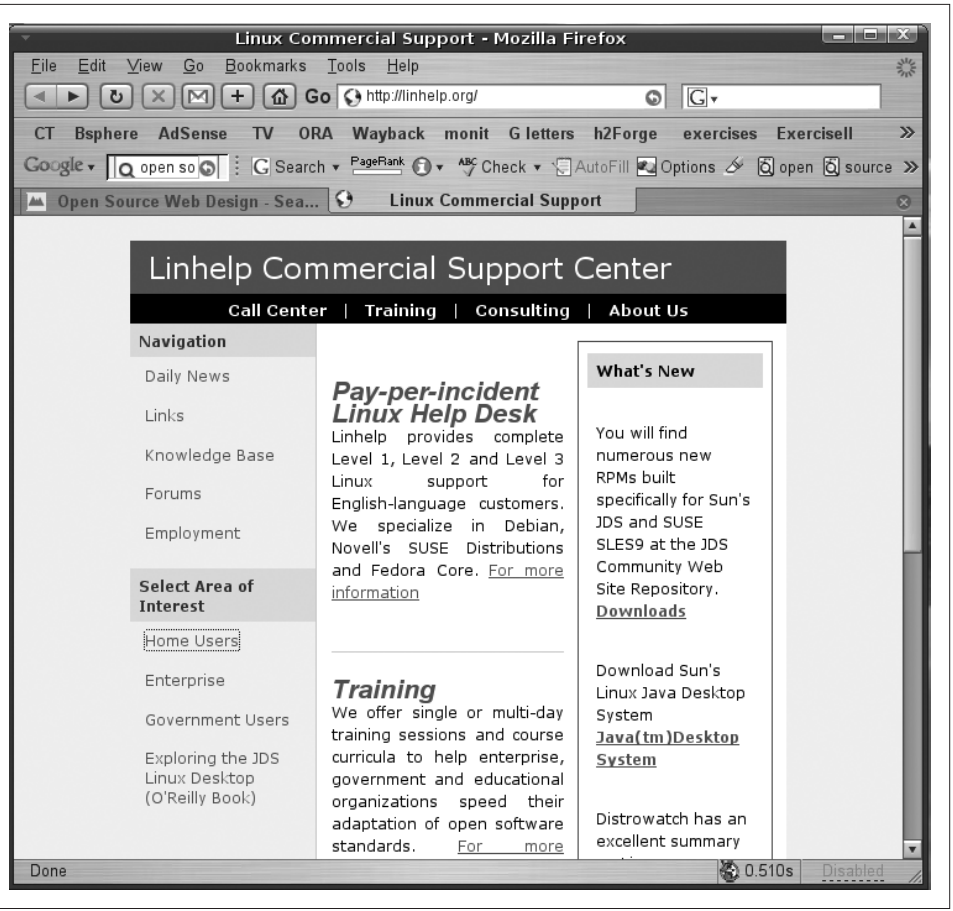
U ovom trenutku imate Web lokaciju koju možete koristiti. Jednostavan način da joj dodate stranice jeste upotreba FTP klijenta, poput *gftp* sa grafičkim korisničkim sučeljem, za prijenos stranica koje ste izradili i smjestili u mapu na radnoj površini, kao što je prikazano na slici 4-11.




*Slika 4-11. Upotreba programa gftp za prijenos datoteka u korijenski direktorij linhelp.org*

Ako preglednik usmjerite na *http://linhelp.org* prikazat će se početna stranica *index.html*. Na slici 4-12 možete vidjeti kako izgleda.

ISPConfig koristi hijerarhijski model s direktorijem */var/www/web1/Web* kao korijenom za ulaz 80. U svakom direktoriju koji izradite na ovoj putanji, Apache stvara drugu granu u koju možete smještati stranice. Podrazumijevano je da, kada preglednik zatraži direktorij, Apache traži i isporučuje HTML datoteku koja se zove *index.html*. Ako ju ne pronađe, prikazat će imena datoteka i direktorija koji se nalaze u korijenskom direktoriju.



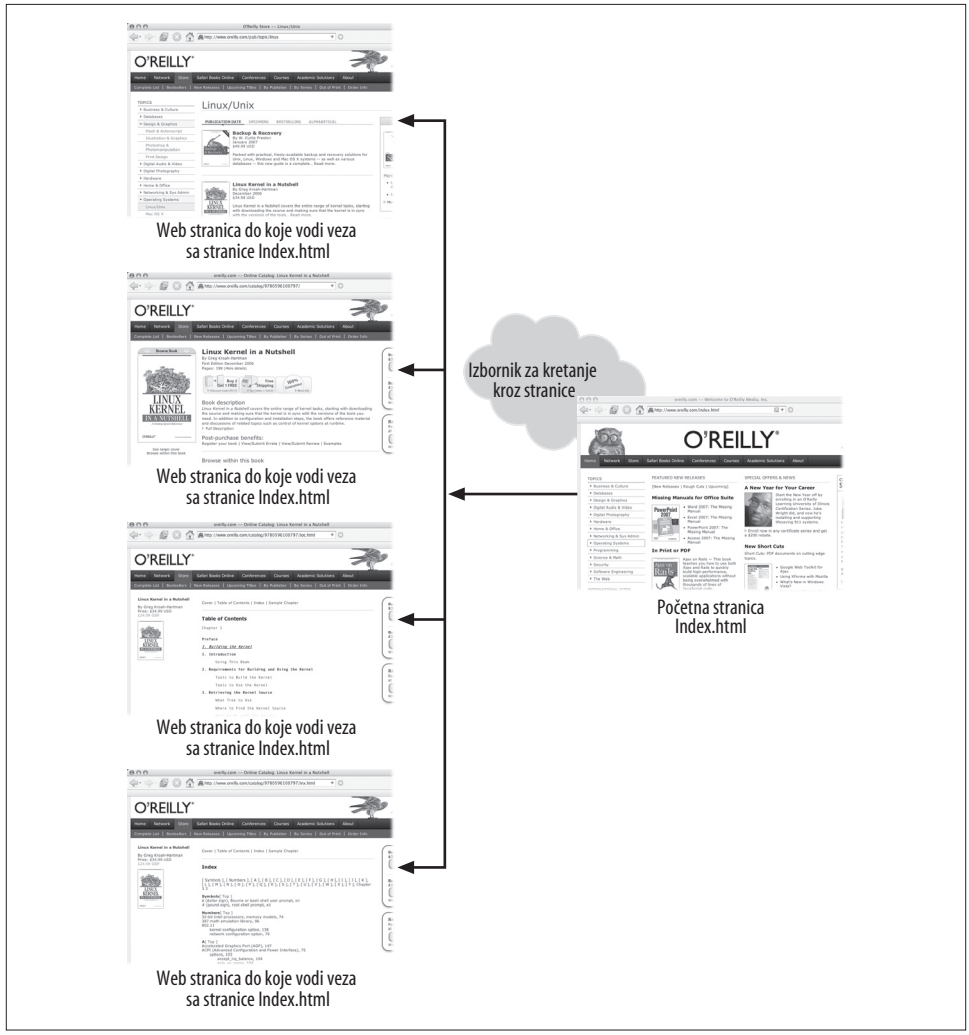

*Slika 4-12. Linhelp.org Web lokacija u pregledniku Firefox*

Slika 4-12 pruža primjer korijenskog direktorija Web lokacije. Početna (Home) stranica prikazuje se kada preglednik zada ime direktorija, zato što ima podrazumijevani naslov. Ona sadrži veze do ostalih stranica lokacije.

Primjer dijagrama na slici 4-13 mogao bi se tumačiti kao dijagram toka. Aktualni kôd početne stranice izgledao bi otprilike ovako:

```
<a href="./about_us.html">About Us </a><br><br>
<a href="./products.html">Products </a><br><br>
<a href="./services.html">Services </a><br><br>
<a href="./support.html">Support </a><br><br>
```

Web tim s kojim radite najčešće će sam napraviti strukturu direktorija i Web stranice. Vjerojatno će koristiti i bazu podataka, no to je tema sljedećeg poglavlja. Za sada je dovoljno da znate kako napraviti Web lokaciju i vidljivost domene.



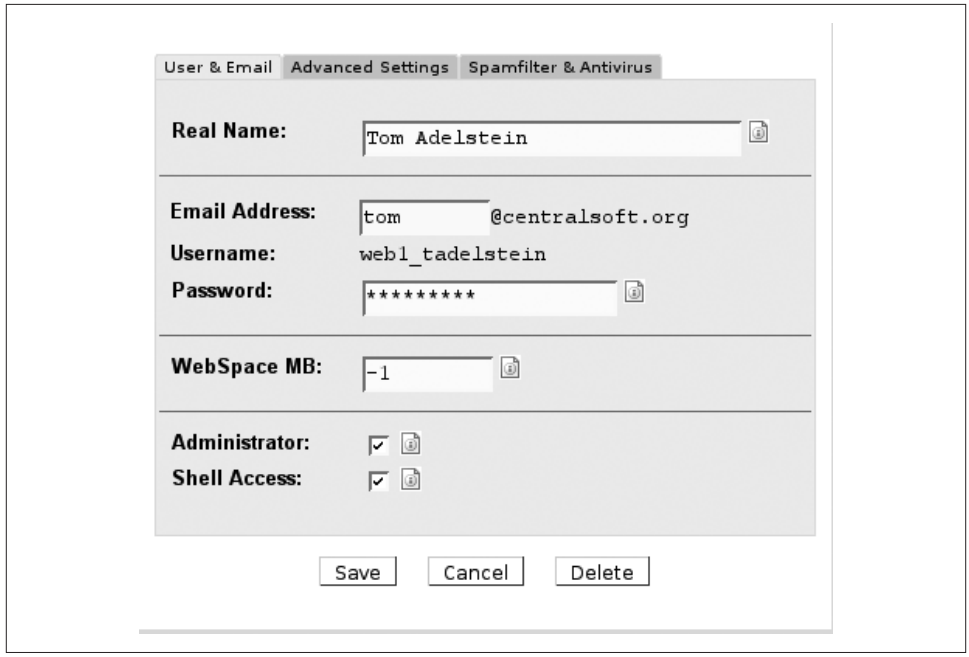

*Slika 4-13. Struktura jednostavne Web lokacije*

## Upravljanje korisnicima i elektroničkom poštom

Jedna od osnovnih zadaća administratora Linux sustava je upravljanje korisnicima i njihovim računima. To možete izvesti korištenjem ISPConfigovog kontrolnog panela.

Kada ste postavili svoje domene, odabir jedne od njih na ISP Manager dijelu alatne vrpce, otvorit će obrazac ISP Site prikazan na slici 4-9. Vratimo se i pogledajmo ga još jednom.

Obrazac ima šest kartica. Druga kartica s lijeve strane zove se Users & Email. Na toj kartici možete dodati nove korisnike te upravljati postojećim. Kada odaberete New, vidjet ćete drugi obrazac kao na slici 4-14.




*Slika 4-14. Obrazac korisnika*

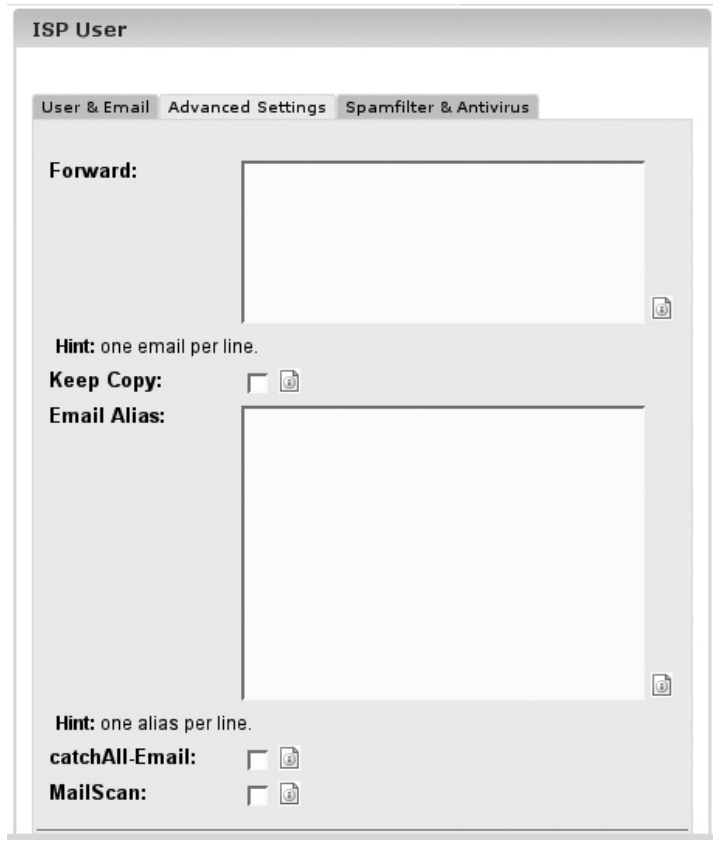
Na tom obrascu možete unijeti detalje o novim korisnicima i postaviti ograničenja upotrebe diskovnog prostora. Vrijednost -1 pruža neograničen prostor, no kvotama možete upravljati kako god želite.

Na kartici Advanced Settings (slika 4-15) možete koristiti polje Forward kako biste omogućili da elektronička poruka poslan korisniku bude proslijeđena na drugu adresu. Drugim riječima, ako korisnik ima primarnu adresu elektroničke pošte koju najradije koristi, ova opcija se može koristi da se poruke koje stižu na drugu adresu preusmjere na primarnu.

Ostale opcije na toj kartici su:

*Keep Copy*
: Ako odaberete ovu opciju, kopija proslijeđene pristigle poruke čuvat će se u korisnikovom lokalnom poštanskom sandučiću. To je korisno u slučaju da proslijeđene poruke ne stignu do ciljne adrese (zbog filtara neželjenih poruka ili nekih drugih problema)

*Email Alias*
: Ako ne želite izložiti javnosti korisnikov poštanski sandučić, posjetitelji lokacije mogu slati poštu na opću adresu kao što je *info@centralsoft.org* ili *webmaster@centralsoft.org*. To možete učiniti pružajući zamjenski korisnički račun.



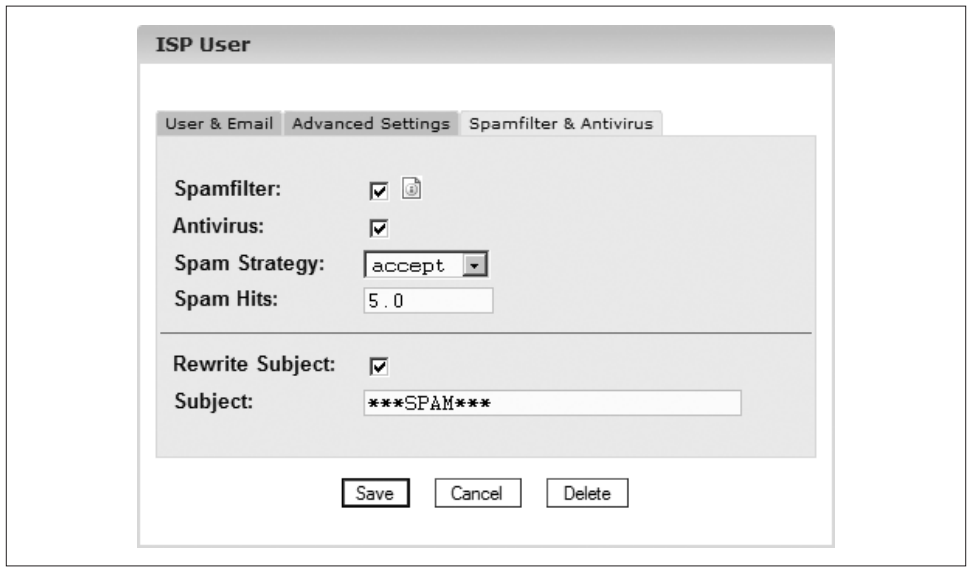

*Slika 4-15. Napredne opcije za elektroničku poštu*

*catchAll-Email*
: Ova opcija preusmjerava sve poruke poslane na nepostojeće adrese u zadani poštanski sandučić. Ljudi ponekad pišu na adrese poput *editor@centralsoft.org* ili *advertising@centralsoft.org* a da prije toga ne provjere postoje li uopće takvi računi. Sve takve poruke možete preusmjeriti u jedan postojeći poštanski sandučić.

*MailScan*
: Ako želite na poslužitelju skenirati elektroničke poruke kako biste provjerili sadrže li viruse ili JavaScript kôd, potvrdite ovo polje.

*Autoresponder*
: Ova opcija omogućava slanje automatskog odgovora na dolazne poruke adresirane na specifičnog korisnika. Najčešće se koristi za obavješćivanje pošiljatelja da je primatelj na odmoru i da neće moći odgovarati na poruke.

Na kartici Spamfilter & Antivirus, prikazanoj na slici 4-16, možete odabrati strategiju borbe protiv neželjenih poruka. Aktivirajte Spamfilter za korisnički račun i zatim ćete moći zadati i ponašanje filtra.




*Slika 4-16. Kartica Spamfilter & Antivirus*

Ako odaberete strategiju *accept,* dopustit ćete neželjenim porukama da budu isporučene u sandučić te da ih agent za slanje pošte sortira. Mnogi administratori vole odabrati ovu strategiju na samom početku, dok korisnik ima vlastitu bazu podataka za identificiranje poruka neželjene pošte. Nakon toga administrator može odabrati strategiju *discard* koja briše sa poslužitelja sve elektroničke poruke koje su identificirane kao neželjene.

Pogledajmo sada ostale opcija za borbu s neželjenim porukama elektroničke pošte:

*Spam Hints*
: Filtar neželjenih poruka izvodi skup testova nad dolazećim porukama i svaki test ocjenjuje bodovima. Ako je suma bodova tih testova blizu vrijednosti zadane u polju Spam Hints, ili ju prelazi, elektronička poruka se kategorizira kao neželjena i obrađuje se prema odabranoj strategiji za neželjene poruke.

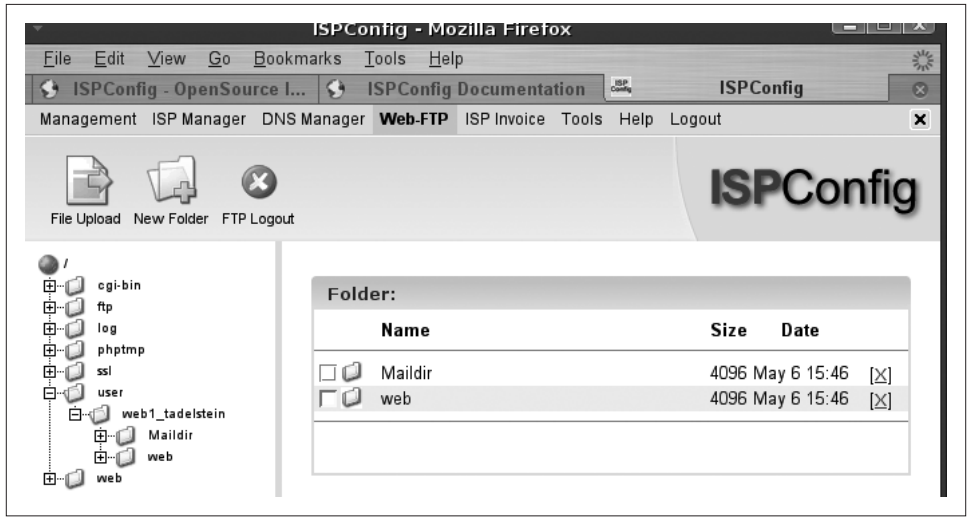
*Rewrite Subject*
: U modu *accept*, odabir ove opcije zadaje da se u polje Subject svake elektroničke poruke koja je identificirana kao neželjena doda prefiks (podrazumijevano je to ***SPAM***). To omogućuje korisniku da sortira potencijalno neželjene poruke u posebnu mapu na temelju polja Subject.

Kako biste omogućili korisniku da mijenja postavke svojeg računa (uključujući lozinku, filtere neželjene pošte i antivirusnu provjeru), morate odbrati opciju Mailuser za određenog korisnika na kartici Basic obrasca ISP User (slika 4-10). Da bi promijenio postavke, korisnik se može jednostavno prijaviti se na sustav s korisničkim imenom poput *http://centralsoft.org:81/mailuser.*

### Direktoriji user, email, home i web

Svaki korisnik domene pod ISPConfigom ima vlastiti direktorij u direktoriju *users*. Ako je na domeni dopušten FTP pristup, korisnici se odmah usmjeravaju u svoje početne direktorije kada se prijavljuju na sustav preko FTP-a. Svaki početni direktorij sadrži poddirektorij *Web* koju korisnik može vidjeti pristupajući URL-u *http://www.centralsoft.org/~*`korisnik` ili *http://www.centralsoft.org/users/*`korisnik`.

Slika 4-17 prikazuje strukturu početnog direktorija korisnika domene centralsoft.org.




*Slika 4-17. Direktorij korisnika lokacije*

## Konfiguriranje klijenta elektroničke pošte

Sada biste već trebali razumjeti osnove postavljanja Web lokacije, otvaranja korisničkih računa i upravljanja njima. No, vjerojatno ćete morati pomagati korisnicima i kada budu konfigurirali svoje klijente elektroničke pošte. Na našem sustavu, ISPConfig koristi *server1.centralsoft.org* kao odlazni, SMTP, poslužitelj te kao ulazni, POP3/IMAP, poslužitelj.

Kod većine modernih klijenata poruke elektroničke pošte možete odabrati opciju Transport Layer Security (TLS). Odaberite TLS kada je to moguće prilikom konfiguriranja odlaznog poslužitelja. Kako većina klijenata elektroničke pošte kao odlazni poslužitelj koristi SMPT poslužitelj svog pružatelja Internet usluga, možete odabrati TLS ako ga vaš pružatelj usluga koristi. U velikoj većini slučajeva, vaše korisničko ime i lozinka putovat će preko Interneta u obliku čistog teksta.

Za primanje poruka postavite dolazni poslužitelj (mi smo koristili *server1.centralsoft.org* ) i odaberite POP3 ili IMAP protokol. Navedite ime svojeg sistema (npr. *web1_adelstein*) i zadajte svoju adresu elektroničke pošte kao alias (npr. *tom@centralsoft.org*).

Ako dobijete poruku o pogrešci „-ERR Unknown AUTHORIZATION state command“ kada pokušate preuzeti poštu preko POP3 protokola, vjerojatno ste prethodno zaboravili aktivirati SSL/TLS enkripciju. Ponovno konfigurirajte klijent elekroničke pošte, aktivirajte POP3 preko SSL-a i pokušajte ponovno.

# Zaštita Linux Web poslužitelja

U današnjem poslovnom okruženju često se događaju nepredviđeni događaji. Zlonamjerne osobe skeniraju IP adrese tražeći žrtve. Koriste se sofisticiranima rječnicim lozinki kako bi pristupili poslužitelju, tako da mogu slati neželjene poruke, viruse i crve. Situacije s kojima se susreću administratori sustava vrlo su zahtjevne i teško se s preciznošću mogu kontrolirati. S obzirom na sve to, administratori se moraju učiti brzo prilagođavati novim (često neprijateljskim) situacijama.

Postoje dva načina prilogođavanja. Prvi, ako ste dovoljno svjesni situacije i možete predvidjeti neke situacije, možete poduzeti sve mjere opreza. To ćemo zvati *predviđanje*.

U drugim slučajevima morate se u trenutku prilagoditi novoj situaciji, bez vremena za pripremu. To podrazumijeva *improvizaciju*. Da biste bili potpuno prilagodljivi, morate biti sposobni i predviđati i improvizirati.

## Uloga servisa koji nadzire servise

Bez obzira na to koliko rigorozno radite na zaštiti svojeg poslužitelja, zbog neke misteriozne kombinacije okolnosti, ponekad se sustav može blokirati. U savršenom svijetu, mogli biste nadzirati svaki servis i sustav bi vas odmah obavijestio o bilo kakvom kvaru. No, ne živimo u takvom svijetu.

Zamislite da ste udomili poslužitelj kod pružatelja usluga 250 km daleko od vašeg ureda. Ako se taj poslužitelj blokira, netko će morati nazvati pružatelja usluga i zamoliti ga da vrati poslužitelj u funkcionalno stanje. Osobe iz tehničke podrške možda vam neće odmah biti na raspolaganju, pa ćete morati čekati, dok će se za to vrijeme na vašem poslužitelju izvoditi sumnjive aplikacije.

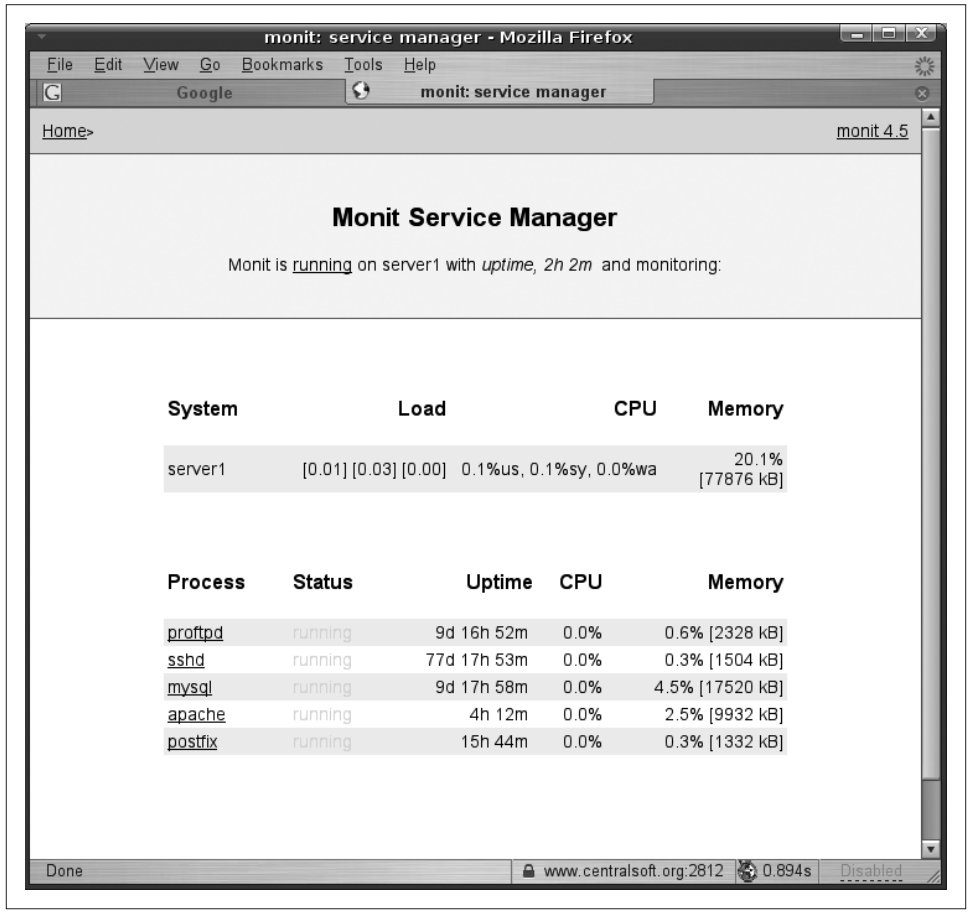
U velikim tvrtkama, možete se ponekad osjećati izolirano, baš kao kada je poslužitelj udaljen 250 km, iako se nalazi možda nekoliko katova ispod ili iznad vašeg ureda. Osoblje podatkovnih centara administratorima sustava rijetko dopušta pristup u prostoriju s poslužiteljima, pa je vrlo važno da administratori znaju kako mogu daljinski upravljati sustavom.

*Daemon-monitoring daemon* (DMD) je pomoćni program koji automatski nadzire vaše servise i automatski ih pokušava ponovno pokrenuti kada s njima nešto nije u redu. Ako se servis ne oporavi, morate se prijaviti na poslužitelj i otvoriti konzolu da biste zadali naredbu poput */etc/init.d/mysql restart*. DMD, međutim, može zadavati takve naredbe bez ikakve intervencije s vaše strane.

Ako se servis ponovno pokrene, to je kraj problema. No, ako se ne pokrene, DMD će ga pokušati pokrenuti nekoliko puta (primjerice, pet). Ako ni tada ne uspije, kontaktirat će vas putem tekstualne poruke, poruke elektroničke pošte ili na neki drugi način kako bi vam dao do znanja da postoji problem. U tom ćete trenutku morati intervenirati i otkriti što nije u redu s vašim servisom.

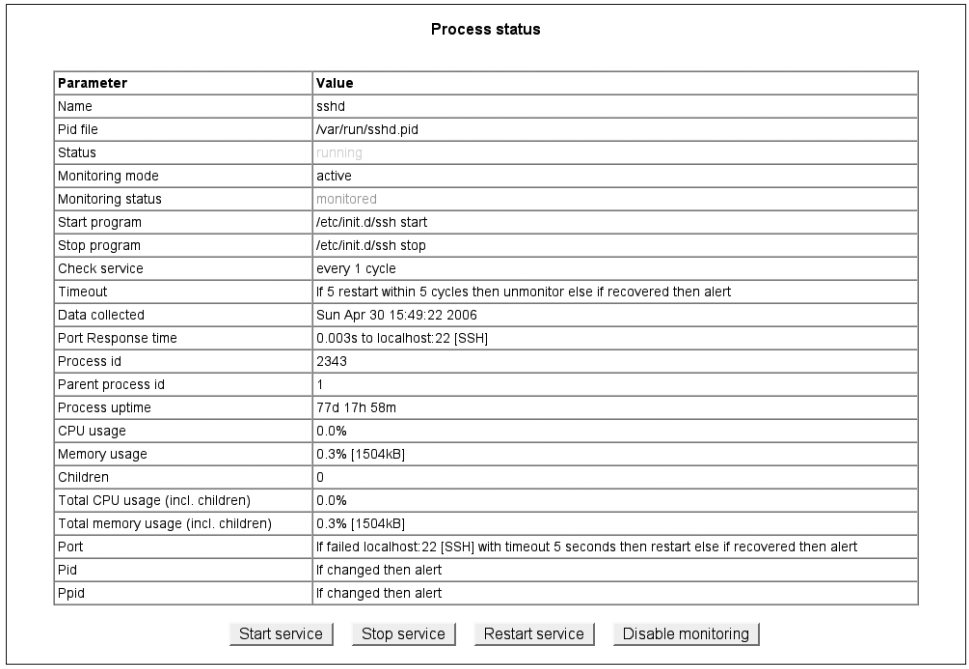
DMD radi kao bilo koji drugi servis na vašem sustavu. Ima konfiguracijsku datoteku koja vam omogućuje da izaberete opcije koje najbolje odgovaraju vašim potrebama. Može se pokretati zajedno sa sustavom ili ga možete ručno pokrenuti.

U nastavku ćemo postaviti DMD *monit* koji ima jednostavno Web sučelje prikazano na slici 4-18.




*Slika 4-18. Web sučelje za nadziranje rada poslužitelja centralsoft.org*

Obratite pozornost na pet servisa pod nadzorom. Na slici 4-19 pokazat ćemo kako sustav upravlja svakim procesom. U ovome slučaju nadzirat ćemo proces *sshd*.



Process status

| Parameter | Value |
|---|---|
| Name | sshd |
| Pid file | /var/run/sshd.pid |
| Status | running |
| Monitoring mode | active |
| Monitoring status | monitored |
| Start program | /etc/init.d/ssh start |
| Stop program | /etc/init.d/ssh stop |
| Check service | every 1 cycle |
| Timeout | If 5 restart within 5 cycles then unmonitor else if recovered then alert |
| Data collected | Sun Apr 30 15:49:22 2006 |
| Port Response time | 0.003s to localhost:22 [SSH] |
| Process id | 2343 |
| Parent process id | 1 |
| Process uptime | 77d 17h 58m |
| CPU usage | 0.0% |
| Memory usage | 0.3% [1504kB] |
| Children | 0 |
| Total CPU usage (incl. children) | 0.0% |
| Total memory usage (incl. children) | 0.3% [1504kB] |
| Port | If failed localhost:22 [SSH] with timeout 5 seconds then restart else if recovered then alert |
| Pid | If changed then alert |
| Ppid | If changed then alert |

Start service | Stop service | Restart service | Disable monitoring

*Slika 4-19. Detaljniji pregled postavki servisa sshd*

Primijetite na slici 4-19 da status servisa *sshd* pokazuje da on trenutno radi i da ga sustav nadzire. U zadnja tri reda prikazane tablice možete vidjeti upute o tome što treba učiniti ako se *sshd* blokira:

```
If failed localhost:22 [SSH] with timeout 5 seconds then restart else if recovered
then alert
```

Ovo pravilo jednostavno ponovno pokreće blokirani servis i šalje poruku kada u tome uspije.

Konačno, *monit* ima četiri gumba na dnu stranice za ručno interveniranje. Pogledajmo sada kako radi ovaj sustav.

## Instaliranje i konfiguriranje monita

Kako biste instalirali *monit*, možete koristiti upravitelj za Linux instalacijske pakete ili preuzeti arhivu sa *http://www.tildeslash.com/monit*. Ako koristite Debian sustav iz drugog poglavlja, jednostavno unesite:

```
# apt-get install monit
```

Nakon instalacije *monita*, uredite */etc/monit/monitrc*. Ta datoteka, izrađena za vrijeme instalacije, sadrži mnogo primjera. Dodatne konfiguracijske primjere možete pronaći na *http://www.tildeslash.com/monit/doc/examples.php*. U našem slučaju želimo:

- Osposobiti *monit* Web sučelje na ulazu 2812
- Nadzirati servise *proftpd*, *sshd*, *mysql*, *apache* i *postfix*
- Izraditi Secure Sockets Layer (*https*) Web sučelje na koje ćemo se moći prijaviti s korisničkim imenom *admin*.
- Zadati *monitu* da šalje poruke upozorenja na *root@localhost*.

Naša konfiguracijska datoteka */etc/monit/monitrc* izgleda ovako:

```
set daemon  60
set log file syslog facility log_daemon
set mailserver localhost
set mail-format { from: monit@server1.centralsoft.org }
set alert root@localhost
set httpd port 2812 and
     SSL ENABLE
     PEMFILE  /var/certs/monit.pem
     allow admin: test
check process proftpd with pidfile /var/run/proftpd.pid
   start program = "/etc/init.d/proftpd start"
   stop program  = "/etc/init.d/proftpd stop"
   if failed port 21 protocol ftp then restart
   if 5 restarts within 5 cycles then timeout
check process sshd with pidfile /var/run/sshd.pid
   start program  "/etc/init.d/ssh start"
   stop program  "/etc/init.d/ssh stop"
   if failed port 22 protocol ssh then restart
   if 5 restarts within 5 cycles then timeout
check process mysql with pidfile /var/run/mysqld/mysqld.pid
   group database
   start program = "/etc/init.d/mysql start"
   stop program = "/etc/init.d/mysql stop"
   if failed host 127.0.0.1 port 3306 then restart
   if 5 restarts within 5 cycles then timeout
check process apache with pidfile /var/run/apache2.pid
   group www
   start program = "/etc/init.d/apache2 start"
   stop program  = "/etc/init.d/apache2 stop"
   if failed host www.centralsoft.org port 80 protocol http
      and request "/monit/token" then restart
   if cpu is greater than 60% for 2 cycles then alert
   if cpu > 80% for 5 cycles then restart
   if totalmem > 500 MB for 5 cycles then restart
   if children > 250 then restart
   if loadavg(5min) greater than 10 for 8 cycles then stop
   if 3 restarts within 5 cycles then timeout
check process postfix with pidfile /var/spool/postfix/pid/master.pid
   group mail
   start program = "/etc/init.d/postfix start"
   stop  program = "/etc/init.d/postfix stop"
   if failed port 25 protocol smtp then restart
   if 5 restarts within 5 cycles then timeout
```

Iskazi i opcije opisane su u dokumentaciji dostupnoj na *http://tildeslash.com/monit/doc/manual.php*.

U odjeljku *apache* konfiguracijske datoteke, vidjet ćete ovaj iskaz:

```
if failed host www.centralsoft.org port 80 protocol http
      and request "/monit/token" then restart
```

To znači da se *monit* pokušava spojiti na *www.centralsoft.org* na ulazu 80 i pokušava pristupiti datoteci */monit/token*. Kako je korijen naše Web lokacije */var/www/www.centralsoft.org/Web*, ime datoteke se proširuje na */var/www/www.centralsoft.org/Web/monit/token*. Ako *monit* ne uspije, znači da ne radi Apache, pa će ga *monit* pokušati ponovno pokrenuti.

Sada moramo izraditi datoteku */var/www/www.centralsoft.org/Web/monit/token* i u nju upisati nešto proizvoljno:

```
# mkdir /var/www/www.centralsoft.org/web/monit
# echo "hello" > /var/www/www.centralsoft.org/web/monit/token
```

Sličnu proceduru možete slijediti i na vašem sustavu.

Nadalje, izradite direktorij koji će sadržavati datoteku sertifikata (*/var/certs/monit.pem*) koja je potrebna za SSL enkripciju Web sučelja *monita*:

```
# mkdir /var/certs
# cd /var/certs
```

Trebat će vam konfiguracijska datoteka za OpenSSL kako biste izradili certifikat. Rezultirajuća datoteka */var/certs/monit.pem* bi trebala ovako izgledati:

```
# create RSA certs - Server
RANDFILE = ./openssl.rnd
[ req ]
default_bits = 1024
encrypt_key = yes
distinguished_name = req_dn
x509_extensions = cert_type
[ req_dn ]
countryName = Country Name (2 letter code)
countryName_default = MO
stateOrProvinceName             = State or Province Name (full name)
stateOrProvinceName_default     = Monitoria
localityName                    = Locality Name (eg, city)
localityName_default            = Monittown
organizationName                = Organization Name (eg, company)
organizationName_default        = Monit Inc.
organizationalUnitName          = Organizational Unit Name (eg, section)
organizationalUnitName_default  = Dept. of Monitoring Technologies
commonName                      = Common Name (FQDN of your server)
commonName_default              = server.monit.mo
emailAddress                    = Email Address
emailAddress_default            = root@monit.mo
[ cert_type ]
nsCertType = server
```

Sada izradite certifikat:

```
# openssl req -new -x509 -days 365 -nodes -config ./monit.cnf -out \
/var/certs/monit.pem -keyout /var/certs/monit.pem
# openssl gendh 512 >> /var/certs/monit.pem
# openssl x509 -subject -dates -fingerprint -noout -in /var/certs/monit.pem
# chmod 700 /var/certs/monit.pem
```

Zatim uredite */etc/default/monit* kako biste pokrenuli *monit* daemon. Promijenite `startup` na 1 i postavite `CHECK_INTERVALS` na interval u sekundama u kojem ćete provjeravati sustav. Mi smo zadali 60. Sada bi datoteka trebala izgledati ovako:

```
# Defaults for monit initscript
# sourced by /etc/init.d/monit
# installed at /etc/default/monit by maintainer scripts
# Fredrik Steen <stone@debian.org>
# You must set this variable to for monit to start
startup=1
# To change the intervals which monit should run uncomment
# and change this variable.
CHECK_INTERVALS=60
```

Konačno, pokrenite *monit*:

```
# /etc/init.d/monit start
```

Sada usmjerite preglednik na *https://*`vaša domena`*:2812/* (ulaz 2812 ne smije biti blokiran na vatrozidu) i prijavite se na sustav korisničkim imenom *admin* i lozinkom *test*. Trebali biste vidjeti *monitovo* Web sučelje, pokazano ranije na slici 4-18.

# Što slijedi

Započeli smo postavljanjem i pokretanjem poslužitelja tako da ga možete koristiti kao internetsku platformu. Instalirali smo tekstualni poslužitelj, bez X Window sustava (zbog sigurnosnih razloga i poboljšanja performansi), te smo postavili Web sučelje kako bismo omogućili da sigurno upravljate svojom platformom te ju nadzirete.

U poglavljima koja slijede pozabavit ćemo se još detaljnije administriranjem Linux sustava. Počevši od petog poglavlja, više nećete ovisiti o samoinstalirajućem administrativnom softveru. Konfigurirat ćemo glavne Linux aplikacije koje se svakodnevno koriste u velikim sustavima ali i malim i srednjim poduzećima.

POGLAVLJE 5

# Elektronička pošta

U ovom ćemo poglavlju opisati postupak postavljanja servisa za elektroničku poštu za lokaciju male do srednje veličine. Elementi servisa uključuju:

- Postfix poslužitelj kao SMTP agent za prijenos pošte koji prihvaća poštu vaših korisnika te komunicira s drugim lokacijama na Internetu u cilju dostave pošte.
- Post Office Protocol (POP) i Interactive Mail Access Protocol (IMAP) poslužitelje za dostavu elektroničke pošte korisnicima vaše lokacije.
- Simple Authentication and Security Layer (SASL) za provjeravanje identiteta korisnika da bi se spriječile prijevare.

Postfix ćemo konfigurirati tako da koristi tradicionalnu provjeru identiteta temeljenu na datotekama koja se može proširiti na tisuće korisnika. Veći sustavi elektroničke pošte mogu pohranjivati korisnička imena i lozinke u relacijsku bazu podataka ili LDAP imenik. Kao primjer izuzetno skalabilnog poslužitelja temeljenog na Postfixu s LDAP provjerom identiteta, pogledajte Zimbru (*http://www.zimbra.com*).

Rješenja u ovom poglavlju povezuju različite komponente kako bi stvorile robustan, siguran i učinkovit sustav dostave elektroničke pošte. U današnje vrijeme stručnjaci poput Wietse Venema (izumitelja Postfixa) uvelike su smanjili složenost i nepouzdanost konfiguriranja sustava elektroničke pošte. Administratori Linux sustava, umjesto da gube vrijeme na složeno konfiguriranje poslužitelja elektroničke pošte, mogu rješavati druge aktualne probleme:

- Kako zaštititi elektroničku poštu od pokušaja zloupotrebe budući da je to medij osjetljiv na napade zlonamjernika.
- Kako zaštititi osjetljive podatke tvrtke.
- Kako omogućiti pristup elektroničkoj pošti korisnicima koji nisu izravno povezani s računalnom mrežom tvrtke.

## Ključni pojmovi vezani uz servis elektroničke pošte

*Agenti za prijenos pošte* (engl. *Mail Transfer Agent*, *MTA*) temelj su internetske komunikacije jer prenose poruke elektroničke pošte između lokacija. Da bi poslao poruku, pošiljatelj povezuje svoje računalo s agentom za prijenos pošte koji tada koristi SMTP kako bi prenio poruku do agenta za prijenos koji će dostaviti poštu primatelju.

Primatelj ima nekoliko opcija za primanje pošte od agenta i niti jedna od njih ne koristi SMTP: može se prijaviti kao korisnik na sustav na kojem se izvodi agent, može se povezati s agentom izravnom vezom (npr. modemskom vezom preko pružatelja Internet usluga) ili stvoriti tunel kroz Internet do udaljenog agenta. (Zanemarili smo metode kao što su primanje elektroničke pošte putem Web sučelja kao što je Gmail ili korištenjem mobilnog telefona).

Bez obzira na to koju od ovih metoda primatelj koristi, poštu će primiti putem *agenta za dostavu pošte* (eng. *Mail Delivery Agent,* MDA) kao što je Courier IMAP. Agent za dostavu pošte komunicira s agentom za prijenos kako bi primio poruke te pruža sandučić za dolaznu poštu iz kojeg ih korisnik preuzima. Nakon toga pošta se prikazuje na korisnikovom računalu putem *korisničkog agenta za elektroničku poštu* (eng. *Mail User Agent*, MUA) kao što je Outlook, Evolution ili Thunderbird.

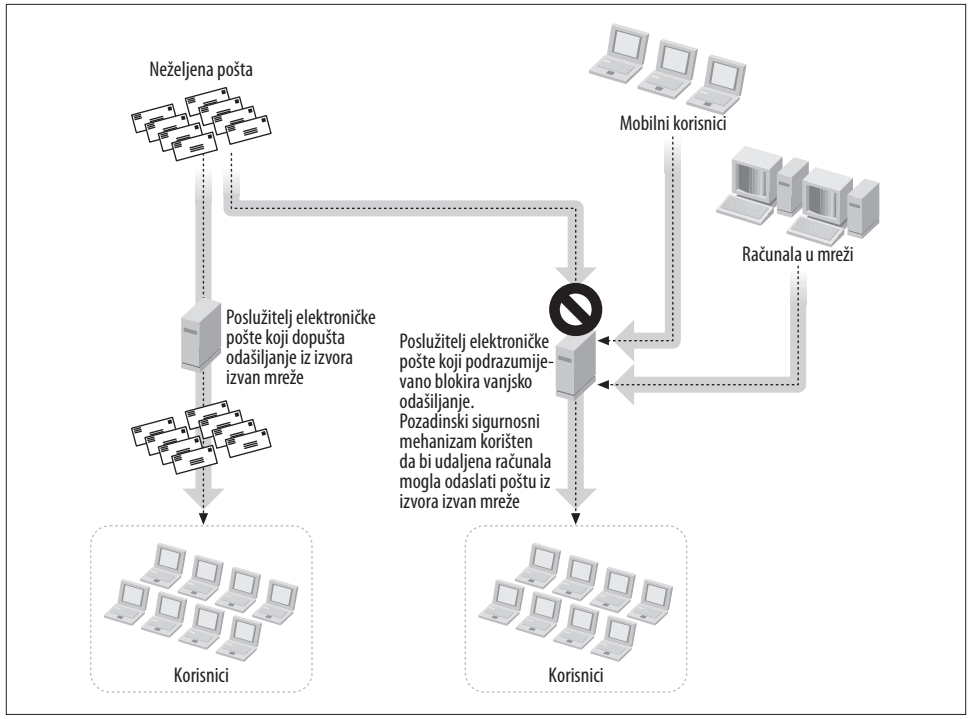
Korisnik obično prima elektroničku poštu koristeći se protokolima POP3 ili IMAP4 preko TCP/IP protokola. Gotovo svi moderni korisnički agenti za poštu podržavaju i POP3 i IMAP4. Korisnički agenti šalju poruke spajanjem s agentima za prijenos pošte te prijenosom poruka preko SMTP-a.

Većina korisnika upotrebljava adresare s popisom kontakata putem kojih korisnički agenti dolaze do adresa elektroničke pošte. U većim tvrtkama adresari su često spremljeni na LDAP poslužiteljima imenika. Većina korisnika niti ne zna da njihovi adresari u pozadini koriste LDAP.

## Postfix, Sendmail i drugi agenti za prijenos pošte

Možda se pitate zašto smo odabrali Postfix kao naš agent za prijenos pošte a ne Sendmail, izvorni Internet poslužitelj elektroničke pošte koji je u ranim osamdesetima razvio Eric Allman na kalifornijskom sveučilištu Berkeley. Sendmail je dugi niz godina bio vodeći agent za prijenos pošte na Internetu iako nismo sigurni da je i danas tako. Mnoga istraživanja ukazuju na to da je Sendmail izgubio popularnost i da ga danas koristi manje od 40% poslužitelja. Premda neki administratori koji snažno podupiru Sendmail tvrde da je fleksibilan i skalabilan, većina ga smatra izuzetno složenim i teškim za postavljanje i održavanje.

Sendmail je razvijen prije pojave neželjene elektroničke pošte i zlonamjernih programa tako da ima i nekoliko sigurnosnih nedostataka. Jedan od najozbiljnijih problema sa Sendmailom je to što on podrazumijevano dopušta *otvoreno odašiljanje* (eng. *open relaying*), tj. on će odašiljati poštu neovisno o njenom izvoru koji može biti i izvan poslužiteljeve lokalne mreže. Ovaj sigurnosni problem ilustriran je na slici 5-1.




*Slika 5-1. Sigurnosni problem s neželjenom poštom*

Neželjena pošta ili neželjena komercijalna pošta trenutno čini više od 50% prometa elektroničke pošte. Ta vrsta pošte ozbiljno ometa promet poruka i rad DNS poslužitelja, opterećuje procesor i kapacitet memorije računala kao i infrastrukturne resurse. Njeni pošiljatelji koriste različite tehnike za skrivanje svog pravog identiteta, uključujući zloupotrebu IP adresa, krivotvorenje zaglavlja poruka te odašiljanje preko otvorenih SMTP poslužitelja.

Dobro konfigurirani agenti za prijenos pošte prihvaćaju (odašilju) samo poruke dobivene s mrežnih adresa legitimnih korisnika koji najčešće pripadaju nekoj podmreži. Međutim, Sendmail standardno prenosi poruke pristigle s bilo kojeg računala. Koristite li Sendmail i ne isključite opciju za otvoreno odašiljanje, pošiljatelji neželjene pošte mogu iskoristiti vaš agent za prijenos pošte kako bi sakrili svoju lokaciju. U tom bi slučaju vaš poslužitelj elektroničke pošte mogao biti stavljen na „crnu listu“ kao otvoreni odašiljač što bi rezultiralo time da sve legitimne poruke koje preko njega dolaze budu tretirane kao neželjene. Ako ilegalni materijal bude poslan preko vašeg poslužitelja, mogli biste imati i problema sa zakonom.

Velika Sendmailova baza korisnika koja često radi s neispravnim, nedokumentiranim i starim inačicama aplikacije uvelike pomaže pošiljateljima neželjene pošte. Stručnjaci koji razvijaju Sendmail svjesni su problema* i trude se da ga učine što sigurnijim

* Pogledajte *BusinessWire*, 25.05.2006., „One in Three Companies Operate Without Email Usage Policies, Risking Damage to Their Systems and Reputations, Sendmail Finds“ (*http://goliath.ecnext.com/coms2/summary_0199-5568576_ITM*)

ali najveći napredak postigli su samo u tržišnoj inačici softvera. Aaron Weiss objašnjava razliku između tržišne i besplatne inačice Sendmaila u članku „The Fee vs. Free Divide“ (*http://www.serverwatch.com/tutorials/article.php/3580006*):

> Sendmail, Inc. osnovana je kako bi se komercijalizirao Sendmail i ponudili dodatni proizvodi koji znatno poboljšavaju njegovu instalaciju. Vodeći proizvod, Sendmail Switch, temelji se na besplatnoj inačici Sendmaila. To je nadogradnja Sendmailove jezgre a dodaje centraliziranu grafičku upravljačku konzolu, neprestano održavanje sustava sigurnosti, filtre sadržaja (uključujući obranu od neželjene elektroničke pošte i virusa), podršku za SSL, SASL i LDAP imenike te mogućnosti za nadzor, grupiranje i upravljanje na daljinu. Sve to dolazi s grafičkim instalacijskim programom i čarobnjacima za obavljanje pojedinačnih zadaća.

Konzorcij Sendmail (odgovoran za besplatnu inačicu Sendmail agenta za prijenos pošte s otvorenim izvornim kodom) sponzorira tvrtka Sendmail, Inc. koja za njega proizvodi dodatke protiv neželjene pošte i virusa te upravljanje pravilima. Slijedi opis poslovnog modela tvrtke Sendmail, Inc. (s *http://www.sendmail.com/company*):

> Sendmail proizvodi poslovna rješenja za sigurno, pouzdano i jednostavno komuniciranje uključujući elektroničku i glasovnu poštu te slanje instant poruka. Sendmailova rješenja kontroliraju sve aspekte sigurnosti elektroničke pošte, a mogu se lako implementirati u softver i uređaje. Sendmailovi proizvodi rade unutar heterogene infrastrukture elektroničke pošte koja podržava Exchange, Notes, Groupwise i druga rješenja.

Postfix je prvenstveno projektiran kao sigurna i robusna zamjena za Sendmail. Debianov podrazumijevani agent za prijenos pošte je Exim 4 no mi preferiramo Postfix jer Exim ima problema sa skalabilnosti. Nedostaje mu centralni upravitelj reda poruka te centralizirano raspoređivanje opterećenja. Pored toga, postoje naznake da će u skoroj budućnosti Postfix postati podrazumijevani Debianov agent za prijenos pošte. U međuvremenu, Exim možete jednostavno zamijeniti Postfixom kao što će biti objašnjeno u sljedećem odjeljku.

# Postfix SMTP poslužitelj elektroničke pošte na Debianu

Da bismo postavili poslužitelj, upotrijebit ćemo novu instalaciju Debiana. Ako odaberete neku drugu distribuciju, iste ćete rezultate postići primjenjivanjem postupaka sličnih ovima opisanim u ovom poglavlju.

## Debian paketi vezani uz Postfix

Instalirajte posljednju stabilnu inačicu Debiana i konfigurirajte ju s minimalnim brojem paketa. Ako nemate Debianov mrežni instalacijski disk, preuzmite ga s *http://www.us.debian.org/CD/netinst*. Tada pokrenite instalaciju preko mreže i svakako unesite potpuno kvalificirano ime domene. Konfigurirajte Debian u skladu s uputama koje slijede.

Prije konfiguracije, Debianov instalacijski program provodi vas kroz standardne skripte. Slijedite standardni postupak instaliranja sve dok se ne prikaže grafički izbornik u kojem se trebate odlučiti za tip instalacije. Izbornik će izgledati ovako:

```
( ) Desktop Environment
( ) Web Server
( ) Print Server
( ) DNS Server
( ) File Server
( ) Mail Server
( ) SQL database
( ) manual package selection
```

Nemojte odabrati niti jednu od ponuđenih opcija obzirom da nećete koristiti podrazumijevani Debianov poslužitelj elektroničke pošte (Exim) već ćete umjesto njega instalirati Postfix. Kad se otvori zaslon pritisnite tabulator (tipku Tab) i zatim gumb OK. Debianov instalacijski program će nastaviti preuzimati i instalirati pakete. Za vrijeme preuzimanja, program će prikazati još jedan grafički zaslon s pitanjem želite li konfigurirati Exim (*Exim-config*). Odaberite „no configuration“. Na pitanje „Really leave the mail system unconfigured?“ odgovorite yes.

Debianov instalacijski program nastavit će preuzimati i konfigurirati pakete. Kad se završi instalacija, na zaslonu će se pojaviti poruka u kojoj se zahvaljuje što koristite Debian.

Sada je potrebno ukloniti neke nepotrebne programe koristeći Debianov pomoćni program *apt-get*. Ako koristite neku drugu distribuciju, pakete obrišite koristeći odgovarajuće postupke. U Debianu zadajte:

```
# apt-get remove lpr nfs-common portmap pidentd pcmcia-cs pppoe \
pppoeconf ppp pppconfig
```

Sada onemogućite neke servisne skripte:

```
# update-inetd --remove daytime
# update-inetd --remove telnet
# update-inetd --remove time
# update-inetd --remove finger
# update-inetd --remove talk
# update-inetd --remove ntalk
# update-inetd --remove ftp
# update-inetd --remove discard
```

i ponovno pokrenite *inetd* superposlužitelj:

```
#/etc/init.d/inetd reload
```

## Instalacija Postfixa na Debian

Sljedeća naredba instalira pakete potrebne za pokretanje Postfixa, zajedno s protokolom TLS i pozadinskim servisom SASL koji omogućava provjeru identiteta korisnika:

```
# apt-get install postfix postfix-doc postfix-tls libsasl2 \
sasl2-bin libsasl2-modules
```

Kad instalirate ove pakete, Debian bi istovremeno mogao instalirati i *libldap2*. *libsasl2* je možda već instaliran na sustavu.

Sada će Debianov instalacijski pomoćni program početi preuzimati i konfigurirati nekoliko datoteka. Za vrijeme ovog procesa primijetit ćete dugačak dijalog koji započinje redovima:

```
Reading Package Lists... Done
Building Dependency Tree... Done
```

Zatim će se prikazati opširan tekst koji započinje s:

```
You have several choices for general configuration at this point...
```

Na dnu zaslona pronaći ćete pitanje koje nas zanima:

```
General type of configuration?
    No configuration
    Internet Site
    Internet with smarthost
    Satellite system
    Local only
     <Ok>          <Cancel>
```

Odaberite „Internet Site“ čak i ako planirate koristiti Postfix samo za lokalnu dostavu pošte.

Dijalog koji se sada pojavljuje govori vam da instalacija izrađuje Postfixovu konfiguracijsku datoteku. Ako već imate poslužitelj koji koristi Sendmail, imat ćete datoteku *aliases*. U ovom poglavlju pretpostavljamo da započinjete od početka, tako da na sljedećem zaslonu upišite `NONE`:

```
The user root (and any other users with a uid of 0) must have mail
redirected via an alias, or their mail may be delivered to /var/mail/nobody.
This is by design: mail is not delivered to external delivery agents as root.
If you already have a /etc/aliases file, then you possibly need to add this
entry.  (I will only add it if I am creating a new /etc/aliases.)
What address should I add to /etc/aliases, if I create the file? (Enter NONE
to not add one.)
Where should mail for root go
NONE_____________________________________________________
                <Ok>                          <Cancel>
```

Sljedeće pitanje tijekom instaliranja odnosi se na potpuno kvalificirano ime vaše domene. Postfix zahtijeva da naredba *hostname* vrati ime domene u obliku *mail.centralsoft.org*. Ali na Debianu naredba *hostname* podrazumijevano vraća samo *mail*. Da biste mogli konfigurirati puno ime domene, instalacijska skripta pruža sljedeći dijalog:

```
Your 'mail name' is the hostname portion of the address to be shown on
outgoing news and mail messages (following the username and @ sign).
This name will be used by other programs besides Postfix; it should be the
single, full domain name (FQDN) from which mail will appear to originate.
Mail name?
mail.centralsoft.org_____________________________________
                <Ok>                          <Cancel>
```

Odgovorite `<OK>` kako biste prihvatili podrazumijevanu vrijednost koja se pojavila u plavom tekstualnom okviru.

Sljedeći dijalog prikazuje popis podrazumijevanih vrijednosti za domene za koje će poslužitelj primati poštu:

```
Give a comma-separated list of domains that this machine should consider itself the
finaldestinationfor.Ifthisisamaildomaingateway,youprobablywanttoinclude
the top-level domain.
Other destinations to accept mail for? (blank for none)
server2.centralsoft.org, localhost.centralsoft.org, , localhost
                <Ok>                          <Cancel>
```

Nabrojane domene pojavit će se u konfiguracijskoj datoteci *main.cf*.

Posljednje pitanje odnosi se na sustave s datotečnim sustavima bez dnevnika transakcija (engl. *non-journaled filesystems*).

```
If synchronous updates are forced, then mail is processed more slowly.
If not forced, then there is a remote chance of losing some mail if the
system crashes at an inopportune time, and you are not using a journaled
filesystem (such as ext3).
The default is "off".
Force synchronous updates on mail queue?
                <Yes>                         <No>
```

Budući da gotovo sve aktualne distribucije podrazumijevano koriste datotečni sustav s dnevnikom transakcija *ext3*, ovdje možete odgovoriti `<No>`.

Instaliranje sada završava i ispisuje Postfixovu konfiguracijsku datoteku. Parametri i vrijednosti koji su ovdje ispisani možda će vam se trenutno činiti besmislenima, ali moći ćete ih pronaći u konfiguracijskoj datoteci i prema potrebi ih promijeniti.

## Osnovna konfiguracija Postfixa

Sljedeći kod predstavlja najosnovniju Postfixovu konfiguracijsku datoteku, */etc/postfix/main.cf*:

```
smtpd_banner = $myhostname ESMTP $mail_name (Debian/GNU)
biff = no
append_dot_mydomain = no
myhostname =
mydomain =
myorigin = $mydomain
inet_interfaces =
mydestination = $mydomain, localhost.$mydomain, localhost
mynetworks = 127.0.0.0/8
```

Da ste Postfix ručno konfigurirali, mnoge od ovih vrijednosti morali biste sami unijeti. Ova datoteka ističe vrijednost Debianovog instalacijskog programa.

Postfix koristi jednostavnu sintaksu u kojoj se svaki red sastoji od konfiguracijskog parametra iza kojeg slijedi znak jednakosti i vrijednost. Nakon što je parametar definiran, na njega se možemo pozivati i u kasnijim redovima tako da ispred naziva parametra stavimo znak dolara. Stoga:

```
mydomain = centralsoft.org
myorigin = $mydomain
```

dodijeljuje vrijednost `centralsoft.org` parametrima `mydomain` i `myorigin`.

Osnovna konfiguracijska datoteka obavlja samo lokalnu dostavu pošte. Očekuje se da primatelji pošte imaju otvorene račune za školjku i korijenske direktorije na samom poslužitelju elektroničke pošte na kojem je instaliran Postfix. Ne zahtijeva se da sustav doda nastavak `@` (što bi bilo zadano parametrom `append_dot_mydomain`). Upravo zato vas Debianov instalacijski program i pita za domene, imena računala i odredišne adrese.

Debianov program za upravljanje paketima sam će konfigurirati mnoge parametre u datoteci */etc/postfix/main.cf*. Tablica 5-1 objašnjava ključne redove. Potpuniji popis parametara možete pronaći na Debian sustavu u datoteci */usr/share/postfix/main.cf.dist*.

*Tablica 5-1. Ključni parametri za konfiguraciju Postfixa*

| Parametar | Objašnjenje |
|---|---|
| `.smtpd_banner = $myhostname ESMPT $mail_name (Debian/GNU)` | Zadaje tekst u naslovu koji identificira poslužitelj prilikom komuniciranja s drugim poslužiteljem preko SMTP-a. Prema specifikaciji SMTP-a korištenje naslova je obavezno. |
| `biff = no` | biff je mali Postfixov proces koji lokalne korisnike može obavještavati da su dobili poruke. Ako nemate lokalnih korisnika, isključite ovu opciju. Podrazumijevana vrijednost u Debianovoj instalaciji je no. |
| `append _dot_mydomain = no` | U okruženju kao što je naše, dodavanje imena domene adresi elektroničke pošte posao je korisničkog agenta za poštu. Ova vrijednost znači da Postfix neće stavljati nastavak poput @centralsoft.org |
| `#delay_warning_time = 4h` | Iz ovog reda obrišite znak komentara da biste generirali upozorenja o kašnjenju pošte. Ovu opciju nećemo uključivati budući da počinjemo s malim brojem korisnika te ne očekujemo kašnjenja. |
| `myhostname = server2.centralsoft.org` | Zadaje ime računala na Internetu za ovaj sustav elektroničke pošte. Podrazumijevano se koristi potpuno kvalificirano ime domene.<br>$myhostname se koristi kao podrazumijevana vrijednost za mnoge druge konfiguracijske parametre. |
| `alias_maps = hash:/etc/aliases`<br>`alias_database = hash:/etc/aliases` | Zadaje pseudonime baza podataka koje koristi lokalni agent za dostavu pošte. Pseudonim je alternativno ime koje se koristi umjesto izvornog. Primjerice, možete zadati admin kao pseudonim za root. Uloge ovih parametara trenutno nije važno razumjeti već samo zapamtite da Postfix čuva popis svih pseudonima u jednoj datoteci i da ovi parametri govore sustavu gdje je ona smještena te koji je format baze podataka korišten. |
| `myorigin = mydomain` | Zadaje domenu iz koje dolaze lokalne poruke. |
| `mydestination = server2.centralsoft.org, localhost.centralsoft.org, , localhost` | Zadaje popis imena računala i domena razdvojenih zarezima i/ili razmacima za koje će poslužitelj prihvaćati poštu. |
| `relayhost =` | Zadaje podrazumijevano računalo koje će poslužitelj koristiti za prosljeđivanje pošte kad ne zna kako doprijeti do primatelja. Ovaj ćemo parametar ostaviti praznim oslanjajući se na parametar mynetworks koji slijedi. |

*Tablica 5-1. Ključni parametri za konfiguraciju Postfixa (nastavak)*

| Parametar | Objašnjenje |
|---|---|
| `mynetworks = 127.0.0.0/8` | Zadaje računala koja poslužitelj neće smatrati pošiljateljima neželjene pošte. Ovdje smo zadali samo naše lokalno računalo. Umjesto toga možete zadati parametar mynetworks_style = class kada Postfix kao pošiljatelje neželjene pošte neće smatrati SMTP klijente u istoj mrežnoj klasi (A/B/C) u kojoj se nalazi računalo. Nemojte imati povjerenja u cijelu klasu na poslužitelju koji prihvaća modemske veze jer bi tad Postfix postao otvoreni odašiljač za čitavu mrežu pružatelja. |
| `mailbox_command = procmail -a "$EXTENSION"` | Zadaje opcionalnu vanjsku naredbu koja se koristi za dostavu pošte u sandučić lokalnog korisnika. Naredba se izvodi za primatelje s ispravnim HOME, SHELL i LOGNAME postavkama. |
| `mailbox_size_limit = 0` | Postavlja kvotu za pohranjene poruke za svakog korisnika. 0 isključuje ograničenje kvotama. |
| `recipient_delimiter = +` | Zadaje separator koji se koristi između korisničkog imena i nastavka adrese u referentnoj tablici. |
| `inet_interfaces = all` | Zadaje adrese mrežnog sučelja (mrežne kartice) na koje ovaj sustav prima poštu. Ovaj je parametar koristan samo ako na računalu postoji više mrežnih kartica. |

Neke jednostavne i korisne prilagodbe koje možete izvesti uključuju sljedeće:

- U pravilu, parametar `mydestination` sadrži popis domena koje se pojavljuju u adresama elektroničke pošte lokalnih korisnika, to jest domena za koje Postfix prihvaća i dostavlja poruke. Podrazumijevano, Postfix prihvaća poruke namijenjene za `$myhostname` i `localhost.$mydomain`, računalo na kojem se izvodi Postfix. Možete zadati da sustav prihvaća poštu za čitavu domenu dodavanjem parametra `$mydomain` na popis:

  ```
  mydestination = $myhostname, localhost.$mydomain, $mydomain
  ```

- Postfixu možete zadati računala kojima želite dopustiti odašiljanje pošte postavljanjem parametra `mynetworks`. (Ako postavite parametar `mynetworks`, Postfix zanemaruje parametar `mynetworks_style`). Možete zadati jednu ili više IP adresa i/ili upotrijebiti zapis mrežne maske (npr. `151.164.28.0/28`). Ovaj je parametar koristan ako želite omogućiti odašiljanje računalima izvan mreže – npr. zaposlenicima koji rade kod kuće, trgovačkim putnicima i sl.

Kasnije ćemo izvesti još nekoliko promjena u datoteci */etc/postfix/main.cf* da bismo dodali provjeru identiteta i šifriranje lozinki.

## Testiranje sustava za slanje elektroničke pošte

Kad je Debianova konfiguracija gotova, možete započeti primati i slati elektroničku poštu koristeći se vašim korisničkim računom. Slijede primjeri dviju pokusnih poruka koje je poslao jedan od autora ove knjige. U prvom je primjeru upotrijebio korisnički račun na Gmailu da bi poslao poruku korisničkom računu u sustavu *server2.centralsoft.org*. Poruku je pročitao iz školjke koristeći standardnu Unixovu naredbu *mail*:

```
~$ mail
Message 1:
Date: Tue, 11 Jul 2006 17:38:32 -0500
From: "Tom Adelstein" <tadelstein@gmail.com>
To: tadelste@server2.centralsoft.org
Subject: Testing simple STMP services
We're sending this email to test our mail server's capability to send
and receive simple SMTP mail.
```

Potom je odgovorio na izvornu poruku i odgovor primio na korisnički račun za Gmail:

```
Delivered-To: tadelstein@gmail.com
Received: from server2.centralsoft.org
        Tue, 11 Jul 2006 16:10:44 -0700 (PDT)
To:tadelstein@gmail.com
Subject: Re: testing simple SMTP mail
In-Reply-To
tadelste@server2.centralsoft.org (Tom Adelstein)

We're sending this email to test our mail server's capability to send
and receive simple SMTP mail
```

Korištenje naredbe *mail* neprikladan je način za rad s većim količinama pošte, čak i u okruženju školjke. Alternativa je naredba *mutt* s robusnijim sučeljem i sa mnogo više značajki. Kao administrator, možda ćete koristiti jedan od tih naredbenih korisničkih agenata za poštu kad budete primali poštu od servisnih računa Linux sustava.

# Dodavanje provjere identiteta i šifriranja

Sada smo konfigurirali podrazumijevani SMTP poslužitelj. Što bismo još mogli učiniti s Postfixom? U ovom ćemo odjeljku konfiguracijskoj datoteci */etc/postfix/main.cf* dodati sustav provjere identiteta (koji koristi SASL) i šifriranje (koji koristi TLS). Zahvaljujući provjeri identiteta možemo biti sigurni da samo korisnici s odgovarajućim vjerodajnicama mogu koristiti SMTP poslužitelj. Šifriranjem osiguravamo da se korisnička imena i lozinke ne šalju preko mreže u obliku čistog teksta.

## SASL provjera identiteta

Slika 5-1 prikazuje grupu mobilnih korisnika koji trebaju odaslati poštu preko poslužitelja elektroničke pošte s lokacije izvan poslužiteljeve lokalne mreže. Ovo je čest slučaj. Da bi se legitimni korisnici mogli razlikovati od slučajnih pošiljatelja neželjene pošte, potrebno je ugraditi sigurnosni mehanizam. Sloj Simple Authentication and Security Layer razvijen je kao dio projekta Cyrus na sveučilištu Carnegie Mellon. On Postfixu pruža način za identificiranje izvora poruka poslanih poslužitelju i kontroliranje njihovog odašiljanja.

Administratori sistema mogu koristiti SASL za dodavanje provjere identiteta na veliki broj procesa u kojima međudjeluju klijent i poslužitelj. Međutim, svaki servis koji koristi SASL na Linux operativnom sustavu zahtijeva zasebnu konfiguracijsku datoteku. Nije dovoljno samo instalirati SASL i konfigurirati ga za cijeli sustav.

Kako je SASL postao dio Postfixovog rješenja? Da bismo pronašli odgovor moramo se vratiti u 1999. godinu kada je IETF napisao standard SMTP's Service Extension for Authentication. Ovo ćete proširenje prepoznati po akronimu *ESMTP* koji možete uočiti u prvom redu datoteke */etc/postfix/main.cf* (pogledajte tablicu 5-1). ESMTP sprječava da pošiljatelji neželjene elektroničke pošte i/ili zlonamjerni korisnici upotrijebe agente za prijenos pošte kao odašiljače za svoje poruke. On također pruža sigurnost putem provjeravanja identiteta korisnika i zapisivanja njihovih aktivnosti u dnevnik.

IETF je proširenje ESMTP utemeljio na SASL-u. Kao dio SMTP protokola, ESMTP jednostavno dodaje naredbu *AUTH* naredbama koje poslužitelji koriste za povezivanje i razmjenu podataka.

SASL-ov kostur za provjeru identiteta dopušta veliki broj načina za pohranjivanje i razmjenu korisničkih vjerodajnica. On može koristiti Linuxove sistemske lozinke *(/etc/passwd*, */etc/shadow* ili Pluggable Authentication Module (PAM) module), zasebne datoteke ili vanjske servise kao što su LDAP, Kerberos ili *sasldb* (imenik razvijen kroz projekt Cyrus i uključen u SASL).

U ovom ćemo poglavlju opisati dva načina za korištenje Postfixa sa SASL-om. Najprije ćemo konfigurirati jednostavnu metodu koja dobro funkcionira na manjim lokacijama, kada možete svakom korisniku elektroničke pošte otvoriti korisnički račun na Linux poslužitelju. Ova metoda koristi PAM, podrazumijevanu provjeru identiteta korisnika prilikom prijavljivanja. Zatim ćemo konfigurirati složeniji sustav koji omogućava provjeru identiteta korisnika koji na poslužitelju nemaju otvoren korisnički račun.

Prijavljivanje se može smatrati procesom koji se odvija u dvije faze. U prvoj fazi se utvrđuje da je korisnik koji se želi prijaviti zaista onaj za kog se izdaje. U drugoj fazi pruža mu se pristup zahtijevanom servisu, što može biti sesija u naredbenoj školjci (*bash*, *tcsh*, *zsh*, *etc*) ili X Windowsima koja se izvodi pod korisnikovim identitetom.

### Konfiguriranje Postfixa sa SASL-om za provjeru identiteta korisnika s otvorenim korisničkim računom

Srećom, Debian u Postfixovu instalaciju uključuje SASL. Možete iskoristiti Debianove SASL biblioteke da biste korisnicima prijenosnih računala omogućili potvrditi svoj identitet i kad nisu spojeni s lokalnom mrežom. U sljedećem primjeru koristit ćemo SASL da bismo provjerili imaju li korisnici koji se pokušavaju spojiti valjan korisnički račun na Linux poslužitelju. Naš sistem će dopustiti da se samo osobe s korisničkim računom na poslužitelju povežu i šalju elektroničku poštu. Da bismo to postigli koristit ćemo podrazumijevani Linux mehanizam za prijavu, PAM.

Prilikom instaliranja paketa dodali ste sva potrebna SASL-ova proširenja i biblioteke (*postfix-tls*, *libsasl2*, *sasl2-bin* i *libsasl2-modules*). Sada trebate konfigurirati datoteku */etc/postfix/main.cf*. Najprije ćemo pokazati kako se dodaju parametri korištenjem naredbi *postconf* a zatim alternativni način na koji se datoteka */etc/postfix/main.cf* izravno uređuje.

Provjeru identiteta u Postfix SMTP poslužitelju uključite dodavanjem poslužiteljskog parametra `smtpd` u datoteku *main.cf* zadajući ovu *postconf* naredbu:

```
# postconf -e 'smtpd_sasl_auth_enable = yes'
```

Zatim dodajte parametar za prihvaćanje nestandardnih klijenata koji ne slijede pravilno SMTP-ov postupak provjere identiteta:

```
# postconf -e 'broken_sasl_auth_clients = yes'
```

Parametar `smtpd_sasl_security_options` omogućava kontroliranje mehanizama za provjeru lozinki kada se klijenti spajaju na SMPT poslužitelj. Sljedeća naredba blokira provjeru identiteta anonimnih korisnika:

```
# postconf -e 'smtpd_sasl_security_options = noanonymous'
```

Postfix podrazumijevano ne dopušta neautorizirano odašiljanje poruka. Stoga, da biste vašim korisnicima elektroničke pošte omogućili pristup poslužitelju preko Interneta, trebate dodati još jedan parametar (on mora biti napisan u jednom redu – ovdje smo ga morali prelomiti da bi stao na stranicu):

```
# postconf -e 'smtpd_recipient_restrictions =
permit_mynetworks,permit_sasl_authenticated,reject_unauth_destination'
```

Na kraju, parametar `smtp_sasl_local_domain` definira ime lokalne domene za provjeru identiteta. Podrazumijevano, Postfix smatra da ime računala ima tu ulogu. Ako želite da tako ostane, zadajte prazan niz:

```
# postconf -e 'smtpd_sasl_local_domain ='
```

Ovime je dovršeno konfiguriranje SASL-a za Postfix. Alternativno, umjesto izvođenja navedenih *postconf* naredbi, možete urediti datoteku */etc/postfix/main.cf*, dodati joj sljedeće redove, te ponovno pokrenuti Postfix:

```
smtpd_sasl_local_domain = $myhostname
smtpd_sasl_auth_enable = yes
broken_sasl_auth_clients = yes
smtpd_sasl_security_options = noanonymous
smtpd_recipient_restrictions =
    permit_sasl_authenticated,permit_mynetworks,reject_unauth_destination
smtpd_sasl_local_domain =
```

Skoro ste dovršili konfiguriranje SASL-a pa ga možete započeti koristiti. Prije nego što opišemo zadnje korake, izvedite naredbe koje slijede da biste izradili SASL konfiguracijsku datoteku u direktoriju u kojem će je Postfix tražiti (opcijom *-p* izbjegavamo poruku o pogreški ako direktorij već postoji):

```
# mkdir -p /etc/postfix/sasl
# cd /etc/postfix/sasl
Create the smtpd.conf file with these two lines:
pwcheck_method: saslauthd
mech_list: plain login
```

Sada možete ponovno pokrenuti Postfix:

```
# postfix reload
```

### Pozadinski servis saslauthd

U datoteci *smtpd.conf* zadali smo *saslauthd* kao metodu za provjeravanje korisničkih vjerodajnica. Zašto smo to učinili?

Naš mehanizam za upravljanje lozinkama koristi PAM i neautorizirani procesi nemaju pristup datotekama s lozinkama. Budući da Postfixov servisni račun ima ograničena dopuštenja, on ne može izravno provjeravati identitet korisnika.

SASL biblioteke instalirane s Debianom rješavaju ovaj problem dodavanjem pozadinskog servisa za provjeru identiteta koji se zove *saslauthd* a obrađuje zahtjeve upućene Postfixu. On se izvodi s dopuštenjima superkorisnika u procesu odvojenom od Postfixa tako da eventualno kompromitirani poslužitelj elektroničke pošte ne može iskoristiti njegove privilegije.

*saslauthd* ne komunicira s računalima izvan lokalne mreže pa utjecaj koji će njegovo izvođenje imati na sigurnost možete smatrati minimalnim iako taj pozadinski servis koristi nezaštićene tekstualne lozinke. *saslauthd* treba stvarne lozinke jer koristi isti servis za prijavljivanje koji koristite za započinjanje sesije na Linux konzoli.

Sada ćemo konfigurirati *saslauthd* da bi radio s poslužiteljem elektroničke pošte. Sljedeće upute oblikovane su za Debian ali ih uz male promjene direktorija i naredbi možete primijeniti i na druge Linux sustave.

Inačica Postfixa za Debian izvodi se preusmjerena u */var/spool/postfix*. Stoga je u isti imenski prostor potrebno smjestiti i pozadinski servis *saslauthd*. Slijedite ove korake:

1. Izradite direktorij za pozadinski servis:

```
#mkdir -p /var/spool/postfix/var/run/saslauthd
```

2. Uredite */etc/default/saslauthd* da biste aktivirali *saslauthd*. Obrišite oznaku komentara (#) iz reda `START=yes` a zatim dodajte red:

```
PARAMS="-m /var/spool/postfix/var/run/saslauthd -r"
```

3. Datoteka bi sada trebala izgledati ovako:

```
# This needs to be uncommented before saslauthd will be run automatically
START=yes
PARAMS="-m /var/spool/postfix/var/run/saslauthd -r"
# You must specify the authentication mechanisms you wish to use.
# This defaults to "pam" for PAM support, but may also include
# "shadow" or "sasldb", like this:
# MECHANISMS="pam shadow"
MECHANISMS="pam"
```

4. Zatim uredite */etc/init.d/saslauthd* da biste promijenili lokaciju identifikacijske datoteke procesa *saslauthd*. Promijenite vrijednost parametra `PIDFILE` kao što slijedi:

```
PIDFILE="/var/spool/postfix/var/run/${NAME}/saslauthd.pid"
```

5. Pokrenite *saslauthd*:

```
# /etc/init.d/saslauthd start
```

Ako koristite neku drugu Linuxovu distribuciju, radit ćete s različitim datotekama, direktorijima i naredbama. Primjerice, u mnogim sustavima standardni način za prvo pokretanje servisa *saslauthd* je zadavanjem naredbe:

```
# saslauthd -a pam
```

Za razliku od drugih sustava, kod Debiana je korištenje PAM-a zadano u konfiguracijskoj datoteci.

#### Konfiguriranje Posfixa sa SASL-om za provjeru identiteta korisnika bez korisničkog računa

Korištenje datoteke s lozinkama za provjeru identiteta kod Postfixa na Linux sustavu zahtijeva da svaka osoba koja odašilje poštu putem poslužitelja ima korisnički račun. Ovom pristupu nedostaje skalabilnost te zahtijeva više vremena za administriranje sustava. Da bi omogućio korištenje SMTP poslužitelja korisnicima koji nemaju otvoren korisnički račun, SASL dopušta upotrebu i drugih opcija za pohranu lozinki. Popularne opcije uključuju *sasldb*, LDAP, Kerberos i MySQL. Pozadinski servis *saslauthd* se ne izvodi kad Postfix koristi neku od tih metoda. Zasebni program s dopuštenjima superkorisnika nije potreban jer SASL-u nije potreban pristup datoteci s lozinkama operativnog sustava.

Kad koristite *saslauthd*, ograničeni ste na prijenos lozinki u obliku čistog teksta te provjeru identiteta prilikom prijave. Zato Postfix nudi i alternativnu metodu *auxprop* koja uz ove dvije podržava i CramMD5, DigestMD5, OPT i NTLM metode za provjeru identiteta.

Od svih mehanizama za provjeru identiteta korisnika o kojima je bilo riječi u ovom poglavlju, LDAP je najrobusniji i najskalabilniji ali je ograničen na upotrebu lozinki u obliku čistog teksta. Da bi riješili ovaj problem, administratori obično koriste protokol Transport Layer Security (TLS) za šifriranje lozinki koje klijent šalje poslužitelju (što će biti opisano u sljedećem odjeljku). Kombinacija LDAP-a i TLS-a trenutno pruža najvišu razinu sigurnosti.

U manjoj mreži *sasldb* pruža jednostavno rješenje za omogućavanje pristupa manjem broju udaljenih korisnika. Kod jako velikih lokacija s više korisnika, MySQL bi se mogao pokazati skalabilnijim i lakšim za korištenje i održavanje.

I *sasldb* i MySQL metode pohrane zahtijevaju instaliranje dodatnog softvera u obliku proširenja za dodavanje svojstava (engl. *auxiliary property plug-ins*). Ako konfigurirate *sasldb* ili MySQL, morate urediti datoteku *smtpd.conf* i promijeniti red:

```
pwcheck_method: saslauthd
```

tako da glasi:

```
pwcheck_method: auxprop
```

što će pružiti kostur za proširenja za dodavanje svojstava.

## TLS šifriranje

Nedostatak korištenja *auxprop* metode za provjeru legitimnosti korisnika je to što provjerava lozinke u obliku čistog teksta, ukoliko nije primijenjena dodatna zaštita. Kad se prijavljujete na vlastito računalo to nije problem. Ali kad preko mreže šaljete svoje kori-

sničko ime i lozinku u obliku čistog teksta da biste poslali poruku, bilo unutar lokalne mreže ili preko Interneta, zlonamjernici lako mogu doći do vaših vjerodajnica.

U drugom poglavlju opisivali smo korištenje protokola TLS, unaprijeđene inačice SSL šifriranja, za sigurno slanje lozinki od osobnog računala do poslužitelja. Ovdje ćemo proširiti to rješenje da bismo šifrirali informacije za identifikaciju izradom certifikata koristeći OpenSSL.

Kao i prethodni odjeljak u kojem se govorilo o SASL-u, i ovaj se odnosi na sigurnost, ali na njen drugi aspekt. Odjeljak o SASL-u govorio je o *provjeri identiteta* kojom se utvrđuje tko ima pravo slati poštu preko poslužitelja. U ovom se odjeljku govori o *zaštiti lozinki* koja osigurava da potencijalni uljezi neće moći pročitati korisnikove vjerodajnice dok se prenose preko mreže. Za siguran sustav elektroničke pošte potrebna su oba servisa.

Započnite izradom direktorija za SSL certifikate. Izradite ga kao poddirektorij u glavnom Postfixovom direktoriju u Debianu:

```
# mkdir /etc/postfix/ssl
# cd /etc/postfix/ssl/
```

Zatim generirajte dva certifikata i dva ključa za šifriranje. Potreban vam je privatni ključ koji nitko drugi ne smije znati i javni ključ koji omogućava da vam drugi korisnici sigurno pošalju svoje vjerodajnice. Započnite s ključem poslužitelja:

```
# openssl genrsa -des3 -rand /etc/hosts -out smtpd.key 1024
293 semi-random bytes loaded
Generating RSA private key, 1024 bit long modulus
.....................................++++++
.....................................++++++
e is 65537 (0x10001)
Enter pass phrase for smtpd.key:
Verifying - Enter pass phrase for smtpd.key:
```

Promijenite dopuštenja za izrađenu datoteku koja sadrži OpenSSL ključ poslužitelja:

```
# chmod 600 smtpd.key
```

Zatim generirajte drugi ključ i certifikat:

```
# openssl req -new -key smtpd.key -out smtpd.csr
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
Country Name (2 letter code) [AU]:
State or Province Name (full name) [Some-State]:
Locality Name (eg, city) []:
Organization Name (eg, company) [Internet Widgits Pty Ltd]: centralsoft.org
Organizational Unit Name (eg, section) []: web
Common Name (eg, YOUR name) []:
Email Address []:
```

```
Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:
An optional company name []:cso
```

Vode se rasprave na temu treba li ili ne treba dati točne informacije kod izrade samogenerirajućih certifikata. Preporučujemo da unesete informacije koje odgovaraju vašim uvjetima.

Sljedeće naredbe generiraju ključ potpisa i zamjenjuju postojeće ključeve s novima:

```
# openssl x509 -req -days 3650 -in smtpd.csr -signkey smtpd.key -out \
smtpd.crt
Signature ok
subject=/C=US/ST=Texas/L=Dallas/O=centralsoft.org/OU=web/CN=Tom_Adelstein/
emailAddress=tom.adelstein@gmail.com
Getting Private key
Enter pass phrase for smtpd.key:
# openssl rsa -in smtpd.key -out smtpd.key.unencrypted
Enter pass phrase for smtpd.key:
writing RSA key
# mv -f smtpd.key.unencrypted smtpd.key
# chmod 600 smtpd.key
# openssl req -new -x509 -extensions v3_ca -keyout cakey.pem -out \
cacert.pem -days 3650
Generating a 1024 bit RSA private key
.....................++++++
.........................++++++
writing new private key to 'cakey.pem'
Enter PEM pass phrase:
Verifying - Enter PEM pass phrase:
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank
Country Name (2 letter code) [AU]:
State or Province Name (full name) [Some-State]:
Locality Name (eg, city) []:
Organization Name (eg, company) [Internet Widgits Pty Ltd]:
Organizational Unit Name (eg, section) []:
Common Name (eg, YOUR name) []:
Email Address []:
```

Sada je potrebno Postfix obavijestiti o ključevima i certifikatima koristeći sljedeće *postconf* naredbe:

```
# postconf -e 'smtpd_tls_auth_only = no'
# postconf -e 'smtp_use_tls = yes'
# postconf -e 'smtpd_use_tls = yes'
```

```
# postconf -e 'smtp_tls_note_starttls_offer = yes'
# postconf -e 'smtpd_tls_key_file = /etc/postfix/ssl/smtpd.key'
# postconf -e 'smtpd_tls_cert_file = /etc/postfix/ssl/smtpd.crt'
# postconf -e 'smtpd_tls_CAfile = /etc/postfix/ssl/cacert.pem'
# postconf -e 'smtpd_tls_loglevel = 1'
# postconf -e 'smtpd_tls_received_header = yes'
# postconf -e 'smtpd_tls_session_cache_timeout = 3600s'
# postconf -e 'tls_random_source = dev:/dev/urandom'
```

Datoteka */etc/postfix/main.cf* sada bi trebala izgledati ovako:

```
# See /usr/share/postfix/main.cf.dist for a commented, more complete version
smtpd_banner = $myhostname ESMTP $mail_name (Debian/GNU)
biff = no
# appending .domain is the MUA's job.
append_dot_mydomain = no
# Uncomment the next line to generate "delayed mail" warnings
#delay_warning_time = 4h
myhostname = server1.example.com
alias_maps = hash:/etc/aliases
alias_database = hash:/etc/aliases
myorigin = /etc/mailname
mydestination = server1.example.com, localhost.example.com, localhost
relayhost =
mynetworks = 127.0.0.0/8
mailbox_command = procmail -a "$EXTENSION"
mailbox_size_limit = 0
recipient_delimiter = +
inet_interfaces = all
smtpd_sasl_local_domain =
smtpd_sasl_auth_enable = yes
smtpd_sasl_security_options = noanonymous
broken_sasl_auth_clients = yes
smtpd_recipient_restrictions = permit_sasl_authenticated,permit_mynetworks,reject_
unauth_destination
smtpd_tls_auth_only = no
smtp_use_tls = yes
smtpd_use_tls = yes
smtp_tls_note_starttls_offer = yes
smtpd_tls_key_file = /etc/postfix/ssl/smtpd.key
smtpd_tls_cert_file = /etc/postfix/ssl/smtpd.crt
smtpd_tls_CAfile = /etc/postfix/ssl/cacert.pem
smtpd_tls_loglevel = 1
smtpd_tls_received_header = yes
smtpd_tls_session_cache_timeout = 3600s
tls_random_source = dev:/dev/urandom
```

Sada možete ponovno pokrenuti Postfix:

```
# /etc/init.d/postfix restart
Stopping mail transport agent: Postfix.
Starting mail transport agent: Postfix.
```

## Konfiguriranje POP3 i IMAP agenata za dostavu pošte

U ovom ćemo dijelu dodati Postfixu agente za dostavu pošte kako bi nadopunili Postfix. Zadajte sljedeću naredbu na Debianu da biste dodali IMAP i POP3 poslužitelje:

```
# apt-get install ipopd-ssl uw-imapd-ssl
```

Odabrali smo *ipopd-ssl* za pružanje POP2 i POP3 agenata za dostavu pošte a *uw-imapd-ssl* za IMAP. Neka vas ne zavara nastavak *ssl* – oba paketa nude servise sa šifriranjem ali i bez njega. Standardni IMAP koristi ulaz 143, a POP3 ulaz 110. Šifrirani protokoli i ulazi su POP3S (ulaz 995) te IMAPS (ulaz 993).

Iako je nastao na sveučilištu u Washingtonu, paket *ipopd-ssl* sada održava Debian. Trebate ga samo instalirati a konfigurira se sam tako da koristi početni direktorij za poštu koji se nalazi na poslužitelju elektroničke pošte kao što smo postavili u poglavlju 4. Davatelji Internet usluga još uvijek koriste POP3, ali se on rijetko koristi u poslovnim okruženjima.

*uw-imapd-ssl* dodaje IMAP poslužitelj. Premda zahtijeva više prostora na disku, IMAP je napredniji od POP-a jer poštu ostavlja na poslužitelju i omogućava korisnicima pregledavanje pošte s bilo koje lokacije koja ima pristup Internetu i klijent za elektroničku poštu. Nije nam poznat niti jedan današnji klijent za poštu koji ne podržava IMAP pa će ga odabrati većina korisnika.

Također možete na svoj poslužitelj dodati i uslugu za rad s elektroničkom poštom putem Web sučelja uz upotrebu SSL-a (*https*) za šifriranje prijenosa podataka.

U našoj konfiguraciji, korisnicima su potrebni standardni Linux korisnički računi na poslužitelju elektroničke pošte čak i ako čitaju poštu koristeći klijent na nekom drugom sustavu. Postfix obično dopušta lokalnu dostavu unutar domene, ali zahtijeva pozadinsko odašiljanje (kao što je objašnjeno u odlomku „Konfiguriranje Posfixa sa SASL-om za provjeru identiteta korisnika bez korisničkog računa“) ako su korisnici izvan domene.

*uw-imapd* ima svojih prednosti, ali i nedostataka. Prednost je to što koristi *mbox* pohranu elektroničke pošte u Unix stilu koja čuva sve korisnikove poruke u jednoj datoteci u njegovom početnom direktoriju. Ovaj se servis i vrlo lako administrira.

Njegov nedostatak je to što ne dopušta pristup pošti virtualnim korisnicima ili onima bez korisničkog računa i početnog direktorija. Pored toga, mnogim se administratorima ne sviđa jednostavni *mbox* format za pohranu pošte već daju prednost hijerarhijskom *maildir* formatu. Budući da je format s jednom datotekom, *mbox* dopušta da u isto vrijeme pošti pristupa samo jedna aplikacija, što zahtijeva zaključavanje datoteke te zbog opterećenja može usporiti sustav.

Zaključavanje datoteke (engl. *file locking*) je mehanizam koji ograničava pristup datoteci na računalu samo jednom korisniku ili procesu u određenom trenutku. Zaključavanje služi za sprječavanje konflikata koji mogu nastati ako više korisnika istovremeno uređuje istu datoteku.

Većina korisnika mehanizam zaključavanja datoteka smatra problematičnim u slučaju elektroničke pošte.Velikom broju distribuiranih datotečnih sustava nedostaje pouzdan mehanizam zaključavanja. Neki također smatraju da zaključavanje nije dovoljna zaštita od povremenih oštećenja *mbox* datoteke. Kod Linuxa, oštećenje je moguće i ako se neki proces vezan uz poštu prekine usred ažuriranja *mbox* datoteke.

Za razliku od *mbox* formata, *maildir* omogućava istovremeni pristup i ne zahtijeva zaključavanje datoteke.

Drugi IMAP poslužitelji, kao što su Cyrus, Courier i Dovecot koriste *maildir* format i dopuštaju pristup pošti virtualnim korisnicima i korisničkim računima bez pristupa školjci i polaznog direktorija. Konfigurirani zajedno s Postfixom, korisnički računi imaju samo poštanske sandučiće. Ovo administratoru omogućava održavanje agenata i za prijenos i za dostavu pošte bez rukovanja standardnim korisničkim računima na samom poslužitelju.

Za razliku od *uw-imapd*, drugi IMAP poslužitelji teški su za ovladavanje i njihovo konfiguriranje zahtijeva veliko znanje pa biste trebali sami procijeniti opravdava li veličina vašeg poduzeća njihovo korištenje. Ako je tako, trebali biste potražiti druge izvore informacija, kao što je knjiga *The Book of Postfix* čiji su autori Ralf Hildebrandt i Patrick Katter (u izdanju No Starch Press).

## Konfiguriranje klijenta za elektroničku poštu

Prilikom predstavljanja Postfixove konfiguracijske datoteke */etc/postfix/main.cf* ranije u ovom poglavlju, ostavili smo korisnikovom klijentu za elektroničku poštu posao dodavanja imena domene nakon što korisnik upiše korisničko ime:

```
append_dot_mydomain = no
```

Ovako postupa većina klijenata. Oni dodaju domenu kao što je *@centralsoft.org* kad korisnik upiše korisničko ime u polje „To“ u poruci elektroničke pošte.

Ako konfigurirate Postfix tako da koristi šifriranje kako je bilo opisano ranije u ovom poglavlju, korisnik mora konfigurirati i svoj korisnički agent za poštu da bi koristio TLS šifriranja prilikom slanja poruka. Većina modernih klijenata ga podržava i pruža grafičko sučelje koje omogućava upotrebu TLS šifriranja za odlazni poslužitelj.

Kad niste na Postfixovoj mreži a stacionarni ste (a ne mobilni) korisnik, koristite SMTP poslužitelj svog davatelja Internet usluga. Tada biste trebali odabrati TLS ako ga pružatelj usluga koristi. U većini slučajeva vaši korisnički podaci putovat će kroz mrežu pružatelja usluga u obliku čistog teksta.

Da bi vaš poslužitelj mogao primati elektroničku poštu, trebat ćete dolazni poslužitelj postaviti s DNS-om, kao što je bilo opisano u poglavlju 3. Kao kratki podsjetnik, to ćete učiniti korištenjem `MX` zapisa. Tipični `MX` zapis izgleda ovako:

```
MX 10 server1.centralsoft.org.
```

Ovaj zapis zadaje da poruka adresirana na domenu *centralsoft.org* treba biti dostavljena na *server1.centralsoft.org* (što je poslužitelj elektroničke pošte za domenu).

# Što slijedi

Sada ste instalirali i konfigurirali Postfix te IMAP i POP3 servis što znači da imate najvažnije komponente sustava elektroničke pošte koji možete koristiti u poslovnom okruženju.

Ako se prvi put susrećete s administriranjem sustava elektroničke pošte, sada možete razumjeti zašto tvrtke troše ogromnu količinu novca na licenciranje gotovih sustava projektiranih prema broju korisnika. Također ćete razumjeti zašto zapošljavaju desetke administratora za održavanje infrastrukture za komuniciranje putem elektroničke pošte. Ovo područje zahtijeva posebnu stručnost. Nakon što ovladate znanjem koje pružaju informacije u ovoj knjizi, vjerojatno ćete se željeti upoznati i s komponentama naprednih sustava elektroničke pošte. Trebali biste naučiti kako instalirati i konfigurirati skalabilan i siguran poslužitelj elektroničke pošte te biti svjesni koliko je truda potrebno da bi se ovladalo tim područjem. Trebali biste upoznati i imeničke servise kao što su OpenLDAP ili Fedora Directory Server za provjeru identiteta velikog broja korisnika i pružanje popisa korisnika vašeg sustava.

Sljedeće poglavlje opisuje servis koji većina ljudi smatra najvažnijim Internetskim servisom tvrtke: Web poslužiteljem. Nakon što vas uvedemo u postavljanje najpopularnijeg Web poslužitelja, Apachea, nastavit ćemo s dodavanjem niza važnih značajki poput podrške za dinamičke Web stranice i statistička izvješća a pružit ćemo vam i nekoliko savjeta za rješavanje problema.

POGLAVLJE 6

# Administriranje Apachea

U ovom poglavlju izgradit ćemo Web poslužitelj pod Linux operativnim sustavom. Naučit ćete kako:

- Instalirati i konfigurirati Apache, PHP i MySQL
- Upravljati sa više Web lokacija pomoću virtualnih poslužitelja
- Šifrirati stranice sa povjerljivim sadržajem pomoću SSL-a
- Omogućiti upotrebu poslužiteljskih uključenih datoteka i CGI skripti
- Testirati performanse i razinu sigurnosti
- Instalirati *vlogger* i Webalizer za pregled statistike Web lokacije
- Instalirati Drupal - program za upravljanje sadržajem koji ćete moći primijeniti u većini okruženja a koristi se većinom nabrojenih elemenata

Ovo poglavlje opisuje okolinu sa samo jednim Web poslužiteljem. U sedmom poglavlju pokazati ćemo kako postaviti par Web poslužitelja za raspoređivanje opterećenja.

Web poslužitelji veliki su i kompleksni, a prilikom njihove konfiguracije nije uvijek jasno zašto i kako sve zajedno funkcionira. Kako budemo napredovali ponekad ćemo se odlučiti i za alternativna rješenja što ćemo adekvatno istaknuti. Da bi objašnjena ostala kratka i jednostavna koristit ćemo standardne Debianove procedure i postavke. Sigurnosnim problemima početi ćemo se baviti još za vrijeme instalacije kako bismo istaknuli važnost sigurnosnog aspekta i stvorili dobre osnove za sigurnu Web lokaciju. Na samom kraju poglavlja nalazi se dio posvećen rješavanju najčešćih problema.

## Statičke i dinamičke datoteke

Klasična Web lokacija sastoji se od HTML datoteka, grafičkih datoteka, JavaScripta, stilova i drugih datoteka. Sadržaj tih datoteka je *statičan* – drugim riječima one se na poslužitelju ne mijenjaju i jedini posao poslužitelja je da ih isporuči na zahtjev preglednika.

Međutim, veliki broj Web stranica ima također i drugi, *dinamički* aspekt. To uključuje generiranje sadržaja, kontrolu pristupa, pohranjivanje podataka u bazu podataka te uzimanje podataka iz nje. Najjednostavniji način da se statička HTML stranica pretvori u

dinamičku je pomoću *poslužiteljskih uključenih datoteka* (engl. *server-side includes*, *SSI*), što su u biti posebno oblikovani HTML komentari koje Apache interpretira da bi ispisao vrijednosti varijabli ili uključio sadržaj drugih HTML stranica. SSI tako možemo iskoristiti kao jednostavan način za definiranje zajedničkog zaglavlja i podnožja Web stranica.

SSI ima svoja ograničenja te mnoge dinamičke stranice koriste uvelike moćnije Common Gateway Interface (CGI) programe. Ti programi mogu biti napisani u bilo kojem jeziku koji Linux podržava, premda su najčešći izbor dinamički (skriptni) jezici kao što su Perl, PHP, Python ili Ruby te nezaobilazna Java. CGI je u biti protokol koji definira kako Web klijent i poslužitelj razmjenjuju zahtjeve i odgovore.

Kad se CGI prvi put pojavio na Webu, CGI programi bili su potpuno odvojeni od poslužitelja. Za svaki klijentov zahtjev bilo je potrebno pokrenuti novi CGI proces. Ako bi se stranice posjećivale u većem broju povećalo bi se i opterećenje sustava što je dovodilo do problema za čije je rješavanje razvijen veliki broj alternativa.

CGI protokol se vrlo često miješa s tim ranije implementiranim metodama te se misli da je CGI i dalje spor. Međutim, CGI standard ne definira implementaciju te postoje brže metode koje slijede isti CGI protokol.

Jedna od bržih metoda je FastCGI gdje se CGI program pokreće u zasebnom procesu koji se izvršava duže vrijeme. Taj proces upravlja dvostranom komunikacijom između samog programa i Web poslužitelja. Time se izbjegavaju troškovi stalnog ponovnog pokretanja procesa, a činjenica da je proces odvojen od poslužitelja osigurava da blokiranje CGI programa neće uzrokovati i blokiranje poslužitelja. Međutim FastCGI ima i jedan nedostatak: FastCGI programi, kao ni samostalni CGI programi, ne mogu pristupati unutarnjim elementima Web poslužitelja što neke kompleksnije primjene zahtijevaju.

Neki CGI programi razvili su se u Apacheove module koji se učitavaju kao dio samog poslužitelja: Perl interpretator postao je *mod_perl*, PHP je postao *mod_php*, a *mod_squad* očajna igra riječi iz sedamdesetih. Performanse FastCGI programa i Apacheovih modula ugrubo su slične. Moduli imaju uglavnom prednosti no imaju i neke nedostatke. Moduli mogu pristupati čitavoj unutarnjoj podatkovnoj strukturi i funkcijama te se tako mogu koristiti u raznim fazama Web transakcija, a ne samo za generiranje HTML sadržaja. Međutim, moduli povećavaju memorijske potrebe Web poslužitelja, a pogreška u modulu vrlo lako može uzrokovati kvar poslužitelja.

## Jednostavna LAMP konfiguracija

Standardna LAMP konfiguracija (Linux, Apache, MySQL, PHP/Perl/Phyton) koristi Apache module za izvođenje CGI funkcija. Ovaj pristup funkcionira jako dobro, no i tu postoje ograničenja. Neka od njih istaknuti ćemo u ovom poglavlju. Ali, ako volite učiti na vlastitim pogreškama te dijelove slobodno možete preskočiti. L smo obradili, sada ćemo proučiti A, a M i P nešto kasnije.

Apache nije najbrži Web poslužitelj a niti najjednostavniji za konfiguriranje. Nije niti najsigurniji no dovoljno je dobar da dominira nad svim drugim Web poslužiteljima.

Prema Netcraftu, na Apache otpada više od 60 posto javnih Web poslužitelja (*http://news.netcraft.com/archives/web_server_survey.html*). Apache se izvodi pod Linuxom, sustavom Mac OS X i drugim sustavima temeljenim na Unixu kao i pod većinom inačica Microsoft Windowsa.

Kao i ostali Unix programi, Apache se može izgraditi sa svim svojim modulima skupljenim u jedan veliki program (*statičko povezivanje*) ili sa modulima koji se učitavaju po potrebi (*dinamički dijeljeni objekti*). Metoda dinamičkih dijeljenih objekata je jednostavnija i mnogo fleksibilnija budući da dozvoljava dodavanje modula i nakon instalacije Apachea. Debianova instalacija PHP-a i drugih Apacheovih modula koristi ovu metodu.

# Instaliranje

U ovom poglavlju instalirati ćemo Apache, PHP i MySQL. Testirat ćemo svaki od njih sa standardnim postavkama da bismo se uvjerili u ispravnost instalacije. U sljedećim dijelovima uroniti ćemo u Apacheovu konfiguracijsku datoteku i istražiti postupak prilagodbe postavki našim potrebama.

## Apache

Da biste instalirali pakete morate biti *root* korisnik. Prvo instalirajte Apache poslužitelj:

```
# apt-get install apache2
```

Ovom naredbom Apache se instalira i pokreće. Da li radi? To možete provjeriti ako unesete URL vaše lokacije u Web preglednik. U ovom poglavlju ćemo kao primjer koristiti naziv našeg testnog poslužitelja (*http://server1.centralsoft.org*). U svim sljedećim primjerima ovaj URL zamijenite sa URL-om vaših stranica. Ako se preglednik izvršava na istom računalu na kojem je instaliran i poslužitelj, možete imati problema s pronalaženjem imena poslužitelja. Ako se taj problem pojavi koristite *http://localhost* ili *http://127.0.0.1*. Ako testirate izvana, koristite poslužiteljevu IP adresu, npr. *http://70.253.158.41*.

Kad ste u Web preglednik unijeli URL trebala bi se prikazati stranica koja počinje s:

```
If you can see this, it means that the installation of the Apache
web server software on this system was successful. You may now add
content to this directory and replace this page.
```

Preglednik bi također trebao pokazati da je Apache preusmjerio unesenu adresu u: *http://server1.centralsoft.org/apache2-default*.

Ovo ćemo objasniti nešto kasnije kad se počnemo baviti Apacheovom konfiguracijskim datotekama. Sada ćemo izraditi našu prvu Web stranicu. Uđite u direktorij koji Apache smatra početnim direktorijem za vašu Web lokaciju i izradite malu tekstualnu datoteku:

```
# cd /var/www
# echo testing > test.html
```

Sada unesite njen URL (npr. *http://server1.centralsoft.org/test.html*) u preglednik.

U pregledniku bi se trebala pojaviti riječ `testing`. Vaš se Apache poslužitelj izvodi bez restrikcija pristupa, poslužujući sve datoteke i direktorije koji se nalaze u direktoriju */var/www*.

## PHP

PHP je najpopularniji Apacheov CGI modul. U ovom poglavlju koristimo PHP 4, koji je popularniji od svojeg nasljednika PHP 5. Korištenje neke od ovih inačica je dobar način za izradu dinamičkih Web stranica a velika biblioteka PHP modula stavlja vam na raspolaganje mnogo korisnih funkcija. Započnimo s instaliranjem PHP programa i biblioteke:

```
# apt-get install php4
```

Sada instalirajte Apacheov PHP modul, *mod_php*. Sljedeća naredba instalira *mod_php* i govori Apacheu da izvršava datoteke s nastavkom imena *.php*:

```
# apt-get install libapache2-mod-php4
```

Izradite testnu PHP skriptu i spremite je kao */var/www/info.php*:

```
<?php
phpinfo();
?>
```

Sada u Web preglednik unesite URL skripte (*http://server1.centralsoft.org/info.php*).

U pregledniku bi se trebala pojaviti stranica s tablicom ispunjenom konfiguracijskim informacijama PHP-a. Ove informacije govore jako puno o vašem računalu i nisu nešto što bi htjeli podijeliti sa ostatkom svijeta, tako da skriptu poslije testiranja obrišite. Ako se u pregledniku nije pojavilo ništa, pogledajte dio o rješavanju najčešćih problema na kraju ovog poglavlja.

Usput, ako se još niste bavili CGI skriptama, upravo ste napisali svoju prvu CGI skriptu! (U kasnijem dijelu o CGI-ju, dati ćemo više detalja o tome kako Web poslužitelj izvršava vanjske programe i skripte).

## MySQL

U slučaju da ne trebate bazu podataka, imate LAP platformu i možete preskočiti ovaj dio. Za puni LAMP instalirajte MySQL poslužitelj i PHP MySQL modul:

```
# apt-get install mysql-server
# apt-get install php4-mysql
```

Ovo je sve što trebate da biste mogli pisati PHP CGI skripte i pristupati MySQL bazi podataka na poslužitelju. Međutim instalirati ćemo i standardni MySQL klijent (*mysql*) koji se pokreće iz odzivnika tako da možemo testirati bazu podataka bez korištenja PHP-a i Apachea.

```
# apt-get install mysql-client
```

Ako pokrenete *mysql* klijent, ali ne zadate MySQL korisničko ime sa *-u* opcijom, *mysql* pokušava koristiti vaše Linux korisničko ime. U našem primjeru, prijavili smo se kao *root* korisnik, tako da će naše ime biti *root*. Korisničko ime MySQL administratora također će se zvati *root* i imati će punu kontrolu nad bazom podataka. Međutim, MySQL-ov korisnički račun *root* i Linuxov korisnički račun *root* nisu u nikakvoj vezi. MySQL svoja korisnička imena i lozinke sprema u bazu podataka kojom upravlja.

Sljedeću naredbu koristite kako bi provjerili je li poslužitelj baze podataka ispravno instaliran:

```
# mysql -u root
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 5 to server version: 4.0.24_Debian-10sarge2-log

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

mysql> show databases;
+----------+
| Database |
+----------+
| mysql    |
| test     |
+----------+
2 rows in set (0.00 sec)

mysql> quit;
Bye
#
```

Ako ova naredba funkcionira, MySQL je ispravno instaliran. Loša vijest je da MySQL korisnik *root* na početku nema lozinku. Zbog toga mu je moramo dodijeliti (upišite lozinku po vašem izboru gdje god smo napisali `novamysqllozinka`):

```
# mysqladmin -u root password novamysqllozinka
```

Sada ponovno pokušajte pristupiti bazi podataka bez lozinke:

```
# mysql -u root
ERROR 1045: Access denied for user: 'root@localhost' (Using password: NO)
```

Drago nam je da je nešto pošlo po krivu, pošto smo to i očekivali. Pokušajmo ponovno:

```
# mysql -u root -p
Enter password: novamysqllozinka
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 8 to server version: 4.0.24_Debian-10sarge2-log

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

mysql> quit;
```

Zapišite lozinku jer ćete ju trebati kasnije u ovom poglavlju kada budete instalirali aplikaciju Drupal, kao kada budete željeli pristupiti MySQL-u kao glavni administrator.

Iz sigurnosnih razloga, standardna instalacija MySQL-a koju smo proveli ograničava MySQL poslužitelj na lokalne klijente kao što su PHP Web skripte ili *mysql* klijent. Da nije tako, bilo tko bi se preko Interneta mogao spojiti s vašom bazom podataka. Možete provjeriti je li adresa MySQL poslužitelja 127.0.0.1 (lokalna ili *loopback* adresa) koristeći naredbu:

```
# netstat -tlnp
Proto Recv-Q Send-Q Local Address   Foreign Address  StatePID/Program name
tcp        0      0 127.0.0.1:3306  0.0.0.0:*        LISTEN25948/mysqld
```

# Apacheove konfiguracijske datoteke

Apache koristi čiste ASCII konfiguracijske datoteke. Njihove lokacije ovise o distribuciji Linuxa, a tablica 6-1 pokazuje gdje ih sprema Debian.

*Tablica 6-1. Apacheove konfiguracijske datoteke*

| Datoteka/direktorij u /etc/apache2 | Svrha |
|---|---|
| *apache2.conf* | Glavna konfiguracijska datoteka. Učitava ostale datoteke kroz sljedeće direktive:<br>`# Include module configuration:`<br>`Include /etc/apache2/mods-enabled/*.load`<br>`Include /etc/apache2/mods-enabled/*.conf`<br><br>`# Include all user configurations:`<br>`Include /etc/apache2/httpd.conf`<br><br>`# Include ports listing`<br>`Include /etc/apache2/ports.conf`<br><br>`# Include generic snippets of statements`<br>`Include /etc/apache2/conf.d/[^.#]*` |
| *conf.d/** | Ovdje možete dodati što god želite. Standardno je ova datoteka prazna. |
| *mods-enabled/*.conf* | Definicije za svaki uključeni modul. Debian sadrži program a2enmod za uključivanje modula i a2dismod za njihovo isključivanje. Bit je u premještanju datoteka xyz.conf i xyz.load između /etc/apache/mods-available i /etc/apache2/mods-enabled za modul nazvan xyz. Datoteka apache2.conf koristi datoteke iz mods-enabled. |
| *sites-enabled/** | Definicije za svaku Web lokaciju. Standardno ime je 000-default premda nema ničeg posebnog oko imena. Ovdje možete imati proizvoljan broj datoteka. |
| *.htaccess* | Definicije za direktorije sadržane u tom istom direktoriju. Premošćuje postavke iz drugih konfiguracijskih datoteka budući se čita posljednja. To je dozvoljeno samo ako je AllowOverride postavljeno na none. Može biti promijenjena bez ponovnog učitavanja Apachea. To je način na koji administratori Web lokacija dozvoljavaju klijentima prilagodbu stranica bez uređivanja glavne Apacheove konfiguracijske datoteke. |

Ako je `AllowOverride` uključeno za bilo koji direktorij, na svaki klijentov zahtjev Apache mora u svim direktorijima, od početnog nadolje, potražiti *.htaccess* datoteku i pročitati je. Iako ovaj proces značajno usporava Apache mnogo je važnija činjenica da je u slučaju korištenja ove datoteke vrlo teško utvrditi koja je opcija važeća. Ako ne trebate *.htaccess* datoteke, nemojte ih koristiti, a standardno nisu ni uključene.

## Direktive konfiguracijske datoteke

Svaka Apacheova konfiguracijska datoteka podijeljena je na dijelove koji sadrže Apacheove *direktive* (naredbe ili postavke) i njihove vrijednosti. Neke su direktive dio Apacheove jezgre, dok druge koriste samo neki specifični moduli. Ako direktiva referira na modul za koji još niste konfigurirali Apache, Apache se neće pokrenuti i u dnevnik će biti upisana poruka o pogrešci zajedno s problematičnim redovima.

Nakon što ste uspješno pokrenuli Apache možete provjeriti koje direktive možete koristiti zadavanjem:

```
# /usr/sbin/apache2 -L
```

Dio o rješavanju problema na kraju ovog poglavlja sadrži upute „korak po korak“ koje će vam pomoći da utvrdite što izaziva probleme s Web poslužiteljem.

Ako testna datoteka radi ispravno možete početi konfigurirati Apache. Slijedi sadržaj standardne Apacheove konfiguracijske datoteke */etc/apache2/sites-enabled/000-default*. Dijelovi započinju i završavaju s oznakama u HTML stilu, primjerice:

```
<VirtualHost *>
...
</VirtualHost>
```

Ovdje je kopija datoteke koju smo dopunili komentarima:

```
# Answer to any name or IP address:
NameVirtualHost *

# For any virtual host at any address, any port:
<VirtualHost *>
    # If Apache has problems, whom should it contact?
    ServerAdminwebmaster@localhost

    # Our web site files will be under this directory:
    DocumentRoot /var/www/

    # Overall directives, in case we move DocumentRoot
    # or forget to specify something later:
    <Directory />
        # Lets Apache follow symbolic links:
        Options FollowSymLinks
        # Disables .htaccess files in subdirectories:
        AllowOverride None
    </Directory>

    # DocumentRoot itself:
    <Directory /var/www/>
        Options Indexes FollowSymLinks MultiViews
        # Forbids .htaccess files:
        AllowOverride None
        Order allow,deny
        allow from all
        # Maps / to /apache2-default, the initial welcome
```

```
        # page that says "If you can see this...":
        RedirectMatch ^/$ /apache2-default/
    </Directory>

    # Permits CGI scripts:
    ScriptAlias /cgi-bin/ /usr/lib/cgi-bin/
    <Directory "/usr/lib/cgi-bin">
        AllowOverride None
        Options ExecCGI -MultiViews +SymLinksIfOwnerMatch
        Order allow,deny
        Allow from all
    </Directory>

    # Error log for a single site:
    ErrorLog /var/log/apache2/error.log

    # Possible values include: debug, info, notice,
    # warn, error, crit, alert, and emerg:
    LogLevel warn

    # Access log for a single site:
    CustomLog /var/log/apache2/access.log combined

    # Sends Apache and PHP version information to browsers;
    # Set to Off if you're paranoid, or have reason to be:
    ServerSignature On

    # Shows Apache docs (only to local users)
    # if you installed apache2-docs;
    # to suppress showing the documents,
    # you can comment these lines or delete them:
    Alias /doc/ "/usr/share/doc/"
    <Directory "/usr/share/doc/">
        Options Indexes MultiViews FollowSymLinks
        AllowOverride None
        Order deny,allow
        Deny from all
        Allow from 127.0.0.0/255.0.0.0 ::1/128
    </Directory>
</VirtualHost>
```

Većina promjena koje ćemo izvesti u konfiguraciji Apachea odnose se upravo na ovu datoteku. Datoteka za općenito konfiguriranje poslužitelja */etc/apache2/apache2.conf* sadrži veliki broj postavki koje se tiču čitavog poslužitelja i koje najčešće nije potrebno mijenjati, a u nastavku ćemo navesti nekoliko značajnih iznimki.

### Direktive User i Group

Ove važne postavke govore Apacheu da se izvršava pod određenim korisničkim, odnosno grupnim identifikatorom. Standardne Debianove postavke u */etc/apache2/apache2.conf* su:

```
User www-data
Group www-data
```

Datoteke i direktoriji koje će Apache posluživati moraju biti čitljive navedenom korisniku i grupi. Pogrešna dopuštenja vrlo često uzrokuju pogreške u radu Apachea, kao što je nemogućnost prikaza stranice (ili pak mogućnost prikazivanja nečega što ne bi trebalo biti prikazano).

### Direktiva Listen

Apache normalno odgovara na zahtjev poslan preko TCP ulaza 80, međutim može se postaviti da radi i na drugim ulazima. Vrlo često se za testiranje koristi neki drugi ulaz – mnogi koriste 81 jer se lako pamti i ne koristi se za neku drugu svrhu. Za zadavanje ulaza koristite jednu ili više `Listen` direktiva:

```
Listen 81
```

Ako ćete za neke stranice koristiti SSL šifriranje, morate uvrstiti i direktivu za osluškivanje standardnog sigurnog ulaza:

```
Listen 443
```

### Direktiva DocumentRoot

Svaka Web lokacija ima korijenski direktorij u kojemu se nalaze datoteke sa sadržajem i skripte, a zadaje se direktivom `DocumentRoot`. U standardnoj Debian/Apache konfiguraciji to je zadano u */etc/apache2/sites-enabled/000-default*:

```
DocumentRoot /var/www/
```

## Provjera identiteta i autorizacija

Neki dijelovi Web lokacije biti će namijenjeni široj javnosti, a vjerojatno ćete htjeti imati i dijelove rezervirane samo za određenu skupinu posjetitelja. *Provjera identiteta* (engl. *authentication*) utvrđuje *tko* je posjetitelj, a *autorizacija* (engl. *authorisation*) *što* taj posjetitelj može raditi, primjerice:

- Čitati datoteku
- Koristiti poslužiteljske uključene datoteke
- Pokrenuti CGI program
- Generirati početnu stranicu za direktorije koji ju nemaju

U Apacheu je standardno mjesto za spremanje informacija o korisnicima *korisnička* tekstualna datoteka (često se zove i *.htpasswd* datoteka, po programu koji se koristi za njeno modificiranje). Ova datoteka sadrži korisnička imena i šifrirane lozinke korisnika. Opcionalna *grupna* datoteka sadrži identifikatore grupe i korisnika. Korisna je za veće lokacije jer omogućava zadavanje dopuštenja za grupu što je ponekad korisnije od zadavanja pojedinačnih dopuštenja za svakog korisnika.

### Korisničke datoteke

Za primjer izradite direktorij zaštićen lozinkom i u njega smjestite malu tekstualnu datoteku:

```
# cd /var/www
# mkdir secret
```

```
# cd secret
# echo "now you see it" > file.html
```

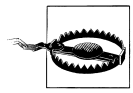
Kako je još niste zaštitili, datoteka će biti vidljiva u pregledniku (*http://server1.centralsoft.org/secret/file.htm*):

```
now you see it
```

Sada izradite korisničku datoteku:

```
# cd /tmp
# htpasswd -c /tmp/users jack
New password: black_pearl
Re-type new password: black_pearl
Adding password for user jack
```

Lozinka neće biti prikazana dok ju unosite. Kada prvi put pokrenete program *htpasswd* s nekom datotekom trebate uključiti argument `-c`. To će izraditi datoteku.



Ako kasnije budete dodavali još korisnika nemojte koristiti argument `-c` jer će to prouzročiti prepisivanje postojeće datoteke.

Ako želite promijeniti lozinku za korisnika *jack* upišite:

```
# htpasswd /tmp/users jack
New password: kraken
Re-type new password: kraken
Updating password for user jack
```

Korisnička datoteka sastoji se od redova koji sadrže korisničko ime i šifriranu lozinku, odvojene dvotočkom, kao u sljedećem primjeru:

```
jack:OSRBcYQOd/qsI
```

Sada uredite Apacheovu konfiguracijsku datoteku */etc/apache2/sites-enabled/000-default* i dodajte (prije završnog reda `</VirtualHost>`):

```
<Location /secret>
    AuthName "test"
    AuthType Basic
    AuthUserFile /tmp/users
    Order deny,allow
    require valid-user
</Location>
```

`AuthName` obavezan je parametar i mora mu slijediti skup znakova u navodnicima. Ovdje koristimo `"test"` a možete koristiti i `""` (prazan skup znakova) ako želite, ali iz nekog razloga ova se direktiva ne može ispustiti. `AuthType Basic` znači da koristimo *htpasswd* korisničku datoteku. `AuthUserFile` zadaje lokaciju korisničke datoteke. Direktiva `Order` govori da Apache standardno treba uskratiti pristup, a dozvoliti ga samo ako se pristupni podaci podudaraju s podacima u korisničkoj datoteci. Najzad, direktiva `require` kaže da su svi korisnici iz korisničke datoteke dozvoljeni. Da biste dozvolili pristup samo korisniku *jack* zadajte sljedeće:

```
require jack
```

Ako želite pristup dozvoliti većem broju korisnika, to radite na način prikazan u sljedećem primjeru:

```
require jack will elizabeth
```

Da bi promjene postale aktivne Apache mora ponovno učitati svoju konfiguracijsku datoteku:

```
# /etc/init.d/apache2 reload
```

Sada pokušajte pristupiti datoteci s tajnim sadržajem (*http://www.example.com/secret/file.html*) koristeći se korisničkim imenom navedenim u korisničkoj datoteci. Vidjet ćete dijalog s otprilike ovakvim sadržajem:

```
Enter username and password for "test" at server1.centralsoft.org
Username:
Password:
```

Unesite korisničko ime i lozinku (dok unosite lozinku umjesto znakova prikazivat će se zvjezdice) i pritisnite na OK. Trebali biste vidjeti:

```
now you see it
```

### Datoteke grupe

Drugi način za rad s više korisnika je pomoću datoteke grupe. Izradite datoteku */tmp/groups* koja će sadržavati naziv grupe, dvotočku te jedno ili više korisničkih imena odvojenih razmacima:

```
pirates: jack will elizabeth
```

Korisnici se mogu pridruživati grupi i pojedinačno:

```
pirates: jack
pirates: will
pirates: elizabeth
```

Sada dodajte direktivu `AuthGroupFile` datoteci *000-default*:

```
<Location /secret>
    AuthName "test"
    AuthType Basic
    AuthUserFile /tmp/users
    Order deny,allow
    AuthGroupFile /tmp/groups
    require group pirates
</Location>
```

Kao i obično, da bi se promjene aktivirale ponovno pokrenite Apache:

```
# /etc/init.d/apache2 reload
```

## Kontejneri i aliasi

Apache primjenjuje autorizacijska ograničenja na *kontejnere* ili datoteke i direktorije na poslužitelju. Jedan od kontejnera je i odjeljak `Location` koji smo ranije objasnili. Sada ćemo se pozabaviti s različitim direktivama kontejnera.

### Apsolutna putanja: Directory

Ova direktiva zadaje direktorij na poslužitelju. Navodimo primjer iz naše konfiguracijske datoteke za Apache:

```
<Directory />
    Options FollowSymLinks
    AllowOverride None
</Directory>
```

### Relativna putanja: Location

Ova direktiva zadaje datoteke i direktorije u odnosu na korijenski direktorij. Primjerice:

```
<Location /cgi>
    Options ExecCGI
</Location>
```

dozvoljava izvršavanje CGI programa u */var/www/cgi*. Ovime ćemo se ponovno baviti u dijelu koji se bavi CGI-om.

### Uspoređivanje uzorka: Files i FilesMatch

Ponekad ćete trebati zadati datoteku ili direktorij prema nekom tekstualnom uzorku. Navodimo primjer koji sprječava preuzimanje slika s vaših stranica bez autorizacije tako da se provjerava izvor zahtjeva za preuzimanjem slike. Koristi se direktiva `FilesMatch` koja dozvoljava zadavanje izraza (uzoraka) unutar navodnika:

```
# Some notes on the regular expression:
#   \. means a literal dot character.
#   (gif|jpg|jpeg|png) means any of these four strings.
#   $ means the end of the filename.
# The regular expression will match files with the suffix
#   .gif, .jpg, .jpeg, or .png.
<FilesMatch "\.(gif|jpg|jpeg|png)$">
    # Set the environment variable local to 1
    # if the referring page (the URL this image
    # was called from) is on this site.
    # Set local to 0 if the URL was on another site
    # that wants to steal our lovely images.
    SetEnvIfNoCase Referer "^http://server1.centralsoft.org/" local=1
    Order Allow, Deny
    # This checks the local variable and
    # allows access only if the referrer was local.
    Allow from env=local
</FilesMatch>
```

### Aliasi

Direktiva `Alias` pridružuje ime direktoriju:

```
Alias /test /tmp/test
```

Alias (novo ime) dolazi prvo u direktivi, nakon njega slijedi stvarna lokacija direktorija. Direktorij se može nalaziti i izvan korijenskog direktorija. U ovom slučaju datoteci */tmp/test/button.gif* moći će se pristupati preko URL-a *http://www.example.com/test/button.gif* čak i ako se ne nalazi u direktoriju */var/www/test*.

## Limiti

Na opterećenom poslužitelju Apache može pokrenuti veliki broj procesa potomaka koji rade simultano i koriste veliku količinu memorije. To može povećati prosječno opterećenje i učiniti sustav sporim, a može se dogoditi i da poslužitelj prestane odgovarati na zahtjeve klijenata. Tablica 6-2 pokazuje kako se mogu ograničiti neke vrijednosti Apacheovih parametara izvršavanja u konfiguracijskoj datoteci.

*Tablica 6-2. Apacheove direktive za ograničavanje resursa*

| Direktiva | Standardna vrijednost | Svrha |
|---|---|---|
| MaxClients | 256 | Maksimalan broj istovremenih zahtjeva. Ako stigne još zahtjeva, biti će odbijeni. |
| MaxRequestPerChild | 0 (infinite) | Maksimalan broj obrađenih zahtjeva prije nego što se proces potomak ponovno pokrene. Koristi se da bi se izbjeglo curenje memorije. |
| KeepAlive | on | Ponovno koristi aktivnu TCP vezu između klijenta i Apachea. Povećava propusnost slanjem čitavog sadržaja stranice preko iste veze. |
| KeepAliveTimeout | 15 | Maksimalno vrijeme (u sekundama) koje će se čekati drugi zahtjev na istoj vezi. |

## Poslužiteljski umetci

Poslužiteljski umetci (engl. *Server-Side Includes, SSI*) se mogu koristiti za umetanje sadržaja datoteke, rezultata nekog programa ili sadržaja varijable okruženja u HTML datoteku. Sintaksa za zadavanje poslužiteljskih umetaka u Apacheu može biti zbunjujuća. Primjerice, da biste dozvolili *samo* poslužiteljske umetke u */var/www/ssi*, bez dodatnih opcija, izradite direktorij:

```
# mkdir /var/www/ssi
```

i kažite Apacheu da dozvoli samo poslužiteljske umetke unutar njega:

```
<Location /ssi>
    Options Includes
</Location>
```

Da biste dodali poslužiteljski umetak postojećim opcijama koristite:

```
<Location /ssi>
    Options +Includes
</Location>
```

Poslužiteljski umetak omogućava uključivanje sadržaja datoteke, ali isto tako može pokrenuti bilo koji program i uključiti njegov rezultat. Ta opcija može biti opasna za

sigurnost sustava. Zbog toga biste trebali ograničiti poslužiteljske umetke samo na umetanje sadržaj datoteka:

```
<Location /ssi>
    Options IncludesNoExec
</Location>
```

Ako želite imate datoteke umetaka na raznim mjestima a ne samo u ovom direktoriju, možete reći Apacheu da umecima pridruži određeni nastavak imena datoteke:

```
AddHandler server-parsed .shtml
```

Da bi umeci radili mora biti učitan Apacheov modul *include*. Budući da se on ne učitava u standardnim Apache i PHP konfiguracijama učinit ćemo sljedeće:

```
# a2enmod include
Module include installed; run /etc/init.d/apache2 force-reload to enable.
# /etc/init.d/apache2 force-reload
```

SSI naredbe izgledaju kao HTML komentari, tj. imaju sljedeći oblik:

```
<!--#naredba argument="vrijednost"-->
```

Moguće vrijednosti za `naredba` su `include` (uključivanje sadržaja datoteke), `echo` (prikazivanje vrijednosti varijable okruženja), `exec` (uključivanje rezultata naredbe) i `config` (oblikovanje neke `echo` varijable). Testirajmo najprije uključivanje sadržaja datoteke. Izradite dvije datoteke:

```
# cd /var/www/ssi
# echo "top stuff" > top.html
# echo "bottom stuff" > bottom.html
```

Sada izradite datoteku *middle.shtml* sa sljedećim sadržajem:

```
<!--#include virtual="top.html"-->
middle stuff!
<!--#include virtual="bottom.html"-->
```

Primijetite da datoteka koja uključuje sadržaj (*middle.shtml*) mora imati nastavak *.shtml*, dok datoteke koje se uključuju (*top.html* i *bottom.html*) ne moraju. Sada usmjerite preglednik na *http://server1.centralsoft.org/middle.shtml*. Trebali biste vidjeti:

```
top stuff
middle stuff!
bottom stuff
```

Ako je opcija `Include` postavljena na kontejner, umetak također može izvršavati naredbe, ali mu korisnik (najčešće u pregledniku) ne može pružiti niti jednu direktivu. Izvršavanje naredbi kroz poslužiteljske umetke koristi se za jednostavnije stvari kao što je primjerice ispisivanje sadržaja direktorija:

```
<!--#exec cmd="ls -l /tmp"-->
```

Glavna primjena poslužiteljskih umetaka je prikaz sadržaja CGI varijabli okruženja i nekih drugih prikladnih varijabli. Brzi način za ispis sadržaja varijabli jest:

```
<!--#printenv-->
```

Za određenu varijablu, red:

```
<!--#echo var="DATE_GMT"-->
```

prikazuje nešto poput ovoga:

```
Tuesday, 01-Aug-2006 02:42:24 GMT
```

Ako imate samo statičke datoteke ili mješavinu statičkih datoteka i CGI skripti, najsigurnije je isključiti izvršavanje naredbi kroz umetke:

```
<Location />
    Options IncludesNoExec
</Location>
```

## CGI

CGI je daleko fleksibilniji (ali i opasniji) način za izvršavanje programa na Web poslužitelju jer korisnik može prosljeđivati informacije programu. Apache podržava dva načina za zadavanje programa koji mogu se mogu izvršavati kao CGI programi.

### Lokacija

Bilo koja od sljedeće dvije direktive pridružit će CGI programima iz direktorija */var/cgi/* URL koji počinje s *http://server1.centralsoft.org/cgi/*:

```
ScriptAlias /cgi /var/cgi
```

ili

```
<Location /cgi>
    Options ExecCGI
</Location>
```

### Nastavak imena datoteke

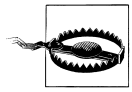
Metoda s nastavcima imena datoteka pridružuje *MIME tip* (standard za imenovanje tipova datoteka) nastavku imena datoteke. PHP modul pod Apacheom koristi ovu metodu da bi datoteke tipa *.php* proslijedio interpretatoru *mod_php*:

```
AddType application/x-httpd-php .php
```

Evo kompletnog sadržaja Apacheove konfiguracijske datoteke za *mod_php* (*/etc/apache2/mods-enabled/php4.conf*) koji obrađuje datoteke s nastavcima *.phtml* i *.php3* kao PHP datoteke:

```
<IfModule mod_php4.c>
    AddType application/x-httpd-php .php .phtml .php3
    AddType application/x-httpd-php-source .phps
</IfModule>
```

Prvi `AddType` red zadaje izvršavanje svake datoteke koja završava s *.php*, *.php3* ili *.phtml* kao PHP CGI programa. Drugi `AddType` red daje Apacheu uputu da ispisuje sadržaj datoteka sa nastavkom *.phps* umjesto da ih izvršava i vraća njihov rezultat. Programeri ovaj postupak koriste kako bi izvršili skriptu (*.php*) i omogućili korisniku pogled na inačicu za ispis (*.phps*). Ako slučajno upotrijebite nastavak imena datoteke *.phps* a trebali ste *.php*, vaša se skripta neće izvršiti već će se umjesto toga prikazati njen sadržaj.



Nikada nemojte stavljati interpretatore skripti kao što su Perl, PHP ili Linux školjka u CGI direktorij. U tom bi ih slučaju bilo tko mogao izvršavati s punim dopuštenjima Apacheova korisnika i grupe.

Kada ste ranije testirali instalaciju PHP-a izradili ste mali PHP CGI program:

```
<?php
phpinfo();
?>
```

Sada pokušajmo nešto mnogo zanimljivije: spojiti ćemo se na MySQL poslužitelj, izvršiti SQL upit i prikazati rezultat kao HTML. Ponovno ćemo trebati MySQL *root* korisničko ime. Spremite ovu datoteku kao */var/www/db.php*:

```
<?php
$link = mysql_connect("localhost", "root", "newmysqlpassword");
if (!$link) {
        echo "Can't connect to database.  Drat.\n";
        exit();
}
$result = mysql_query("show databases");
if (!$result) {
        echo "Arggh, a database error: ", mysql_error();
        exit();
}
# print_r prints all of a variable's contents
while ($row = mysql_fetch_assoc($result))
        print_r($row);
?>
```

Upišite URL *http://server1.centralsoft.org/db.php* u preglednik. Prikazat će se:

```
Array ( [Database] => mysql ) Array ( [Database] => test )
```

Da ste zadali iste SQL naredbe u *mysql* klijentu dobili biste isti rezultat (dvije baze podataka pod nazivima *mysql* i *test*) ali u različitom formatu:

```
$ mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 2996 to server version: 4.0.24_Debian-10sarge2-log

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

mysql> show databases;
+----------+
| Database |
+----------+
| mysql    |
| test     |
+----------+
2 rows in set (0.00 sec)
```

## Direktive specifične za PHP modul

PHP direktive mogu biti smještene u PHP-ovu vlastitu konfiguracijsku datoteku (*/etc/php4/apache2/php.ini*) ili u Apacheovu konfiguracijsku datoteku. Obično s ovim datotekama ne trebate raditi osim kada instalirate PHP module ili želite promijeniti mjesto na kojem PHP traži biblioteke te poboljšati sigurnosne postavke (npr. sigurni način rada). Klasični Apacheovi moduli imaju konfiguracijske datoteke s nastavkom *.conf*, smještene u direktoriju */etc/apache2/mods-enabled*.

## Virtualni poslužitelji

Iako Apache možete koristiti za posluživanje samo jedne Web lokacije, vjerojatnije je da ćete ga koristiti za nekoliko lokacija. Apache takvu konfiguraciju, kada poslužuje više lokacija, naziva *virtualni poslužitelj* i pruža nekoliko načina za njihovo zadavanje. Kad Web klijent uspostavlja vezu s Web poslužiteljem preko HTTP-a on šalje IP adresu odredišta i (u trenutnoj inačici HTTP 1.1 protokola) ime poslužitelja koji se nalazi na toj adresi.

U standardnoj konfiguraciji Apachea nema nezavisnih virtualnih poslužitelja. Apache će posluživati stranice bez obzira koliko poslužitelj ima imena, a sva imena domena dijele istu konfiguraciju.

U slijedećim primjerima pretpostavit ćemo da smo svaku Web lokaciju smjestili u zaseban poddirektorij u */var/www/vhosts*.

### Virtualni poslužitelji na temelju IP adrese

Ako na poslužitelju imate više od jedne IP adrese i želite određenu adresu pridružiti određenoj Web lokaciji koristit ćete virtualne poslužitelje temeljene na IP adresi:

```
<VirtualHost 192.168.6.1>
    ServerName "www1"
    DocumentRoot "/var/www/vhosts/www1.example.com"
</VirtualHost>
<VirtualHost 192.168.6.2>
    ServerName "www2"
    DocumentRoot "/var/www/vhosts/www2.example.com"
</VirtualHost>
```

Ovo se veoma često koristilo u ranim danima Weba budući da HTTP 1.0 nije nudio način da zadate koji poslužitelj želite koristiti na toj adresi. Dolaskom inačice protokola HTTP 1.1 virtualni poslužitelji temeljeni na imenu postala su daleko popularniji.

### Virtualni poslužitelji na temelju imena

Kod ove metode direktiva `NameVirtualHost` definira adrese koje mogu biti virtualni poslužitelji. * znači da bilo koje ime ili adresa poslužitelja, uključujući *localhost*, *127.0.0.1*, *www.centralsoft.org*, *www2.centralsoft.org*, ili druge. Pojedinačne `ServerName` direktive

pridružuju ime poslužitelja iz preglednikovog zahtjeva direktoriju u kojem su smještene potrebne datoteke:

```
# Accept any site name on any port:
NameVirtualHost *
<VirtualHost *>
    ServerName www1.example.com
    DocumentRoot "/var/www/vhosts/www1.example.com"
</VirtualHost>
<VirtualHost *>
    ServerName www2.example.com
    # A virtual host can have multiple names:
    ServerAlias backup.example.com
    DocumentRoot "/var/www/vhosts/www2.example.com"
</VirtualHost>
```

### mod_vhost_alias

Ako želite administrirati više poslužitelja bez zadavanja svih njihovih imena u konfiguracijskim datotekama možete uključiti Apacheov modul *mod_vhost_alias*:

```
# a2enmod vhost_alias
```

i konfigurirati imena za posluživanje u za to namijenjenoj datoteci. U prethodnoj naredbi, *vhost_alias* kratica je za */etc/apache2/mods-enabled/vhost_alias.conf*. Njen sadržaj može biti:

```
UseCanonicalName    Off
VirtualDocumentRoot /var/www/vhosts/%0
```

Direktiva `VirtualDocumentRoot` izuzetno je fleksibilna. Oznaka `%0` koju smo ovdje zadali proširuje se do punog imena lokacije (*server1.centralsoft.org)*. Mogli smo koristiti `%2` da bismo dobili drugi dio s lijeva (*centralsoft*), `%-2` za drugi dio s desna (također *centralsoft*), `%2+` za dio od drugog dijela do kraja (*centralsoft.org*) i tako dalje. Ove su alternative korisne ako imate puno virtualnih poslužitelja. Ako uvijek imate isto bazno ime domene kao što je *centralsoft.org* i lokacije nazvane *www1.centralsoft.org*, *www2.centralsoft.org*, itd. možete koristiti `%1` da biste dobili direktorije */var/www/vhosts/www1*, */var/www/vhosts/www2* itd.

Za sada, koristite samo `%0` za puno ime i izradite direktorij za svaki virtualni poslužitelj:

```
# cd /var/www/vhosts
# mkdir www1.centralsoft.org
# echo "test www1.centralsoft.org" > www1.centralsoft.org/index.html
# mkdir www2.centralsoft.org
# echo "test www2.centralsoft.org" > www2.centralsoft.org/index.html
```

Ponovno pokrenite Apache da bi promjene stupile na snagu:

```
# /etc/init.d/apache2 reload
```

Ako imate DNS zapise koji *www1.centralsoft.org* i *www2.centralsoft.org* upućuju na vaš poslužitelj, možete u preglednik upisati *http://www1.centralsoft.org/index.html* i *http://www2.centralsoft.org/index.html* čime će se prikazati sadržaj datoteke *index.html* koju ste upravo izradili.

# Datoteke dnevnika

Apache vodi dva tipa ASCII datoteka dnevnika: *dnevnik pristupa* (zahtjevi upućeni poslužitelju) i *dnevnik pogrešaka* (pogreške koje su se dogodile prilikom izvršavanja zahtjeva). Moguće je kontrolirati koliko će se informacija zapisivati u ove datoteke, ovisno o tome koliko želite znati o posjetiteljima stranica, koliko diskovnog prostora imate na raspolaganju (datoteke dnevnika se s vremenom povećavaju) i koji ćete alat koristiti za analizu podataka iz dnevnika.

Tipičan zapis iz dnevnika pristupanja izgleda ovako (prelomljen je u nekoliko redova kako bi stao na stranicu):

```
192.168.0.1 - - [22/Sep/2006:15:04:05 -0400] "GET / HTTP/1.1"
200 580 "-" "Mozilla/5.0 (Windows; U; Windows NT 5.0; en-US;
rv:1.8.0.7) Gecko/20060909 Firefox/1.5.0.7"
```

Tipična poruka o grešci izgleda ovako:

```
[Fri Sep 29 10:13:11 2006] [error]
[clientwww.centralsoft.org]
File does not exist: /var/www/index.html
```

Standardne datoteke dnevnika su */var/log/apache2/access.log* i */var/log/apache2/error.log*.

## Dijeljenje i rotacija dnevnika

Standardna instalacija Apachea uključuje *cron* zadatak koji na dnevnoj bazi rotira dnevnike pristupa i pogrešaka. Rotacija se izvodi prema sljedećoj proceduri:

1. Imena datoteka *access.log* i *error.log* promijene se u *access.log.1* i *error.log.1*
2. Brojčana oznaka starih datoteka se poveća (npr. *access.log.1* postaje *access.log.2*)
3. Brišu se datoteke *access.log.7* i *error.log.7*
4. Izrađuju se nove datoteke *access.log* i *error.log*

Svi virtualni poslužitelji podrazumijevano dijele isti dnevnik. Ako imate više virtualnih poslužitelja vjerojatno ćete htjeti podijeliti dnevnike tako da se za svaki poslužitelj može izvoditi posebna analiza.

Apache ima dva standardna formata pristupnog dnevnika: *obični* i *kombinirani*. Njihove ćete definicije pronaći u glavnoj Apacheovoj konfiguracijskoj datoteci */etc/apache2/apache.conf*:

```
# The following directives define some format nicknames for use with
# a CustomLog directive (see below).
LogFormat "%h %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User-Agent}i\"" combined
LogFormat "%h %l %u %t \"%r\" %>s %b" common
LogFormat "%{Referer}i -> %U" referer
LogFormat "%{User-agent}i" agent
```

Svi znakovi % stoje za Apacheove konfiguracijske varijable, primjerice %h znači *hostname*. Kombinirani format nije ništa drugo do obični format kojem je dodan *referer* i *korisnički agent* (preglednik). Nažalost niti jedan format ne sadrži ime virtualnog poslužitelja (varijabla %v) koje trebate da biste dnevnike mogli podijeliti prema računalima. Da biste to mogli učiniti morate definirati novi tip datoteke dnevnika.

Umjesto da mijenjate glavnu Apacheovu konfiguracijsku datoteku, promjene unosite u konfiguracijskoj datoteci Web lokacije koju smo do sada koristili (*/etc/apache2/sites-enabled/000-default*). Dodajte ove redove iznad svake direktive `VirtualHost`:

```
# Define a new virtual host common log format:
LogFormat "%v %h %l %u %t \"%r\" %s %b" vcommon
```

## Podjela dnevnika pomoću programa vlogger

Možda se pitate je li bolje podijeliti unose u dnevnik u odvojene datoteke kako se Apacheu pristupa ili je bolje podijeliti datoteku pristupnog dnevnika jednom na dan pomoću alata kao što je Apacheov *split-logfile*. Smatramo da je prva opcija bolja jer se zapisi odmah usmjeravaju u odgovarajuće dnevnike i nije potrebno programirati *cron* zadatak koji će to činiti po nekom vremenskom rasporedu. Dobar program koji to radi je *vlogger*. Apache dozvoljava da čitate dnevnik putem nekog vanjskog programa a to je upravo ono što želimo. Dodajte ovaj unos odmah ispod reda `LogFormat` koji ste upravo unijeli:

```
# Split log on the fly into virtual host directories
# under /var/log/apache2:
CustomLog "| /usr/sbin/vlogger -s access.log /var/log/apache2" vcommon
```

Budući da *vlogger* nije dio standardnog Debianovog paketa morate ga najprije instalirati:

```
# apt-get install vlogger
```

Nakon toga ponovno pokrenite Apache:

```
# /etc/init.d/apache2 restart
```

*vlogger* će izraditi direktorij na lokaciji */var/log/apache2* za svako virtualno računalo koje ste definirali. Napravit će dnevne pristupne dnevnike s vremenskim oznakama svakog unosa i simboličkom vezom od datoteke *access.log* do najnovije datoteke dnevnika:

```
# cd /var/log/apache2/www1.example.com
# ls -l
total 4
-rw-r--r--  1 root root 984 Aug  3 23:19 08032006-access.log
lrwxrwxrwx  1 root root  19 Aug  3 23:19 access.log -> 08032006-access.log
```

## Analiziranje dnevnika programom Webalizer

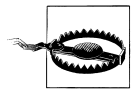
Na raspolaganju nam je veliki broj alata otvorenog izvornog koda i komercijalnih Apacheovih alata za analizu dnevnika. Smatramo da je Webalizer dobar izbor budući se lako instalira, dobro radi i daje korisne informacije.

Isprobajmo ga:

```
# apt-get install webalizer
...
Which directory should webalizer put the output in?
/var/www/webalizer
Enter the title of the reports webalizer will generate.
Usage Statistics forserver1.centralsoft.org
What is the filename of the rotated webserver log?
/var/log/apache2/access.log.1
```

Pristupite mu preko URL-a *http://server1.centralsoft.org/webalizer*.

Sljedećeg dana (nakon što se dnevni *cron* zadatak */etc/cron.daily/webalizer* prvi put pokrenuo) trebali biste vidjeti stranicu s tablicama koje opisuju pristupe vašem Web poslužitelju. Ne morate uređivati konfiguracijsku datoteku (*/etc/webalizer.conf*) osim ako ne želite promijeniti postavke koje ste zadali tijekom instalacije.



Pošiljatelji neželjene elektroničke pošte znaju veliki broj načina za manipuliranje dnevnicima, tako da je dobra praksa ograničiti pravo pristupa Webalizerovim izvješćima.

## SSL/TLS šifriranje

Willie Sutton jednom je rekao da je orobio banku jer "je to mjesto gdje se čuva novac". Napadi putem Interneta sve su više usmjereni na aplikacijsku razinu iz istog razloga. Postalo je nužno šifrirati povjerljive podatke kao što su brojevi kreditnih kartica i lozinke.

Kada od poslužitelja zatražite Web stranicu s prefiksom *http://* svi podaci između poslužitelja i preglednika putuju bez šifriranja i svatko tko ima pristup posredničkim mrežama ima uvid u sadržaj. Pristupanje stranicama na ovaj način (kao i slanje elektroničke pošte) možete usporediti sa slanjem obične poštanske dopisnice.

Standard Secure Socket Layer (SSL) razvijen je radi šifriranja Web prometa i upravo je on bio ključan za golem razvoj komercijalnih Web stranica i elektroničkog poslovanja. Apache može šifrirati Web prometa pomoću SSL-a što je uz neke modifikacije poznato kao Transport Layer Security (TLS). Ovo šifriranje se koristi kada pristupate stranicama s prefiksom *https://*. Šifrirani Web promet može se figurativno shvatiti kao zapečaćena koverta poslana običnom poštom.

Sada ćemo konfigurirati SSL pod Apacheom. Uredite datoteku */etc/apache2/ports.conf* i dodajte joj red:

```
Listen 443
```

Uključite Apacheov SSL modul i kažite Apacheu da ga počne koristiti:

```
# a2enmod ssl
Module ssl installed; run /etc/init.d/apache2 force-reload to enable.
# /etc/init.d/apache2 force-reload
```

Sada možete pokušati pristupiti vašoj početnoj stranici zadajući *https://* prefiks (primjerice *https://server1.centralsoft.org*).

Da bi SSL radio, vaš poslužitelj treba *certifikat*. To nije ništa drugo nego šifrirana datoteka koja korisnikovu pregledniku potvrđuje da poslužitelj zaista jeste onaj za koji se predstavlja. Kako preglednik zna kome vjerovati? Preglednici imaju ugrađen popis izdavatelja certifikata (engl. *certificate authorities, CA*) kojima vjeruju. Možete ga pogledati ako odaberete sljedeće opcije/kartice:

*Firefox 2.0*
: Tools → Advanced → Encryption → View Certificates → Authorities

*Internet Explorer 6.0*
: Tools → InternetOptions → Content → Certificates → Trusted Root Certification Authorities

Izdavatelji certifikata su poduzeća koja prodaju certifikate i žele novac za verificiranje vašeg identiteta. Komercijalne Web stranice gotovo uvijek koriste komercijalne izdavatelje certifikata budući da preglednici prešutno prihvaćaju takve certifikate.

Alternativno, možete biti sam svoj izdavatelj certifikata i izraditi *samostalno potpisani certifikat*. Što se tiče SSL-a ovo funkcionira kao i komercijalni certifikat, ali će preglednik upitati korisnika da li želi prihvatiti vaš certifikat ili ne. U malim projektima s otvorenim izvornim kodom i u testiranju većih projekata ovakvi certifikati su uobičajena praksa.

## suEXEC podrška

Apache može usluživati više lokacija u isto vrijeme iako svaka lokacija ima različite stranice, CGI skripte, korisnike, itd. Pošto se Apache izvodi pod određenim korisničkim i grupnim identitetom, taj korisnik može čitati i mijenjati sadržaj svih lokacija. Međutim, obično se želite osigurati da samo član određene lokacije može izvršavati definirane programe i pristupati pripadnim podacima. Kao i obično, ima nekoliko načina da se to postigne koristeći razne kombinacije Apachea, PHP-a i drugih alata.

Popularna metoda je korištenje programa *suEXEC*. To je program koji se izvršava sa *root* dozvolama pristupa i omogućava da se CGI programi izvršavaju sa korisničkim i grupnim identitetima zadanih korisnika, a ne pod identitetom pod kojima se izvršava sam Apache poslužitelj. Primjerice koristeći naš drugi virtualni poslužitelj nemaštovita imena *www2.example.com*, korisnički račun *www-user2* i grupu *www-group2*, možemo promijeniti dopuštenja za taj virtualni poslužitelj zadavanjem:

```
<VirtualHost www2.example.com>
    SuExecUserGroup www-user2 www-group2
</VirtualHost>
```

## Testiranje performansi

Naš je primarni cilj instalirati i konfigurirati Web poslužitelj tako da radi brzo i sigurno. Osim toga, želimo biti sigurni da poslužitelj može podnijeti očekivano opterećenje za određenu Web lokaciju. Web lokacija ima mnogo pokretnih dijelova i veoma lako se može dogoditi da se neki dio zaglavi ili izgubi. Da bismo vidjeli kako naš sustav radi koristimo alate za testiranje performansi da bismo simulirali rad stotina korisnika koji izuzetno brzo tipkaju (što je mnogo jeftinije nego ih stvarno zaposliti).

Apache se može izvršavati u nekoliko različitih verzija koje se zovu *modeli*. Standardna instalacija pod Debianom je *prefork* model koji pokreće nekoliko Apacheovih procesa kako bi se obradili zahtjevi. Čini se da je to model koji pod Linuxom radi najbolje.

Za provjeru performansi potrebna je barem jedna statička HTML datoteka. Izradite datoteku */var/www/bench.html*. Trebala bi ugrubo biti iste veličine kao i tipična Web stranica vaše lokacije. Možete impresionirati svoje prijatelje koristeći se specijaliziranim stranicama *http://www.lipsum.com* sa kojih tekst možete kopirati u datoteku *bench.html*. Program za provjeru performansi, *ab* nalazi se u paketu *appache2-utils* i trebao bi biti instaliran zajedno s Apacheom. Pošaljimo 1000 odvojenih zahtjeva za istu datoteku s konkurentnošću (brojem istovremenih zahtjeva) 5:

```
# ab -n 1000 -c 5 http://server1.centralsoft.org/bench.html
This is ApacheBench, Version 2.0.41-dev <$Revision$> apache-2.0
Copyright (c) 1996 Adam Twiss, Zeus Technology Ltd, http://www.zeustech.net/
Copyright (c) 1998-2002 The Apache Software Foundation, http://www.apache.org/
Benchmarking server1.centralsoft.org (be patient)
Completed 100 requests
Completed 200 requests
Completed 300 requests
Completed 400 requests
Completed 500 requests
Completed 600 requests
Completed 700 requests
Completed 800 requests
Completed 900 requests
Finished 1000 requests


Server Software:        Apache/2.0.54
Server Hostname:        server1.centralsoft.org
Server Port:            80

Document Path:          /bench.html
Document Length:        1090 bytes

Concurrency Level:      5
Time taken for tests:   2.799386 seconds
```

```
Complete requests:      1000
Failed requests:        0
Write errors:           0
Non-2xx responses:      1000
Total transferred:      1425000 bytes
HTML transferred:       1090000 bytes
Requests per second:    357.22 [#/sec] (mean)
Time per request:       13.997 [ms] (mean)
Time per request:       2.799 [ms] (mean, across all concurrent requests)
Transfer rate:          496.89 [Kbytes/sec] received

Connection Times (ms)
              min  mean[+/-sd] median   max
Connect:        0    0   0.1      0       3
Processing:     6   11   2.2     11      22
Waiting:        5   10   2.3     11      18
Total:          6   11   2.2     11      22

Percentage of the requests served within a certain time (ms)
  50%     11
  66%     12
  75%     13
  80%     13
  90%     14
  95%     14
  98%     15
  99%     16
 100%     22 (longest request)
```

Korisnici obično žele vidjeti broj zahtjeva po sekundi ili recipročnu vrijednost, vrijeme potrebno za jedan zahtjev. Ovi brojevi reći će vam što se najbolje može postići s vašim hardverom i Apacheovom konfiguracijom.

## Instaliranje i administracija Drupala

Kad smo pokrenuli Apache, PHP i MySQL instalirajmo paket koji koristi sve te servise. Kako nam reklamiranje nekog komercijalnog proizvoda nije dozvoljeno, odlučili smo se za program otvorenog izvornog koda, dovoljno velik i koristan da može reprezentirati stvarnu primjenu. Prema njegovim Web stranicama: (*http://www.drupal.org*):

> Drupal je softver koji pojedincu ili grupi korisnika omogućava jednostavno objavljivanje i uređivanje širokog spektra sadržaja na Web stranici te upravljanje njime.

To uključuje Web dnevnike (blogove), forume, upravljanje dokumentima, galerije fotografija, biltene novosti i druge oblike kolaboracije na Webu.

Sljedeća dva dijela opisuju dvije metode instaliranja Drupala:

*apt-get*
: Ova je metoda jednostavnija pa ju prvu isprobajte. Međutim, imali smo nekoliko problema sa Debianovim Drupal paketima.

*Izvorni kôd*
: Više posla, ali možete pratiti razvoj događaja. Ovu metodu koristite ako *apt-get* metoda ne uspije.

## Instalacija Drupala sa programom apt-get

Najjednostavniji način instalacije Drupala je korištenjem programa *apt-get*. Možete otići na Drupalove Web stranice i potražiti paket za preuzimanje ili putem programa *apt-cache* provjeriti da li se nalazi u Debianovu repozitoriju:

```
# apt-cache search drupal
drupal - fully-featured content management/discussion engine
drupal-theme-marvinclassic - "Marvin Classic" theme for Drupal
drupal-theme-unconed - "UnConeD" theme for Drupal
```

Prva stavka je ono što tražimo te možemo započeti s instalacijom:

```
# apt-get install drupal
```

Instalacijska procedura reći će vam da treba neke pakete koje nemate, pribaviti će ih te vas pitati još nekoliko sitnica dok ih instalira. Zatim će tražiti da konfigurirate Drupal kroz niz tekstualnih izbornika. Koristite se tipkom Tab kako biste se kretali između ponuđenih opcija, razmaknicu (tipku Space) za mijenjanje odabrane opcije a tipku Enter za prijelaz na iduću stranicu. Prikazat ćemo samo zadnje redove svakog zaslona i preporučene opcije:

```
Automatically create Drupal database?
Yes

Run database update script?
Yes

Database engine to be used with Drupal
MySQL

Database server for Drupal's database
localhost

Database server administrator user name on host localhost
root

Password for database server administrator root on localhost
newmysqlpassword

Drupal database name
drupal

Remove Drupal database when the package is renoved?
No

Remove former database backups when the package is removed?
Yes
```

```
Web server(s) that should be configured automatically
[ ] apache
[ ] apache-ssl
[ ] apache-perl
[*] apache2
```

Instalacija će kopirati programske datoteke, izraditi MySQL bazu podataka te izraditi Apacheovu konfiguracijsku datoteku (*/etc/apache2/conf.d/drupal.conf*):

```
Alias /drupal /usr/share/drupal
<Directory /usr/share/drupal/>
    Options +FollowSymLinks
    AllowOverride All
    order allow,deny
    allow from all
</Directory>
```

Ako vam se prikaže čudna poruka poput:

```
An override for "/var/lib/drupal/files" already exists, but –force
specified so lets ignore it.
```

možete razbijati glavu neko vrijeme, kao što smo mi činili, ili instalirati Drupal pomoću izvornog koda. Ako sve izgleda dobro, preskočite sljedeći dio.

## Instaliranje Drupala pomoću izvornog koda

Preuzmite najnoviju distribuciju izvornog koda Drupala i premjestite njen direktorij u korijenski direktorij Web lokacije:

```
# wget http://ftp.osuosl.org/pub/drupal/files/projects/drupal-4.7.3.tar.gz
# tar xvzf drupal-4.7.3.tar.gz
# mv drupal-4.7.3 /var/www/drupal
# cd /var/www/drupal
```

Prikazati ćemo instalacijske postupke iz datoteka *INSTALL.txt* i *INSTALL.mysql.txt*. Izradite Drupalovu bazu podataka (nazvati ćemo je *drupal*), administrativne korisnike (korisničko ime neka opet bude *drupal*, u skladu s našom maštovitošću) te administrativnu lozinku (molimo da upotrijebite nešto drugo a ne *drupalpw*):

```
# mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 37 to server version: 4.0.24_Debian-10sarge2-log

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

mysql> create database drupal;
Query OK, 1 row affected (0.00 sec)

mysql> GRANT SELECT, INSERT, UPDATE, DELETE, CREATE, DROP,
    -> INDEX, ALTER, CREATE TEMPORARY TABLES, LOCK TABLES
    -> on drupal.* to
    -> "drupal"@"localhost" identified by "drupalpw";
```

```
Query OK, 0 rows affected (0.01 sec)

mysql> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.00 sec)

mysql>quit;
Bye
```

Zatim učitajte Drupalove definicije baze podataka u MySQL:

```
# mysql -u root -p drupal < database/database.4.0.mysql
Enter password:
#
```

Nakon toga uredite datoteku */sites/default/config.php* i promijenite red:

```
$db_url = 'mysql://username:password@localhost/databasename';
```

u:

```
$db_url = 'mysql://drupal:drupalpw@localhost/drupal';
```

## Konfiguriranje Drupala

U preglednik unesite adresu *http://server1.centralsoft.org/drupal*. Na prvoj se stranici prikazuje (u inačici koju smo testirali):

```
Welcome to your newDrupal website!
Please follow these steps to set up and start using your website:
Create your administrator account
Tobegin,createthefirstaccount.Thisaccountwillhavefulladministrationrights
and will allow you to configure your website.
```

Pritisnite vezu „create the first account“. Na stranici koja se pojavila unesite vaše korisničko ime u polje „Username“ i vašu adresu elektroničke pošte u polje „E-mail address“. Sada pritisnite gumb „Create new account“. Vratit ćete se na prvu stranicu na čijem gornjem dijelu sada stoji:

```
Your password and further instructions have been sent to your e-mail address.
```

Na unesenu adresu elektroničke pošte dobit ćete jednokratnu lozinku pomoću koje ćete se prijaviti na Drupal u dijelu „User login“. Sustav će vas preusmjeriti na stranicu na kojoj ćete zadati trajnu lozinku. Nakon ovih postavki možete se vratiti na početnu stranicu gdje se možete odlučiti za neku od sljedećih opcija:

1. **Create your administrator account**

   Da biste počeli s radom izradite prvi korisnički račun. Ovaj korisnički račun će imati puna administrativna prava koja dozvoljavaju konfiguriranje Web lokacije.

2. **Configure your website**

   Kad se prijavite, posjetite administratorski dio gdje možete prilagoditi i konfigurirati sve aspekte lokacije.

3. **Enable additional functionality**

   Ovdje možete vidjeti popis modula i eventualno uključiti mogućnosti koje će vam trebati.

4. **Customize your website design**

   Da biste promijenili izgled stranica pogledajte dio „themes“. Možete se odlučiti za neku od uključenih tema ili preuzeti dodatne teme u dijelu „download“.

5. **Start posting content**

   Konačno možete početi dodavati sadržaj na stranice. Ova će poruka nestati čim počnete objavljivati sadržaj.

   Za više informacija pogledajte dio „Help“ ili Drupalov priručnik na Web stranicama *http://drupal.org*. Također se možete koristit Drupalovim forumom ili nekom drugom opcijom podrške korisnicima.

Budući ste izradili prvi (administratorski) korisnički račun na vama je da samostalno isprobate sve druge funkcije.

# Rješavanje problema

Ako volite dijagnosticirati probleme zasigurno ćete voljeti Web poslužitelje. Tu ima toliko mnogo stvari koje mogu krenuti krivim putem i to na jako mnogo mjesta i isto toliko mnogo načina. Mogli biste godinama samo rješavati probleme.

Pogledajmo neke klasične probleme Web poslužitelja. (Poruke o pogreškama koje se pojavljuju u pregledniku preuzeli smo iz Firefoxa ali su i poruke Internet Explorera slične).

## Web stranica se ne prikazuje u pregledniku

Pretpostavimo da je vaš početni direktorij */var/www*, početna datoteka *test.html* a poslužitelj *server1.centralsoft.org*. Kada ste koristili vanjski preglednik da bi pristupili stranici *http://server1.centralsoft.org/test.html* dobili ste poruku o grešci u prozoru preglednika.

Poruka o pogrešci oblika „Poslužitelj nije pronađen“ koja se pojavljuje u prozoru preglednika implicira problem s DNS poslužiteljima. Najprije provjerite ima li *server1.centralsoft.org* unos u javnom DNS poslužitelju imena:

```
# dig server1.centralsoft.org
...
;; ANSWER SECTION:
server1.centralsoft.org.        106489  IN      A       192.0.34.166
...
```

Nakon toga provjerite može li se poslužitelju pristupiti preko Interneta. Ako vaš vatrozid dozvoljava naredbu ping, provjerite da li je poslužitelj aktivan koristeći naredbu:

```
# ping server1.centralsoft.org
PING server1.centralsoft.org (192.0.34.166) 56(84) bytes of data.
64 bytes from server1.centralsoft.org (192.0.34.166): icmp_seq=1 ttl=49
time=81.6 ms
```

Provjerite je li ulaz 80 otvoren i da možda nije blokiran. Na vanjskom računalu zadajte naredbu *nmap*:

```
# nmap -P0 -p 80 server1.centralsoft.org

Starting nmap 3.81 ( http://www.insecure.org/nmap/ ) at 2006-07-25 23:50 CDT
Interesting ports on server1.centralsoft.org (192.0.34.166):
PORT   STATE SERVICE
80/tcp open  http

Nmap finished: 1 IP address (1 host up) scanned in 0.186 seconds
```

Ako nemate *nmap* pravite se kao da ste preglednik. Koristeći se *telnetom* spojite se na standardni Web ulaz (80) i pošaljite najjednostavniji mogući HTTP zahtjev:

```
# telnet server1.centralsoft.org 80
Trying 192.0.34.166...
Connected to server1.centralsoft.org.
Escape character is '^]'.
HEAD / HTTP/1.0

HTTP/1.1 200 OK
Date: Wed, 26 Jul 2006 04:52:13 GMT
Server: Apache/2.0.54 (Fedora)
Last-Modified: Tue, 15 Nov 2005 13:24:10 GMT
ETag: "63ffd-1b6-80bfd280"
Accept-Ranges: bytes
Content-Length: 438
Connection: close
Content-Type: text/html; charset=UTF-8

Connection closed by foreign host.
```

Ako ovo ne radi provjerite da li u datoteci */etc/apache2/ports.conf* postoji red:

```
Listen 80
```

i pogledajte da li nešto drugo blokira ulaz 80:

```
# lsof -i :80
COMMAND   PID     USER   FD   TYPE DEVICE SIZE NODE NAME
apache2 10678 www-data    3u  IPv6 300791       TCP *:www (LISTEN)
apache2 10679 www-data    3u  IPv6 300791       TCP *:www (LISTEN)
apache2 10680 www-data    3u  IPv6 300791       TCP *:www (LISTEN)
apache2 20188     root    3u  IPv6 300791       TCP *:www (LISTEN)
apache2 20190 www-data    3u  IPv6 300791       TCP *:www (LISTEN)
apache2 20191 www-data    3u  IPv6 300791       TCP *:www (LISTEN)
apache2 20192 www-data    3u  IPv6 300791       TCP *:www (LISTEN)
apache2 20194 www-data    3u  IPv6 300791       TCP *:www (LISTEN)
apache2 20197 www-data    3u  IPv6 300791       TCP *:www (LISTEN)
apache2 20198 www-data    3u  IPv6 300791       TCP *:www (LISTEN)
apache2 20199 www-data    3u  IPv6 300791       TCP *:www (LISTEN)
```

Ako u dobivenom ispisu ne vidite *apache2* provjerite da li se Apache uopće izvršava:

```
# ps -efl | grep apache2
```

Ako se u ispisu pojavi red poput ovog:

```
5 S root      7692     1  0  76   0 -  2991 415244 Jul16 ?        00:00:00
/usr/sbin/apache2 -k start -DSSL
```

Apche se izvršava. Ako ne, pokrenite ga:

```
# /etc/init.d/apache2 start
```

Tada ponovno zadajte naredbu *ps*. Ako se Apache još uvijek ne pojavljuje, pogledajte u dnevnik pogrešaka:

```
# tail -f /var/log/apache2/error.log
```

Ako nemate dopuštenje za čitanje ove datoteke očito imate loš dan. Ako je datoteka prazna, također možete imati nedovoljna dopuštenja za pristup. Provjerite postoji li direktorij */var/log/apache2* i datoteka */var/log/apache2/error.logfile*:

```
# ls -l /var/log/apache2
total 84
-rw-r-----  1 root adm 31923 Jul 25 23:09 access.log
-rw-r-----  1 root adm 32974 Jul 22 20:50 access.log.1
-rw-r-----  1 root adm   379 Jul 23 06:25 access.log.2.gz
-rw-r-----  1 root adm  1969 Jul 25 23:09 error.log
-rw-r-----  1 root adm  1492 Jul 23 06:25 error.log.1
-rw-r-----  1 root adm   306 Jul 23 06:25 error.log.2.gz
```

Ako su u dnevniku pokazane stare informacije možda vam nedostaje prostora na disku. Iznenađujuće je kako često zaboravljamo provjeriti količinu slobodnog prostora na disku prije nego krenemo u istraživanje egzotičnijih uzroka, poput vatrozida. Unesite:

```
# df
Filesystem           1K-blocks      Used Available Use% Mounted on
/dev/hda1            193406200    455292 183126360   1% /
tmpfs                   453368         0    453368   0% /dev/shm
```

Ako ste koristili drugačije `User` ili `Group` direktive u konfiguraciji Apachea provjerite da li korisnik i grupa uopće postoje:

```
# id www-data
uid=33(www-data) gid=33(www-data) groups=33(www-data)
```

Ako preglednik prikaže Apacheovu poruku o pogrešci morat ćete još poprilično tražiti. Ako se pak u pregledniku prikaže poruka:

```
Not Found
The requested URL /wrong.html was not found on this server.
```

vjerojatno ste krivo upisali URL adresu. Ako vidite poruku:

```
Forbidden
You don't have permission to access /permissions.html on this server.
```

datoteka je prisutna, ali je Apacheov korisnik ne može čitati:

```
# cd /var/www
# ls -l permissions.html
-rw-------  1 root root 0 Jul 26 00:01 permissions.html
```

Problemi s dopuštenjima za pristup mogu se riješiti promjenom vlasnika datoteke tako da vlasnik postane proces koji pokreće Apache.

## Virtualni poslužitelj ne radi

Zadajte naredbu:

```
# apache2ctl -S
```

za brzu provjeru direktiva virtualnih poslužitelja.

## SSI ne radi

Ako u dnevniku pogrešaka (*/var/log/apache2/error.log*) uočite red:

```
[error] an unknown filter was not added: INCLUDES
```

niste uključili *mod_include*. Zadajte naredbu:

```
# a2enmod include
```

## CGI program ne radi

Ako ne možete pokrenuti CGI program, provjerite sljedeće:

- Je li CGI uključen pomoću metoda koje smo ranije objasnili?
- Je li CGI program smješten u direktorij kao što je */var/cgi-bin* i ima li nastavak imena datoteke kao što je *.php*?
- Je li datoteka čitljiva? Ako nije, upotrijebite naredbu *chmod*.
- Što kaže Apacheov dnevnik grešaka?
- Provjerite dnevnik pogrešaka sustava */var/log/messages*?

## SSL ne radi

Provjerite jeste li uključili Apacheov SSL modul (*a2enmod ssl*) i rekli Apacheu da prati ulaz 443 u datoteci */etc/apache2/ports.conf*:

```
Listen 443
```

Ako direktive nema, dodajte je i ponovno pokrenite Apache. Nakon toga pokušajte iz preglednika pristupiti URL-u *https://server1.centralsoft.org*. Ako SSL još uvijek ne radi vjerojatno je ulaz 443 blokiran vatrozidom, što možete provjeriti naredbom *nmap*:

```
# nmap -P0 -p 443 server1.centralsoft.org

Starting nmap 3.70 ( http://www.insecure.org/nmap/ ) at 2006-08-01 22:38 CDT
Interesting ports on ... (...):
PORT    STATE SERVICE
443/tcp open  https

Nmap run completed -- 1 IP address (1 host up) scanned in 0.254 seconds
```

## Dodatna literatura

Svoje znanje o webu možete proširiti čitajući knjige kao što su *Apache Cookbook* autora Kena Coara i Richa Bowena (u izdanju O'Reilly Media), *Pro Apache* autora Petera Wainwrighta (u izdanju Apress) te *Run Your Own Web Server Using Linux & Apache* čiji su autori Stuart Langridge i Tony Steidler-Dennison (u izdanju SitePoint).

POGLAVLJE 7

# Grozdovi s raspoređivanjem opterećenja

Prije više od 10 godina otkrivena je mogućnost spajanja nekoliko jeftinijih računala radi izvođenja proračuna za koje bi u pravilu trebalo imati središnje računalo ili čak superračunalo. Jedan od prvih primjera grozda, a koji se koristi još i danas, NASA-in je Beowulf (*http://www.beowulf.org*). U Wikipedijinu zapisu (*http://en.wikipedia.org/wiki/Computer_cluster*) sažete su glavne karakteristike računalnih grozdova:

> *Grozd računala* grupa je labavo povezanih računala koja usko surađuju te se u većini aspekata mogu promatrati kao jedno samostalno računalo. Računala koja čine grozdove su najčešće, ali ne uvijek, spojena putem brzih lokalnih mreža. Grozdovi se obično razvijaju da bi se povećala brzina i/ili pouzdanost koju pruža samo jedno računalo, a obično su daleko isplativiji nego samo jedno računalo koje može postići zahtijevanu brzinu ili pouzdanost.

Grozdovi su dobro rješenje ako želite uz razumnu cijenu povećati brzinu, pouzdanost i skalabilnost. Amazon, Yahoo! i Google razvili su svoj poslovni uspjeh na tisućama poslužitelja u redudantnoj konfiguraciji grozdova. Jeftinije je i lakše širiti se *horizontalno* (dodavanjem novih poslužitelja) nego *vertikalno* (nabavkom skupljih računala). Danas postoji veliki broj Linuxovih rješenja za grozdove kako u komercijalnim verzijama tako i u verzijama otvorenog izvornog koda. U ovom ćemo poglavlju objasniti grozdove koristeći besplatni Linux Virtual Server (*http://www.linuxvirtualserver.org*). Pokazati ćemo kako kombinirati kutiju žitarica, gumenu traku i tri računala da bi dobili Apacheov Web poslužitelj izveden kao grozd s raspoređivanjem opterećenja. Također ćemo razmotriti visoku razinu dostupnosti i na kraju neka alternativna rješenja. Nećemo se baviti grozdovima visokih performansi, mrežnim računanjem, paralelizacijom ili distribuiranim računanjem; u područjima koja koriste ove tehnologije razvijena se specijalizirana hardverska i softverska rješenja (primjerice u meteorološkom modeliranju i grafici).

## Raspoređivanje opterećenja i visoka razina dostupnosti

*Raspoređivanje opterećenja* (engl. *load balancing, LB*) omogućuje *skalabilnost* – raspodjelu zahtjeva na više poslužitelja. Raspoređivanje opterećenja se sastoji od prosljeđivanja paketa i nekih informacija o servisu koji se raspoređuje (u ovom poglavlju je to HTTP). Oslanja se na vanjsko praćenje opterećenja na temelju kojeg odlučuje kamo slati pakete.

*Visoka razina dostupnosti* (engl. *high availability, HA*) omogućuje *pouzdanost*: održavanje aktivnosti servisa. Visoka razina dostupnosti počiva na redudantnim poslužiteljima, tzv. *razmjeni „otkucaja srca"* odnosno poruka u kojima poslužitelji potvrđuju svoju aktivnost te *proceduri oporavka* tj. proceduri zamjene neispravnih poslužitelja ispravnima.

U ovom se poglavlju uglavnom bavimo raspoređivanjem opterećenja što administratori općenito smatraju važnijim i češće koriste. Kada Web lokacije postanu ključne za funkcioniranje tvrtki neophodna postaje i visoka razina dostupnosti. Pred kraj ovog poglavlja dati ćemo i nekoliko korisnih veza prema informacije o postavljanju kombiniranih sustava s raspoređivanjem opterećenja i visokom razinom dostupnosti.

Konfiguracija sa raspoređivanjem opterećenja koju ćemo koristiti u ovom poglavlju jednostavna je i sastoji se od tri javne i jedne virtualne adrese, kao što je navedeno u tablici 7-1.

*Tablica 7-1. Adrese i uloge poslužitelja u našem grozdu*

| Ime | IP adresa | Opis |
|---|---|---|
| *lb* | 70.253.158.44 | Raspoređivač opterećenja – javna adresa Web servisa |
| *web1* | 70.253.158.41 | Prvi Web poslužitelj – jedna od stvarnih IP adresa |
| *web2* | 70.253.158.45 | Drugi Web poslužitelj – druga stvarna IP adresa |
| *(VIP)* | 70.253.158.42 | Virtualna IP adresa (VIP) koju dijele lb, web1 i web2 kao dodatak svojim stvarnim IP adresama |

Virtualna IP adresa je adresa koju raspoređivač opterećenja prikazuje vanjskim klijentima i preko koje stižu zahtjevi Web poslužiteljima.

## Softver za raspoređivanje opterećenja

Najjednostavniji oblik raspoređivanja opterećenja je *round-robin DNS*, gdje se više `A` zapisa definira za isto ime. Zbog toga se poslužitelji smjenjuju prilikom odgovaranja na pristigle zahtjeve. Metoda prestaje djelovati čim se jedan poslužitelj pokvari a i ne vodi računa o posebnostima servisa. Kod HTTP-a ćemo ponekad morati čuvati podatke sesije kao što su provjera autentičnosti ili kolačići i klijentu osigurati spajanje na isti poslužitelj. Da bi udovoljili ovim zahtjevima biti ćemo malo sofisticiraniji i upotrijebiti dva alata:

- IP Virtual Server (IPVS), modul za raspoređivanje opterećenja na transportnoj razini (TCP) koji je danas standardna Linuxova komponenta.
- *ldirectord*, pomoćni program koji prati ispravnost poslužitelja s raspoređivanjem opterećenja

Instalacijske upute odnose se na Linux distribuciju Debian 3.1 (Sarge).

## IPVS na raspoređivaču opterećenja

Budući se IPVS nalazi u Linuxovoj jezgri nije potrebno instalirati nikakav softver, međutim potrebno ga je konfigurirati.

Na poslužitelju *lb*, dodajte sljedeće redove u */etc/modules*.

```
ip_vs_dh
ip_vs_ftp
ip_vs
ip_vs_lblc
ip_vs_lblcr
ip_vs_lc
ip_vs_nq
ip_vs_rr
ip_vs_sed
ip_vs_sh
ip_vs_wlc
ip_vs_wrr
```

Nakon toga učitajte module u jezgru:

```
# modprobe ip_vs_dh
# modprobe ip_vs_ftp
# modprobe ip_vs
# modprobe ip_vs_lblc
# modprobe ip_vs_lblcr
# modprobe ip_vs_lc
# modprobe ip_vs_nq
# modprobe ip_vs_rr
# modprobe ip_vs_sed
# modprobe ip_vs_sh
# modprobe ip_vs_wlc
# modprobe ip_vs_wrr
```

Da biste omogućili prosljeđivanje paketa u Linuxovoj jezgri na raspoređivaču opterećenja uredite datoteku */etc/sysctl.conf* tako što ćete joj dodati red:

```
net.ipv4.ip_forward = 1
```

Poslije toga učitajte ovu postavku u jezgru:

```
# sysctl -p
net.ipv4.ip_forward = 1
```

## ldirectord

Premda se može nabaviti samostalno, *ldirectord* ćemo preuzeti kao dio paketa *Ultra Monkey* koji uključuje softver za slanje „otkucaja srca“ za održavanje visoke dostupnosti. Budući da Ultra Monkey nije dio standardne Debianove distribucije trebate u Debianovu datoteku repozitorija (*/etc/apt/sources.list*) na računalu *sa raspoređivačem opterećenja* dodati sljedeća dva reda:

```
deb http://www.ultramonkey.org/download/3/ sarge main
deb-src http://www.ultramonkey.org/download/3 sarge main
```

Nakon toga ažurirajte repozitorij i preuzmite paket:

```
# apt-get update
# apt-get install ultramonkey
```

Instalacijski proces zatražit će odgovore na nekoliko pitanja:

```
Do you want to automatically load IPVS rules on boot?
No
Select a daemon method.
none
```

Naša konfiguracija imat će jedan virtualni poslužitelj (adresa koju vide klijenti, *ldirectord*) koji ćemo nazvati *upravitelj* i dva *stvarna poslužitelja* (Apache). Stvarni poslužitelji mogu s upraviteljem biti spojeni na jedan od tri načina:

*LVS-NAT*
: Stvarni poslužitelji su u NAT podmreži iza upravitelja i svoje odgovore usmjeravaju natrag preko preko upravitelja.

*LVS-DR*
: Stvarni poslužitelji usmjeravaju svoje odgovore izravno prema klijentima. Sva su računala u istoj podmreži i mogu pronaći adresu 2. razine (Ethernet) svakog računala. Ne moraju biti dostupna računalima izvan svoje podmreže.

*LVS-TUN*
: Stvarni poslužitelji i upravitelj ne moraju biti u istoj mreži. Komuniciraju stvaranjem tunela s IP-over-IP (IPIP) učahurivanjem.

Koristit ćemo drugi način (DR) jer je jednostavan, brz i lako se skalira. Pomoću ove metode određujemo VIP (virtualnu IP adresu) koju dijele raspoređivač opterećenja i stvarni poslužitelji. Međutim tu se pojavljuje jedan problem: ako sva računala dijele istu VIP adresu, kako ćemo iz nje razlučiti fizičku MAC adresu? Ovo se zove *ARP problem* budući da sustavi na istoj LAN mreži za međusobno pronalaženje koriste Address Resolution Protocol (ARP), a ARP očekuje da svaki sustav ima jedinstvenu IP adresu.

Većina rješenja zahtjeva zakrpe ili dodatne module za jezgru koji se mijenjaju s promjenama Linuxove jezgre. Od verzije 2.6 na dalje popularno je rješenje dozvoliti raspoređivaču opterećenja upravljanje ARP-om za VIP, a na stvarnim poslužiteljima konfigurirati VIP na aliase povratnog uređaja (engl. *loopback device*). Ovo rješenje počiva na činjenici da povratni uređaji uređaji ne odgovaraju na ARP zahtjeve.

Taj ćemo pristup koristiti, ali prije toga moramo konfigurirati Web poslužitelje.

## Konfiguriranje stvarnih poslužitelja (Apache čvorova)

Na svakom stvarnom poslužitelju (*web1* i *web2*) učinite sljedeće:

1. Ako na poslužitelju već nije instaliran Apache, instalirajte ga

   ```
   # apt-get install apache2
   ```

   Ako niste kopirali datoteke vaše Web lokacije, to možete učiniti sada ili nakon postavljanja raspoređivanja opterećenja.

2. Instalirajte *iproute* (Linuxov mrežni paket sa mnogo više mogućnosti od starih alata kao što su *ifconfig* i *route*):

   ```
   # apt-get install iproute
   ```

3. U datoteku */etc/sysctl.conf* dodajte redove:

```
net.ipv4.conf.all.arp_ignore = 1
net.ipv4.conf.eth0.arp_ignore = 1
net.ipv4.conf.all.arp_announce = 2
net.ipv4.conf.eth0.arp_announce = 2
```

4. Unesite promjene u jezgru:

```
# sysctl -p
net.ipv4.conf.all.arp_ignore = 1
net.ipv4.conf.eth0.arp_ignore = 1
net.ipv4.conf.all.arp_announce = 2
net.ipv4.conf.eth0.arp_announce = 2
```

5. Pod pretpostavkom da je vaš stvarni poslužitelj sustav s Debianom, uredite datoteku */etc/network/interfaces* pridružujući VIP adresi (70.253.15.42) alias povratnog uređaja `10:0`:

```
auto lo:0
iface lo:0 inet static
    address 70.253.15.42
    netmask 255.255.255.255
    pre-up sysctl -p > /dev/null
```

6. Omogućite alias povratnog uređaja:

```
# ifup lo:0
```

7. Izradite datoteku */var/www/ldirector.html* sljedećeg sadržaja:

```
I'm alive!
```

8. Na *web1* zadajte:

```
# echo "I'm web1" > /var/www/which.html
```

9. Na *web2* zadajte:

```
# echo "I'm web2" > /var/www/which.html
```

10. Pokrenite Apache, a ako se već izvršava ponovno ga pokrenite:

```
# /etc/init.d/apache2 restart
```

Apacheov dnevnik pristupa još uvijek ne bi trebao pokazivati nikakvu aktivnost budući da *lb* s njime još nije započeo komunikaciju.

## Konfiguriranje raspoređivača opterećenja

Na poslužitelju *lb*, napravite konfiguracijsku datoteku raspoređivača opterećenja, */etc/ha.d/ldirectord.cf*:

```
checktimeout=10
checkinterval=2
autoreload=no
logfile="local0"
quiescent=no
virtual=70.253.158.42:80
        real=70.253.158.41:80 gate
        real=70.253.158.45:80 gate
        service=http
        request="director.html"
        receive="I'm alive!"
```

```
        scheduler=rr
        protocol=tcp
        checktype=negotiate
```

Ako je `quiescent` postavljeno na `yes`, neispravan stvarni poslužitelj dobit će težinu `0`, ali će ostati u LVS-ovoj tablici usmjeravanja. Da smo odabrali postavku `no`, neispravni poslužitelj više ne bi bio u igri. Težina poslužitelja odražava njegov kapacitet u odnosu na druge poslužitelje. Kod jednostavnijih shema raspoređivanja opterećenja kao što je naša, svi ispravni poslužitelji imaju težinu `1`, a neispravni težinu `0`.

Ako je `checktype` postavljeno na `negotiate`, upravitelj će uputiti HTTP zahtjev prema svakom stvarnom poslužitelju za URL `request` i provjeriti da li vraćeni sadržaj sadrži znakovnu vrijednost `receive`. Ako je pak vrijednost `checktype` postavljena na `check`, biti će izvedena samo brza TCP provjera, a `request` i `receive` će biti zanemareni.

Datoteke za pokretanje sustava *ldirectord* trebale su biti izrađene u direktoriju */etc* još za vrijeme instaliranja. Ultra Monkey također instalira Heartbeat kojeg još nećemo koristiti tako da ga za sada možemo isključiti:

```
# update-rc.d heartbeat remove
update-rc.d: /etc/init.d/heartbeat exists during rc.d purge (use -f to force)
```

Raspoređivač opterećenja prati ispravnost Web poslužitelja tako što redovito zahtjeva datoteku koju smo zadali u *ldirectord.cf* (`request="director.html"`).

Budući da će poslužitelj odgovarati na Web zahtjeve na VIP adresi (70.253.158.42) trebamo ga o tome obavijestiti. Uredite */etc/network/interfaces* i dodajte sljedeće redove kako biste izradili alias uređaja `eth0:0`:

```
auto eth0:0
iface eth0:0 inet static
        address 70.253.158.42
        netmask 255.255.255.248
    # These should have the same values as for eth0:
        network ...
        broadcast ...
        gateway ...
```

Sada aktivirajte ovu novu IP adresu:

```
# ifup eth0:0
```

Na kraju, pokrenite *strojeve* na raspoređivaču opterećenja:

```
# /etc/init.d/ldirectord start
Starting ldirectord... success
```

## Testiranje sustava

Provjerimo da li se na računalu *lb* izvodi raspoređivač opterećenja:

```
# ldirectord ldirectord.cf status
```

Trebali biste vidjeti poruku sličnu ovoj:

```
ldirectord for /etc/ha.d/ldirectord.cf is running with pid:
1455
```

Ako umjesto gornje poruke vidite nešto poput:

```
ldirectord is stopped for /etc/ha.d/ldirectord.cf
```

pojavio se problem. Zaustavite upravitelj te ga ponovno pokrenite koristeći se zastavicom za uklanjanje pogrešaka `-d`. Pogledajte pojavljuju li se u ispisu nekakve pogreške:

```
# /usr/sbin/ldirectord /etc/ha.d/ldirectord.cf stop
# /usr/sbin/ldirectord -d /etc/ha.d/ldirectord.cf start
DEBUG2: Running exec(/usr/sbin/ldirectord -d /etc/ha.d/ldirectord.cf start)
Running exec(/usr/sbin/ldirectord -d /etc/ha.d/ldirectord.cf start)
DEBUG2: Starting Linux Director v1.77.2.32 with pid: 12984
Starting Linux Director v1.77.2.32 with pid: 12984
DEBUG2: Running system(/sbin/ipvsadm -A -t 70.253.158.42:80 -s rr )
Running system(/sbin/ipvsadm -A -t 70.253.158.42:80 -s rr )
DEBUG2: Added virtual server: 70.253.158.42:80
Added virtual server: 70.253.158.42:80
DEBUG2: Disabled server=70.253.158.45
DEBUG2: Disabled server=70.253.158.41
DEBUG2: Checking negotiate: real
server=negotiate:http:tcp:70.253.158.41:80:::\/director\.html:I\'m\ alive\!
(virtual=tcp:70.253.158.42:80)
DEBUG2: check_http: url="http://70.253.158.41:80/director.html"
virtualhost="70.253.158.41"
LWP::UserAgent::new: ()
LWP::UserAgent::request: ()
LWP::UserAgent::send_request: GET http://70.253.158.41:80/director.html
LWP::UserAgent::_need_proxy: Not proxied
LWP::Protocol::http::request: ()
LWP::Protocol::collect: read 11 bytes
LWP::UserAgent::request: Simple response: OK
45:80/director.html is up
```

Ispis će biti kraći ako je `checktype` postavljeno na `check`.

Znatiželje radi, pogledajmo što kaže IP virtualni poslužitelj niže razine:

```
# ipvsadm -L -n
IP Virtual Server version 1.2.0 (size=4096)
Prot LocalAddress:Port Scheduler Flags
  -> RemoteAddress:Port           Forward Weight ActiveConn InActConn
TCP  70.253.158.42:80 rr
  -> 70.253.158.45:80             Route   1      1          2
  -> 70.253.158.41:80             Route   1      0          3
```

Ovo nam pokazuje da je naš prvi stvarni poslužitelj aktivan, ali da je drugi neaktivan.

Provjerimo i dnevnik sustava na raspoređivaču opterećenja *lb*:

```
# tail /var/log/syslog
Sep 11 22:59:45 mail ldirectord[8543]: Added virtual server:
70.253.158.44:80
Sep 11 22:59:45 mail ldirectord[8543]: Added fallback server: 127.0.0.1:80
( x 70.253.158.44:80) (Weight set to 1)
Sep 11 22:59:45 mail ldirectord[8543]: Added real server: 70.253.158.41:80
( x 70.253.158.44:80) (Weight set to 1)
Sep 11 22:59:45 mail ldirectord[8543]: Deleted fallback server: 127.0.0.1:80
```

```
( x 70.253.158.44:80)
Sep 11 22:59:46 mail ldirectord[8543]: Added real server: 70.253.158.45:80
( x 70.253.158.44:80) (Weight set to 1)
```

Vratimo se na *web1* i *web2* te provjerimo Apacheove dnevnike pristupa. Upravitelj bi trebao zahtijevati datoteku svakih `checkinterval` sekundi:

```
70.253.158.44 - - [11/Sep/2006:22:49:37 -0500] "GET /director.html HTTP/1.1"
200 11 "-" "libwww-perl/5.803"
70.253.158.44 - - [11/Sep/2006:22:49:39 -0500] "GET /director.html HTTP/1.1"
200 11 "-" "libwww-perl/5.803"
```

U pregledniku posjetite virtualnu lokaciju *http://70.253.158.42/which.html*. Trebali biste ugledati:

```
I'm web1
```

ili

```
I'm web2
```

Ako raspoređivač opterećenja ne radi ili neki od poslužitelja nije uključen uvijek ćete dobiti odgovor od istog Web poslužitelja.

Sada zaustavite Apache na *web1*:

```
# /etc/init.d/apache stop
```

Opet učitajte stranicu u pregledniku kako bi ponovno pristupili datoteci *http://70.253.158.42/which.html*. Svaki puta biste trebali dobiti odgovor:

```
I'm web2
```

## Osiguravanje visoke razine dostupnosti

Raspoređivač opterećenja je mogući uzrok problema u sustavu. Ako raspoređivač počne otkazivati svi poslužitelji iza njega postaju nedostupni. Da bi sustav bio pouzdaniji možete u konfiguraciju instalirati još jedan raspoređivač opterećenja koji bi radio istovremeno s postojećim i koji bi povećao razinu dostupnosti. Detaljne upute kako se to može izvesti s pomoću paketa Ultra Monkey koji ste već instalirali, možete pronaći u članku „How To Set Up A Loadbalanced High-Availability Apache Cluster“ (*http://www.howtoforge.com/high_availability_loadbalanced_apache_cluster*).

Za same Apache poslužitelje posebno bavljenje razinom dostupnosti nije nužno potrebno budući da *ldirectord* tako i tako svakih `checkinterval` sekundi provjerava njihov status i podešava težinu što je vrlo slično *heartbeat* procesu.

## Dodavanje ostalih servisa konfiguraciji sa raspoređivanjem opterećenja

U ovom smo poglavlju kao primjer koristili Apache Web poslužitelje jer je najvjerojatnije da će oni biti dio farme poslužitelja. Ostali servisi koji mogu imati koristi od raspoređivanja opterećenja i osiguravanja visoke razine dostupnosti su MySQL, poslužitelji elektroničke pošte ili LDAP poslužitelji. Za MySQL primjer pogledajte članak „How To Set Up A Load-Balanced Mysql Cluster“ na adresi *http://www.howtoforge.com/loadbalanced_mysql_cluster*.

## Prilagođavanje potrebama bez servisa za raspoređivanje opterećenja i postizanje visoke razine dostupnosti

Iako ste možda ponudili izvrstan servis, pitanje je može li poslužitelj preživjeti Slashdotting tj. period visokog opterećenja. Ako ne, može vam opasti kredibilitet i većina korisnika više nikada neće posjetiti vaše stranice. No kako implementacija servisa za raspoređivanje opterećenja i osiguravanje dostupnosti zahtjeva značajan napor i ulaganje u hardver nije loše razmotriti i alternativna rješenja. Postoji mnogo načina da iz svojeg poslužitelja izvučete više. Primjerice, možete onemogućiti *.htaccess* datoteke u Apacheovoj konfiguraciji (`AllowOverride None`) i koristiti *mod_expires* da bi izbjegli *stat* pozive za datotekama koje se rijetko mijenjaju kao što su slike. Knjige i Web stranice koje se bave Apacheom sadrže jako puno takvih savjeta za optimizaciju.

Kad dosegnete granicu vašeg softvera Web poslužitelja razmislite o alternativama. U mnogo slučajeva, Web poslužitelji kao što su *lighttpd* (*http://www.lighttpd.net*), *Zeus* (*http://www.zeustech.net*) i *litespeed* (*http://litespeedtech.com*) brži su od Apachea i koriste mnogo manje memorije.

Od velike koristi može biti i smještanje u privremenu memoriju. Programi za organiziranje privremene memorije, u koje spadaju PHP ubrzivači kao što je e-accelerator (*http://eaccelerator.net*) i APC (*http://apc.communityconnect.com*), spremaju PHP-ov binarni kod i time se izbjegava parsiranje prilikom svakog pristupa stranici. Privremena memorija za podatke, primjerice rezultate MySQL upita, sprema rezultate identičnih upita. Replikacija je oblik raspoređivanja opterećenja. *memcached* (*http://danga.com/memcached*) je brzi način smještanja rezultata pretraživanja baze podataka u privremenu memoriju. *Squid* (*http://www.squid-cache.org*), kad se koristi kao inverzni posrednik za spremanje u privremenu memoriju, stranični je program za upravljanje privremenom memorijom koji može zaobići čitav Web poslužitelj.

Kad su poslužitelji u odvojenim slojevima (npr. MySQL → PHP → Apache), poboljšanja su višestruka. Primjerice, prezentacija „Getting Rich with PHP 5“ (*http://talks.php.net/show/oscon06*) kombinira puno malih dorada da bi na jednom računalu skalirala PHP aplikaciju sa 17 poziva po sekundi na 1100 poziva po sekundi.

Ako već koristite ove tehnike, a niste postigli željeni cilj definitivno trebate isprobati raspoređivanje opterećenja i dodati visoku razinu dostupnosti ako je stabilnost od kritične važnosti.

## Dodatna literatura

Više detalja o softveru korištenom u ovom poglavlju možete pronaći na Web stranicama:

- The Linux Virtual Server Project (*http://www.linuxvirtualserver.org*)
- Ultra Monkey (http://www.ultramonkey.org)
- Heartbeat/The High-Availability Linux Project (*http://linux-ha.orgS*)

Možda ćete htjeti pogledati i komercijalni softver za Linux, temeljen na LVS-u – Red Hat Cluster Suit (*http://www.redhat.com/software/rha/cluster*). Isti se softver može besplatno dobit u CentOS-u ali bez tehničke podrške.

# POGLAVLJE 8
# Servisi lokalne mreže

U ovom ćemo se poglavlju baviti nekim vještinama koje su administratorima sustava potrebne za održavanje računala iza vatrozida ili mrežnog prolaza tvrtke, organizacije ili čak kućne mreže.

Neki informatičari više vole čitati o razvoju internetskih tehnologija nego o lokalnim mrežama za koje smatraju da su stvar rutine i da kod njih nema ništa izazovno. Međutim, kad treba konfigurirati ili popraviti nešto iznimno važno za radno okruženje, lokalne mreže itekako dobivaju na važnosti. Primjerice, jako malo drugih stvari može biti važnije od elektroničke pošta koja ne radi.

Održavanje lokalne mreže administratoru sustava može uzeti jako puno vremena ako ne posjeduje dovoljnu količinu znanja o radu s njima. Prema tome, ako ste tek počeli raditi kao administrator sustava dobro će vam doći početni udžbenik o lokalnim mrežama i o tome kako instalirati, konfigurirati i održavati veliki broj različitih poslužitelja koji će se tu naći. Za osnove pogledajte najnovije izdanje knjige *Linux Network Administrator's Guide* čiji je autor Terry Dawson sa suradnicima (O'Reilly). Ako znate osnove Linuxa, čak i bez velikog poznavanja lokalnih mreža moći ćete pratiti ovo nama iznimno zanimljivo poglavlje.

U ovom poglavlju bavit ćemo se distribuiranim sustavima datoteka. Naučit ćemo kako postaviti DHCP i mrežni prolaz (uključujući usmjeravanje između lokalne mreže i Interneta). Upoznat ćemo problematiku mrežnog ispisa i upravljanje korisnicima. Lokalni servis elektroničke pošte također pripada području lokalnih mreža, no s njime smo se bavili u petom poglavlju.

Za ovo poglavlje koristit ćemo se distribucijom Fedora Core Linux. Red Hat sponzorira Fedora projekt i obično ga koristi za testiranje novih komercijalnih verzija. Iako Fedora nije najstabilnija verzija Red Hat Enterprise Linuxa, dovoljno je stabilna i što je najvažnije robusna. Red Hat osigurava izvorne pakete većine alata za Fedoru stavljajući je tako na vodeće mjesto besplatnih Linux distribucija dostupnih i za komercijalnu upotrebu.

Bez obzira na to sviđa li vam se Red Hatov model ili ne, moći ćete primijeniti materijal iz ovog poglavlja i na druge distribucije Linuxa. Preporučamo vam da ozbiljno pristupite čitanju ovog poglavlja: zabavno je, stečeno znanje trebat će vam praktički u svakom okruženju u kojem možete raditi, a i na drugim mjestima nećete pronaći ovoliko korisnog materijala.

# Distribuirani sustavi datoteka

Vjerojatno vam je teško zamisliti vrijeme u kojem su računala funkcionirala samostalno, bez uživanja pogodnosti mrežnog rada i veze s Internetom. Međutim PC računala su originalno i projektirana da rade samostalno, bez razmišljanja o umrežavanju. Možda se sjećate vremena kad su ljudi prenosili datoteke šećući s disketama od ureda do ureda ili prebacivali prekidač kako bi više korisnika moglo koristiti isti pisač. Bila su to teška vremena.

Puno je godina trebalo proći i puno se inovacija trebalo dogoditi da bi zaživjele osnovne mrežne pogodnosti kao što su distribuirani sustavi datoteka. Ovi sustavi datoteka su doslovce promijenili način poslovanja jer su omogućili da se računala postave na radni stol svakog zaposlenika. Više nije trebalo ručno ispunjavati obrasce za bušenje kartica koje bi se obrađivale na središnjem računalu.

Umrežavanje je postalo dostupnije tek kad je IBM-ov istraživač Barry Feigenbaum prebacio DOS-ov lokalni sustav datoteka u distribuirani. Njegov rad pomogao je u izradi aplikativnog protokola Server Message Block (SMB) čime je započelo razdoblje administratora sustava i mrežnih inženjera.

Distribuirani sustavi datoteka dozvoljavaju korisnicima otvaranje, čitanje i pisanje datoteka spremljenih, osim na onom računalu na kojem korisnik radi, i na drugim računalima u lokalnoj mreži. U nekim okruženjima na jednom velikom računalu spremaju se datoteke kojima pristupaju svi korisnici lokalne mreže. Središnje računalo može spremati i početne direktorije korisnika u koje se sprema sav njihov rad. U drugim okruženjima korisnici spremaju datoteke na svoja računala ali dozvoljavaju ostalim korisnicima pristup tim datotekama. Ova dva okruženja se također mogu pomiješati, a bez obzira na konfiguraciju takav način rada zovemo *dijeljenje datoteka*. Direktorije kojima korisnici mogu pristupiti na udaljenim računalima obično zovemo *dijeljeni resursi* (engl. *shares*).

Krajem osamdesetih godina računala su postala široko rasprostranjena u poslovanju, a čim su korisnici otkrili prednosti dijeljenja resursa, u poslovna okruženja su ušle i lokalne mreže.

Pokušajte zamisliti kako je uvođenje lokalne mreže izgledalo grupi korisnika koja nikad nije radila s mrežnim servisima. Najednom su mogli dijeliti dokumente, ispisivati ih na pisačima koji nisu na njihovim stolovima, odgovarati na elektroničke poruke šefova koji su mogli biti u bilo kojem dijelu ureda, zgrade ili čak zemlje. Ove nove mogućnosti većini su korisnika otvorile oči i približile im informatičke tehnologije.

U današnje vrijeme kod većine mrežnih lokacija osobito važne datoteke spremljene su na središnjem poslužitelju koji može dopustiti ili uskratiti pristup informacija određenom korisniku. Nešto kasnije u ovom poglavlju bavit ćemo se i upravljanjem korisnicima.

# Uvod u Sambu

SMB dijeljenje datoteka i pisača razvilo se, pod vodstvom Microsofta, u protokol Common Internet File System (CIFS). CIFS je objavljen kao standard no izuzetno je slabo dokumentiran i sadrži mnogo tajnih postupaka koje Microsoft dalje razvija.

Međutim, neustrašivi tim razvojnih programera počeo je provoditi obrnuti inženjering te je razvio jedan od najpopularnijih besplatnih alata za implementaciju Microsoftovog dijeljenja datoteka na sustavima koji ne koriste Microsoftov softver – *Sambu*. Samba je sve popularnija, ima podršku za Windowse i Linux, a čak se koristi i na sustavu Mac OS X.

Kao administrator Linux sustava morat ćete dobro razumjeti Sambu ako se njome želite ozbiljno baviti (a trebali biste). Postoji veliki broj knjiga i Web priručnika na stranicama *http://samba.org*. Upotrijebit ćemo uobičajenu frazu: „Daljnje ulaženje u ovu temu nadilazi okvire ove knjige". Ne vidimo razloga za ponavljanje izvrsnog materijala koji već postoji. No, usprkos tome, Sambu ćemo objasniti dovoljno detaljno tako da ju možete primijeniti u okruženju u kojem radite. Srećom, većina distribucija pruža jednostavno grafičko sučelje za rad sa Sambom čime ćemo se također baviti.

Neke središnje funkcije u CIFS mrežama (uglavnom načini na koje se sustavi međusobno traže) smještene su na *kontrolorima domene*: poslužiteljima koji pružaju datoteke i pisače te provode različite upravljačke operacije. Samba može uključiti Linux računala u Microsoftovu mrežu kao poslužitelje datoteka i ispisa, kontrolore domena i članove radne grupe.

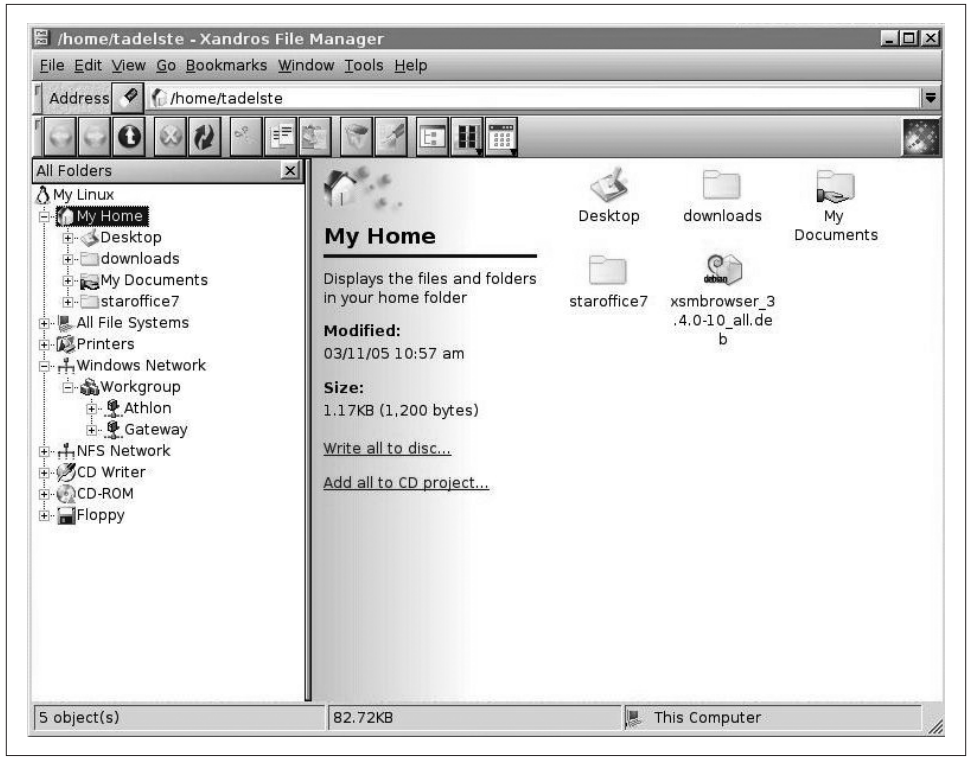
Posljednja inačica Sambe surađuje s Microsoftovim protokolom Active Directory. Samba kombinirana s LDAP-om također može biti robustan poslužitelj za provjeru autentičnosti zamjenjujući i Microsoft NT kontrolore domene i Active Directory poslužitelje.

Samba, u manjim okruženjima u kojima su računala povezana u mrežu bez poslužitelja, može upravljati i dijeljenjem datoteka. Korisnici mogu dijeliti pisače i datoteke bez provjere autentičnosti. Ako se osjetljive operacije, kao što je financijsko računovodstvo i čuvanje podataka, izvode na jednom računalu, mogu se implementirati jača sigurnosna pravila na razini računala čime se ono štiti bez ograničavanja mogućnost pristupa resursima.

Pogledajmo sada jednu Linux/Windows mrežu i način postavljanja Sambe u takvom okruženju.

## Konfiguriranje mreže

Slika 8-1 prikazuje mrežu kako bi se mogla vidjeti iz Linux sustava (Xandros distribucija, koja je pogodna za poslovna okruženja). Stablasti pogled na lijevoj strani zaslona pokazuje četiri računala nazvana *Athlon*, *Atlanta*, *Dallas* i *Dell*. *Dallas* drugim računalima nudi pisač, zajedno s nekoliko direktorija. *Dell* također može pružiti uslugu ispisa. Jedno od preostalih računala radi pod Windowsima XP a druga dva pod Windowsima 98. Linux ih sve povezuje. Linux sustav izgleda isto kao i Windows sustav kada se gleda u prozoru Network Neighborhood ili My Network Places.




*Slika 8-1. Dijeljene datoteke i direktoriji iz Linux sustava na računalu pod Windowsima*

Desno okno na slici 8-1 pokazuje dijeljene direktorije na čvoru *Dallas* koji radi pod Windowsima XP. Također možete vidjeti i datoteku *xsmbrowser_3.4.0-10_all.deb*.

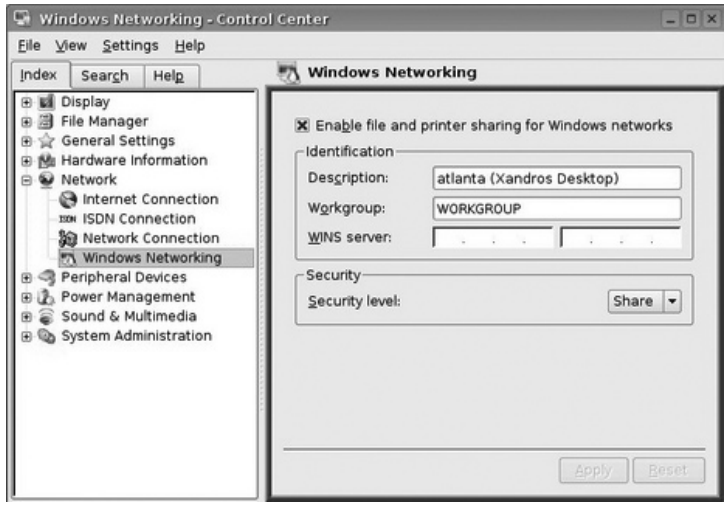
Koliko je teško postaviti ovu mrežu? Osim postavljanja i povezivanja kabela te instaliranja vatrozida i modema, sustav se praktički instalira sam. Slijedili smo standardne procedure za instalaciju na oba računala pod Windowsima 98. Sustavi koriste DHCP za uzimanje IP adresa, adresa DNS poslužitelja i mrežnog prolaza. Usmjerivač pruža DHCP servise i shemu privatnih IP adresa koristeći mrežu klase C (192.168.0.0 do 192.168.0.255). (DHCP ćemo objasniti u idućem odjeljku).

Kad sustav pod Windowsima uspostavi svoju mrežnu konfiguraciju i dobije pristup Internetu, desnom tipkom miša ćemo pritisnuti ikonu Network Neighborhood, odabrati opciju Properties te promijeniti dinamičku adresu u statičku. Ovo nam omogućava da računalo radi kao poslužitelj za ispis i dijeli svoju vezu na Internet s ostalim računalim u mreži.

Postavljanje sustava Windows XP nešto je kompliciranije budući se u početku računala pod Windowsima XP i Windowsima 98 međusobno ne vide. Da bismo riješili taj problem, moramo omogućiti Simple File Sharing kroz XP-ov Control Panel i pokrenuti čarobnjaka Network Setup. Čarobnjak nas pita da li želimo omogućiti dijeljenje na

drugim računalima, referirajući na računala pod Windowsima 98. Ako odgovorimo potvrdno moći ćemo izraditi disketu putem koje ćemo na računala s Windowsima 98 instalirati XP protokole. Ovaj proces unapređuje stari sustav tako da može raditi s novim protokolima koji omogućavaju komunikaciju između Windowsa 98 i XP. (Instalacijski program izradio je Microsoft i zove se *netsetup.exe*).

Sada možemo instalirati Xandros Linux i na njemu omogućiti Windows Networking, kao što je prikazano na slici 8-2.

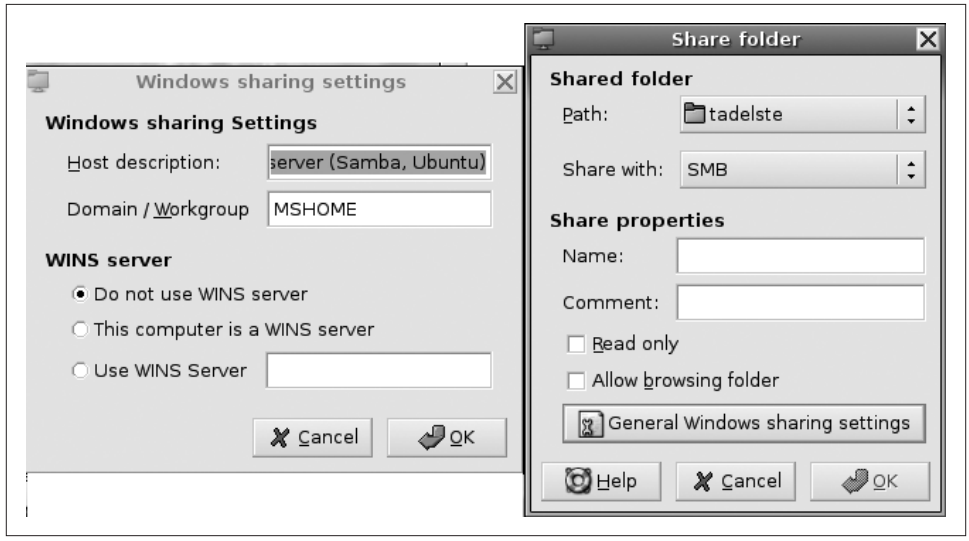



*Slika 8-2. Konfiguriranje mrežnog rada s Windowsima*

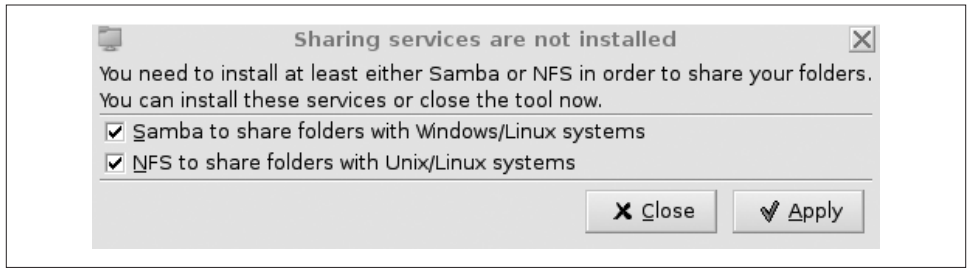
Primijetite da smo Windows Networking mogli konfigurirati kroz dijaloški okvir. Linux dozvoljava da uključimo dijeljenje datoteka i pisača, imenujemo računalo, definiramo radnu grupu i uključimo sigurnosni sustav za dijeljene resurse tako da mrežni čvorovi pod Windowsima mogu koristiti CIFS funkcionalnost.

Druge Linuxove distribucije, kao što su Fedora i Ubuntu, također nude alate za jednostavno postavljanje dijeljenja datoteka. Slika 8-3 prikazuje dva konfiguracijska zaslona Ubuntu Linuxa.

Ubuntu također pruža opcije za postavljanje sustava Network File System (NFS), popularnog sustava za dijeljenje datoteka tipa Unix-to-Unix koji je kompatibilan s CIFS-om. Dijaloški okvir na slici 8-4 dozvoljava da se odlučite za jedan ili oba sustava. Možete koristiti Sambu za povezivanje Windows i Mac OS X sustava, a NFS za komunikaciju između sustava pod Linuxom. Servisi dijeljenja resursa nisu podrazumijevano instalirani na Ubuntu Linuxu, no ako odaberete Shared Folders (pod izbornikom Administration u Ubuntuu 6.10) Ubuntu će preuzeti potrebne datoteke. Nakon toga računalo može postati član domene ili radne grupe.




*Slika 8-3. Postavljanje Ubuntu dijeljenih resursa u Windows okruženju*




*Slika 8-4. Zaslon za postavljanje servisa dijeljenja datoteka na Ubuntu Linuxu*

Nešto kasnije u ovom poglavlju, u dijelu „Servisi za dijeljenje pisača“ detaljnije ćemo se pozabaviti Sambom.

# DHCP

Servisi Dynamic Host Configuration Protocol (DHCP) mogu vam pomoći pri rješavanju brojnih problema u lokalnim mrežama, uključujući dodjeljivanje IP adresa i administriranje. Teško je zamisliti mrežu u kojoj se ne koristi DHCP.

Pogledajmo neke probleme s kojima ćete se susresti i kako DHCP može pomoći pri njihovom rješavanju:

- Računala i radne stanice zahtijevaju jedinstvene IP adrese, DNS informacije i adrese mrežnih prolaza.
- Ručno praćenje IP adresa zahtjeva ogromnu količinu rada.
- Slučajna ponavljanja IP adresa izazivaju konflikte u mreži.

- Rješavanje problema s adresama (kao što su ponovljene adrese) i promjene mjesta stvaraju nepotreban posao.
- Promjene osoblja obično znače da će netko trebati provjeriti sva računala i sastaviti novu bazu podataka o pridruženim IP adresama.
- Česta seljenja korisnika s prijenosnim računalima zahtijevaju ponovno konfiguriranje mrežnih postavki.

DHCP rješava ove probleme tako što dodjeljuje IP adresu svakom računalu u mreži u trenutku kada se ono spoji na mrežu. DHCP poslužitelj osigurava i jedinstvenost IP adresa. Servis zahtjeva malo administratorske pomoći, točnije administratori trebaju samo sastaviti konfiguracijsku datoteku i ostatak posla prepustiti DHCP poslužitelju (*dhcpd*). Ovaj će poslužitelj upravljati IP adresama i osloboditi administratora te zahtjevne zadaće.

## Instaliranje DHCP-a

Da biste mogli koristiti DHCP trebate instalirati DHCP poslužitelj. Buduči da je ovo poglavlje orijentirano na Fedoru, možete instalirati RPM paket s Yumom ili upravitelj paketa */usr/bin/gnome-app-install*. Trenutna inačica paketa je *dhcp-3.0.3-28.i386*. (Korisnici Debiana mogu instalirati paket *dhcp3-server* i urediti konfiguracijsku datoteku */etc/dhcp3/dhcpd.conf*.) Softver je projektirala organizacija Internet Systems Consortium.

Kad ga instalirate, konfigurirajte DHCP s pomoću konfiguracijske datoteke */etc/dhcpd.conf*. Prvi korak je kopiranje sadržaja datoteke */usr/share/doc/dhcp/dhcpd.conf.sample* u datoteku */etc/dhcpd.conf*. Sada možete prilagoditi datoteku potrebama svoje mreže kao u sljedećem tipičnom primjeru. Sintaksa koristi znak povisilice (#) za komentare:

```
ddns-update-style interim;
ignore client-updates;

subnet 192.168.1.0 netmask 255.255.255.0 {

# --- default gateway
    option routers 192.168.1.1;
    option subnet-mask 255.255.255.0;

# ---    option nis-domain "domain.org";
# ---    option domain-name "domain.org";
    option domain-name-servers 192.168.1.1;

# ---    option time-offset -18000;    # Eastern Standard Time
#    option ntp-servers 192.168.1.1;
#    option netbios-name-servers 192.168.1.1;
# --- Selects point-to-point node (default is hybrid). Don't change this
# -- unless you understand Netbios very well
#    option netbios-node-type 2;
```

```
# ---    range dynamic-bootp 192.168.0.128 192.168.0.254;
    default-lease-time 21600;
    max-lease-time 43200;

    # we want the nameserver to appear at a fixed address
    host ns {
        next-server server1.centralsoft.org;
        hardware ethernet 00:16:3E:63:C7:76;
        fixed-address 70.253.158.42;
    }
}
```

Nakon što smo konfiguracijsku datoteku kopirali u direktorij */etc* moramo zadati još nekoliko stvari:

```
subnet 192.168.1.0 netmask 255.255.255.0 {
    option routers 192.168.1.1;
    option domain-name-servers 192.168.1.1;
    option subnet-mask 255.255.255.0;
    default-lease-time 21600;
    max-lease-time 43200;
```

Prvi red postavlja opseg IP adresa koje korisnik ima na raspolaganju u podmreži lokalne mreže. U ovom slučaju koristili smo rezerviranu mrežu klase C 192.168.1.0 koja osigurava prostor za 254 čvora (s adresama od 192.168.1.1 do 192.168.1.254). Ova mrežna maska mora odgovarati onoj koja definira vašu lokalnu mrežu.

U drugom redu zadali smo adresu mrežnog prolaza, `option routers`, a u trećem poslužitelj imena, `option domain-name-servers`. IP adresa je u oba reda ista, što je uobičajena praksa.

Jedan poslužitelj sa dvije mrežne kartice vrlo često u lokalnoj mreži ima ulogu mrežnog prolaza. Jedna kartica, s imenom *eth0*, ima adresu na Internetu, dok druga kartica (nazovimo je *eth1*) ima adresu na privatnoj mreži.

Kada se paket prosljeđuje, a uključen je vatrozid *iptables*, svaki poslužitelj pod Linuxom može raditi kao mrežni prolaz/vatrozid. U ovom smo slučaju uključili BIND za pružanje DNS usluga.

Zadnja dva reda zadaju trajanje adrese klijenta, mjereno u sekundama.

U našoj DHCP konfiguracijskoj datoteci dodali smo i klauzulu koja zadaje statičku adresu za tvrtkin DNS poslužitelj:

```
# we want the nameserver to appear at a fixed address
host ns {
        next-server server1.centralsoft.org;
        hardware ethernet 00:16:3E:63:C7:76;
        fixed-address 70.253.158.42;}
```

U sljedećem dijelu „Pridruživanje IPv6 adresa pomoću programa radvd“ objasnit ćemo kako koristiti *dhcpd* za dodjeljivanje statičke IP adrese bazirane na MAC adresi klijentove mrežne kartice. Prije nego što to učinimo, pogledajmo jednostavnu inačicu datoteke */etc/dhcpd.conf*:

```
ddns-update-style interim;

default-lease-time              600;
max-lease-time                  7200;

subnet 192.168.1.0 netmask 255.255.255.0 {
    option routers 192.168.1.1;
    option subnet-mask 255.255.255.0;
    option domain-name-servers server.centralsoft.org,
        server2.centralsoft.org;
    range 192.168.1.2 192.168.1.254;
}
```

Kod jednostavnijih DHCP poslužitelja, održavanje će biti jednostavnije ako ispustite komentare i tako dobijete kraću konfiguracijsku datoteku.

## Pokretanje DHCP servisa

Neki DHCP servisi zahtijevaju datoteku *dhcpd.leases*. Koristite naredbu *touch* da biste izradili praznu datoteku u istom direktoriju u kojem se nalazi datoteka *dhcpd.conf*:

```
# touch /var/lib/dhcp/dhcpd.leases
```

Sada možete pokrenuti DHCP poslužitelj kako biste provjerili ispravnost konfiguracije. Također biste trebali konfigurirati poslužitelj tako da se pokrene prilikom pokretanja računala. Da biste izveli prvi od ova dva zadatka unesite:

```
[root@host2 ~]# service dhcpd start
Starting dhcpd:                                        [  OK  ]
[root@host2 ~]#
```

Možete provjeriti da li se DHCP proces izvodi sa sljedećim naredbama (ako se servis izvodi, bit će prikazan red sa podacima o njemu):

```
# ps aux | grep dhcpd
root   9028  0.0  0.0   2552   636    Ss   09:40   0:00 /usr/sbin/dhcpd
```

Zadajte naredbu *chkconfig* da biste zadali pokretanje DHCP-a zajedno s računala:

```
# chkconfig dhcpd on
# chkconfig --list
....from the list:
dhcpd           0:off   1:off   2:on    3:on    4:on    5:on    6:off
```

Kao i druge Linux servise, trebate ponovno pokrenuti DHCP svaki put kada promijenite konfiguracijsku datoteku. Možete zadati i druge opcije u datoteci *dhcpd.conf* koje mogu djelovati globalno, odnosno samo za određeno računalo ili podmrežu. To znači

da možete formirati dobre standardne za čitavu mrežu, a nakon toga ih premostiti za određenu skupinu računala ili samo jedno računalo. Navodimo primjer iz odjeljka za globalnu konfiguraciju na početku datoteke *dhcpd.conf*:

```
option domain name "host2.centralsoft.org";
```

## Statička IP adresa

Radne stanice obično dobro funkcioniraju s dinamičkim adresama (tj. adresama koje se dodjeljuju prilikom pokretanja računala), međutim poslužiteljima su korisnije statičke adrese budući da se neće promijeniti usred sesije s klijentom. DHCP dozvoljava zadavanje statičke IP adrese za pojedinačna računala u datoteci *dhcpd.conf*. Učinimo to kroz nekoliko koraka.

Najprije postavite mrežnu masku, adresu emitiranja i usmjerivače:

```
subnet 192.168.1.0 netmask 255.255.255.0
option broadcast-address 192.168.1.255;
option routers 192.168.1.1;
```

Nadalje, dodajte odjeljak `host` za svako računalo u mreži. Da biste to učinili morate znati hardversku adresu (*MAC adresu*) svake mrežne kartice, koja se može saznati zadavanjem naredbe *ifconfig* na odgovarajućem računalu. Slijedi primjer odjeljka `host`:

```
# ethernet MAC address as follows (Host's name is "laser-printer"):

host laser-printer {
    hardware ethernet 08:00:2b:4c:59:23;
    fixed-address 192.168.1.10;
}

host1.centralsoft.com {
    hardware ethernet 01:0:c0:2d:8c:33;
    fixed-address 192.168.1.5;
}
```

Izradite ovakvu konfiguracijsku klauzulu za svaki poslužitelj koji treba statičku IP adresu te ju dodajte u konfiguracijsku datoteku.

## Pridruživanje IPv6 adresa pomoću programa radvd

1995. godine Steve Deering i Robert Hinden uočili su potrebu za novim sustavom adresiranja na Internetu. Njihov prvi sus tav IPv6 nastao je 1995. u sklopu projekta IETF Request for Comments (RFC) 1883. Drugi se pojavio 1998. u RFC 2460. Deering i Hinden obznanili su ono što je već većina stručnjaka znala: IPv4 32-bitni adresni prostor ograničava eksplozivan rast Interneta.

Nekolicina administratora sustava shvatilo je da IPv6 i njegove nove metode pridruživanja IP adresa postaju sve popularniji. Međutim, mnogo stručnjaka ismijava IPv6,

smatrajući ga nepotrebnim, odnosno vjeruju da postojeća praksa neće poticati njegov razvoj budući da ga zahtjeva vrlo malo aplikacija i okruženja.

U ovoj knjizi ne možemo detaljno objašnjavati IPv6 zbog nedostatka prostora. Da biste saznali više o ovom protokolu i sistemskom servisu kao i o načinu dobivanja javnih IPv6 adresa morat ćete potražiti druge izvore informacija.

IPv6 adrese često uključuju hardversku adresu mrežne kartice. To IPv6 korisnicima omogućava dobivanje statičke IP adrese bez konfiguriranja na strani poslužitelja. Automatsko pridruživanje IPv6 adresa može se izvesti uz pomoć usmjerivačkog sistemskog servisa *radvd*.

Korisnici Fedore mogu instalirati *radvd-0.9.1* paket iz Yum repozitorija. Korisnici Debiana će instalirati *radvd* paket i potražiti datoteku */usr/share/doc/radvd/README.Debian*.

*radvd* prati zahtjeve usmjerivača i šalje njegove obavijesti kao što je opisano u RFC 2461, „Neighbor Discovery for IP Version 6 (IPv6)“. Računala mogu automatski konfigurirati svoje adrese i odabrati podrazumijevane usmjerivače na temelju ovih obavijesti.

*radvd* podržava jednostavni protokol, a vidjet ćete da je i njegova konfiguracija jednostavna. Evo primjera potpuno konfigurirane datoteke */etc/radvd.conf*:

```
interface eth0
{
    AdvSendAdvert on;
    prefix 0:70:lf0Q:96::/64
    {
    };
};
```

Ako želite koristiti *radvd* trebat ćete promijeniti `prefix` u onaj koji odgovara vašoj mreži i postaviti servis. Također ćete trebati zasebno konfigurirati DNS na klijentskim radnim stanicama.

Web stranica projekta *radvd* nalazi se na adresi *http://www.litech.org/radvd*.

## Servisi mrežnog prolaza

Linux korisnicima lokalne mreže pruža mogućnost pristupanja Internetu bez otkrivanja IP adrese. U tipičnoj situaciji aktivnosti unutar organizacije skrivaju se od javnosti upotrebom Linux usmjerivača. S privatne strane usmjerivača lokalne aktivnosti se odvijaju tako da ih nitko na javnoj strani ne može detektirati.

Mrežni prolaz se ponekad naziva i *utvrđeno računalo* (engl. *bastion host*). Možete ga shvatiti kao mrežni objekt koji ima jedan ulaz i jedan izlaz usmjeren ka Internetu. Ovakva računala pomažu prevenciji upada u mrežu tako što postavljaju barijeru između privatnih i javnih dijelova mreže. Servise koji ovo omogućavaju zovemo *servisima mrežnog prolaza* (engl. *gateway services*).

Administratori Linux sustava implementiraju servise mrežnog prolaza kombiniranjem usmjeravanja paketa i pravila vatrozida poznata što se naziva *iptables*. Postoje i druga imena za servis mrežnog prolaza kao što je *maskiranje* (engl. *masquerading*) ili Network Address Translation (NAT).

U malim organizacijama i kućnim mrežama, mrežni prolaz može postojati na jednom poslužitelju i uključivati osnovne sigurnosne servise, vatrozid, DHCP, DNS i servise elektroničke pošte. U većim organizacijama ovakvi servisi rasprostiru se na nekoliko poslužitelja sa demilitariziranom zonom (DMZ) koja izolira mrežni prolaz.

### Uloga DMZ-a

U sigurnosti računalnih sustava, pojam demilitarizirane zone označava međumrežu – podmrežu ili mrežu između unutarnje mreže i Interneta. Primjerice, vaša privatna mreža može koristiti unutarnju mrežu 192.168.1.0, DMZ 10.0.0.0, a javni Internet blok 70.253.158.0.

DMZ obično sadrži poslužitelje kojima se može pristupiti izvan lokalne mreže kao što su poslužitelj elektroničke pošte, Web poslužitelji i DNS poslužitelj. Vezu između Interneta i DMZ-a obično kontrolira *Port Address Translation* (PAT).

Izvor i odredište svakog IP paketa sadrži IP adresu i ulaz. PAT mijenja adresu i u dolaznim i u odlaznim paketima podataka. Brojevi ulaza, a ne IP adrese, koriste se za razlikovanje računala u unutrašnjoj mreži.

DMZ je obično smještena između dva mrežna prolaza ili vatrozida te je spojen na oba tako što je jedna mrežna kartica spojena na unutarnju mrežu, a druga na Internet. DMZ može spriječiti slučajne pogreške u konfiguraciji koje bi dozvolile pristup unutarnjoj mreži s Interneta. Ovo se zove *vatrozid sa skrivenom podmrežom*.

Za naše potrebe ograničit ćemo konfiguraciju mrežnog prolaza na prosljeđivanje paketa. Nećemo gubiti vrijeme na DMZ jer zahtjeva puno opreme i truda. Da biste postavili mrežni prolaz, trebate:

- Računalo koje će imati ulogu mrežnog prolaza
- Vezu na Internet i dvije mrežne kartice
- Mali preklopnik prkeo kojeg će se klijenti spajati na mrežni prolaz
- Instaliran *iptables*

Pretpostavit ćemo da je u vašoj konfiguraciji *eth0* veza na Internet, a *eth1* unutarnji mrežni prolaz. Uredite konfiguracijsku datoteku za *eth0* koja se nalazi u direktoriju */etc/sysconfig/networking/devices/ifcfg-eth0* i dodajte sljedeće redove:

```
ONBOOT=yes
USERCTL=no
IPV6INIT=no
PEERDNS=yes
GATEWAY=70.253.158.46
```

```
TYPE=Ethernet
DEVICE=eth0
HWADDR=00:04:61:43:75:ee
BOOTPROTO=none
NETMASK=255.255.255.248
IPADDR=70.253.158.43
```

Slično, konfiguracija za *eth1* treba izgledati ovako:

```
ONBOOT=yes
USERCTL=no
IPV6INIT=no
PEERDNS=yes
TYPE=Ethernet
DEVICE=eth1
HWADDR=00:13:46:e6:e5:83
BOOTPROTO=none
NETMASK=255.255.255.0
IPADDR=192.168.1.1
```

Informacije o ovim konfiguracijskim parametrima mogu se pronaći u datoteci *sysconfig.txt* koja se nalazi u direktoriju */usr/share/doc/initscripts-7.93.7.* ili nekom drugom direktoriju sličnog imena.

Kad ste konfigurirali mrežnu karticu trebate se uvjeriti da je servis *iptables* instaliran. Trebali biste dobiti ovakav rezultat:

```
[root@host2 devices]# rpm -q iptables
iptables-1.3.5-1.2
[root@host2 devices]#
```

Ako nemate instaliran *iptables*, instalirajte ga i učitajte module.

Fedora 5 instalirat će *iptables* koristeći aplikaciju Add/Remove Software koja se nalazi iznad izbornika Application na GNOME paleti. Ova aplikacija tijekom instalacijskog procesa učitava i module jezgre.

Sada pokrenite:

```
# iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE
# service iptables save
# echo 1 > /proc/sys/net/ipv4/ip_forward
```

Sada uredite datoteku *etc/sysctl.conf* tako da promijenite `net.ipv4.ip_forward = 0` u `1` kako bi se aktivirala nakon ponovnog pokretanja. Podesite sustav tako da ponovno učita datoteku *etc/sysctl.conf* zadavanjem:

```
# sysctl -p
```

Konačno, ako imate malu organizaciju, možete dodati DHCP poslužitelj koristeći jednostavnu verziju *dhcpd.conf*.

```
ddns-update-style interim;

default-lease-time              600;
max-lease-time                  7200;
```

```
subnet 192.168.1.0 netmask 255.255.255.0 {
    option routers 192.168.1.1;
    option subnet-mask 255.255.255.0;
    option domain-name-servers server1.centralsoft.org,
        server2.centralsoft.org;
    range 192.168.1.2 192.168.100.254;
}
```

## Drugi pristupi pružanju servisa mrežnog prolaza

Ovaj dio poglavlja pokriva upotrebu proizvoda koji kombiniraju mrežni prolaz i vatrozid te njihove brojne prednosti. Postoji nekoliko besplatnih paketa, kao što su Firestarter, IPCop, Netfilter i Shorewell. Vidjet ćete da se u literaturi o Linuxu spominju Smoothwall i ClarkConnect, ali to su komercijalni proizvodi koji instaliraju čitav Linux a ne samo samostalne aplikacije.

Za potrebe ovog poglavlja odlučili smo se za Firestarter. Međutim, možete pogledati i Shorewall, konfiguracijski alat za Netfilter (alat za odzivnik).

Firestarter možete preuzeti sa Fedorinog repozitorija. Naša instalacija ima sljedeći paket:

```
[root@host2 ~]# rpm -q firestarter
firestarter-1.0.3-11.fc5
[root@host2 ~]#
```

Čarobnjak Firestarter Firewall (slika 8-5) otvara se prilikom prvog pokretanja programa. Možete ga ponovno pokrenuti odabirom opcije s izbornika Firewall u glavnom sučelju, kao i promijeniti odabrane opcije u postavkama.

*Slika 8-5. Čarobnjak Firestarter Firewall*

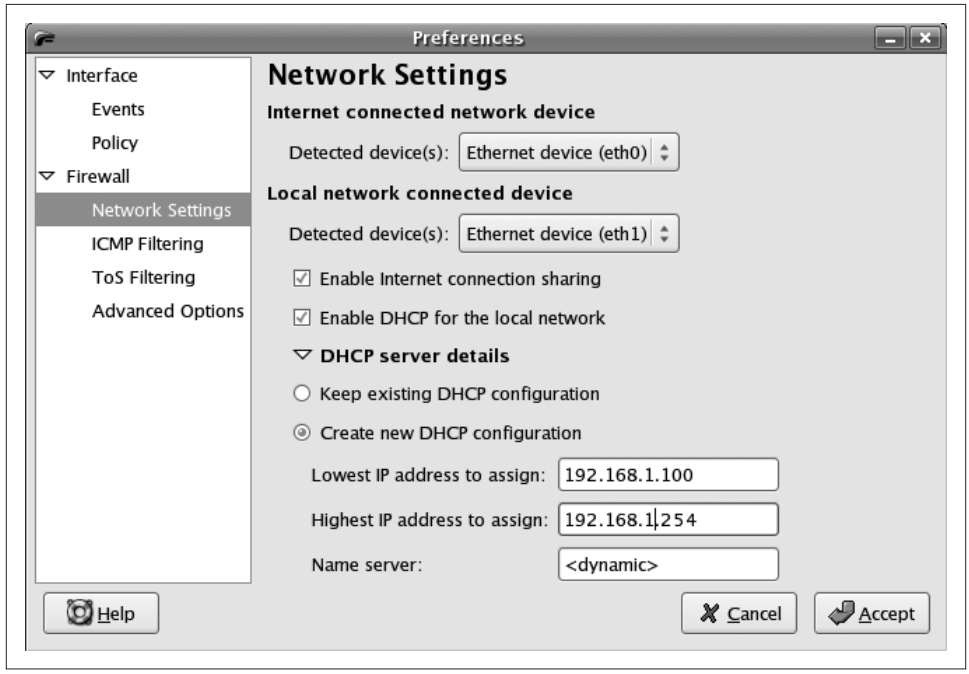
Nakon uvodnog zaslona pojavljuje se niz konfiguracijskih zaslona, počevši sa zaslonom za postavljanje mrežnih uređaja (slika 8-6). Na tom zaslonu možete postaviti dvije mrežne kartice.

*Slika 8-6. Zaslon za postavljanje mrežnih uređaja*

Firestarterova primarna funkcija je *dijeljenja veze*. Međutim, budući da koristi NAT, funkcionira i kao mrežni prolaz, tako da računala u lokalnoj mreži gledano s Interneta izgledaju kao jedno računalo s jednom IP adresom. To postaje očito na zaslonu za promjenu odabranih postavki prikazanom na slici 8-7. Primijetite da prvi opis uređaja upućuje na „Internet connected network device“, a drugi na „Local network connected device“.

Na dnu slike 8-7. možete vidjeti da Firestarter omogućava administratoru korištenje postojeće DHCP konfiguracije ili izradu nove. Ovdje je naveden sadržaj Firestarterove *dhcpd.conf* datoteke:

```
# DHCP configuration generated by Firestarter
ddns-update-style interim;
ignore client-updates;

subnet 192.168.1.0 netmask 255.255.255.0 {
    option routers 192.168.1.1;
    option subnet-mask 255.255.255.0;
    option domain-name-servers 70.253.158.42, 70.253.158.45, 151.164.1.8;
    option ip-forwarding off;
    range dynamic-bootp 192.168.1.10 192.168.1.254;
    default-lease-time 21600;
    max-lease-time 43200;
}
```




*Slika 8-7. Firestarterov zaslon za promjenu odabranih postavki*

Datoteka *resolv.conf* na mrežnom prolazu pojavljuje se u DHCP konfiguracijskim postavkama klijenta kada Firestarter čita tu datoteku i upisuje adresu DNS poslužitelja iz *dhcpd.conf*.

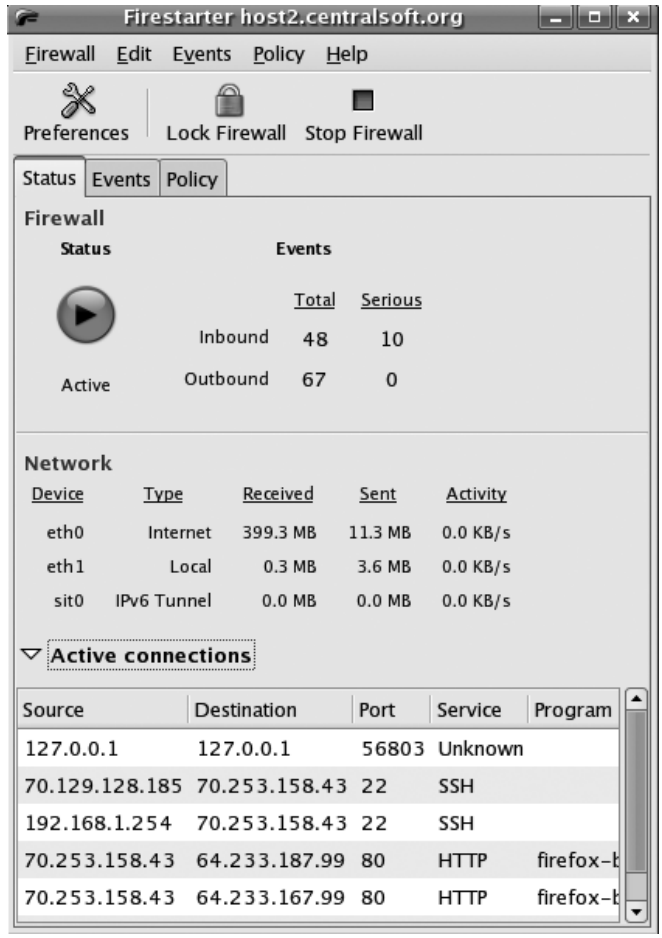
Glavno sučelje Firestartera pruža pregled statusa mrežnog prolaza i veza do DHCP računala. Također se pruža sažetak događaja i aktivnosti, kao što je prikazano na slici 8-8.

Na slici 8-9 možete vidjeti pregled događaja s druge kartice glavnog sučelja. U ovom pogledu možete vidjeti i blokirane veze.

Kartica Events prikazuje dnevnik napada na vatrozid. Može vam biti koristan kada zlonamjernici pokušaju provaliti u vaš sustav. Ako su uljezi uporni dodajte IP adresu s koje pristupaju u datoteku */etc/hosts.deny*. Ako netko pokuša pristupiti računalu preko *ssh* ulaza 22 koristeći napad s rječnikom, taj ulaz možete vrlo jednostavno zatvoriti pomoću Firestartera.

Ikona Firestartera postaje crvena kad detektira potencijalni napad. Primijetite poruku iznad nje na slici 8-10: „Hit from 221.237.38.69 detected“. To je vrijedno istraživanja.

Treća kartica glavnog sučelja dozvoljava postavljanje pravila za servise koje ćete dozvoliti ili zabraniti. Primjerice, dozvoljavamo vanjske SSH veze kroz vatrozid pa smo otvorili ulaz 22.

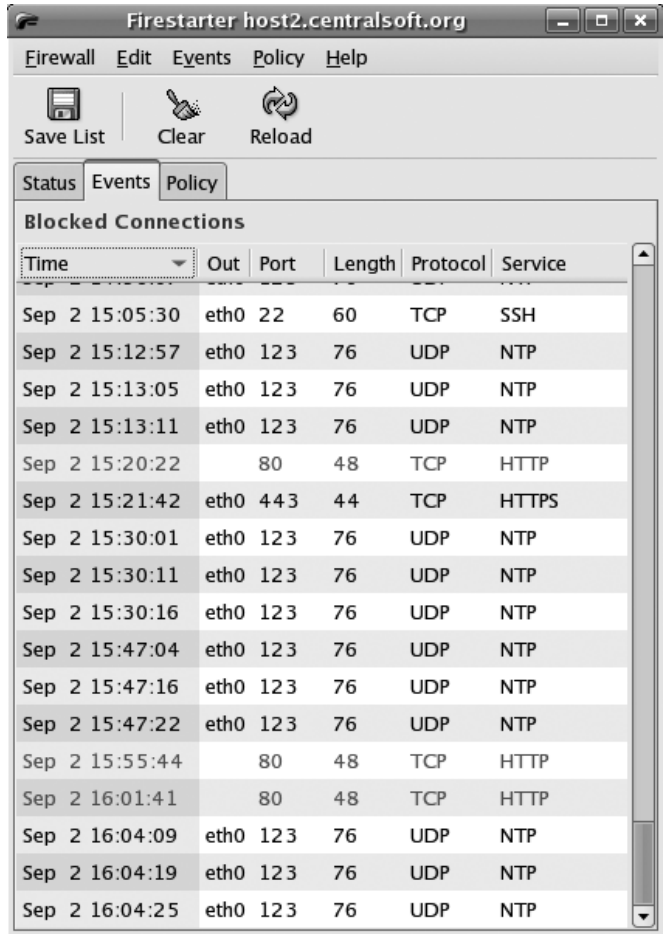



*Slika 8-8. Glavno sučelje Firestartera*

Firestarter koristi čarobnjaka za konfiguriranje pravila mrežnog prolaza. Na slici 8-11 možete vidjeti što vam je sve na raspolaganju.

Slika 8-11 prikazuje prozor nazvan „Add new inbound rule“. On se otvara nakon što odaberete Add Rule na kartici Policy. U ovom prozoru možete vidjeti opcije koje možete koristiti za omogućavanje servisa u mreži. Sličan prozor postoji za izlazne servise koje pružate korisnicima.

Vidjet ćete da je Firestarter aplikacija koja se jednostavno konfigurira. Projektni tim odlično je dokumentirao procedure u jezgrovito pisanom korisničkom priručniku koji možete preuzeti s adrese *http://fs-security.com/docs.php*.




*Slika 8-9. Firestarterova kartica Events*

*Slika 8-10. Ikona na paleti poslova pokazuje pokušaj nedozvoljenog upada u sustav*

Možda vam je čudno zašto smo uključili aplikaciju koja ovisi o GNOME-u. Prisjetimo se da koristimo Fedoru za postavljanje lokalne mreže zbog velikog broja alata koje vam stavlja na raspolaganje. Dodavanje Firestartera uklapa se u pristup, bez gubitka mogućnosti korištenja tekstualnog sučelja.

*Slika 8-11. Konfiguriranje pravila u Firestarteru*

# Servisi za ispis

Kao administratoru Linux sustava, pisači vam mogu prouzročiti ozbiljnu glavobolju. Vjerojatno će hardver, softver i operativni sustav biti nekompatibilni. Budući da postoji široka paleta sustava i metoda za konfiguriranje pisača, ovo područje administracije na najboljem je putu da vam kvari raspoloženje mjesecima – ili barem dok ne uspijete ovladati situacijom.

Počnimo s hardverom. Većina administratora radi s četiri tipa hardvera za ispis preko mreže. U postojećoj mreži možete naići na bilo koju od ovih kombinacija:

- Pisači su spojeni na pojedinačna korisnička računala
- Neka računala koriste se kao poslužitelji za ispis

- Postoji mrežni pisač s ugrađenom Ethernet karticom
- Specijalizirani poslužitelj za ispis spaja pisač izravno na lokalnu mrežu

U većini uredskih zgrada srednje veličine naići ćete na neko od ovih rješenja praktički iza svakog ugla. Fleksibilnost koju pružaju moderni sustavi često je uzrok problema.

Pretpostavimo da jedna korisnica, Sanja, dobila tintni pisač i spojila ga na svoje računalo. Božo, koji sjedi za susjednim stolom, pitao ju je smije li koristiti njen pisač. Da bi mu to dozvolila, Sanja desnom tipkom miša pritisne ikonu pisača i odabere opciju „Share“. Božo se pokušava spojiti na Sanjin pisač, ali ne uspjeva. Zašto? Nema instaliran upravljački program.

Ovo dvoje korisnika zovu administratora sustava (a to ste vi) kako bi riješio ovaj problem. Instalirate upravljački program na Božino računalo i gle čuda – sve radi. Nešto kasnije zove vas Sanja sa žalbom da njeno računalo treba više memorije i brži procesor. Zašto? Njen pisač sad koristi deset ljudi jer je uključila neograničeno dijeljenje resursa i to usporava računalo.

Kada ste pregledavali status mreže, uočili ste da nitko ne koristi veliki laserski pisač. Pitate se zašto korisnici ne ispisuju dokumente na tom pisaču? Ispada da se taj pisač ne vidi na mreži jer ga nitko nije dodao na upravljač domene.

Iz ove hipotetske situacije možete vidjeti da administrator sustava mora imati spremnu strategiju za upravljanje infrastrukturom ispisa. U ovom dijelu dobit ćete dobar globalni pregled ali i dovoljno praktičnih informacija da možete početi raditi. Proces možete započeti popisom hardvera i nekim odlukama u vezi softvera i operativnog sustava.

Budući ima mnogo tipova pisača, operativnih sustava i softvera morati ćete učiti u hodu. Najbolji način za učenje o organiziranju ispisa jeste razvijanje strategije za vlastitu infrastrukturu. To će smanjiti količinu informacija koje trebate obraditi.

## Odluka o izboru softvera za ispis

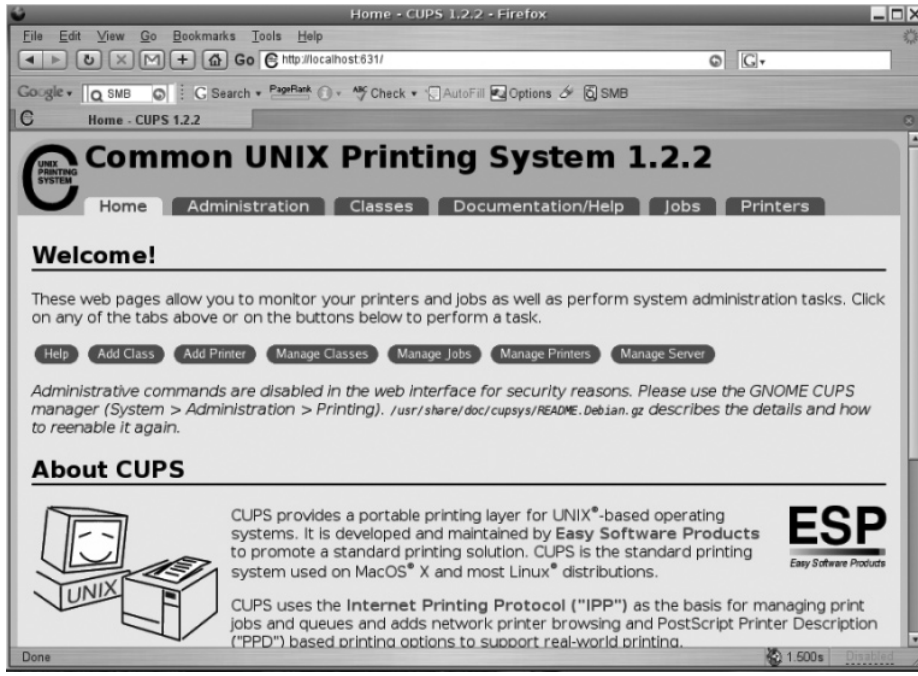
Linux i Windowsi započeli su s potpuno različitim modelima ispisa. Srećom, postignut je napredak pa ti modeli mogu međusobno dobro surađivati. No, da bi suradnja na mreži bila uspješna, najprije morate konfigurirati pisače.

Originalno, Linux koristi Unixov standard za ispis poznat kao Line Printer Deamon (LPD). Kasnije je dodano poboljšanje ovog sistemskog servisa koje se zove LPRng. Linuxove distribucije također koriste LPD alate za ispis i međusobnu operabilnost s različitim varijantama Unixa.

Linuxovi distributeri i dalje koriste LPD i njegove alate, ali su dodali i podršku za novi sustav poznat kao Common Unix Printing System (CUPS). Za razliku od LPD-a, CUPS je kompatibilan s Windows i Macintosh operativnim sustavima. CUPS i LPD

koriste različite mrežne protokole ispisa. Dok LDP ne može komunicirati s procesom ispisa u cilju dobivanja osnovnih značajki, CUPS može. CUPS radi bez problema u heterogenim mrežama, a ako je to potrebno, može surađivati i sa Sambom. Ne donose sve Linux distribucije ovo sučelje ali Red Hat standardno uključuje CUPS u Fedoru.

Kao administrator sustava morat ćete se pobliže upoznati s CUPS-ovim administrativnim alatima. Ako koristite Fedoru, upišite *http://localhost:631* u preglednik i vidjeti ćete upravljačko sučelje prikazano na slici 8-12.




*Slika 8-12. CUPS-ovo konfiguracijsko sučelje*

Sučelje je vrlo intuitivno tako da vam prepuštamo da se sami upoznate s njim. Ako se želite dodatno baviti CUPS-om, pogledajte upravljačko sučelje ili posjetite Web stranice projekta na adresi *http://www.cups.org/book/index.php* i pročitajte priručnik.

## Ispis s više platformi

Pogledajmo sada nekoliko problema s ispisom s kojima ćete se vrlo vjerojatno susresti u današnjem poslovnom okruženju. Gotovo ćete sigurno naići na situacije u kojima morate dijeliti pisač spojen na Linux sa računalima pod Windowsima. (U biti, vjerojatno ćete htjeti koristiti Linux kao poslužitelj ispisa u Windows mreži da biste

uštedjeli na troškovima licence). Također ćete vrlo vjerojatno morati dijeliti pisače spojene na Windows računala s računalima pod Linuxom. Kako to izvesti?

Pogledajmo najprije kako Windows korisnicima omogućiti pristup do pisača spojenih na Linux računalo. Najčešće trebate postaviti Sambinu radnu grupu ili domenu i instalirati CUPS na računalo pod Linuxom. Također ćete trebati konfigurirati CUPS za Sambu, što radite zadavanjem sljedeće naredbe:

```
# ln -s `which smbspool` /usr/lib/cups/backend/smb
```

Uredite datoteku */etc/samba/smb.conf* kako biste aktivirali dijeljenje pisača na Samba poslužitelju. U stvarnim ćete situacijama nekim sustavima ili korisnicima ograničiti pristup pojedinom pisaču, ali u sljedećem primjeru Linux računalo dijeli sve svoje pisače sa svim računalima u mreži koja Samba poslužuje:

```
[printers]
    comment = All Printers
    printing = cups
    printcap name = cups
```

Windows računala sada mogu pristupati pisačima na čitavoj mreži. Vjerojatno ćete trebati upravljačke programe za pisače, a njih možete naći na CD-u koji ste dobili uz pisač ili već postoje u bazi upravljačkih programa na Windowsima.

U sljedećem scenariju trebate omogućiti Linux korisnicima korištenje pisača spojenog na Windows poslužitelj. Ponovno će vam trebati CUPS i Samba da biste to učinili. Na Windows računalima uključite opciju dijeljenja pisača kao što to normalno radite: pod Windowsima NT, 2000 i XP definirajte *guest* korisnički račun i dozvolite svim korisnicima pristup dijeljenom pisaču. Sada instalirajte CUPS na Samba poslužitelju i konfigurirajte ga za Sambu kao što je ranije objašnjeno.

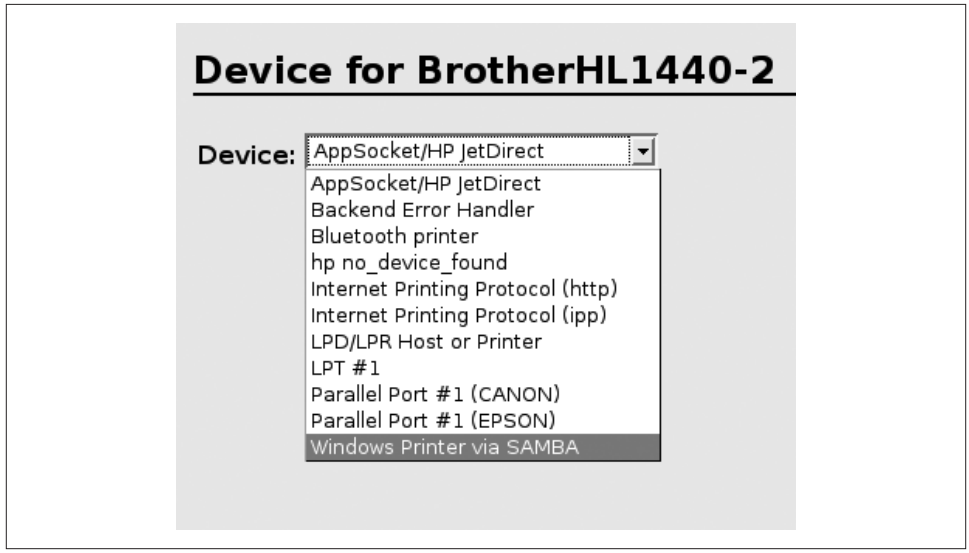
Sada možete instalirati Windows pisače koje želite učiniti dostupnim na Samba poslužitelju s CUPS-om, koristeći CUPS-ovo Web sučelje.

Trebate se prijaviti kao *root* korisnik. Na nekim Linux sustavima morate postaviti *root* za administratora CUPS-a. To možete učiniti pomoću naredbe *adduser*:

```
~$ su
Password:
# adduser cupsys shadow
Adding user `cupsys' to group `shadow'...
Done.
# /etc/init.d/cupsys restart
Restarting Common Unix Printing System: cupsd                    [ ok ]
#
```

Sada se možete prijaviti kao *root*.

Pritisnite „Add Printer“ i upišite naziv pisača iz Windows sustava. Mi ćemo koristiti „BrotherHL1440“ (pogledajte sliku 8-13). Nakon toga unesite lokaciju i opis. Kad dođete do prozora uređaja s izbornika odaberite „Windows Printer via Samba“.




*Slika 8-13. Dodavanje Windows pisača*

U sljedećem prozoru „Device URI for“ upišite URI uređaja. „BrotherHL1440-2“ spojen je na računalo Philadelpha sa instaliranim Windowsima 2003, tako da morate unijeti korisničko ime „guest“ i ime računala:

```
smb://guest@philadelphia/brotherhl1440-2
```

Sada trebate odabrati upravljački program za pisač. Također bi bilo dobro ispisati i testnu stranicu. Na Linux klijentu otvorite CPUS sučelje i trebali biste vidjeti pisač. Linux klijenti na lokalnoj mreži sada mogu koristiti pisač.

## Nadzor nad redovima za ispis iz odzivnika

Naredbom *ssh* se možete spojiti na Linux poslužitelj za ispis i koristit CUPS naredbe za upravljanje redom dokumenata koji čekaju na ispis. CUPS CLI naredbe obično zahtijevaju dopuštenja *root* korisnika.

Pogledajmo ukratko te naredbe:

*lpc*
: Dozvoljava različite oblike kontrole pisača. Sa *lpc status* naredbom možete vidjeti popis aktualnih redova za ispis i status svakog od njih.

*lpstat*
: Prikazuje popis svih započetih procesa ispisa na pisačima sustava. Možete koristiti različite opcije modificiranja rezultata ove naredbe.

*lpq*
: Prikazuje status trenutnog reda ili reda zadanog u `-P red` opciji.

*lppassws*
: Mijenja lozinku CUPS sustava. Postavite `AuthType` na `Digest` u konfiguracijskoj datoteci *cupsd.conf*.

*enable i disable*
: Pokreće odnosno zaustavlja odabrani red. Najčešće se koristi naredba *disable* s opcijom `-c` da bi se red zaustavio i da bi se odustalo od ispisa svih dokumenata koji čekaju u redu.

*accept i reject*
: Zadaje da red za ispis prihvaća ili odbacuje nove poslove ispisa.

*lprm*
: Uklanja posao ispisa iz reda. Možete zadati red (`-P red`) i identifikator posla (dobiven pomoću naredbe *lpstat*).

*lpmove*
: Premješta poslove ispisa iz jednog reda u drugi pomoću identifikatora posla i imena reda (npr. `lpmove red1-46 red2`).

Nožete sami isprobati ove naredbe. Navodimo primjer upotrebe prve naredbe na pisaču koji smo upravo postavili koristeći CUPS sučelje:

```
# lpc status
BrotherHL1440:
        printer is on device 'parallel' speed -1
        queuing is enabled
        printing is enabled
        no entries
        daemon present
```

# Upravljanje korisnicima

Linux pruža brojne načine za administriranje korisnika (dodavanje, mijenjanje, brisanje). Na početku ovog dijela pretpostavit ćemo da svaki poslužitelj koji administrirate ima vlastitu bazu korisnika u datoteci */etc/passwd*. Pretpostavit ćemo također da poznajete osnove dodavanja i brisanja korisničkih računa pomoću naredbi *adduser* i *useradd* ovisno o distribuciji koju imate.

Različite Linux distribucije promijenile su ponašanje naredbi *adduser/useradd*. Možete pronaći stranicu s uputama za svaku od ove dvije naredbe, ali vjerojatno nećete uspjeti naći ništa o razlikama između naredbi. Najbolje je eksperimentirati da vidite kako se vaša distribucija ponaša. U Fedori se ove dvije naredbe ponašaju isto: obje dodaju korisnički račun i korisnički direktorij. Upišete li *adduser tadelste* ili *useradd tadelste* dodati ćete korisnika i napraviti njegov početni direktorij. Međutim, neće tražiti privremenu lozinku niti će vam postavljati standardna pitanja koja očekujete.

Na drugim distribucijama možete vidjeti rezultat poput ovog:

```
... # adduser tadelste
Adding user `tadelste'...
Adding new group `tadelste' (1001).
Adding new user `tadelste' (1001) with group `tadelste'.
Creating home directory `/home/tadelste'.
Copying files from `/etc/skel'
Enter new UNIX password: lozinka1
Retype new UNIX password: lozinka1
passwd: password updated successfully
Changing the user information for tadelste
Enter the new value, or press ENTER for the default
        Full Name []: Novi korisnik
        Room Number []:
        Work Phone []: 999-555-1212
        Home Phone []:
        Other []:
Is the information correct? [y/N] y
```

Kod Fedore dijalog staje kod reda „Copying files ...“. Od administratora se tada očekuje da samostalno zada lozinku za korisnika. No, što se dogodi ako administrator ne zada lozinku odmah? Može li dodani korisnik pristupiti poslužitelju koristeći *ssh*? Pokušajmo:

```
$ ssh tadelste@host2.centralsoft.org
tadelste@host2.centralsoft.org's password:
Permission denied, please try again.
tadelste@host2.centralsoft.org's password:
Permission denied, please try again.
tadelste@host2.centralsoft.org's password:
Permission denied (publickey,gssapi-with-mic,password).
$
```

Kao što vidite odgovor je negativan. Korisnik ne može koristiti praznu lozinka, tj. ona mu uopće nije zadana. U datoteci *ssh_config* zadano je da korisnik mora upisati lozinku i zbog toga se ne može prijaviti.

*root* mora zadati lozinku za korisnika, što administrator radi na ovaj način:

```
[root@host2 ~]# passwd tadelste
Changing password for user tadelste.
New UNIX password: lozinka1
Retype new UNIX password: lozinka1
passwd: all authentication tokens updated successfully.
[root@host2 ~]#
```

U rezultatu stoji da naredba *passwd* mijenja korisničku lozinku, no to nije tako budući da ne traži unos (nepostojeće) trenutno važeće lozinke.

Kad korisniku bude dodijeljena lozinka, moći će ju samostalno promijeniti:

```
$ passwd
Changing password for user tadelste.
Changing password for tadelste
```

```
(current) UNIX password: lozinka1
New UNIX password: lozinka1
Password unchanged
New UNIX password: lozinka2
Retype new UNIX password: lozinka2
passwd: all authentication tokens updated successfully.
$
```

Fedora najprije provjerava imate li uopće lozinku (ako je nemate nećete se moći prijaviti na poslužitelj). Također provjerava razlikuje li se nova lozinka od postojeće. Ako unesete istu lozinku, Fedora je neće prihvatiti i tražit će da ponovite unos.

Budući da Fedora koristi Red Hat protokol morate pretpostaviti da postoje sigurnosne okolnosti prilikom dodavanja korisnika i postavljanja lozinki.

Kad ste instalirali Fedoru instalacijska skripta tražila je da zadate lozinku za *root* korisnički račun i jedan opcionalni primarni korisnički račun. Osim toga, ne treba vam nikakvo drugo iskustvo s dodavanjem korisnika i vrlo malo trebate znati o administraciji grupa.

Administratori sustava trebaju znati:

- Kako izraditi i postaviti korisničke račune
- Kako obrisati i isključiti korisničke račune
- Poznavati potencijalne sigurnosne probleme vezane uz administriranje korisnika i postupke za njihovo rješavanje

Morate znati da korisnički računi imaju čitav niz uloga na Linux sustavima i da korisnici ne mroaju biti osobe. Postoje dvije glavne vrste korisničkih računa:

*Korisnički računi za stvarne osobe*
: Svakom je korisniku izrađen korisnički račun kojem su pridružene konfiguracijske opcije kao što su lozinka, početni direktorij i školjka koja će se izvršavati kada se korisnik prijavi. Otvaranje različitih korisničkih računa za različite korisnike omogućava postavljanje dozvola za pristupa datotekama.

*Korisnički računi za sistemske servise poput poslužitelja elektroničke pošte ili baze podataka*
: Ovi korisnički računi omogućavaju izvođenje servisa koji mogu pristupati samo onim datotekama koje im trebaju. Zbog pogrešaka u programu ili neovlaštenog upada te se servise pokušava prisiliti da rade s dijelovima sustava za koje nisu projektirani. Obično, kada se servis instalira, instalacijski proces ili administrator sustava definira korisnika ili grupu istog imena (*postfix*, *mysql*, itd.) i pridružuje mu sve datoteke i direktorije koji su mu potrebn. Servisima se ne pridružuju lozinke, početni direktoriji ili školjka budući da im ne trebaju, a uljezi bi ih mogli zloupotrijebiti.

Kao što je prije rečeno, ako čitate ovu knjigu već biste trebali znati kako dodavati korisnike, postavljati lozinke i slične stvari. Sada bismo htjeli istaknuti one stvari koji bi administrator trebao znati o korisničkim računima sa sigurnosne točke gledišta.

## Brisanje korisnika

U mnogim organizacijama fluktuacija zaposlenika je svakodnevna pojava. Tako da (osim ako ne vodite malu trgovinu sa stabilnom bazom korisnika) morati ćete naučiti kako administrirati odlazak korisnika. Mnogi samozvani administratori sustava ne shvaćaju ozbiljno administriranje korisnika. Nezadovoljni bivši djelatnici vrlo često izazivaju probleme tako što upadaju u mrežu.

Brisanje korisnika nije proces od samo jednog koraka. Trebate se pozabaviti njegovim datotekama, elektroničkom poštom, aliasima, poslovima ispisa, automatskim procesima kao što je arhiviranje podataka i stvarima koje se tiču suradnje s drugim korisnicima. Dobra je ideja najprije onemogućiti korisnički račun u */etc/passwd*. Poslije toga mogu se potražiti sve korisnikove datoteke. Nakon što su svi tragovi korisnika obrisani može se potpuno ugasiti korisnički račun (ako uklonite red iz datoteke */etc/passwd* prije nego što obrišete sve tragove teško ćete ih kasnije sve pronaći).

Kada brišete korisnički račun dobro je slijediti unaprijed definiran protokol djelovanja tako da nešto ne zaboravite. Dobro je sastaviti i pisani obrazac prema kojem će se ta operacija izvoditi.

Prva zadaća je onemogućiti korisničku lozinku čime se sprječava pristup sustavu. To možete učiniti zadavanjem sljedeće naredbe:

```
# passwd -l tadelste
```

Ponekad je potrebno samo privremeno onemogućiti korisnički račun, bez trajnog brisanja, primjerice kada korisnik ode na porodiljni dopust ili neplaćeni godišnji odmor. Koristeći dnevnik možete utvrditi je li netko pokušao neautorizirano preuzeti korisnički račun pogađajući lozinku. I u ovim situacijama je naredba *passwd –l* izuzetno korisna.

Nadalje, morate odlučiti što s korisnikovim datotekama. Ne zaboravite da korisnici mogu imati datoteke i izvan svog početnog direktorija. Naredba *find* pronalazi sve korisnikove datoteke:

```
# find / -user tadelste
[root@host2 ~]# find / -user tadelste
/home/tadelste
/home/tadelste/.zshrc
/home/tadelste/.bashrc
/home/tadelste/.bash_profile
/home/tadelste/.gtkrc
/home/tadelste/.bash_logout.......
```

Sada možete odlučiti hoćete li ove datoteke obrisati ili zadržati. Ako ih odlučite obrisati izradite sigurnosnu kopiju u slučaju da ih kasnije zatrebate.

Dobra dodatna sigurnosna mjera je korisnikovu školjku postaviti na lažnu vrijednost. Jednostavno ćete posljednje polje u datoteci *passwd* postaviti na `/bin/false`.

Ako vaša organizacija koristi SSH (najčešće putem *OpenSSH* poslužitelja) i dozvoljava udaljenu RSA ili DSA provjeru autentičnosti, korisnik može imati pristup sustavu čak iako mu je lozinka onemogućena. Ovo se događa jer SSH koristi odvojene ključeve.

Primjerice, nakon što ste onemogućili lozinku Toma Adelsteina, on preko nekog drugog računala može zadati naredbu:

```
$ ssh -f -N -L8000:intranet.vasatvrtka.com:80 moja.domena.com
```

Ovime se promet na ulazu 80 (ulaz koji Web poslužitelj obično osluškuje) prosljeđuje na vaš unutarnji poslužitelj.

Očito, ako vaš sustav pruža SSH, trebate ukloniti autorizirane ključeve iz odgovarajućih direktorija *(~tadelste/.ssh* ili *.~tadelste/.ssh2*) da biste onemogućili korisniku pristup na opisani način:

```
$ cd .ssh
:~/.ssh$ ls
authorized_keys known_hosts
:~/.ssh$ rm authorized_keys
:~/.ssh$ ls
known_hosts
:~/.ssh$
```

Isto tako potražite datoteke *.shosts* i *.rhosts* u korisnikovom početnom direktoriju (primjerice *~tadelste/.shosts* i *~tadelste/.rhost*).

Također provjerite postoji li proces koji se izvršava pod korisnikovim identitetom. Takvi procesi također mogu poslužiti kao zaobilazni put za neovlašteni upad u mrežu. Sljedeća će vam naredba pokazati ima li korisnik aktivnih procesa:

```
# ps aux |grep -i ^tadelste
```

Neke druge stvari koje administrator sustava može provjeriti vezano za korisnika koji napušta organizaciju uključuju:

- Može li korisnik izvršavati CGI skripte iz svog početnog direktorija ili na tvrtkinim Web poslužiteljima.
- Postoje li datoteke za prosljeđivanje elektroničke pošte poput *~tadelste/.forward*? Korisnik ih može koristiti da bi poslao poštu na svoj račun što može uzrokovati izvršavanje programa na sustavu na koji korisnik nema pristup.

## Zapečaćivanje početnog direktorija

Vrlo često uprava želi zadržati informacije iz početnog direktorija korisnika koji odlazi. Sve poruke elektroničke pošte i drugi dokumenti u korisničkom direktoriju u principu pripadaju tvrtki. U slučajevima kad nezadovoljni bivši zaposlenik želi tužiti tvrtku, tvrtkini odvjetnici će vjerojatno htjeti pregledati te datoteke. Većina analitičara smatra da je čuvanje korisničkih početnih direktorija dobra praksa.

Sadržaj korisnikova početnog direktorija možete spremiti tako da mu promijenite ime. Jednostavno zadajte naredbu:

```
# mv /home/tadelste /home/tadelste.locked
```

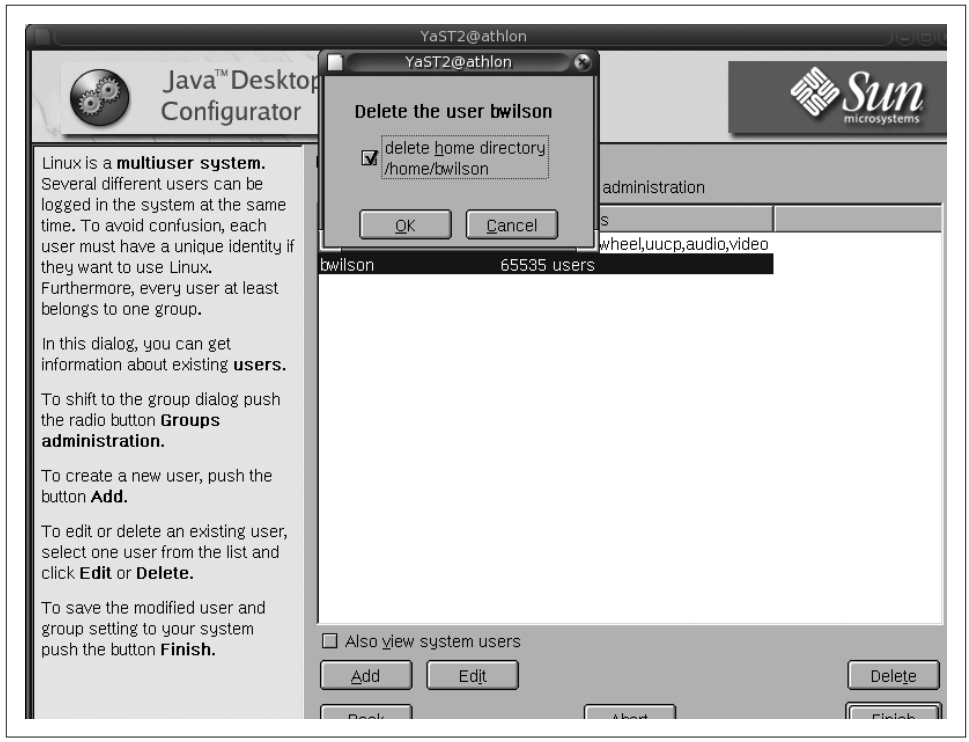
Ovo će spriječiti bivšeg zaposlenika da se prijavi na sustav i zloupotrijebi konfiguracijske datoteke, primjerice *.forward*, kao što je objašnjeno u prethodnom dijelu. Sadržaj ostaje netaknut u slučaju da vam kasnije zatreba.

## Grafički alati za administriranje korisnika

Kako se Linux sve više probijao na tržište početkom ovog desetljeća, tvrtke kao što su Sun Microsystems, Novell, Computer Associates, HP i IBM počele su prilagođavati svoje administrativne alate za Red Hat, SUSE i druge Linux platforme. Dodatno, administrativni alati koji su dolazili s Linuxovim distribucijama počeli su sazrijevati i u broju funkcija i korisnosti.

Budući da ste stekli neka znanja o naredbama i procesima potrebnim za uklanjanje korisničkih računa, bilo bi dobro napraviti alat za njihovo jednostavnije korištenje. Općenito su naredbe u takvim alatima manje fleksibilne nego kad se koriste iz odzivnika.

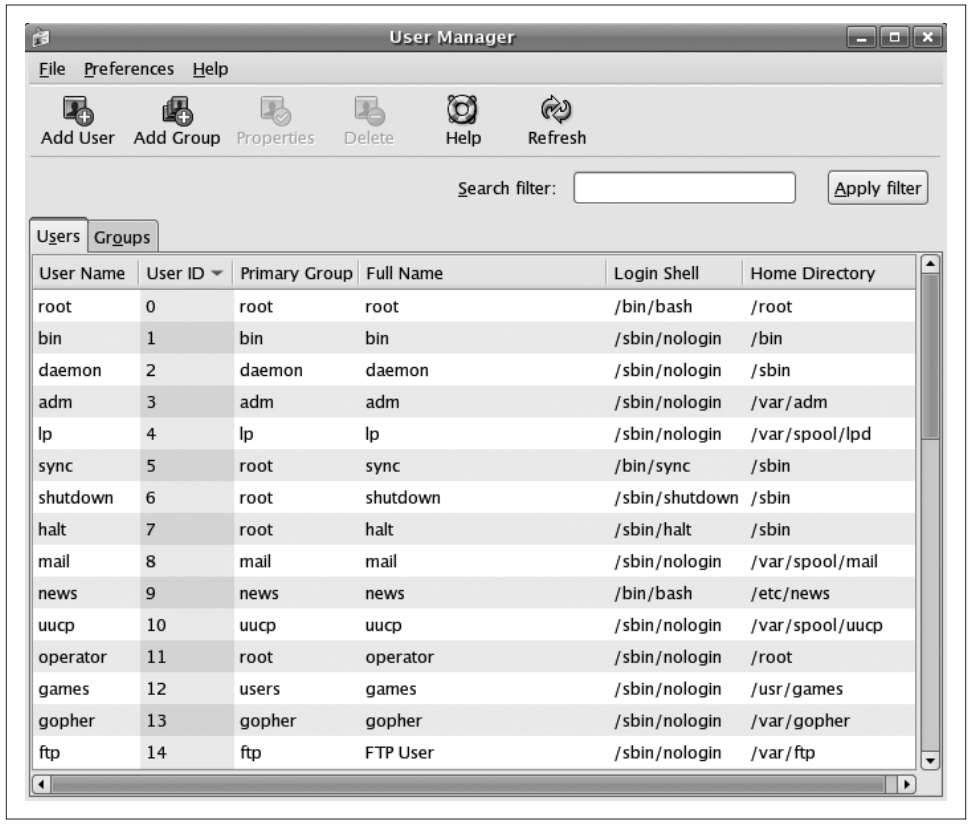
Pogledajmo primjer jednog takvog alata, YaST2, koji je originalno razvijen za SUSE Linux. Sučelje Sunovog Java Desktop Configuratora izgleda kao na slici 8-14. Opisi funkcija koje možete koristiti s ovim alatom nalaze se na lijevoj paleti.




*Slika 8-14. JDS User Manager tvrtke Sun Microsystems*

Primijetite da dijaloški okvir na vrhu pita da li želite obrisati direktorij */home/tadelste*. Kao što smo prije razjasnili, tvrtka želi sačuvati dokumente bivših zaposlenika. U ovom slučaju, alat s grafičkim korisničkim sučeljem daje vam dvije opcije: da ga obrišete ili da ga sačuvate. Ne pruža vam opciju promjene imena direktorija, što je najsigurniji i najjednostavniji način za njegovo čuvanje.

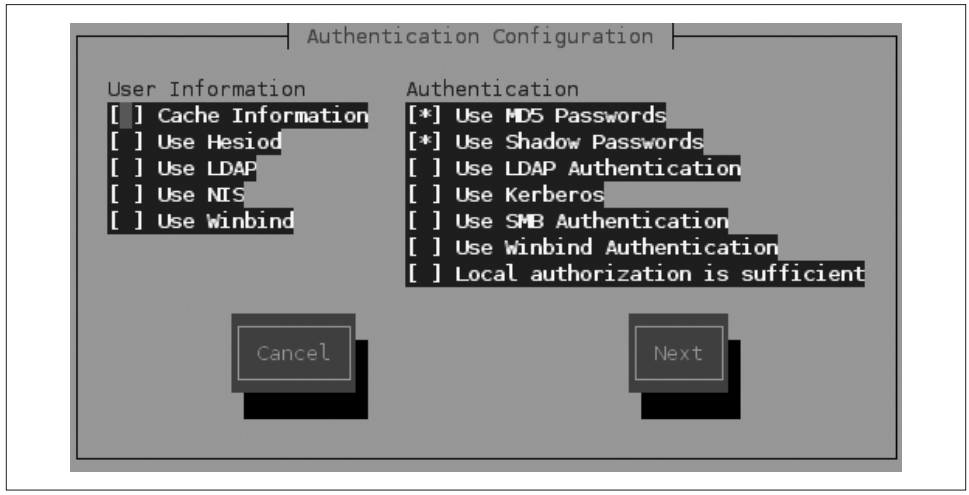
Na slici 8-15 možete vidjeti još jedan primjer iz Fedorinog sustava.




*Slika 8-15. Fedora User Manager – grafički alat za administriranje korisnika*

S Fedorinim grafičkim alatom za administriranje korisnika možete obaviti iste osnovne funkcije kao što su one izložene na slici 8-14. Opet, nemate sve opcije koje vam trebaju za propisno upravljanje korisničkim računima korisnika u odlasku.

Premda tehnički to nije program za administriranje korisnika, Fedora nudi još jedan alat za konfiguriranje velikog broja servisa za upravljanje korisnicima. Pogledajte sliku 8-16, grafički alat koji pruža Fedora kada zadate naredbu *setup* u školjci.




*Slika 8-16. Rad Hatov Authentication Configurator*

Ovo je još jedan primjer brojnih načina koje Linux pruža za upravljanje korisničkim računima. Ne zahtjeva X Windows System.

POGLAVLJE 9

# Virtualizacija u modernom poduzeću

U ovom poglavlju govorimo o području koje je izazvalo eksplozivan rast potražnje za Linux sistem administratorima. Linux virtualizacija nalazi se u samom srcu današnjih trendova – konsolidaciji podatkovnih centara, izradi računalnih sustava vrhunskih performansi, kontinuitetu poslovanja i upravljanju ukupnim poslovnim obavezama. Poduzeća očekuju stvarna sniženja troškova kroz Linux virtualizaciju a analitičari primjećuju da ta tehnologija mijenja poslovno okruženje.

Virtualizacija je koncept koji je stekao popularnost zahvaljujući uspješnoj kompaniji Vmware *(http://www.vmware.com)* i projektu otvorenog koda Xen *(http://www.cl.cam.ac.uk/research/srg/netos/xen)*. On se odnosi na jednu hardversku jedinicu koja izvodi višestruke jezgre (ponekad sve jednake, a ponekad iz potpuno različitih operativnih sustava) povrh nižeg sloja softvera koji upravlja njihovim pristupom hardveru. Svaka jezgra, zvan *gost*, ponaša se kao da ima cijeli procesor za sebe.

Različiti gosti su izolirani jedan od drugoga mnogo više nego što su procesi izolirani unutar jednog operativnog sustava. Ta izolacija omogućuje sigurnost i robusnost jer zakazivanje ili poremećaj u jednom gostu ne utječe na ostale. Virtualizacijski sloj obavlja mnoge funkcije operativnog sustava, upravljajući pristupom procesorskom vremenu, uređajima i memoriji za svakog gosta.

U vrijeme pisanja ove knjige Linux razvojni inžinjeri radili su na novom sustavu nazvanom Kernel-based Virtual Machine (KVM), koji će biti dio jezgre.

## Zašto je virtualizacija popularna

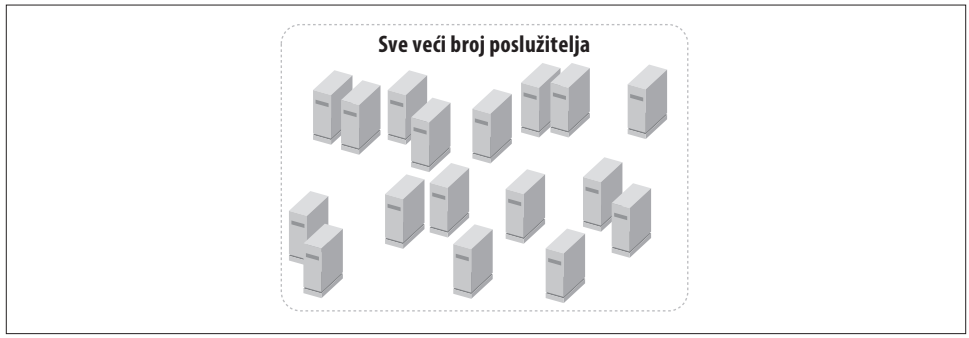
Da biste razumjeli tko koristi virtualizaciju, kao i okruženja u kojima je ona od velike vrijednosti, trebate razumjeti ponešto o sadašnjim poslovnim potrebama. Ovaj odjeljak pruža tu osnovu, prije nego što objasnimo kako funkcionira Linux virtualizacija.

Cijelo područje informatičke tehnologije eksponencijalno je naraslo od pojave uobičajenih distribuiranih sustava datoteka. Organizacije su promatrale kako im se infrastuktura

proširuje iz godine u godinu. Mnogi pripisuju ovaj rast stalnom usavršavanju računalnih komponenti i softvera. No, to nije cijela slika.

Računalna tehnologija je evoluirala od usmjerenosti na upravljanje transakcijama do objedinjavanja poslovnih procesa. Neka poduzeća specijaliziraju se za upravljanje ljudskim resursima, druga za financije i računovodstvo i mnoštvo drugih za proizvodnju i upravljanje lancima dostave. Ta je specijalizacija stvorila područja dominacije u centrima podataka i među IT osobljem.

Tradicionalne mreže su sad sposobne primiti i obraditi više različitih vrsta transakcija nego ikad prije, što je stvorilo potrebu za povećanjem računalne snage i posljedično, za više prostora za pohranu. Porastao je i broj mjesta i načina na koja pohranjujemo podatke, što je samo po sebi pokrenulo povećanje broja poslužitelja (pogledajte sliku 9-1).




*Slika 9-1. Bujanje farme poslužitelja sa po jednim operativnim sustavom po računalu*

Sad dodajte još jedan sastojak ovoj mješavini: specijalizirane aplikacije za područja kao što su računovodstvo i financije skoro uvijek se izvode na odvojenim, pouzdanim poslužiteljima sa redundantnim hardverom u svrhu osiguravanja kontinuiteta poslovanja. Ova kombinacija faktora transformirala je IT okruženje u zbrkanu masu izoliranih, specijaliziranih i nedovoljno iskorištenih poslužitelja.

Povrh svega toga dolazi i rastući teret udovoljavanja propisima, što izaziva ponovni rast troškova: morate povećati kapacitet za pohranjivanje i pronalaženje pohranjenih podataka a u mnogim slučajevima očekuje se da ih čuvate i do 25 godina.

Razmotrite što to znači. Vaši nasljednici možda neće imati na raspolaganju tehnologiju za produciranje dokumenata koje bi revizor ili odvjetnik mogao tražiti za jedno desetljeće, a još manje za četvrt stoljeća.

Pogledajmo još jednom rezultate računalnog razvoja. Imamo:

- Specijalizirane poslužitelje i aplikacije (često znane kao „silosi“) s nedovoljno iskorištenim kapacitetom.
- Dodatni porast troškova zbog kompleksnosti softvera i potrebe za upravljanjem stalno rastućim količinama podataka.

- Potrebu za specijalizacijom osoblja u funkcionalnim područjima u kojima ćete naići na nedostatak dokumetacije i visoku razinu smjena osoblja.
- Potrebu za obučavanjem i podrškom korisnicima i administratorima i održavanjem softvera aktualnim.

Sad biste mogli razumjeti zašto je virtualizacija postala popularna u poduzećima i jedno od nekoliko područja u kojima tehnologija može promijeniti poslovno okruženje. Koristeći virtualne snimke lako možete komprimirati podatke zajedno sa svim programima, konfiguracijskim postavkama, bibliotekama operativnog sustava i ostalim metapodacima koji čine jedan cijeli sustav. Ponovno postavljanje nekog od snimaka vraća sustav u isto stanje u kojem je radio u određeno vrijeme te tako olakšava reproduciranje dokumenata. Virtualizacija ima sljedeće prednosti:

- Zamjenjuje mnogobrojne neučinkovite sustave manjim brojem bolje iskorištenih sustava.
- Pojednostavljuje administriranje, jer su odvojene jezgre na kojima se izvodi po jedan program sigurnije i jednostavnije za upravljanje nego jedna jezgra koja izvodi mnogo aplikacija. To također održava okruženje u kojem su dokumenti izrađeni, kako bi se udovoljilo propisima.
- Reducirana količina hardvera i pojednostavljivanje omogućuju smanjenje broja osoblja.
- Virtualizacija može pomoći preokrenuti trend bujanja poslužitelja.

## Sustavi visokih performansi

Linux je postao preferirani operativni sustav za virtualne strojeve zbog svoje sposobnosti da upravlja golemim klasterima i mrežama računala. Trebalo je nešto vremena da najveći trgovci hardverom uhvate korak, ali kad su to postigli, ostvarili su velike zarade. Tijekom nekoliko godina Linux je uživao podršku kompanija voljnih da doprinesu osobljem i naprednom tehnologijom u njegovim razvojnim nastojanjima. Među te tvrtke spadaju IBM, Intel, AMD, HP, Novell, Red Hat, Unisys, Fujitsu i desetak drugih.

Na primjer, IBM je trebao pomoćni operativni sustav za svoju OpenPower inicijativu. Odjednom, Linux je radio na Big Blue-ovom virtualizacijskom stroju u obliku hipervizora otvorenog koda i pratećih tehnologija. IBM-ov stroj dozvoljava Linuxu da stvara particije, upravlja njima te im dinamički dodjeljuje ulazno/izlazne resurse.

Nakon toga su razvojni inžinjeri Linux jezgre objavili svoju novu simultano višenitnu i hipernitnu tehnologiju. Linux je sada omogućavao istovremeno izvođenje dvije niti na istom procesoru, a to je esencijalna tehnologiju za ulogu domaćina gostujućim operativnim sustavima. Tako, VMware radi dobro povrh Linuxa i pruža virtualizacijski sloj za ostale instance Linuxa ili drugih operativnih sustava. User-mode Linux (UML) je još jedan primjer kako Linux tvori osnovu za virtualizaciju.

Verzija 2.6 Linux jezgre dobro se slaže s IBM-ovom SMT tehnologijom. Prije te inačice jegre, Linux nije imao dobro razvijene sustave za raspoređivanje niti i posredovanje. Inačica jezgre 2.6 riješila je taj problem i uvelike proširila broj procesora na kojima jezgra može raditi.

To je značajno iz dva razloga. Prvo, da bi bio domaćin virtualnim strojevima, Linux se trebao dobro izvršavati i upravljati hardverom. Drugo, kao gost koji je razdvojen od hardvera, treba imati dobre performanse i sposobnost da obrađuje različite procese kao domaćin. Linux je danas i jedno i drugo – odličan domaćin i odličan gostujući OS. Upravlja hardverom i virtualnim particioniranjem te dobro radi u gostujućim particijama zahvaljujući HP-u i IBM-u.

Ako ste se ikad zapitali zašto su se kompanije poput XenSource i Virtual Iron odjednom pojavile niotkuda, sada znate: to je zbog doprinosa hipervizora otvorenog koda. Kao i trgovci hardverom koji su shvatili da Linux može unaprijediti prodaju PC komponenti i komponenti za podatkovne centre i trgovci softverom ukrcali su se na taj „vlak“. Čak je i Microsoft s vremenom shvatio da treba ući u Linux igru, doprinoseći objema kompanijama, XenSource i Virtual Iron.

## Poslovni kontinuitet i upravljanje ukupnim poslovnim obavezama

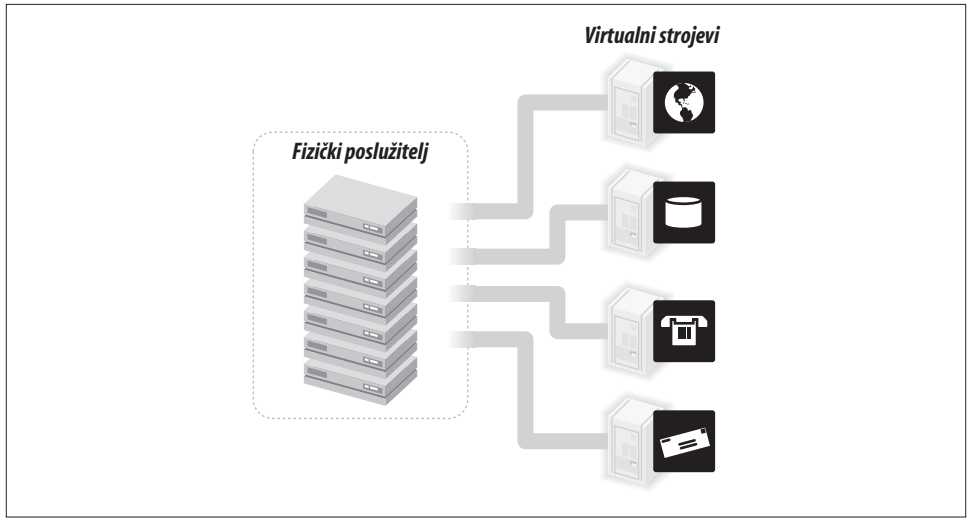
Čak i u malim razmjerima, vaša orgaizacija će imati koristi od razdvajanja poslužitelja elektroničke pošte, DNS i Web poslužitelja te imenika, prolaza i baza podataka. Smještanje svake od ovih usluga na zaseban poslužitelj osigurava da, ako jedan poslužitelj otkaže, vaša cjelokupna infrastruktura ne kolabira. Ali, razdvajanje usluga na poseban hardver zahtijeva puno vremena, prostora i novca. Također, trebate izraditi rezervne kopije i kasnije vratiti podatke, osigurati se od katastrofa i odabrati najbolji hardver za taj posao.

Sa Linux virtualizacijom možete podijeliti jedan fizički poslužitelj na nekoliko virtualnih. Svaki virtualni poslužitelj administratorima sustava izgleda kao fizički. Možete izraditi zasebnu poslužiteljsku instancu za svaku uslugu koju želite ponuditi: elektroničku poštu, DNS, Web i tako dalje. Ako jedan iznevjeri, ostali neće pretrpjeti štetu.

Dijeljenje fizičkog domaćina također omogućuje da izradite različite konfiguracije za svaki virtualni poslužitelj na istom hardveru. U jednom okruženju, na primjer, izradili smo manje virtualne strojeve za DNS poslužitelje i veće za elektroničku poštu i Web. To nam je omogućilo da rasporedimo opterećenje bolje iskoristimo hardver. Slika 9-2 ilustrira što se sve može postići sa samo jednim fizičkim poslužiteljem.

## Brzo instaliranje

Najprije smo postigli virtualizaciju na našoj mreži instaliranjem minimalne konfiguracije Debiana na virtualni stroj. Kad smo je podesili u skladu s našim potrebama, komprimirali smo je i snimili na CD-R medij. Zatim smo postavili dodatne virtualne strojeve koristeći VMware s različitim konfiguracijama i kopirali komprimirane slike u svaki direktorij koji smo zadali kao virtualni stroj.




*Slika 9-2. Dijeljenje jednog fizičkog poslužitelja na više virtualnih strojeva*

Svaki virtualni stroj živi u jednom direktoriju. Na primjer, naš glavni direktorij, */var/lib/vmware/Virtual Machines*, sadrži nekoliko poddirektorija, poput *debian-31r0a-i386-netinst-kernel2.6*. Jednostavno smo komprimirali taj poddirektorij i koristili ga za instalacije u druge poddirektorije sa malo drugačijim imenima.

Također, postavili smo Xen virtualne strojeve koristeći minimalnu instalaciju Fedore. Zatim smo dodali komponente potrebne za svaku uslugu koju želimo pružati. Na primjer, naš primarni DNS poslužitelj izvodi se na Xen virtualnom stroju, dok se Web posužitelj i poslužitelj elektroničke pošte izvode na odvojenim instancama VMwarea.

Nakon što smo pokrenuli posužitelj (recimo, elektroničke pošte), napravili smo komprimiranu kopiju i snimili je na CD-R. Regularno i sistematično snimamo rezervne kopije svih virtualnih poslužitelja na medije kao što su CD i DVD. Isprobali smo i premještanje slika na različite distribucije Linuxa i radile su sasvim isto kao i prije.

## Kako virtualizacija pomaže

Što smo postigli virtualizacijom? Prvo, uklonili smo značajan broj fizičkih poslužitelja. Postavili smo odabrani operativni sustav kao sliku, pa je kroz proces instalacije trebalo proći samo jednom. Zatim smo izradili virtualne strojeve na rezervnom hardveru i sistematčno kopirali virtualne slike da bismo osigurali trenutačni povrat podataka u slučaju blokiranja sustava.

Virtualizacija dobro funkcionira u malim tvrtkama jer im omogućava da izgrade infrastrukturu koristeći besplatan softver. Zamislite kolika je ušteda samo na licencama! Sada zamislite na koje bi sve načine velike tvrtke mogle primijeniti Linux.

Sad biste već mogli biti nestrpljivi da vidite kako to sve funkcionira. Pa, prođimo kroz proces instaliranja i konfiguriranja Xena i VMwarea i pokažimo kako virtualizirati jednu mrežu poslužitelja.

## Instaliranje Xena na Fedoru 5

U ovom dijelu poglavlja pokazat ćemo kako instalirati Xen na samo jedan stroj kako bi upravljao s dva operativna sustava. Kako se Xen bude sve bolje integrirao sa standardnim Linux distribucijama instalacija će postajati lakša. Ali, sada je ponešto potrebno obaviti ručno.

Koristimo Fedoru Core 5 (FC5) kao operativni sustav domaćin za Xen jer podržava Xen 3.0 bez dodatnih podešavanja. Zapitajmo *yum* (program za upravljanje paketima sličan Debianovom *apt-get* ili Red Het-ovom *up2date*) o Xenu:

```
# yum info xen
Loading "installonlyn" plugin
Setting up repositories
core                                                      [1/3]
updates                                                   [2/3]
extras                                                    [3/3]
Reading repository metadata in from local files
Available Packages
Name   : xen
Arch   : i386
Version: 3.0.2
Release: 3.FC5
Size   : 1.4 M
Repo   : updates
Summary: Xen is a virtual machine monitor
Description:
 This package contains the Xen hypervisor and Xen tools, needed to
run virtual machines on x86 systems, together with the kernel-xen*
packages.  Information on how to use Xen can be found at the Xen
project pages.

Virtualisation can be used to run multiple versions or multiple
Linux distributions on one system, or to test untrusted applications
in a sandboxed environment.  Note that the Xen technology is still
in development, and this RPM has received extremely little testing.
Don't be surprised if this RPM eats your data, drinks your coffee
or makes fun of you in front of your friends.
```

To zvuči ohrabrujuće. Hajde da to isprobamo, samo prvo provjerimo neke preduvjete:

- Sustav mora imati bar 256 MB RAM-a.
- Program za učitavanje i podizanje sustava mora biti *grub*.
- `SELINUX` mora biti `disabled` ili `permissive` ali ne `enforcing`.

Pokrenite program *system-config-securitylevel* ili uredite */etc/selinux/config* da izgleda kako slijedi:

```
# This file controls the state of SELinux on the system.
# SELINUX= can take one of these three values:
#       enforcing - SELinux security policy is enforced.
#       permissive - SELinux prints warnings instead of enforcing.
#       disabled - SELinux is fully disabled.
SELINUX=Disabled
# SELINUXTYPE= type of policy in use. Possible values are:
#       targeted - Only targeted network daemons are protected.
#       strict - Full SELinux protection.
SELINUXTYPE=targeted
```

Ako ste promijenili vrijednost `SELINUX` tako da nije `enforcing` trebat ćete ponovno pokrenuti Fedoru prije nego što nastavite dalje.

Sljedeća naredba će instalirati Xen hipervizor – modificiranu Fedorinu jezgru pod imenom *domain 0* i razne pomoćne programe:

```
# yum install kernel-xen0
```

Potreba za posebnom modificiranom Linux jezgrom bi mogla nestati u budućnosti, kada Intel i AMD uvedu podršku za virtualizaciju u svoje čipove. Očekuje se i da Windows Vista podržava virtualizaciju na procesorskoj razini.

Ovo dodaje *xen0* kao prvu moguću jezgru u */boot/grub/grub.conf*, ali ne i kao podrazumijevani izbor:

```
# grub.conf generated by anaconda
#
# Note that you do not have to rerun grub after making changes to this file
# NOTICE:  You have a /boot partition.  This means that
#          all kernel and initrd paths are relative to /boot/, eg.
#          root (hd0,0)
#          kernel /vmlinuz-version ro root=/dev/VolGroup00/LogVol00
#          initrd /initrd-version.img
#boot=/dev/hda
default=1
timeout=5
splashimage=(hd0,0)/grub/splash.xpm.gz
hiddenmenu
title Fedora Core (2.6.17-1.2157_FC5xen0)
        root (hd0,0)
        kernel /xen.gz-2.6.17-1.2157_FC5
        module /vmlinuz-2.6.17-1.2157_FC5xen0 ro root=/dev/VolGroup00/LogVol00
        module /initrd-2.6.17-1.2157_FC5xen0.img
title Fedora Core (2.6.17-1.2157_FC5)
        root (hd0,0)
        kernel /vmlinuz-2.6.17-1.2157_FC5 ro root=/dev/VolGroup00/LogVol00
        initrd /initrd-2.6.17-1.2157_FC5.img
title Fedora Core (2.6.15-1.2054_FC5)
        root (hd0,0)
```

```
        kernel /vmlinuz-2.6.15-1.2054_FC5 ro root=/dev/VolGroup00/LogVol00
        initrd /initrd-2.6.15-1.2054_FC5.img
default=0
```

Da biste postavili Xen jezgru kao podrazumijevani izbor, promijenite ovaj red:

```
default=1
```

u:

```
default=0
```

Sad možete ponovno pokrenuti sustav. Xen bi se trebao automatski pokrenuti, ali ipak provjerimo:

```
# /usr/sbin/xm list
Name                              ID Mem(MiB) VCPUs State  Time(s)
Domain-0                           0      880     1 r-----    20.5
```

Rezultat bi trebao pokazivati da se `Domain-0` izvršava. Domain 0 kontrolira sve gostujuće operativne sustave koji se izvršavaju na procesoru, slično kao što jezgra kontrolira procese u nekom operativnom sustavu.

## Instaliranje gostujućeg operativnog sustava

Xen sada kontrolira procesor ali trebate dodati barem jedan operativni sustav kao gosta. Počet ćemo s instaliranjem Fedora Core 5 gosta, jer to olakšava posao, a zatim ćemo ponuditi neke savjete za druge varijante Linuxa.

### Fedora Core 5

Fedora Core 5 ima skriptu za instalaciju Xen gosta koja pojednostavljuje proces, premda instalira samo FC5 goste. Skripta očekuje pristup FC5 instalacijskom stablu preko FTP-a, Weba ili NFS-a. Iz nekog razloga ne može se zadati direktorij ili datoteka. Mi ćemo koristiti naš FC5 instalacijski DVD i poslužiti ga preko Apachea:

```
# mkdir /var/www/html/dvd
# mount -t iso9660 /dev/dvd /var/www/html/dvd
# apachectl start
```

Sad ćemo pokrenuti instalacijsku skriptu i odgovoriti na njena pitanja:

```
# xenguest-install.py
What is the name of your virtual machine? guest1
How much RAM should be allocated (in megabytes)? 256
What would you like to use as the disk (path)? /xenguest
What is the install location? http://127.0.0.1/dvd
```

U ovom trenutku počinje instalacija FC5. Odaberite između tekstualnog režima i grafičkog režima (ako se izvršava X) preko *vnc*. Ako odaberete tekstualni režim, bit ćete spojeni na konzolu. Nastavite kao što biste to učinili i za Fedora ili Red Hat instalaciju. Na zaslonu za IP adresu zadajte gostu adresu različitu od adrese domaćina ili koristite DHCP (ako ste zadali `dhcp="dhcp"` u Xen konfiguracijskoj datoteci, što je objašnjeno u sljedećem odjeljku). Zadnji zaslon će zatražiti da ponovno pokrenete sustav. Odjavite DVD i izvadite ga iz čitača. Ponovno ćete pokrenuti samo gostujući operativni sustav ali ne i Xen ili računalo.

Xen ne pokreće automatski gostujući operativni sustav. Trebate utipkati sljedeću naredbu:

```
# xm create guest1
```

Sada imate dva operativna sustava (*host 1* i *guest 1*) koji rade nezavisno i žive u harmoniji, svaki sa svojim sustavom datoteka, mrežnim vezama i memorijom. Da biste dokazali da se oba poslužitelja izvršavaju, zadajte sljedeće naredbe:

```
# xm list
Name                                ID Mem(MiB) VCPUs State  Time(s)
Domain-0                             0      128     1 r-----   686.0
guest1                               3      256     1 -b----    14.5
# xentop
xentop - 21:04:38   Xen 3.0-unstable
2 domains: 1 running, 1 blocked, 0 paused, 0 crashed, 0 dying, 0 shutdown
Mem: 982332k total, 414900k used, 567432k free    CPUs: 1 @ 2532MHz
NAME STATE CPU(sec)CPU(%) MEM(k)MEM(%) MAXMEM(k)MAXMEM(%)VCPUSNETS
NETTX(k) NETRX(k) SSID
Domain-0-----r 686 0.3 131144 13.4 nolimit n/a 1 8
1488528    80298    0
guest1--b--- 14 0.1 261996 26.7 262144 26.7 1 1
129      131    0
```

Da biste automatski pokretali Xen domene koristite ove naredbe:

```
# /sbin/chkconfig --level 345 xendomains on
# /sbin/service xendomains start
```

### Drugi gosti

Ako želite neki drugi gostujući operativni sustav, različit od FC5, trebat ćete urediti konfiguracijsku datoteku Xen gosta. To je tekstualna datoteka (zapravo Python skripta) u */etc/xen* direktoriju. *xmexample1* i *xmexample2* su komentirani primjeri datoteka. Za punu sintaksu datoteke pogledajte:

```
# man xmdomain.cfg
```

Kad smo pokrenuli *xenguest-install.py* u prethodnom odjeljku, to je generiralo konfiguraciju Xen gosta */etc/xen/guest1* s nekoliko dodatnih redova:

```
# Automatically generated Xen config file
name = "guest1"
memory = "256"
disk = [ 'file:/xenguest,xvda,w' ]
vif = [ 'mac=00:16:3e:63:c7:76' ]
uuid = "bc2c1684-c057-99ea-962b-de44a038bbda"
bootloader="/usr/bin/pygrub"

on_reboot   = 'restart'
on_crash    = 'restart'
```

Ovo sadrži neke, ali ne i sve, upute koje gost treba. Minimalna konfiguracijska datoteka gosta sadržavala bi nešto poput ovoga:

1. Jedinstveno ime domene gosta:

   ```
   name="vm01"
   ```

2. Putanja slike jezgre prilagođene za Xen za gostujuću domenu:

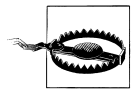
   ```
   kernel="/boot/vmlinuz-2.6.12.6-xenU"
   ```

3. Korijenski uređaj za gostujuću domenu:

   ```
   root="/dev/hda1"
   ```

4. Inicijalna alokacija memorije za gosta, u megabajtima:

   ```
   memory=128
   ```



Ukupna količina memorije dodijeljena svim Xen gostima ne smije premašiti ukupnu memoriju sustava minus 64 MB za sam Xen.

5. Prostor na disku za gostujuću domenu. Definira se u jednoj ili više *stanci* navedenih unutar jednostrukih ili dvostrukih navodnika:

   ```
   disk= [ 'stanca1', 'stanca2' ]
   ```

   Stanca se sastoji od niza tri parametra (`'host_dev`, `guest_dev`, `mode'`). `host_dev` je područje za pohranu domene, kako je vidi domaćin. To može biti:

   `file:`*`putanja`*
   : Slika *loopback* datoteke (lokalna datoteka koju Xen tretira kao sustav datoteka). Ona se stvara kad pokrenete *xm create* ili *xen-create-image* program.

   `phy:`*`uređaj`*
   : Fizički uređaj.

   `guest_dev` je fizički uređaj kako ga vidi gostujuća domena. `mode` može biti `r` za samo za čitanje (read-only) ili `w` za čitanje i pisanje. Primjer direktive `disk` za dva gosta bio bi:

   ```
   disk=['file:/vserver/images/vm01.img, hda1, w', 'file:/vserver/images/vm01-swap.
   img, hda2, w']
   ```

6. Informacije o mrežnom sučelju u `vif` direktivi. Ova direktiva može sadržavati jednu stancu za svaki mrežni uređaj. Podrazumijevana mreža se zadaje sa:

   ```
   vif=[ '' ]
   ```

   `dhcp` direktiva zadaje hoće li se koristiti DHCP ili će informacije o sučelju biti izravno upisane. Sljedeće zadaje upotrebu DHCP-a:

   ```
   dhcp="dhcp"
   ```

   Ukoliko `dhcp` direktiva nedostaje ili je postavljena na `"off"`, podatke o mreži morate zadati statički, kao što činite kad konfigurirate sustav:

   ```
   ip="192.168.0.101"
   netmask="255.255.255.0"
   gateway="192.168.0.1"
   hostname="vm01.example.com"
   ```

Stranica sa uputama (manpage) za *xm* daje sljedeći primjer gminimalnog osta, sa slikom loopback datoteke na domaćinu koja se pojavljuje kao korijenski uređaj na gostu:

```
kernel = "/boot/vmlinuz-2.6-xenU"
memory = 128
name = "MyLinux"
root = "/dev/hda1 ro"
disk = [ "file:/var/xen/mylinux.img,hda1,w" ]
```

Kad imate konfiguracijsku datoteku za gosta, kreirajte Xen gosta ovom naredbom:

```
# xm create -c ime_gosta
```

gdje *ime_gosta* može biti cijela putanja ili relativno ime datoteke (u kojem slučaju je Xen smješta u */etc/xen/*ime_gosta). Xen će kreirati gostujuću domenu i pokušati je pokrenuti iz zadane datoteke ili uređaja. *-c* opcija priključuje konzolu domeni kad se pokrene, pa možete odgovoriti na pitanja o instalaciji koja se pojave.

# Instaliranje Wmwarea

Tvrtka VMware besplatno nudi svoj poslužitelj sa otvorenim izvornim kodom. Možete ga naći na *http://www.vmware.com/products/server*. Smatramo da je robustan i istovremeno jednostavan za korištenje. O VMwareovim inicijativama otvorenog i zajedničkog koda možete čitati na njihovoj Web lokaciji.

Kao što smo ranije spomenuli, projekti kao što su XenSource i Virtual Iron iskoristili su prednosti podrške IBM-ovoj hipervizorskoj tehnologiji koju pruža Linux jezgra. Pod pritiskom Xena, VMware je također dao svoj doprinos razvojnim inžinjerima jezgre, shvaćajući da će VMware bolje raditi na Linuxu ako malo pomogne u razvoju jezgre.

Dok smo izvodili Xen koristeći Fedora Core 5, odlučili smo instalirati VMware na Ubuntu domaćinskom poslužitelju i koristiti Debian kao gostujući operativni sustav. Također smo upravljali udaljenim VMware instancama sa Ubuntu radne površine koristeći VMware konzolu. Kasnije smo instalirali FC5 pod VMware virtualnim strojem.

Preuzeli smo s interneta *Vmware-server-1.0.1-29996.tar.gz* i raspakirali ga u instalacijski direktorij pod imenom *vmware-server-distrib*. Unutar direktorija pronašli *smo vmware-install.pl* i pokrenuli ga naredbom *./vmware-install.pl*. Ubrzo nakon toga instalacijski program se pokrenuo i prikazao sljedeće poruke:

```
Creating a new installer database using the tar3 format.

Installing the content of the package.

In which directory do you want to install the binary files?
[/usr/bin]
```

Instalacija VMware poslužitelja počinje s nekoliko sličnih pitanja, zavisno o tome što instalacijska skripta „otkrije“ u vašem operativnom sustavu i rasporedu datoteka.

Za vrijeme instalacijskog procesa skripta traži da prihvatite VMware licencu. Trebali biste je pročitati prije nego što je prihvatite. Nakon što se složite s licencom, VMware verificira da su program za prevođenje i datoteke zaglavlja na vašem sustavu međusobno kompatibilne i gradi binarne datoteke koristeći program za prevođenje. Vidjet ćete poruke poput:

```
The path "/usr/lib/vmware" does not exist currently. This program is going
to create it, including needed parent directories. Is this what you want?
[yes]
```

Također, vidjet ćete kompilacije koda kao u ovom primjeru:

```
make[1]: Entering directory '/usr/src/linux-headers-2.6.15-26-k7'
  CC [M]  /tmp/vmware-config0/vmnet-only/driver.o
  CC [M]  /tmp/vmware-config0/vmnet-only/hub.o
  CC [M]  /tmp/vmware-config0/vmnet-only/userif.o
  CC [M]  /tmp/vmware-config0/vmnet-only/netif.o
  CC [M]  /tmp/vmware-config0/vmnet-only/bridge.o
  CC [M]  /tmp/vmware-config0/vmnet-only/procfs.o
  CC [M]  /tmp/vmware-config0/vmnet-only/smac_compat.o
  SHIPPED /tmp/vmware-config0/vmnet-only/smac_linux.x386.o
  LD [M]  /tmp/vmware-config0/vmnet-only/vmnet.o
  Building modules, stage 2.
  MODPOST
```

Pri kraju instalacije skripta će vas obavijestiti da je instalacija koda završena i ponuditi vam naredbu koju možete koristiti ako ikad poželite deinstalirati taj poslužitelj:

```
The installation of VMware Server 1.0.1 build-29996 for Linux completed
successfully. You can decide to remove this software from your system at any
time by invoking the following command: "/usr/bin/vmware-uninstall.pl".
```

Instalacijska skripta će također tražiti da zadate konfiguracijsku naredbu:

```
Before running VMware Server for the first time, you need to configure it by
invoking the following command: "/usr/bin/vmware-config.pl". Do you want
this program to invoke the command for you now? [yes]
```

Kako instalacijski proces završava, vidjet ćete sljedeće poruke:

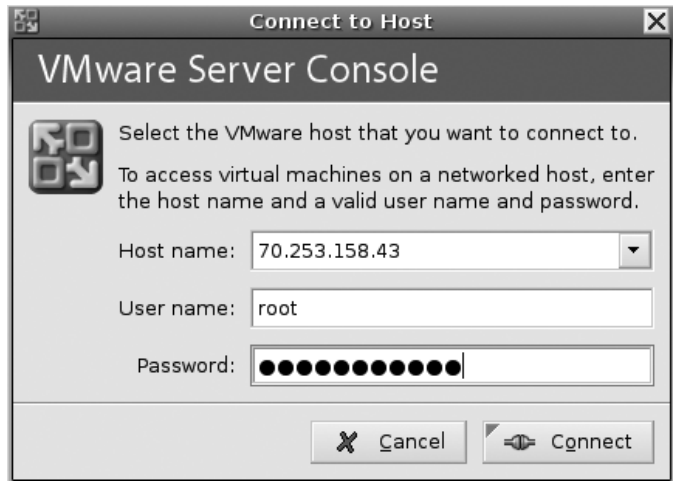
```
Starting VMware services:
Virtual machine monitor                          done
Virtual Ethernet                                 done
Bridged networking on /dev/vmnet0                done
Host-only networking on /dev/vmnet1 (background) done
Host-only networking on /dev/vmnet8 (background) done
NAT service on /dev/vmnet8                       done
Starting VMware virtual machines                 done

The configuration of VMware Server 1.0.1 build-29996 for Linux for this
running kernel completed successfully.
```

Možete preuzeti s Interneta postojeću sliku operativnog sustava, koju VMware zove *appliance*, sa adrese *http://www.vmware.com/vmtn/appliances/directory*. Mi smo odabrali *debian-31r0a-i386-netinst-kernel2.6.zip*, koju smo smjestili u direktorij */var/lib/vmare/Virtual Machines* i raspakirali.

Nakon što smo pripremili osnovnu sliku pokrenuli smo VMware upravljačku konzolu na udaljenoj Ubuntu radnoj površini, iza vatrozida na udaljenoj lokaciji. Zadali smo naredbu:

```
$ gksu vmware-server-console
```

Zatim smo konfigurirali konzolu za daljinsko spajanje s gostujućim operativnim sustavom. Kad je VMware konzola pokrenuta spojili smo se na udaljeni virtualni stroj i prijavili se kao *root*, kao što je prikazano na slici 9-3.




*Slika 9-3. Spajanje na udaljeni virtualni poslužitelj*

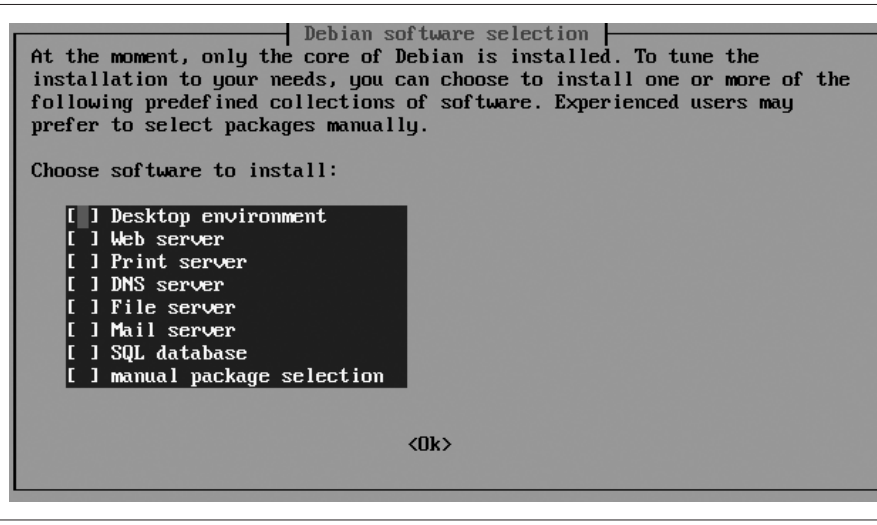
Nakon što smo se spojili s udaljenim domaćinom VMware je zatražio da izradimo virtualni stroj. Kako smo već jednu izradili, umjesto toga smo pritisnuli izbornik File i otvorili direktorij u kojem se nalazila postojeću instancu Debiana. Taj postupak dodao je Debian na popis virtualnog stroja. Naša konzola se tad pojavila slično kao na slici 9-4, što nam je dalo uvid u dostupne operativne funkcije.

Tad smo mogli pokrenuti Debian. Kako se sustav podizao, Debian je počeo izvršavati kasnije faze svojih instalacijskih skripti. Pustili smo ga da radi i uskoro smo došli do zaslona na slici 9-5.

Odabrali smo da konfiguriramo Debian manualno umjesto da odaberemo jednu od predefiniranih konfiguracija. To nam je omogućilo da izradimo pretpostavljeni Debian poslužitelj za korištenje na dodatnim instancama VMware poslužitelja. Slika 9-6 prikazuje Debian sustav u radu.

Snimak zaslona pokazuje nam izvođenje *ifconfig* naredbe. Testirali smo ovu instancu da se uvjerimo da su naše virtualne Ethernet kartice ispravno povezane s IP adresama koje smo zadali.

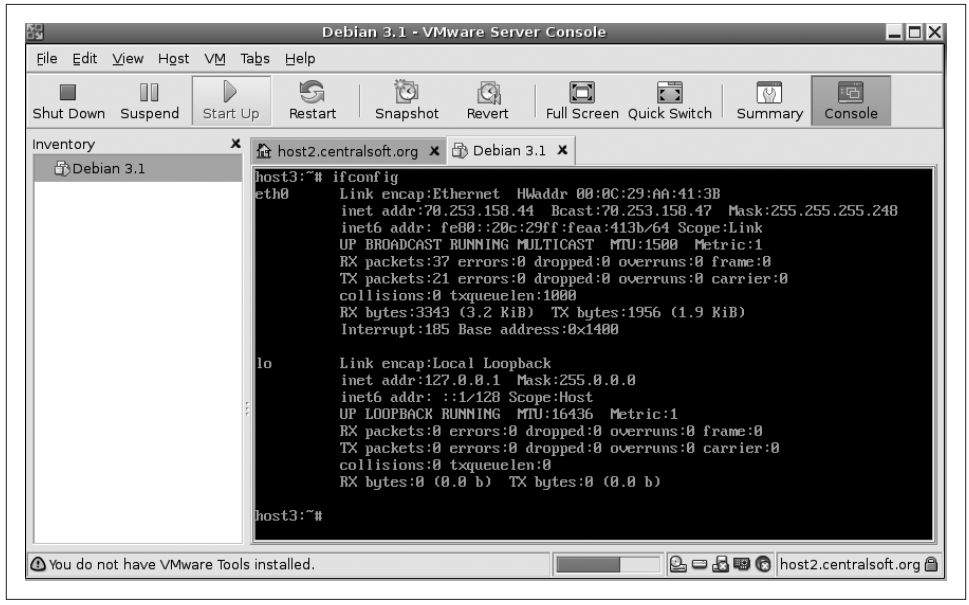
*Slika 9-4. Spojeni smo s udaljenim domaćinom spreman za pokretanje*



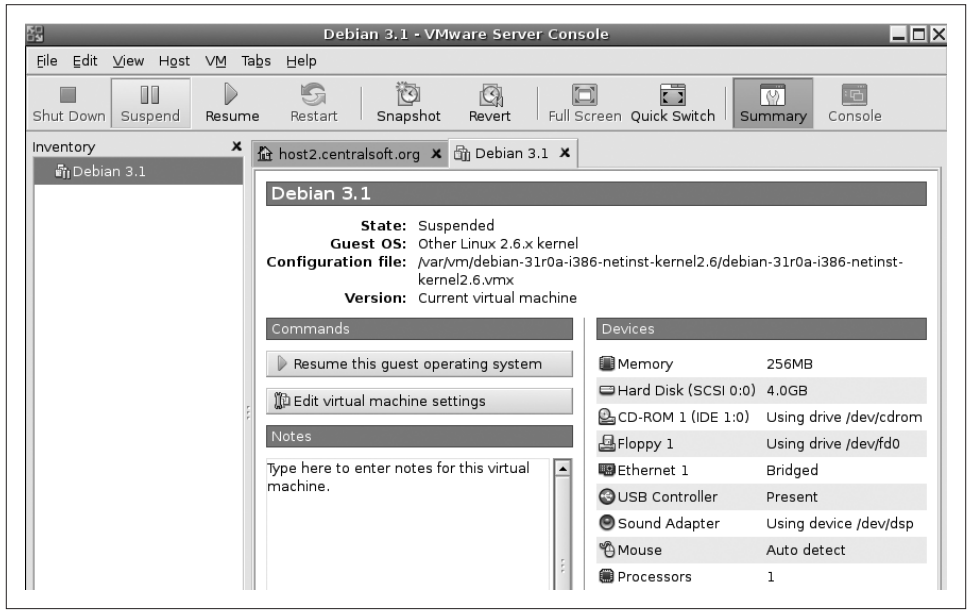

*Slika 9-5. Debian instalacijska skripta koja se izvodi na udaljenom virtualnom stroju*

Kad smo izradili osnovnu Debian sliku, komprimirali smo je i snimili na CD-R medij. Zatim smo upotrijebili tu sliku na drugim domaćinima, nakon što smo za svakog gosta utvrdili ulogu u sustavu i potrebne resurse.

Slika 9-7 pruža sažetak Debian slike. Na desnoj strani možete vidjeti konfiguraciju domaćina. Možemo dinamički mijenjati virtualni poslužitelj da bismo mu dodali memoriju, prostor na disku, Ethernet kartice, procesore i razne uređaje kako se pojavi potreba i kako budemo postavljali dodatne strojeve.



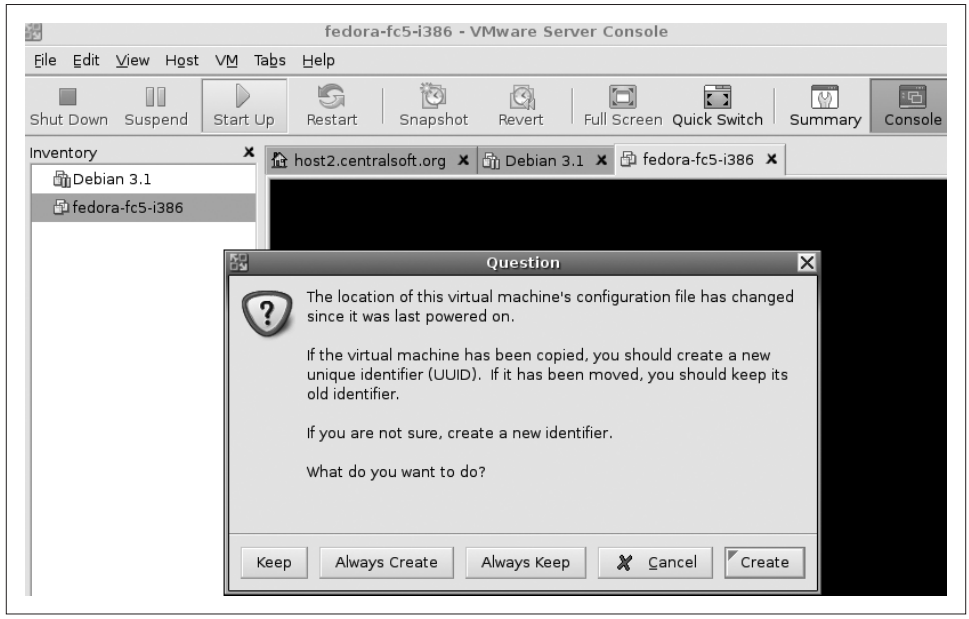

*Slika 9-6. Instalirana instanca Debiana na njegovom udaljenom domaćinu*




*Slika 9-7. Konzola sa sažetkom naše osnovne Debian slike gosta*

## Instaliranje VMware operativnog sustava gosta

Za naš konačni zadatak – instaliranje još jednog operativnog sustava – preuzeli smo Fedoru Core 5 sa Web lokacije VMware zajednice, premjestili je u direktorij *Virtual Machines* i raspakirali kao što smo to učinili i s Debianom. Zatim smo je dodali inventoriju preko izbornika File. Slika 9-8 prikazuje pitanje o jedinstvenom identifikatoru. Možete zadržati postojeći.

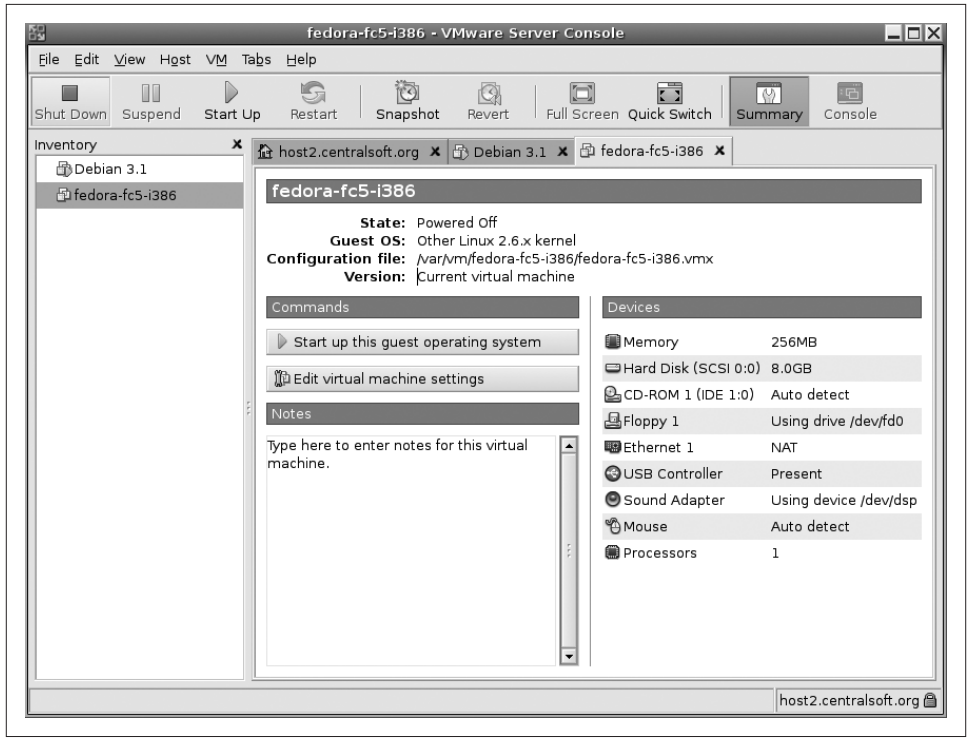



*Slika 9-8. VMware pita za jedinstveni identifikator virtualnog stroja*

VMware upravljačka konzola primijetila je da smo dodali sliku. Kako bi razlikovala moguće višestruke slike, zatražila je jedinstveni identifikator (UUID) u dijaloškom okviru na slici 9-8. Kako smo kopirali Fedoru 5 i imamo sve datoteke koje čine sliku, nije bilo važno koju opciju smo odabrali u dijalogu.

Kad otvorite novi virtualni stroj, VMware pruža mogućnost da verificirate konfiguraciju virtualnog hardvera. Slika 9-9 daje predodžbu o inventoriju virtualnog hardvera dostupnog za Fedoru Core 5.

Osim preuzimanja slika s interneta i učitavanja u upravljačku konzolu, možete instalirati neki Linux operativni sustav i sa standardnih Linux distribucija na CD-ROM-u.




*Slika 9-9. VMware virtualna hardverska konfiguracija za Fedoru Core 5*

# Virtualizacija: samo prolazni hir?

Mnogi analitičari kažu da će sjediti sa strane i čekati da vide hoće li se Linux virtualizacija održati. Kao administrator sustava, možda biste željeli odvagnuti rizike i prednosti ovladavanja ovom tehnologijom.Virtualizacija nije ekvivalent IBM-ovom ili Microsoftovom uvođenju distribuiranih sustava datoteka. Utjecaj hipervizor tehnologije ne može se čak ni usporediti s utjecajem ERP programa kao što su SAP, PeopleSoft ili Oracle Financials.

U svakom slučaju, tehnologije kao što su Xen i VMware imaju neporecive prednosti. Virtualizacija unapređuje primjenu poslužitelja i reducira potrebe za hardverom konsolidiranjem resursa sustava. Izvođenjem postojećeg softvera u virtualnom okruženju možete ne samo sačuvati svoju investiciju u taj softver, već i iskoristiti prednosti jeftinih, standardiziranih poslužitelja.

Nadamo se da vam je ovo poglavlje pružilo znanje i vještine koje trebate da biste implementirali vlastito virtualno okruženje. Sad imate priliku eksperimentirati i zabavljati se besplatnom virtualizacijskom tehnologijom. To bi vas moglo dovesti u poziciju specijalista u području koje samo rijetki razumiju.

# POGLAVLJE 10
# Skripte

Kao administrator Linux sustava koristit ćete dva alata više nego ostale: program za uređivanje teksta, da biste izradili i uredili tekstualne datoteke i školjku za zadavanje naredbi. U jednom trenutku ćete se umoriti od upisivanja naredbi i tražiti načine da poštedite prste i smanjite broj pogrešaka. Tada ćete kombinirati program za uređivanje teksta i školjku da biste izradili najjednostavnije Linux programe: skripte za školjku.

Linux i sam posvuda koristi skripte, pogotovo za prilagodljive zadatke kao što je upravljanje servisima i procesima. Ako razumijete kako su te sistemske skripte napisane, možete interpretirati korake koje poduzimaju i prilagoditi ih svojim potrebama.

Sama *školjka* (sučelje operativnog sustava) je jedna od mnogih inovacija naslijeđenih od Linuxova pradjeda Unixa. 1978. godine, istraživač Bell laboratorija Stephen Bourne razvio je Bourne Shell za Version 7 Unix. Nazvan je *sh* (Unixova cijenjena iritirajuća sažetost) i definirao je standardne značajke koje sve školjke i danas imaju. Školjke su evolvirale na tim temeljima što je dovelo do razvoja Korn školjke (*ksh*, naravno), C školjke (*csh*) i napokon do Bash školjke (*bash*) koja je sada standard na GNU/Linux sustavima. *bash* je igra riječi/akronim za *B*ourne-*A*gain *S*hell i podržava skripte napisane za originalnu Bourne školjku.

Ovo poglavlje počinje sa osnovama *bash* školjke: odzivnicima, naredbama i argumentima, varijablama, izrazima i preusmjeravanjem ulaza i izlaza. Ako su vam one već poznate, nećete mnogo propustiti preskačući par stranica unaprijed (osim možda lijek za nesanicu).

Svaki alat ima svoja ograničenja i u nekom trenutku biste mogli otkriti da *bash* nije najbolje rješenje za sve vaše probleme. Pred kraj ovog poglavlja ispitat ćemo malu aplikaciju napisanu u brojnim skriptnim jezicima: *bash*, kao i Perl, PHP i Python (tri slova P povezana s LAMP akronimom spomenutim u poglavlju 6). Možete usporediti njihove stilove, sintaksu, izražajnost, lakoću korištenja i primjenjivost u različitim domenama. Nije svaki problem čavao, ali dovoljno velik čekić može ga tretirati kao da jest.

# bash počeci

Mnogi operativni sustavi ponudili su sučelja s odzivnikom u svojim ranim danima i tipično su dozvoljavali da naredbe budu spremljene u tekstualnim datotekama i pokretane kao *bash jobs* (spremno shvaćen koncept u to vrijeme). Uskoro je postalo prirodno uvesti načine za prosljeđivanje parametara skriptama i omogućavanje da skripte mijenjaju svoje ponašanje pod različitim okolnostima. Unixova školjka postala je znatno fleksibilnija, pretvarajući se u pravi programski jezik.

Naši interaktivni primjeri pokazivat će *odzivnik školjke*, *naredbu* s opcionalnim *argumentima* i *rezultat*. Na primjer:

```
admin@server1:~$ date
Thu Aug 24 09:16:56 CDT 2006
```

Sadržaj skripte pokazat ćemo ovako:

```
#!/bin/bash
sadržaj skripte...
```

Prvi red je poseban u Linux skriptama: ako počinje s dva znaka `#!`, ostatak prvog reda je naziv datoteke naredbe koju treba pokrenuti da bi se obradio ostatak skripte. (Ako iza znaka # na dolazi !, red se tumači kao komentar koji se nastavlja do kraja reda). Ovaj trik omogućava vam korištenje bilo kojeg programa za interpretiranje skripti. Ako je program tradicionalna školjka, kao *sh* ili *bash*, datoteka se naziva skripta za školjku (engl. *shell script*). Na kraju poglavlja pokazat ćemo skripte za Perl, PHP i Python.

Microsoft Windows koristi nastavak imena datoteke da bi zadao tip datoteke i koji interpretator bi ju trebao otvoriti. Ako promijenite nastavak imena datoteke, mogla bi prestati raditi. Na Linuxu nastavci imena datoteka nemaju nikakve veze s izvršavanjem (premda praćenje konvencija može biti korisno iz nekih drugih razloga).

U svom omiljenom programu za uređivanje teksta (ili čak neki do kojeg vam nije stalo) izradite ovu datoteku od tri reda i spremite ju pod nazivom *hello*:

```
#!/bin/bash
echo hello world
echo bonjour monde
```

Ova datoteka još nije skripta koja radi. Pokazat ćemo kako je zaista pokrenuti u sljedećem odjeljku, ali prvo trebamo objasniti osnovna pravila sintakse.

*/bin/bash* školjka će interpretirati ovu skriptu red po red. Očekuje da svaka naredba bude u jednom redu, ali ako završite red obrnutom kosom crtom (\), *bash* će tretirati sljedeći red kao nastavak:

```
#!/bin/bash
echo \
hello\
world
```

Ovo je dobar način da dugače redove učinite čitljivijima.

Školjka se ne obazire na redove ispunjene bijelim prostorom (razmaci, tabulatori, prazni redovi). Također, ignorira sve od znaka za komentar (#) do kraja reda. Kad *bash* čita drugi red ove skripte (`echo hello world`), tretira prvu riječ (`echo`) kao naredbu koju treba izvršiti, a ostale riječi (`hello world`) kao njezine argumente. Tako naredba *echo* samo kopira svoje argumente u izlaz. Treći red izvršava još jednu naredbu *echo*, ali sa drugačijim argumentima.

Kako biste vidjeli što ste stavili u datoteku *hello*, možete ispisati njen sadržaj na zaslonu:

```
admin@server1:~$ cat hello
#!/bin/bash
echo hello world
echo bonjour monde
```

## Putanje i dopuštenja

Datoteka *hello* može se izvršiti pokretanjem naredbe *bash* s argumentom *hello*:

```
admin@server1:~$ bash hello
hello world
bonjour monde
admin@server1:~$
```

Sad pokušajmo pokrenuti *hello* bez njegovog *bash* pratitelja:

```
admin@server1:~$ hello
bash: hello: command not found
```

Zašto ju *bash* ne može naći? Kad zadate neku naredbu Linux pretratražuje popis direktorija koji se naziva *putanja* (engl. *path*), kako bi našao datoteku s tim imenom i pokreće prvu koju pronađe. U ovom slučaju, *hello* nije bila niti u jednom od direktorija s popisa. Ako sustavu kažete u kojem direktoriju se *hello* nalazi, pokrenut će ju. Putanja može biti apsolutna (*/home/admin/hello*), ili relativna (*./hello* znači datoteka *hello* u trenutačnom direktoriju). U sljedećem odjeljku opisat ćemo kako zadati direktorije u putanji, ali prvo se moramo pozabaviti *dopuštenjima*.

Skripta neće raditi bez određenih dopuštenja. Provjerimo dopuštenja za *hello*:

```
admin@server1:~$ ls -l hello
-rw-r--r--  1 admin admin 48 2006-07-25 13:25 hello
```

Crtica (-) označava da zastavica nije postavljena. Prva crtica je zastavicu direktorija. Može biti `d` za direktorij ili - za datoteku. Zatim dolaze dopuštenja za vlasnika datoteke, grupu kojoj vlasnik pripada i za sve ostale. Vlasnik (*admin*) može čitati (`r`) i pisati (`w`) u ovu datoteku, dok je ostali iz grupe (u ovom slučaju, također nazvane *admin*) i svi ostali, mogu samo čitati (`r--`). Nitko ne može izvršiti datoteku, jer je treći znak u svakom setu od tri znaka - (crtica) umjesto znaka `x`.

Sad pokušajmo pokrenuti *hello* s relativnom putanjom:

```
admin@server1:~$ ./hello
bash: ./hello: Permission denied
```

Ovaj put Linux ju je pronašao, ali ju nije pokrenuo. To se dogodilo zato što datoteka *hello* nema dopuštenja za izvršavanje. Trebate odlučiti kome će biti dozvoljeno izvršavanje: samo vama (vlasniku), bilo kome iz vaše grupe i/ili korisnicima iz drugih grupa. To je praktična sigurnosna odluka koju administratori moraju često donositi. Ako su dopuštenja preširoka, ostali mogu pokretati vašu skriptu bez vašeg znanja. Ako su pak preuska, skripta se možda uopće neće moći pokrenuti.

Naredba za promjenu dopuštenja zove se *chmod* (kratica od change mode) i može koristiti staromodne oktalne brojeve ili slova. Pokušajmo na oba načina, dajući dopuštenja za čitanje/pisanje/izvršavanje vama, dopuštenja za čitanje/izvršavanje vašoj grupi i ukidanje svih dopuštenja ostalim korisnicima (što su oni vama ikada dali?). Za oktalni stil, čitanje=4, pisanje=2 i izvršavanje=1. Tako će broj dopuštenja za korisnika biti 4+2+1 (7), za grupu 4+1 (5), a za ostale korisnike 0:

```
admin@server1:~$ chmod 750 hello
admin@server1:~$ ls -l hello
-rwxr-x---  1 admin admin 50 2006-08-03 15:44 hello
```

Drugi način, zadavanje dopuštenja korištenjem slova, vjerojatno je intuitivniji:

```
admin@server1:~$ chmod u=rwx,g=rx hello
admin@server1:~$ ls -l hello
-rwxr-x---  1 admin admin 50 2006-08-03 15:44 hello
```

Da biste brzo dodali dopuštenje za izvršavanje za sebe, svoju grupu i ostale korisnike unesite:

```
admin@server1:~$ chmod +xr hello
admin@server1:~$ ls -l hello
-rwxr-xr-x  1 admin admin 50 2006-08-03 15:44 hello
```

Sada možemo pokrenuti skriptu iz odzivnika:

```
admin@server1:~$ ./hello
hello world
bonjour monde
```

## Podrazumijevana putanja

Popis direktorija u kojima *bash* treba tražiti naredbe zadane je u varijabli okruženja školjke PATH. Da biste vidjeli što se nalazi u vašoj putanji, unesite:

```
admin@server1:~$ echo $PATH
/bin:/usr/bin
```

Linux rezervira posebna imena: `.` za trenutačni direktorij i `..` za roditeljski direktorij trenutačnog direktorija. Ako želite da Linux uvijek pronađe naredbe kao što je *hello* u vašem trenutačnom direktoriju, dodajte ga u PATH:

```
admin@server1:~$ PATH=$PATH:.
```

Da biste izvodili promjene kao što je ovaj mali dodatak, trebat ćete trajno promijeniti svoju varijablu `PATH`. To može učiniti svaki pojedinačni korisnik u *.bashrc* datoteci smještenoj u njegovom početnom direktoriju, ili to može učiniti administrator sustava u datoteci koja se pokreće zajedno sa sustavom (obično je smještena u */etc* direktoriju). Potrebno je samo dodati red kao ovaj koji smo upravo prikazali.

Alternativno, možete premjestiti skriptu u jedan od direktorija koji su već navedeni u `PATH`. Međutim, ti su direktoriji obično zaštićeni, tako da samo *root* može u njih stavljati datoteke, kako se ne bi narušila sigurnost sustava.

Za skriptu kompleksniju od *hello*, (tj. skoro svaku skriptu), bilo koja od ovih metoda ima implikacije na sigurnosni sustav. Ako je **`.`** naveden u vašem `PATH`-u izlažete se opasnosti da, ukoliko je netko smjestio drugu skriptu *hello* u neki drugi direktorij i vi slučajno nabasate na taj direktorij i upišete *hello*, pokrenete *hello* nekog drugog korisnika, a ne onu koju ste namjeravali.

Ispravnost skripte je također jedna briga. Prilično smo sigurni oko toga što naša *hello* skripta sada radi, ali možda nećemo biti nakon što dodamo još stotinjak redova.

Uobičajena je praksa smjestiti skripte u direktorij poput */usr/local/bin* ili privatni *~/bin*, radije nego u sistemski direktorij poput */bin*, */sbin* ili */usr/bin*. Kako biste taj direktorij permanentno dodali u `PATH`, dodajte red poput ovoga koji slijedi na kraj vaše *.bashrc* datoteke:

```
export PATH=$PATH:/usr/local/bin
```

## Preusmjeravanje izlaza i ulaza

*Preusmjeravanje ulaza i* izlaza te *cijevi* (engl. *pipes*) su još neke od Unix inovacija koje su Microsoft i mnogi drugi besramno kopirali. Školjka pruža pristup ovim značajkama na vrlo intuitivan način.

Kad utipkavate naredbu u konzoli ili u prozoru s tekstom, vaši prsti predstavljaju *standardni ulaz* dok oči čitaju *standardni rezultat* naredbe ili *standardnu obavijest o pogrešci*. Međutim, možete pružiti ulaz iz datoteke ili uhvatiti rezultat u datoteku. Zadajmo naredbu *ls* sa standardnim prikazom rezultata na zaslonu, a potom preusmjerenim (pomoću >) u datoteku:

```
admin@server1:~$ ls
hello
admin@server1:~$ ls > files.txt
admin@server1:~$
```

U drugom primjeru, preusmjeravanje se događa u tišini. Ali, ako bi se pojavile bilo kakve pogreške, vidjeli biste ih na zaslonu, a ne u datoteci:

```
admin@server1:~$ ls ciao > files.txt
ls: ciao: No such file or directory
admin@server1:~$
```

Trebate biti svjesni da će, ako *files.txt* postoji i prije nego što zadate ove naredbe, ta datoteka biti zamijenjena novom datotekom sa istim imenom. Ako biste radije dodali sadržaj u postojeću datoteku nego je zamijenili novom, koristite znak za dodavanje sadržaja u datoteku (>>):

```
admin@server1:~$ ls -l >> files.txt
```

Ukoliko *files.txt* još ne postoji, bit će izrađena prije nego što dodavanje započne.

Možete također preusmjeriti i standardne pogrešku. Evo primjera koji istovremeno preusmjerava i standardni rezultat i standardnu pogrešku:

```
admin@server1:~$ ls -l > files.txt 2> errors.txt
```

Ovaj neelegantni 2> je čarolija koja preusmjerava standardne pogreške. Preusmjeravanje standardne pogreške može biti korisno kod dugotrajnih procesa kao što je prevođenje izvornog koda, pa možete pregledati poruke o pogreškama kasnije, umjesto da „bdijete" nad zaslonom.

Ako želite preusmjeriti standardni rezultat i standardnu pogrešku u istu datoteku, učinite ovo:

```
admin@server1:~$ ls -l > files.txt 2>&1
```

&1 znači „isto mjesto kao i standardni rezultat", što je u ovom slučaju datoteka *files.txt*. Kratica za prethodnu naredbu je:

```
admin@server1:~$ ls -l >& files.txt
```

Koristite >> umjesto > gdje god biste radije dodali sadržaj nego ga zamijenili novim.

Standardni ulaz također može biti preusmjeren. Evo jednostavnog primjera koji traži imena datoteka koja sadrže niz *foo*:

```
admin@server1:~$ ls -l > files.txt
admin@server1:~$ grep foo < files.txt
admin@server1:~$ rm files.txt
```

Prvi korak stvara privremenu datoteku *files.txt*. Drugi korak čita iz nje, a u trećem koraku vježbamo dobru higijenu diska, pa je se rješavamo. Život privremene datoteke bio je kratak, ali produktivan.

Možemo kombinirati ta tri koraka u jedan i izbjeći privremenu datoteku pomoću Unixove najbolje inovacije, *cijevi* (*pipe*). Jedna cijev spaja rezultat jedne naredbe sa ulazom neke druge naredbe. Sibol za cijev je |, koji izgleda kao > i < koje se susreću pri velikoj brzini. Standardni rezultat prve naredbe postaje standardni ulaz druge naredbe, pojednostavljujući naše ranije korake:

```
admin@server1:~$ ls -l | grep foo
```

Možete, također, povezati cijevi u lanac:

```
admin@server1:~$ ls -l | grep foo | wc -l
```

Ova naredba će izbrojiti koliko puta se niz *foo* pojavljuje u datotekama u trenutačnom direktoriju.

## Varijable

*bash* je programski jezik, a programski jezici imaju zajedničke značajke. Jedna od najosnovnijih je *varijabla*: neki simbol koji ima vrijednost. *bash* varijable su nizovi, osim ukoliko zadate drugačije koristeći izraz `declare`. Ne morate zadati ili definirati *bash* varijable prije nego što ih počnete koristiti, što nije slučaj s mnogim drugim jezicima.

Ime neke varijable je niz koji počinje slovom i sadrži slova, brojeve i donje crte (_). Vrijednost neke varijable određuje se stavljanjem znaka `$` ispred imena varijable. Evo jedne skripte koja dodjeljuje vrijednost niza varijabli `hw` i zatim je ispisuje:

```
#!/bin/bash
hw="hello world"
echo $hw
```

Varijabla `hw` je stvorena dodjeljivanjem vrijednosti u redu 2. U redu 3 sadržaj varijable `hw` zamijenit će referencu `$hw`. Kako *bash* i druge školjke tretiraju znakove bijelog prostora (razmake i tabulatore) kao razdvajatelje argumenata naredbe, a ne kao normalne znakove argumenata, da biste ih sačuvali, morate okružiti cijeli niz dvostrukim („) ili jednostrukim (‘) navodnim znakovima. Razlika je u tome što su varijable školjki (i druge posebne sintakse školjki) proširene unutar dvostrukih navodnih znakova i tretirane doslovno unutar jednostrukih navodnih znakova. Pogledajte razliku u rezultatu između dvije *echo* naredbe u sljedećoj skripti:

```
admin@server1:~$ cat hello2
#!/bin/bash
hw="hello world"
echo "$hw"
echo '$hw'
admin@server1:~$ ./hello2
hello world
$hw
admin@server1:~$
```

Standardni rezultat naredbe možete pridružiti varijabli koristeći sintaksu `$(naredba)` ili `` `naredba` `` (s malim znakovima za naglasak):

```
admin@server1:~$ cat today
#!/bin/bash
dt=$(date)
dttoo=`date`
echo "Today is $dt"
echo "And so is $dttoo"
admin@server1:~$ ./today
Today is Tue Jul 25 14:56:01 CDT 2006
And so is Tue Jul 25 14:56:01 CDT 2006
admin@server1:~$
```

Posebne varijable predstavljaju argumente odzivnika. Znak `$` iza kojeg slijedi broj `n` odnosi se na n-ti argument u odzivniku, počevši od 1. Varijabla `$0` je ime same skripte.

Varijabla $* sadrži sve argumente u jednom nizu. Ove varijable se zatim mogu proslijediti naredbama koje skripta izvršava:

```
admin@server1:~$ cat files
#!/bin/bash
ls -Alv $*
admin@server1:~$ ./files hello hello2 today
-rwxr-xr-x  1 admin admin 48 2006-07-25 13:25 hello
-rwxr-xr-x  1 admin admin 51 2006-07-25 14:45 hello2
-rwxr-xr-x  1 admin admin 45 2006-07-25 14:49 today
admin@server1:~$
```

Specijalna varijabla $$ sadržava identifikator trenutačnog procesa. Ona se može koristiti za stvaranje jedinstvenog privremenog imena datoteke. Ako se više kopija iste skripte izvršava istovremeno, svaka će imati drugačiji identifikator procesa i stoga drugačije privremeno ime datoteke.

Još jedna korisna varijabla je $? koja sadrži povratni status naredbe koja je zadnja izvršena. Koristit ćemo to kasnije u ovom poglavlju, kako bismo provjerili uspjeh ili neuspjeh izvršenja programa u nekoj skripti.

# Korisni elementi za bash skripte

Predstavili smo osnovne *bash* elemente koje ćete koristiti u svakodnevnom izvođenju interaktivnih naredbi. Pogledajmo sada neke stvari koje će vam pomoći da napišete učinkovite skripte.

## Izrazi

*bash izrazi* sadrže varijable i *operatore* poput == (znakovi jednakosti) i > (veće od). Ovi znakovi se obično koriste u testovima, što se može zadati na nekoliko načina:

```
test $file == "test"
[ $file == "test" ]
[[ $file == "test" ]]
```

Ako koristite naredbu *test*, zapamtite da neki simboli imaju višestruka značenja (na primjer, u jednom ranijem odlomku, koristili smo znak > za preusmjeravanje), pa zato trebaju biti okruženi navodnim znakovima. Ne trebate brinuti o navodnim znakovima ako koristite sintaksu s jednostrukim ili dvostrukim uglatim zagradama. Dvostruke zagrade rade isto što i jednostruke i još ponešto više, pa je najsigurnije koristiti dvostruke zagrade.

*bash* ima nekoliko korisnih ugrađenih operatora:

```
-a datoteka  # true ako datoteka postoji
-d datoteka  # true ako datoteka postoji i direktorij je
-f datoteka  # true ako datoteka postoji i datoteka je
-r datoteka  # true ako datoteka postoji i može se čitati
-w datoteka  # true ako datoteka postoji i u nju se može pisati
-x datoteka  # true ako datoteka postoji i može se izvršiti
```

## Aritmetika

*bash* je uvelike orijentiran prema tekstu kao što su naredbe, argumenti i imena datoteka. On može procjenjivati uobičajene aritmetičke izraze (koristeći +, -, *, / i druge operatore) okružujući ih parom dvostrukih zagrada: ((izraz)). Kako se mnogi aritmetički znakovi – uključujući *, ( i ) – tumače na poseban način u školjci, najbolje je argumente školjke staviti u navodne znakove, ako će u skripti biti tretirani kao matematički izrazi:

```
admin@server1:~$ cat arith
#!/bin/bash
answer=$(( $* ))
echo $answer
admin@server1:~$ ./arith "(8+1)*(7-1)-60"
-6
admin@server1:~$ ./arith "2**60"
1152921504606846976
admin@server1:~$
```

Najnovija inačica *bash*-a podržava 64-bitne cjelobrojne vrijednosti (-9223372036854775808 do 9223372036854775807). Starije verzije podržavaju samo 32-bitne cjelobrojne vrijednosti (sa skromnim rasponom od - 2147483648 do 2147483647). Brojevi s pomičnim zarezom, izraženi kao faktori broja 10) nisu podržani. Skripte koje trebaju brojeve s pomičnim zarezom ili naprednije operatore, mogu koristiti vanjski program, kao što je *bc*.

U aritmetičkim izrazima možete koristiti varijable bez znaka $ koji bi se inače koristio za zamjenu njihovih vrijednosti u drugim okolnostima:

```
admin@server1:~$ cat arithexp
#!/bin/bash
a=$1
b=$(( a+2 ))
echo "$a + 2 = $b"
c=$(( a*2 ))
echo "$a * 2 = $c"
admin@server1:~$ ./arithexp 6
6 + 2 = 8
6 * 2 = 12
admin@server1:~$
```

## If...

Kada imate spremne izraze, možete izvršavati različite dijelove koda, ovisno o rezultatima testa. *bash* koristi `if ... fi` (obrnuto `if`) sintaksu, uz opcionalne `elif` (else if) i `else` dijelove:

```
if izraz1 ; then
    (naredbe)
elif izraz2 ; then
    (naredbe)
        ...
elif izrazN ; then
```

```
    (naredbe)
else (naredbe)
fi
```

Fraza `; then` na kraju reda može također biti izražena kao obični `then` u sljedećem redu:

```
if izraz
then
    (naredbe)
fi
```

Ako se nalazite u istom direktoriju kao i *hello* skripta koju ste ranije izradili, pokušajte ovo:

```
admin@server1:~$ if [[ -x hello ]]
> then
> echo "hello is executable"
> fi
hello is executable
admin@server1:~$
```

Evo jedne kompliciranije skripte koja pretražuje datoteku */etc/passwd* u potrazi za imenom nekog korisničkog računa:

```
#!/bin/bash
USERID="$1"
DETECTED=$( egrep -o "^$USERID:" < /etc/passwd )
if [[ -n  "${DETECTED}" ]] ; then
    echo "$USERID is one of us    :-)"
else
    echo "$USERID is a stranger   :-("
fi
```

Nazovimo tu skriptu *friendorfoe*, dodijelimo joj dopuštenja za izvršavanje i isprobajmo s poznatim korisničkim računom na našem sustavu (*root*) i zatim sa izmišljenim korisničkim računom (*sasquatch*):

```
admin@server1:~$ ./friendorfoe root
root is one of us   :-)
admin@server1:~$ ./friendorfoe sasquatch
sasquatch is a stranger  :-(
```

Prvi argument dodijeljen je varijabli školjke `USERID`. Naredba *egrep* izvršava se unutar `$()` da bi se njen rezultat dodijelio varijabli školjke `DETECTED`. *egrep -o* ispisuje samo niz koji se podudara, a ne cijeli red. `"^$USERID:"` se podudara sa sadržajem `USERID` varijable samo ako se sadržaj varijable pojavljuje na početku reda i odmah nakon njega slijedi dvotočka. Izraz `if` okružen je dvostrukim uglatim zagradama da bi bio zadržan, procijenjen i da bi se vratio njegov rezultat. Izraz `-n "${DETECTED}` vraća `true` ako je varijabla školjke `DETECTED` neprazan niz. Na kraju, varijabla `DETECTED` stavljena je u navodne znakove (`"${DETECTED}"`) da bi bila tretirana kao samo jedan niz.

Gdje god `if` tvrdnja uzima izraz, možete ubaciti naredbu, ili čak niz naredbi. Ako je zadnja naredba u slijedu uspješno izvedena, tvrdnja `if` smatra da je izraz vratio pozitivan rezultat (true). Ako je zadnja naredba u nizu neuspješna, smatra se da je izraz vratio negativan rezultat (false) i izvršit će se izraz `else`. Vidjet ćemo primjere u dolazećim odjeljcima.

## Rješavanje problema s jednostavnim skriptama

Idemo izvesti kirurški zahvat nad skriptom koja bi trebala izbrisati svoj argument (datoteku ili direktorij), ali ima nekoliko problema:

```
admin@server1:~$ cat delete
#!/bin/bash
if rm $1
    then
    echo file $1 deleted
else
    if rmdir $1
        then
        echo directory $1 deleted
    fi
fi
```

Skripta bi trebala izbrisati datoteku proslijeđenu u argumentu koristeći *rm* i ispisati poruku ako uspije. Ako *rm* ne uspije, skripta pretpostavlja da se argument odnosi na direktorij i pokušava s *rmdir*.

Evo nekih rezultata:

```
admin@server1:~$ ./delete hello2
file hello2 deleted
admin@server1:~$ ./delete hello2
rm: cannot remove `hello2': No such file or directory
rmdir: `hello2': No such file or directory
admin@server1:~$ mkdir hello3
admin@server1:~$ ./delete hello3
rm: cannot remove `hello3': Is a directory
directory hello3 deleted
admin@server1:~
```

Koristeći ove poruke o pogreškama, pokušajmo popraviti skriptu. Prvo, koristit ćemo preusmjeravanje ulaza i izlaza da bismo spremili rezultate u dnevnik i datoteku s popisom pogrešaka, koje kasnije možemo pregledati. Zatim ćemo uhvatiti povratnu vrijednost naredbe *rm*, kako bismo generirali poruku o uspjehu ili neuspjehu izvršavanja. Također ćemo zabilježiti trenutni datum i vrijeme kako bismo i te podatke uključili u dnevnik rezultata:

```
admin@server1:~$ cat removefiles
#!/bin/bash
# removefiles deletes either files or directories
echo  "$0 ran at" $(date) >> delete.log
if rm $1 2>> delete-err.log
    then
    echo "deleted file $1" >> delete.log
elif rmdir $1 2>> delete-err.log
    then
    echo "deleted directory $1" >> delete.log
else
    echo "failed to delete $1" >> delete.log
fi
```

Skripta ima još nekoliko nedostataka: ne provjerava postoji li datoteka i ne razlikuje datoteku od direktorija. Možemo koristiti neke od operatora koje smo ranije spomenuli kako bismo riješili ove probleme:

```
admin@server1:~$ cat removefiles
#!/bin/bash
# removefiles deletes either files or directories
echo  "$0 ran at" $(date) >> delete.log
if [ ! -e $1 ]
    then
    echo "$1 does not exist" >> delete.log
elif [ -f $1 ]
    then
    echo -n "file $1 " >> delete.log
    if rm $1 2>> delete-err.log
        then
        echo "deleted" >> delete.log
    else
        echo "not deleted" >> delete.log
    fi
elif [ -d $1 ]
    then
    echo "directory $1 " >> delete.log
    if rmdir $1 2>> delete-err.log
        then
        echo "deleted" >> delete.log
    else
        echo "not deleted" >> delete.log
    fi
fi
```

Ovo izgleda prilično dobro, ali imamo još jedan zavoj za savladati: što ako ime datoteke ili direktorija sadži razmake? Jamčimo vam da ćete se susresti s time ako razmjenjujete datoteke s Windows ili Macintosh korisnicima). Izradite datoteku *my file* a zatim je pokušajte izbrisati našom skriptom:

```
admin@server1:~$ ./removefiles my file
```

Zadnji red u dnevniku *delete.log* će sadržavati:

```
my does not exist
```

Kako nismo stavili navodne znakove oko *my file*, školjka je podijelila *my* i *file* na varijable `$1` i `$2`. Zbog toga ćemo staviti *my file* u navodne znakove kako bismo ime zadržali u `$1`:

```
admin@server1:~$ ./removefiles "my file"
./removefiles: [: my: binary operator expected
./removefiles: [: my: binary operator expected
```

Ups! Dobili smo niz *my file* unutar varijable `$1`, ali ga *unutar* skripte trebamo ponovno staviti u navodne znakove da bismo ga sačuvali za test imena i naredbu za brisanje:

```
admin@server1:~$ cat removefiles
#!/bin/bash
# removefiles deletes either files or directories
```

```
echo  "$0 ran at" $(date) >> delete.log
if [ ! -e "$1" ]
    then
    echo "$1 does not exist" >> delete.log
elif [ -f "$1" ]
    then
    echo -n "file $1 " >> delete.log
    if rm "$1" 2>> delete-err.log
        then
        echo "deleted" >> delete.log
    else
        echo "not deleted" >> delete.log
    fi
elif [ -d "$1" ]
    then
    echo -n "directory $1 " >> delete.log
    if rmdir "$1" 2>> delete-err.log
        then
        echo "deleted" >> delete.log
    else
        echo "not deleted" >> delete.log
    fi
fi
```

Napokon, kada zadate naredbu:

```
admin@server1:~$ ./removefiles "my file"
```

zadnj red *delete.log* dnevnika će glasiti:

```
file my file deleted
```

## Petlje

Ako jedan postupak želite izvesti više puta, treba vam *petlja*. *bash* nudi tri vrste petlji: `for`, `while` i `until`.

Simpatična i talentirana petlja `for` ima ovaj opći oblik:

```
for argument in popis
do
naredbe
done
```

Ona izvršava ***naredbe*** (koje mogu obuhvaćati koliko god želite redova i naredbi) zadane između `do` i `done` za svaku stavku na ***popisu***. Kako se naredbe izvršavaju, one mogu pristupiti trenutačnoj stavci sa `popisa` putem varijable $***arg***. Sintaksa može biti malo zbunjujuća u početku: u iskazu `for` morate zadati `argument` bez znaka `$`, ali u ***naredbama*** morate zadati $***arg*** sa znakom dolara.

Neki jednostavni primjeri su:

```
admin@server1:~$ for stooge in moe larry curly
> do
> echo $stooge
```

```
> done
moe
larry
curly

admin@server1:~$ for file in *
> do
> ls -l $file
> done
-rw-r--r--  1 admin admin 48 2006-08-26 14:12 hello

admin@server1:~$ for file in $(find / -name \*.gif)
> do
> cp $file /tmp
> done
```

Petlja `while` se izvodi dok je rezultat testa `true`:

```
while izraz
do
naredbe
done
```

Evo primjera skripte koja koristi aritmetičke izraze za `while` petlju u C-stilu (uvlačenje nije potrebno, ali nama se sviđa):

```
#!/bin/bash
MAX=100
((cur=1))  # Treat cur like an integer
while ((cur < MAX))
    do
    echo -n "$cur "
    ((cur+=1)) # Increment as an integer
    done
```

Petlja `until` je suprotnost petlji `while`. Ona se izvodi dok je rezultat testa `false`:

```
until izraz
do
naredbe
done

Evo i primjera:

#!/bin/bash
gameover="q"
until [[ $cmd == $gameover ]]
    do
    echo -n "Your commmand ($gameover to quit)? "
    read cmd
    if [[ $cmd != $gameover ]]; then $cmd; fi
    done
```

Da biste izašli iz petlje, koristite `break` (prekid). Napišimo sada naš primjer `until` kao `while` petlju sa dodatkom `break`:

```
#!/bin/bash
gameover="q"
while [[ true ]]
    do
    echo -n "Your commmand ($gameover to quit)? "
    read cmd
    if [[ $cmd == $gameover ]]; then break; fi
    $cmd
    done
```

Da biste preskočili ostatak petlje i vratili se na početak, koristite `continue`:

```
#!/bin/bash
gameover="q"
while [[ true ]]
    do
    echo -n "Your commmand ($gameover to quit)? "
    read cmd
    if [[ $cmd != $gameover ]]; then $cmd; continue; fi
    break
    done
```

## cron poslovi

Skripte za školjku se često koriste za povezivanje programa. Uobičajen primjer u Linuxu je definicija *cron poslova*. *cron* je standardni Linuxov upravljač poslovima. Ako želite da se neki posao obavlja svakog trećeg utorka u mjesecu, u neciviliziranо doba od 01:23, možete zadati *cronu* da to učini umjesto vas. *cron* pozadinski servis svake minute provjerava je li vrijeme da nešto učini te jesu li se neke specifikacije *cron* posla promijenile.

*cron* poslove zadajete uređivanjem *crontab* datoteke. Sadržaj *crontab* datoteke, ako ga ima, možete vidjeti na sljedeći način:

```
admin@server1:~$ crontab -l
no crontab for admin
```

Da biste uredili svoju *crontab* datoteku, zadajte:

```
admin@server1:~$ crontab -e
```

Svaki red *crontab* datoteke sadrži dan/vrijeme i naredbu u ovom formatu:

```
minuta sat dan_u_mjesecu mjesec dan_u_tjednu naredba
```

Ovaj format zahtijeva malo detaljnije objašnjenje:

- *minuta* je između `0` i `59`
- *sat* koristi 24-satni sat i može imati vrijednost između `0` i `23`
- *dan_u_mjesecu* je u rasponu između `1` i `31`
- *mjesec* je broj između `1` i `12` ili ime, npr. `February`

- *dan_u_tjednu* je broj između 0 i 7 (0 ili 7 je nedjelja, 6 je subota) ili ime, kao npr. Tuesday
- *dan_u_mjesecu* i *dan_u_tjednu* su povezani logičkim operatorom OR, što može izazvati iznenađenja. Na primjer, ako svako polje sadrži 1, *cron* će izvršiti naredbu u siječnju, kao i ponedjeljcima. Obično je zadano samo jedno od ovih polja.
- U bilo kojem polju, neka vrijednost znači točno podudaranje; na primjer, 1 u polju month znači samo siječanj.
- Zvjezdica (*) znači *bilo koja vrijednost* (*any value*)
- Dvije vrijednosti odvojene crticom označavaju opseg. Stoga, 11-12 u month polju znači od studenog do prosinca.
- Da biste zadali više vrijednosti, odvojite ih zarezima. Npr. ako za mjesec zadate 2, 3, 5-6 znači veljača, ožujak i svibanj do lipnja.
- *Modifikator koraka* dolazi iza vrijednosti i kose crte (/) i zadaje za koliko jedinica se vrijednost povećava. Vrijednost */3 za mjesec znači svaki treći mjesec dok se vrijednost 4-9/2 odnosi na mjesece 4, 6 i 8.

Školjka izvršava naredbu, pa može koristiti značajke spomenute u ovom poglavlju. Neki primjeri korištenja izravnih naredbi umjesto skripti su:

```
5 * * * * rm /tmp/*.gif # briše sve GIF datoteke svakih 5 minuta
5 * * * * rm -v /tmp/*.gif >> /tmp/gif.log # obavlja isto ali uz bilježenje u dnevnik
```

Kada *cron* izvrši naredbu, on šalje poruku elektroničke pošte sa standardnim rezultatom i standardnom porukom o pogreškama vlasniku *crontab* datoteke. Da biste izbjegli primanje velike količine takvih poruka, možete preusmjeriti standardni rezultat i standardne poruke o pogreškama na neko mjesto gdje sunce ne sja:

```
naredba > /dev/null 2>&1
```

# Završna bitka skriptnih jezika

Glavna primjena školjke je izvršavanje naredbi. Druge zadaće, poput izvođenja aritmetičkih proračuna, su teže jer tekst u njima treba biti zaštićen od prijeloma riječi i proširivanja. U složenim skriptama hrpa zagrada, uglatih zagrada i ostalih simbola počinje ličiti na psovke likova iz crtanih filmova.

Nekada davno („Tada smo imali nule i jedinice i bili smo sretni što ih uopće imamo!), u priručnicima su se često navodile dugačke skripte za dodavanje korisnika, učitavanje i izgradnju paketa, sigurnosno kopiranje datoteka i tako dalje. U današnje vrijeme, ove poslove prikladnije je obavljati korištenjem naprednijih skriptnih jezika, iz nekoliko razloga:

- S vremenom, aplikacije poput *adduser* i *apt-get* su automatizirale neke poslove koje su tradicionalno obavljale skripte.
- Skripte se teško proširuju i postaju teške za održavanje

- Skripte se izvode sporije
- Sintaksa školjke je kompliciranа

Perl je u početku popunio ovu prazninu, ali sad je PHP prešao izvan okvira Web stranica a Python stekao reputaciju alata visoke produktivnosti. Napisat ćemo jednu aplikaciju na svakom od tih jezika. Na Linuxu je dostupno i nekoliko drugih jezika kao što su Ruby i Tcl.

Naša aplikacija će pretraživati datoteku */etc/passwd* tražeći ime, identifikator korisnika i što god drugo može tamo pronaći. Vidjet ćete kako otvoriti datoteku, čitati zapise, parsirati formate, tražiti uzorke i ispisivati rezultate. Potom ćemo pogledati načine izbjegavanja nekih od ovih poslova, jer znoj != produktivnost. Bit ćete sposobni primijeniti te tehnike na druge datoteke, poput dnevnika ili Web stranica. To je jedan primjer *žongliranja podacima* (*data munging*) i vjerojatno to već često i radite.

Izmislimo neke zahtjeve za našu aplikaciju i izrazimo ih ovim pseudokodom:

```
učitaj niz za pretraživanje koji je zadao korisnik
otvori datoteku
u svakom redu:
    parsiraj polja (stupce)
    potraži u polju ime niz koji je zadao korisnik
    ako nešto pronađeš:
        ispiši i ostala polja u čitljivom formatu
```

Do sada bi mnogi programeri već požurili i počeli pisati kod (neki i prije nego što su pročitali format podataka ili uvjete koji moraju biti ispunjenu). Čitatelji ove knjige su, međutim, discipliniraniji, kao i zgodniji. Oni su već morali čistiti nered koje su napravili drugi programeri, pa ne žele sami raditi iste pogreške.

## Format podataka: /etc/passwd datoteka

Datoteka s lozinkama obično sadržava standardne sistemske korisničke račune, poput svemoćnog računa *root*, račune aplikacija kao što je *apache* i račune običnih korisnika. Evo odlomka iz jedne takve datoteke:

```
# System
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/sbin/nologin
daemon:x:2:2:daemon:/sbin:/sbin/nologin
...
# Applications
postgres:x:26:26:PostgreSQL Server:/var/lib/pgsql:/bin/bash
apache:x:48:48:Apache:/var/www:/bin/false
...
# Users
adedarc:x:500:500:Alfredo de Darc:/home/adedarc:/bin/bash
rduxover:x:501:501:Ransom Duxover:/home/rduxover:/bin/bash
```

```
cbarrel:x:502:502:Creighton Barrel:/home/cbarrel:/bin/bash
cmaharias:x:503:503:C Maharias:/home/cmaharias:/bin/bash
pgasquette:x:504:504:Papa Gasquette:/home/pgasquette:/bin/bash
bfrapples:x:505:505:Bob Frapples:/home/bfrapples:/bin/bash
```

Polja razdvojena dvotočkom su:

- Ime računa
- Šifrirana lozinka, ili x ako se koristi */etc/shadow*
- Identifikator korisnika (uid)
- Identifikator grupe (gid)
- Puno ime ili opis
- Početni direktorij
- Školjka

Zanima nas peto polje (puno ime ili opis). U prastarim Unix svicima, to se zvalo *gecos* polje, iz razloga koji su već i tad bili staromodni. Ime traje i dobro ga je znati.

## Inačice skripte

Odlomke koji sljede započet ćemo sa po jednom malom skriptom koja traži niz u datoteci */etc/passwd* i ispisuje odgovarajući red. Znamo da je ovo preširoko, ali želimo da skripta bude funkcionalna prije nego što postanemo previše maštoviti.

U nastavku ćemo podijeliti ulazne redove na polja i ograničiti poklapanje uzoraka na polje *gecos* koje sadrži imena naših korisnika.

Zatim ćemo dalje ograničiti pretraživanje na redove u kojima je vrijednost *uid* polja veća od 500. U našem slučaju, normalni korisnički identifikatori počinju od 501, pa će taj postupak isključiti sistemske račune i druge automatone.

Do ove točke ćemo se prilično umoriti od prethodnih koraka, pa ćemo potražiti alate koji mogu obaviti dio tog posla umjesto nas.

### bash skripta

Većina jezika pruža biblioteke funkcija za obavljanje raznih zadataka. Programi ispunjavaju ovu ulogu za školjku, a većina iskusnih autora skripti su upoznati s većinom korisnih Linux pomoćnih programa (*cat*, *head*, *tail*, *awk*, *cut*, *grep*, *egrep* i drugi). Mi ćemo koristiti neke od njih za našu *bash* skriptu.

Evo brze i grube verzije (*finduser.sh*) koja uzima korisnikov niz za pretragu kao argument, traži podudaranje neovisno o malim ili velikim slovima, bilo gdje u redu i ispisuje u cjelosti svaki red:

```
#!/bin/bash
grep -i "$1" /etc/passwd
```

```
admin@server1:~$ chmod +x finduser.sh
admin@server1:~$ ./finduser.sh alf
adedarc:x:500:500:Alfredo de Darc:/home/adedarc:/bin/bash
```

To nije bilo ništa brže nego upisivanje:

```
admin@server1:~$ grep -i alf /etc/passwd
```

Ali što bi se dogodilo da se *alf* podudaralo sa sistemskim korisničkim računom *gandalf*, ili nizom u nekom drugom polju? Ako želimo ograničiti pretraživanje na polje sa imenima i normalne korisničke račune (npr. račune s korisničkim identifikatorom većim od 500), naša skripta će ponešto narasti.

Pretraživanje *bash* dokumentacije otkriva da *bash* može razdvojiti svoj ulaz na znakovima koji nisu bijeli prostor koristeći `IFS` varijablu. U sljedećoj inačici skripte čitamo */etc/passwd* red po red, razdvajajući svaki red na varijable polja. Ako pronađemo poklapanje trebamo ponovno izgraditi red da bismo ga ispisali u originalnom obliku:

```
#!/bin/bash
pattern=$1
IFS=":"
while read account password uid gid name directory shell
    do
    # Exact case-sensitive matches only!
    if [[ $name == $pattern ]]; then
        echo "$account:$password:$uid:$gid:$name:$directory:$shell"
        fi
        done < /etc/passwd
```

No, sad smo naletjeli na problem sa poklapanjem: za razliku od *grepa*, *bash* nema ugrađenu podršku za djelomično poklapanje niza s razlikovanjem velikih i malih slova. Morat ćemo ugraditi napredniji mehanizam za pronalaženje uz pomoć vanjskog alata *egrep*:

```
#!/bin/bash
pattern=$1
IFS=":"
while read account password uid gid name directory shell
    do
    if [[ $(echo $name | egrep -i -c "$pattern") -gt 0 ]]; then
        echo "$account:$password:$uid:$gid:$name:$directory:$shell"
        fi
        done < /etc/passwd
```

Za konačnu inačicu skripte dodajmo provjeru *uid* brojeva:

```
#!/bin/bash
pattern=$1
IFS=":"
while read account password uid gid name directory shell
    do
    # Exact matches only!
    if [[ $uid -gt 500 && $(echo $name | egrep -i -c "$pattern") -gt 0 ]]; then
        echo "$account:$password:$uid:$gid:$name:$directory:$shell"
        fi
        done < /etc/passwd
```

Ako pokrenete skriptu sa opcijom *-v* ili *-x*, *bash* će ispisati svaku naredbu prije nego što je izvrši. To može pomoći da vidite što skripta zapravo radi.

### Perl skripta

Perl je koncizan i jako, jako dobro radi s tekstom. Perl ekvivalent naše prve *bash* skripte bio bi:

```
admin@server1:~$ perl -ne 'print if /alf/i' /etc/passwd
```

Izraz `/uzorak/` pronalazi `uzorak`, dok `i` koji slijedi zadaje zanemarivanje velikih i malih slova. Evo ekvivalentne verzije skripte koju ćemo koristiti da pojačamo program kako bi odgovarao našim ostalim zahtjevima:

```
#!/usr/bin/perl
my $pattern = shift;
while (<>) {
    if (/$pattern/i) {
        print;
    }
}
```

Mnogi elementi Perl sintakse su tajnoviti, ali neki podsjećaju na sintaksu školjke (ili druge poznate Unix alate) te ih stoga nije teško zapamtiti jednom kad ih upoznate. U prethodnoj skripti možete vidjeti iskaze `while` i `if` i oni se ponašaju onako kako ste naučili da se ponašaju ekvivalentni izrazi za školjku. Sintaksa sa znakovima `< >` podsjeća na znakove < i > za preusmjeravanje koji se koriste u školjci. Zbog toga svaka iteracija `while` petlje čita jedan red ulaza. Obratite pozornost na to da, za razliku od *basha*, varijable u Perlu zahtjevaju početni `$` čak i kada pridružujete vrijednosti. Iskaz `print` prikazuje ono što `< >` pronađe.

Perl ima i alternativnu obrnutu `if` sintaksu koja štedi nekoliko znakova:

```
#!/usr/bin/perl
my $pattern = shift;
while (<>) {
    print if /$pattern/i;
}
```

Skripta (zovite je *finduser.pl*) pretpostavlja da se datoteka s lozinkama čita iz standardnog ulaza, pa biste je izvodili ovako:

```
admin@server1:~$ ./finduser.pl alf < /etc/passwd
```

Sljedeća inačica izravno otvara datoteku s lozinkama:

```
#!/usr/bin/perl
my $fname = "/etc/passwd";
my $pattern = shift;
open(FILE, $fname) or die("Can't open $fname\n");
while (<FILE>) {
    if (/$pattern/i) {
        print;
    }
}
close(FILE);
```

Da bismo ograničili pretraživanje na polje imena, kao što smo učinili u *bash* primjeru, igramo na Perlovu jaču stranu:

```
#!/usr/bin/perl
my $fname = "/etc/passwd";
my $pattern = shift;
open(FILE, $fname) or die("Can't open $fname\n");
while (<FILE>) {
    $line = $_;
    @fields = split/:/;
    if ($fields[4] =~ /$pattern/i) {
        print $line;
    }
}
close(FILE);
```

Argument koji zadaje korisnik učitava se u varijablu `$pattern` pomoću iskaza `shift`. Skripta također definira još jednu vrstu varijable: polje `@fields`. Perlova funkcija `split` stavlja svaki element odvojen zarezom u nekom redu u pojedinačni element polja. Tad možemo izdvojiti element `4` (koji je zapravo peti element, jer se elementi broje počevši od 0) i usporediti ga (bez razlikovanja velikih i malih slova) s argumentom koji je zadao korisnik.

Sve ove skripte izvodile su učitavanje ulaznih redova teksta i uspoređivanje. Kako je */etc/passwd* vrlo važna datoteka u Linuxu, pomislili biste da je netko već automatizirao nešto od tog posla. Srećom, netko i jest: stari dobri Perl ima ugrađenu funkciju `getpwent` koja vraća sadržaj */etc/passwd* red po red, kao polje nizova. U sljedećoj inačici naše skripte zadajemo svakom polju njegovu varijablu. U inačici nakon nje koristit ćemo polje `@list` kako bismo obuhvatili sve varijable. U svakom slučaju želimo *gecos* polje (u Perl dokumentaciji zove se *gcos*). Obratite pozornost na to da je to polje `6`, kako ga vraća `getpwent`, a ne polje `4`, jer `getpwent` podržava druga dva polja koja se pojavljuju u *passwd* datotekama na nekim sustavima:

```
#!/usr/bin/perl
$pattern = shift;
while (($name,$passwd,$uid,$gid,
       $quota,$comment,$gcos,$dir,
       $shell,$expire) = getpwent) {
    if ($gcos =~ /$pattern/i) {
        print "$gcos\n";
    }
}

#!/usr/bin/perl
$pattern = shift;
while (@fields = getpwent) {
    if ($fields[6] =~ /$pattern/i) {
        print "$fields[6]\n";
    }
}
```

Za kraj, ograničimo pretraživanje samo na normalne korisnike (*uid* > 500). To je samo jedan kratak dodatak:

```
#!/usr/bin/perl
$pattern = shift;
while (@fields = getpwent) {
    if ($fields[6] =~ /$pattern/i and $fields[2] > 500) {
        print "$fields[6]\n"
    }
}
```

### PHP skripta

PHP se može izvršavati na Web poslužitelju (koristeći CGI) ili samostalno (koristeći CLI). Mi ćemo koristiti CLI inačicu. Ako nemate CLI inačicu, možete je instalirati na Debian sustavima pomoću *apt-get install php4-cli*.* Naša prva PHP skripta izgledat će kao naše prve Perl skripte:

```
#!/usr/bin/php
<?
$pattern = $argv[1];
$file = fopen("/etc/passwd", "r");
while ($line = fgets($file, 200)) {
    if (eregi($pattern, $line))
        echo $line;
}
fclose($file);
?>
```

Zahvaljujući svom porijeklu iz Web okruženja, PHP čini neuobičajenu pretpostavku da podrazumijevani sadržaj neke datoteke treba tumačiti kao običan tekst te da se PHP kôd nalazi samo između početne oznake `<?` ili `<?php` i završne oznake `?>`. On ispisuje tekst na standardni ulaz. Funkcija `eregi` uspoređuje regularne izraze uz razlikovanje malih i velikih slova.

Kako je PHP posudio mnogo toga od Perla, nije iznenađenje da ima i funkciju `split`:

```
#!/usr/bin/php
<?
$pattern = $argv[1];
$file = fopen("/etc/passwd", "r");
while ($line = fgets($file, 200)) {
    $fields = split(":", $line);
    if (eregi($pattern, $fields[4]))
        echo $line;
}
fclose($file);
?>
```

* Ili PHP5-cli kada je dostupan.

No, možemo li pozvati neku funkciju poput Perlove `getpwent` da umjesto nas "secira" datoteku s lozinkama? PHP, čini se, nema ekvivalentnu funkciju pa ćemo se držati pristupa s parsiranjem da bismo ograničili pretraživanje na *uid* vrijednosti iznad 500:

```
#!/usr/bin/php
<?
$pattern = $argv[1];
$file = fopen("/etc/passwd", "r");
while ($line = fgets($file, 200)) {
    $fields = split(":", $line);
    if (eregi($pattern, $fields[4]) and $fields[2] > 500)
        echo $line;
}
fclose($file);
?>
```

### Python skripta

Python skripte izgledaju drugačije od Perl i PHP skripti jer su iskazi zaključene bijelim prostorom, a ne točka-zarezom ili vitičastim zagradama u C-stilu. Znakovi tabulatora su također značajni. Naša prva Python skripta, kao i naši raniji pokušaji u drugim jezicima, pretražuje datoteku s lozinkama i ispisuje bilo koji red koji sadrži odgovarajući tekst:

```
#!/usr/bin/python
import re, sys
pattern = "(?i)" + sys.argv[1]
file = open("/etc/passwd")
for line in file:
    if re.search(pattern, line):
        print line
```

Python podržava *imenske prostore* (kao i Perl) za grupiranje funkcija i zbog toga ispred funkcija u ovoj skripti stoje nizovi `sys.` i `re.`. Ovo pomaže da moduli koda budu malo modularniji. `"(?i)"` u trećem redu skripte čini da se u pronalaženju ne razlikuju velika i mala slova, slično kao `/i` u Perlu.

Sljedeća iteracija, koja dijeli ulazni red na polja, ima jednostavan dodatak:

```
#!/usr/bin/python
import re, sys
pattern = "(?i)" + sys.argv[1]
file = open("/etc/passwd")
for line in file:
    fields = line.split(":")
    if re.search(pattern, fields[4]):
        print line
```

Python ima ekvivalent Perlovoj funkciji `getpwent` koji omogućava da ograničimo pretraživanje na polje koje sadrži imena. Spremite sljedeću skriptu kao *finduser.py*:

```
#!/usr/bin/python
import re, sys, pwd
pattern = "(?i)" + sys.argv[1]
for line in pwd.getpwall():
    if re.search(pattern, line.pw_gecos):
        print line
```

Pogledajmo sada kako to radi:

```
admin@server1:~$ ./finduser.py alf
('adedarc', 'x', 501, 501, 'Alfredo de Darc', '/home/adedarc', '/bin/bash')
```

U ovoj skripti, red koji smo ispisali bio je Python popis a ne niz, i bio je lijepo ispisan. Da biste ispisali taj red u njegovom originalnom formatu, koristite ovo:

```
#!/usr/bin/python
import re, sys, pwd
pattern = "(?i)" + sys.argv[1]
for line in pwd.getpwall():
    if re.search(pattern line.pw_gecos):
        print ":".join(["%s" % v for v in line])
```

Zadnji red je potreban kako bi pretvorio svako polje u niz (`pw_uid` i `pw_gid su` cjelobrojne vrijednosti) prije nego što ih spoj u jedan dugačak niz odvojen dvotočkama. Premda Perl i PHP dozvoljavaju da tretirate neku varijablu kao niz ili broj, Python je strog.

Završni korak je ograničavanje potrage na korisničke račune kod kojih je *uid* > 500:

```
#!/usr/bin/python
import re, sys, pwd
pattern = "(?i)" + sys.argv[1]
for line in pwd.getpwall():
    if line.pw_uid > 500 and re.search(pattern line.pw_gecos):
        print ":".join(["%s" % v for v in line])
```

## Odabir skriptnog jezika

Odabir programskog jezika, kao i odabir programa za uređivanje teksta ili operativnog sustava, je umnogome stvar ukusa. Neki ljudi smatrju Perl kôd nečitljivim, a drugi se opiru Pythonovim pravilima za bijeli prostor. Često usporedba ne ide dalje od toga; ako ne volite grašak, zašto biste ga jeli?

Ako vam odgovara stil jezika, najvažniji kriterij je produktivnost. *bash* je brz način za pisanje kratkih skripti, ali kada skripta postane veća od oko stotinu redova, počinje se sporo izvršavati. Perl može biti težak za čitanje, ali je moćan i na raspolaganju imate veliku CPAN biblioteke. PHP izgleda kao C, nedostaju mu imenski prostori, lako pomiješa kôd i rezultat te ima nekoliko dobrih biblioteka. Python bi mogao biti najlakši za čitanje i pisanje, što je posebna prednost za velike skripte.

## Dodatna literatura

U dodatku se nalazi nekoliko duljih *bash* skripti koje bi mogle biti korisne administratorima sustava. Knjiga *Linux Shell Scripting with Bash* od Kena Burtcha (u nakladi Sams) i *Advanced Bash-Scipting Guide* (*http://www.tldp.org/LDP/abs/html*), su dobri izvori dodatnih informacija. Ako se upustite u proučavanje drugih skriptne jezike, bilo koja knjiga sa slikom životinje na koricama trebala bi biti siguran pogodak (osim ako ne naiđete na *Curious George Learns COBOL* na dječjem odjelu knjižnice).

POGLAVLJE 11

# Rezervne kopije podataka

Računala zakazuju – diskovi se kvare, čipovi se tope, u žicama se događaju kratki spojevi a pića prolijevaju po prijenosnicima. Ponekad su računala ukradena ili su žrtve ljudske pogreške. Mogli biste izgubiti ne samo hardver i softver, nego još važnije, podatke. Vraćanje izgubljenih podataka troši vrijeme i novac. U međuvremenu, klijenti će biti nesretni a vlada bi vam mogla zaračunati kamatu ako zbog izgubljenih podataka niste na vrijeme prijavili porez. Izrada rezervnih kopija svih važnih podataka je jeftino osiguranje protiv skupih katastrofa, a kontinuitet poslovanja također zahtjeva plan za izradu rezervnih kopija podataka.

U ovom poglavlju obradit ćemo nekoliko alata za izradu rezervnih kopija podataka koji mogu biti korisni u različitim okolnostima:

*rsync*
: Dovoljan za većinu korisničkih datoteka. Učinkovito premješta datoteke preko mreže na drugi sustav s kojeg ih možete vratiti ako katastrofa pogodi lokalni sustav

*tar*
: Tradicionalni Unix program za izradu komprimiranih kolekcija datoteka. Izrađuje prikladne pakete podataka koje možete kopirati koristeći druge alate opisane u ovom poglavlju

*cdrecord/cdrtools*
: Zapisuje datoteka na CD ili DVD

*Amanda*
: Automatizira izradu rezervnih kopija na vrpcu. Koristan je u okruženju s velikim količinama podataka

*MySQL*
: Pruža načine za rješavanje specifičnih zahtjeva baza podataka

# Kopiranje korisničkih podataka na poslužitelj pomoću alata rsync

Podaci čije rezervne kopije svakako trebate izraditi su podaci koje je nemoguće ili vrlo teško ponovno izraditi. To su obično *korisnički podaci* koji su se nastajali tijekom mjeseci ili godina rada. *Sistemske podatke* možete relativno lako vratiti ponovnim instaliranjem sa originalnih distribucijskih medija.

Ovdje ćemo se usredotočiti na izradu rezervnih kopija korisničkih podataka sa stolnih računala koja rade pod Linuxom. Poslužitelj za rezervne kopije treba imati dovoljno prostora na disku za pohranu svih korisničkih datoteka. Preporučamo da za tu ulogu namijenite posebno računalo. U velikom uredu diskovi mogu biti postavljeni u RAID (Redundant Array of Independent Disks) konfiguraciju da bi se dodatno zaštitili od gubitka podataka uslijed kvara.

Pomoćni program *rsync* napisan je za kopiranje velikih količina podataka. On može preskočiti datoteke i fragmente koje ste već kopirali te šifrirati prijenos podataka koristeći *ssh* stvarajući tako udaljene rezervne kopije brže i sigurnije nego kada biste koristili tradicionalne alate poput *cp*, *cpio* ili *tar*. Da biste provjerili je li *rsync* instaliran na vašem sustavu, unesite:

```
# rsync --help
bash: rsync: command not found
```

Ako vidite ovu poruku, trebate nabaviti *rsync* paket. Da biste ga instalirali na Debianu unesite:

```
# apt-get install rsync
```

Obično ćete željeti da rezervne kopije zadrže originalne vlasnike i dopuštenja. Stoga ćete trebati osigurati da svi korisnici imaju račune i početne direktorije na poslužitelju za pohranu rezervnih kopija.

## Osnove alata rsync

Sintaksa *rsync* naredbe je sljedeća:

```
rsync opcije izvor odredište
```

Najvažnije opcije za *rsync* su:

*-a*
: Arhiva. Ova opcija ispunjava većinu ranije spomenutih zahtjeva i lakše ju je utipkavati i izgovoriti nego njezin ekvivalent *-Dgloprt*.

*-b*
: Zadaje izradu rezervnih kopija datoteka koje već postoje na odredištu umjesto da ih zamjenjuje novima. Obično nećete koristiti ovu opciju, osim ako ne želite čuvati stare inačice svih datoteka. Ako za odlučite za to, poslužitelj za pohranu rezervnih kopija vrlo brzo će se popuniti.

*-D*
: Zadaje očuvanje uređaja. Ova opcija se koristi pri kopiranju sistemskih datoteka i nije potrebna za korisničke datoteke. Radi samo kad se *rsync* izvršava kao *root*. Podrazumijeva opciju *-a*.

*-g*
: Čuva grupno vlasništvo nad datotekama koje se kopiraju. Ovo je vrlo važno za izradu rezervne kopije. Uključena je u opciju *-a*.

*-H*
: Čuva čvrste veze (engl. *hard link*). Ako se dva imena koja se kopiraju odnose na isti indeksni čvor datoteke (engl. *file inode*), ova opcija održava isti odnos na odredištu. Usporava *rsync* ali njena upotreba se preporuča.

*-l*
: Kopira simboličke veze kao simboličke veze. Skoro uvijek ćete htjeti uključiti ovu opciju. Bez nje, simbolička veza s datotekom bila bi kopirana kao regularna datoteka. Uključena je u opciju *-a*.

*-n*
: „Suho“ kopiranje: vidite koje datoteke bi bile kopirane, ali se kopiranje ne odvija.

*-o*
: Čuva vlasništvo nad datotekama koje se kopiraju. Ovo je važno za rezervne kopije. Uključena je u opciju *-a*.

*-p*
: Čuva dopuštenja za datoteke. Ovo je važno za rezervne kopije. Uključena je u opciju *-a*.

*-P*
: Uključuje opcije *--partial* i *-- progress*.

*--partial*
: Omogućuje djelomični prijenos datoteka. Ako je *rsync* prekinut u radu, moći će završiti prijenos kasnije, kada nastavi.

*--progress*
: Prikazuje napredovanje prijenosa datoteka.

*-r*
: Omogućava rekurziju za prijenos podataka iz svih poddirektorija. Uključena je u opciju *-a*.

*--rsh='ssh'*
: Koristiti SSH za prijenos datoteka. Preporuča se jer podrazumijevani protokol za prijenos (*rsh*) nije siguran. Možete također posatviti varijablu okruženja `RSYNC_RSH` na `ssh` da biste postigli isti učinak.

*-t*
: Čuva vrijeme modifikacije za svaku datoteku. Uključena je u *-a*.

*-v*
: Prikazuje popis datoteka koje se prenose.

*-vv*
: Kao *-v*, ali popisuje i datoteke koje se preskaču.

*-vvv*
: Kao *-vv*, ali se ispisuju i *rsync* informacije za uklanjanje pogrešaka.

*-z*
: Omogućava komprimiranje. Korisnija je ako se rezervne kopije prenose preko sporijeg Interneta nego preko lokalne mreže velike brzine.

Postoji još mnoštvo *rsync* opcija koje bi mogle biti korisne u specijaliziranim situacijama. Možete ih pronaći na stranicama s uputama.

Nakon opcija treba navesti izvor i odredište. `izvor` i `odredište` mogu biti putanje do datoteka na lokalnom računalu na kojem se *rsync* izvodi, *rsync* oznaka poslužitelja (općenito se koristi za poslužitelje s kojih se preuzimaju datoteke), ili oznake `korisnik@računalo:putanja` za *ssh*. Kako *rsync* prima mnogo opcija i dugačkih argumenata koji se neće često mijenjati, u nastavku ćemo napisati *bash* skriptu za njegovo pokretanje.

## Skripta za izradu rezervnih kopija

U ovom odlomku predstavljamo jednostavnu *bash* skriptu koja pravi rezervne kopije podataka s računala na poslužitelj za rezervne kopije. Ime poslužitelja za rezervne kopije je pridruženo varijabli `dest`. Varijabli `user` je dodijeljeno korisničko ime računa koji izvodi skriptu zadavanjem naredbe *whoami* i hvatanjem rezultata u niz. Naredba *cd* mijenja trenutačni direktorij u korisnikov početni direktorij. Testni uvjet `OR` koji slijedi nakon naredbe *cd* prekida skriptu ako dođe do problema. Jedna točka (`.`) sama po sebi zadaje trenutačni direktorij kao argument `izvor`. Za argument `odredište` zadajemo korisničko ime i ime računala da bismo se prijavili. Nakon toga slijedi točka koja zadaje trenutačni početni direktorij na odredištu.

Evo skripte:

```
#!/bin/bash
export RSYNC_RSH=/usr/bin/ssh
dest=backup1
user=$(whoami)
cd || exit 1
rsync -aHPvz . "${user}@${dest}:."
```

Varijabla okruženja `RSYNC_RSH` sadrži ime školjke koju će *rsync* koristiti. Pretpostavljena je */usr/bin/rsh*, pa smo je ovdje promijenili u */usr/bin/ssh*. Skripta kopira sve datoteke iz početnog direktorija korisnika koji ju je pokrenuo u početni direktorij tog korisnika na poslužitelju za smještaj rezervnih kopija. Pogledajmo kako radi tako što ćemo je pokrenuti za našeg pokusnog korisnika (nakon što smo se prijavili na lokalno računalo):

```
amy@desk12:~$ ./backup
Password:
building file list ...
14 files to consider
./
new-brochure.sxw
       37412 100%  503.91kB/s    0:00:00  (1, 62.5% of 16)
sales-plan-2006-08.sxw
       59513 100%    1.46MB/s    0:00:00  (2, 68.8% of 16)
sales-plan-2006-09.sxw
       43900 100%  691.47kB/s    0:00:00  (3, 75.0% of 16)
sales-plan-2006-10.sxw
       41285 100%  453.00kB/s    0:00:00  (4, 81.2% of 16)
```

```
vacation-request.sxw
       15198 100%  154.60kB/s    0:00:00  (5, 87.5% of 16)

sent 185942 bytes  received 136 bytes  24810.40 bytes/sec
total size is 210691  speedup is 1.13
amy@desk12:~$
```

*rsync* kaže da razmatra 14 datoteka. Međutim, izradit će rezervne kopije samo pet datoteka jer se ostalih devet već nalaze na poslužitelju za rezervne kopije i nisu se mijenjale. Ovaj ispis prikazuje napredak prijenosa u postocima i te trajanje prijenosa svake datoteke. Na lokalnim mrežama prijenos će najčašće biti kraći od sekunde za male datoteke i datoteke srednje veličine. Za prijenos preko sporijih veza te za vrlo velike datoteke vidjet ćete podatak koliki je dio datoteke prenesen i procjenu vremena potrebnog za završetak prijenosa.

## Prikazivanje popisa datoteka na poslužitelju za smještaj rezervnih kopija

*rsync* može izraditi popis datoteka na poslužitelju za rezervne kopije. Ovo je korisno za provjeravanje jesu li nove i važne datoteke zaista tamo, kao i za pronalaženje datoteka koje treba vratiti na lokalno računalo jer su izgubljene ili je korisniku potrebna stara inačica.

Da biste dobili popis nemojte navesti argumente `izvor` i `odredište`. Evo jednostavne *bash* skripte koja daje željene rezultate:

```
# !/bin/bash
dest=server1
user=$(whoami)
cd || exit 1
rsync "${user}@${dest}:." | more
```

Izvođenje ove skripte daje rezultate slične sljedećim:

```
amy@desk12:~$ ./backlist
Password:
drwx------  4096 2006/08/09 13:20:41 .
-rw------- 10071 2006/08/09 12:35:21 .bash_history
-rw-r--r--   632 2006/07/27 23:03:06 .bash_profile
-rw-r--r--  1834 2006/07/26 19:59:08 .bashrc
-rwxr-xr-x   108 2006/07/27 23:06:51 .path
-rwxr-xr-x    79 2006/08/09 13:18:34 backlist
-rwxr-xr-x   137 2006/08/09 13:19:29 backrestore
-rwxr-xr-x    88 2006/08/09 13:03:46 backup
-rw-r--r-- 37412 2006/07/17 14:40:52 new-brochure.sxw
-rw-r--r-- 59513 2006/07/19 09:16:41 sales-plan-2006-08.sxw
-rw-r--r-- 43900 2006/07/19 22:51:54 sales-plan-2006-09.sxw
-rw-r--r-- 41285 2006/07/17 16:24:19 sales-plan-2006-10.sxw
-rw-r--r-- 15198 2006/07/10 14:42:23 vacation-request.sxw
drwx------  4096 2006/08/09 13:12:25 .ssh
amy@desk12:~$
```

## Obnavljanje izgubljenih ili oštećenih datoteka

Ni jedan sustav za izradu rezervnih kopija nije dobar ako izgubljene datoteke ne mogu biti obnovljene. Ne samo da moramo biti spremni u slučaju da nas pogodi katastrofa, nego moramo testirati i planove za oporavak i povratak na prethodno stanje da bismo bili sigurni da će sve raditi kada bude bilo potrebno.

Naša skripta za povratak na prethodno stanje je samo malo kompliciranija od prethodne skripte. Dodali smo način za zadavanje pojedinačnih datoteka koje će biti obnovljene:

```
#!/bin/bash
dest=server1
user=$(whoami)
cd || exit 1
for file in "$@" ; do
    rsync -aHPvz "${user}@${dest}:./${file}" "./${file}"
done
```

Da bismo vratili datoteke iz rezervne kopije, jednostavno pokrenemo skriptu navodeći imena datoteka koje treba vratiti u prethodno stanje kao argumente. U sljedećem primjeru namjerno ćemo ukloniti jednu od datoteka i potom sve vratiti u prethodno stanje:

```
amy@desk12:~$ rm sales-plan-2006-10.sxw
amy@desk12:~$ ./backrestore sales-plan-2006-10.sxw
Password:
receiving file list ...
1 file to consider
sales-plan-2006-10.sxw
       41285 100%    6.56MB/s    0:00:00  (1, 100.0% of 1)

sent 42 bytes  received 39299 bytes  6052.46 bytes/sec
total size is 41285  speedup is 1.05
amy@desk12:~$
```

Možemo također vratiti u prethodno stanje sve datoteke odjednom navodeći točku kao ime datoteke.

## Automatska izrada rezervnih kopija

Izrada rezervnih kopija se može automatizirati korištenjem skripti sličnih onima koje se izvode kao *cron poslovi* (o njima smo govorili u prethodnom poglavlju). SSH zahtijeva da se unese korisnička lozinka, pa ćete morati uključiti javne ključeve vaših korisnika u njihove SSH konfiguracije kako bi SSH korisnički računi radili kad korisnici nisu nazočni (primjerice svake noći u 3 sata ujutro).

Na raspolaganju su vam mnoge opcije za izradu rezervnih kopija. Možete izvršavati skriptu *cron* posla na poslužitelju jednom dnevno ili tjedno da sprema rezervne kopije na drugom poslužitelju. Poduzeća s udaljenim uredima mogu redovito raditi rezervne kopije podataka preko Interneta. Rezervne kopije se također mogu spremati na CD-ove, DVD-ove ili vrpce za izradu dugotrajnih arhivskih kopija koje se mogu čuvati na drugim lokacijama.

## tar arhive

Naredba *tar* izrađuje arhivsku datoteku od jedne ili više zadanih datoteka ili direktorija. Također može ispisati sadržaj arhive ili raspakirati datoteke i direktorije iz neke arhive. Datoteka *tar* arhive poznata je još i kao *tarfile* ili *tarball*.

*tar* datoteka pruža nekoliko prednosti u odnosu na direktorij u koji su pohranjene datoteke. Na primjer, znatno pojednostavljuje slanje cijelog direktorija putem elektronske pošte. Direktoriji koji sadrže puno sličnih datoteka komprimiraju se mnogo učinkovitije kada su sve datoteke obuhvaćene u jednoj arhivskoj datoteci.

Uobičajena upotreba *tar* arhiva je da olakšaju distribuciju izvornih datoteka programa otvorenog izvornog koda. U većini slučajeva, *tar* arhive su komprimirane pomoću *gzip* ili *bzip2* programa. Međutim, ako su sve datoteke koje se arhiviraju već komprimirane (što je obično točno za audio, video i OpenOffice.org datoteke), komprimiranje same arhive neće bitno smanjiti njenu veličinu.

Možete nazvati *tar* datoteku kako god želite, ali nastavak imena bi trebao biti jedan od konvencionalnih da bi korisnici znali kako raspakirati datoteku. Najčešći nastavci imena datoteka su:

*.tar*
: Za nekomprimirane *tar* arhive.

*.tar.gz ili .tgz*
: Za *tar* arhive koje su komprimirane *gzip* programom za komprimiranje.

*.tar.bz2 ili .tbz*
: Za *tar* arhive koje su komprimirane *bzip2* programom za komprimiranje.

Sintaksa *tar* naredbe je sljedeća:

```
tar opcije argumenti
```

Opcije se tradicionalno zadaju kao samostalna slova, bez crtice (-), premda mnoge inačice *tara* prihvaćaju i crticu. Najkorisnije opcije su:

*-b*
: Zadaje veličinu bloka (podrazumijevane su jedinice od 512 bajtova)

*-c*
: Zadaje izradu nove arhive.

*-f* `datoteka`
: Zadaje čitanje iz arhive `datoteka` ili pisanje u nju. Ako je argument `datoteka` izostavljen ili je `-`, datoteka arhive se ispisuje na standardni izlaz ili čita sa standardnog ulaza.

*-j*
: Zadaje komprimiranje ili raspakiravanje arhive korištenjem *bzip2* ili *bunzip2*. Arhive komprimirane s *bzip2* obično imaju nastavak imena datoteke *.bz2*.

*-p*
: Čuva dopuštenja za datoteke.

*-t*
: Prikazuje popis datoteka u postojećoj arhivi.

*-v*
: Pri izradi ili raspakiravanju arhiva popisuje sadržaj. Uz opciju *-t* pruža više detalja o popisanim datotekama.

*-x*
: Raspakirava (čita) datoteke iz postojeće arhive.

*-z*
: Zadaje komprimiranje ili raspakiravanje arhive korištenjem pomoćnih programa *gzip* ili *gunzip*. Arhive komprimirane s programom *gzip* obično imaju nastavak imena datoteke *.gz*.

## Izrada nove arhive

*tar* arhivu možete izraditi samo da biste spremili skupinu datoteka u jednu datoteku za vlastite potrebe arhiviranja, ili da biste ih poslali elektroničkom poštom ili objavili, na primjer, na nekom FTP poslužitelju). Evo nekoliko naredbi koje se mogu koristiti za arhiviranje direktorija *work-docs*:

- Za izradu arhive *work-docs.tar* od direktorija *work-docs*:
  ```
  $ tar -cf work-docs.tar work-docs
  ```
- Za izradu komprimirane arhive *work-docs.tar.gz* od direktorija *work-docs*:
  ```
  $ tar -czf work-docs.tar.gz work-docs
  ```
- Za izradu komprimirane arhive *work-docs.tar.bz2* od direktorija *work-docs*:
  ```
  $ tar -cjf work-docs.tar.bz2 work-docs
  ```

## Raspakiravanje arhive

Do sada ste vjerojatno već raspakiravali datoteke iz neke arhive koju ste ranije izradili (kao što je rezervna kopija podataka), iz arhive koju vam je netko poslao elektroničkom poštom ili iz arhive koju ste preuzeli s Interneta (recimo, izvorni kod programa koji trebate).

Prije nego što raspakirate arhivu, trebali biste pregledati njezin sadržaj. Ne želite slučajno zamijeniti datoteke na vašem sustavu s datotekama iz arhive, niti želite napraviti zbrku koju ćete kasnije morati rješavati.

Datoteke u arhivi trebale bi biti smještene unutar direktorija, ali ne rade svi tako, pa trebate biti pažljivi da biste izbjegli raspakiravanje datoteka u trenutačni direktorij. Obično je dobra ideja izraditi novi direktorij na računalu u koji ćete raspakirati *tar* arhivu. To drži raspakirane datoteke podalje od ostalih datoteka, pa se neće pomiješati. To također može spriječiti da se raspakiravanjem prebrišu postojeće datoteke.

Opcija *-t* prikazuje popis datoteka u arhivi i direktorije u kojima će se nalaziti kada se arhiva raspakira. Dodavanje opcije *-v* povećava količinu informacija koje se prikazuju o svakoj datoteci u *tar* arhivi pa ćete vidjeti veličinu svake datoteke i vrijeme kada je zadnji put modificirana. Evo nekoliko primjera naredbi:

- Da biste vidjeli popis datoteke u arhivi *collection.tar:*

  ```
  $ tar -tf collection.tar
  ```

- Da biste vidjeli popis datoteka u arhivi *collection.tar.bz2* s dodatnim detaljima:

  ```
  $ tar -tvjf collection.tar.bz2
  ```

  Da biste raspakirali datoteke iz arhive *collection.tar* u trenutačni direktorij uz zadržavanje originalnih dopuštenja:

  ```
  $ tar -xpf collection.tar
  ```

  Opcija *-x* raspakirava datoteke u trenutačni direktorij. *tar* radi u tišini osim ako nije zadana i opcija *-v* za prikazivanje popisa datoteka. Opcija *-p* čuva originalna dopuštenja, pa će raspakirane datoteke imati ista dopuštenja kao i datoteke koje su bile arhivirane.

- Da biste raspakirali datoteke iz *collection.tar.gz* u trenutačni direktorij, uz zadržavanje originalnih dopuštenja:

  ```
  $ tar -xpzf collection.tar.gz
  ```

- Da biste raspakirali datoteke iz *collection.tar.bz2* u trenutačni direktorij, uz zadržavanje originalnih dopuštenja:

  ```
  $ tar -xpjf collection.tar.bz2
  ```

- Da biste popisali datoteke iz arhive *collection.tar.bz2* i raspakirali ih u trenutačni direktorij, uz zadržavanje originalnih dopuštenja:

  ```
  $ tar -xpvjf collection.tar.bz2
  ```

## Kompletan primjer pakiranja i raspakiravanja s tarom

Sljedeća sesija školjke prikazuje izradu *tar* arhive na temelju direktorija u kojem se nalazi nekoliko datoteka:

```
amy@desk12:~$ ls -dl monthly-reports
drwxr-xr-x  2 amy amy 4096 2006-08-11 14:15 monthly-reports
amy@desk12:~$ ls -l monthly-reports
total 228
-rw-r--r--  1 amy amy 50552 2006-05-09 11:09 mr-2006-04.sxw
-rw-r--r--  1 amy amy 51284 2006-06-06 15:44 mr-2006-05.sxw
-rw-r--r--  1 amy amy 51428 2006-07-06 14:30 mr-2006-06.sxw
-rw-r--r--  1 amy amy 54667 2006-08-07 10:06 mr-2006-07.sxw
amy@desk12:~$ tar -czf monthly-reports-aug.tar.gz monthly-reports
amy@desk12:~$ ls -l monthly-reports-aug.tar.gz
-rw-r--r--  1 amy amy 199015 2006-08-14 12:46 monthly-reports-aug.tar.gz
```

Sljedeća sesija školjke prikazuje ispis sadržaja *tar* arhive:

```
amy@desk12:~$ ls -l monthly-reports-aug.tar.gz
-rw-r--r--  1 amy amy 199015 2006-08-14 12:46 monthly-reports-aug.tar.gz
```

```
amy@desk12:~$ tar -tzf monthly-reports-aug.tar.gz
monthly-reports/
monthly-reports/mr-2006-04.sxw
monthly-reports/mr-2006-05.sxw
monthly-reports/mr-2006-06.sxw
monthly-reports/mr-2006-07.sxw
amy@desk12:~$ tar -tvzf monthly-reports-aug.tar.gz
drwxr-xr-x amy/amy           0 2006-08-11 14:15:12 monthly-reports/
-rw-r--r-- amy/amy       50552 2006-05-09 11:09:12 monthly-reports/mr-2006-04.sxw
-rw-r--r-- amy/amy       51284 2006-06-06 15:44:33 monthly-reports/mr-2006-05.sxw
-rw-r--r-- amy/amy       51428 2006-07-06 14:30:19 monthly-reports/mr-2006-06.sxw
-rw-r--r-- amy/amy       54667 2006-08-07 10:06:57 monthly-reports/mr-2006-07.sxw
amy@desk12:~$
```

Sljedeća sesija prikazuje raspakiravanje sadržaja *tar* arhive:

```
amy@desk12:~$ mkdir extract.dir
amy@desk12:~$ cd extract.dir
amy@desk12:~/extract.dir$ tar -xzf ../monthly-reports-aug.tar.gz
amy@desk12:~/extract.dir$ tar -xvzf ../monthly-reports-aug.tar.gz
monthly-reports/
monthly-reports/mr-2006-04.sxw
monthly-reports/mr-2006-05.sxw
monthly-reports/mr-2006-06.sxw
monthly-reports/mr-2006-07.sxw
amy@desk12:~/extract.dir$ tar -xvvzf ../monthly-reports-aug.tar.gz
drwxr-xr-x amy/amy           0 2006-08-11 14:15:12 monthly-reports/
-rw-r--r-- amy/amy       50552 2006-05-09 11:09:12 monthly-reports/mr-2006-04.sxw
-rw-r--r-- amy/amy       51284 2006-06-06 15:44:33 monthly-reports/mr-2006-05.sxw
-rw-r--r-- amy/amy       51428 2006-07-06 14:30:19 monthly-reports/mr-2006-06.sxw
-rw-r--r-- amy/amy       54667 2006-08-07 10:06:57 monthly-reports/mr-2006-07.sxw
amy@desk12:~/extract.dir$ cd
amy@desk12:~$
```

## Sažetak

Najvažnije što trebate zapamtiti o alatu *tar* je sljedeće:

- Opcija `-c` čita datoteke i *izrađuje* (piše u) *tar* datoteku.
- Opcija `-x` *raspakirava* (čita iz) *tar* datoteke i sprema pojedinačne datoteke koje su tvorile arhivu.

Većina Unix i Linux administratora pomiješala je ove opcije barem jednom.

# Spremanje datoteka na optičke medije

CD i DVD mediji na koje se može snimati, tj. CD-R, DVD-R i DVD+R mediji, omogućuju spremanje datoteka u prigodnom i kompaktnom obliku. Mogu se koristiti za izradu rezervnih kopija koje će se pohraniti u posebnu zaštićenu prostoriju ili za distribuciju softvera i podataka korisnicima ili klijentima. Na jedan CD-R može se pohraniti do 700 MB podataka, dok DVD-R ili DVD+R mogu pohraniti do 4,7 GB. Postoje i dvoslojni DVD+R mediji s kapacitetom od 8,55 GB.

Razlika između DVD-R i DVD+R je u tehnologiji koja se koristi za pozicioniranje lasera na putanju za snimanje. Te dvije tehnologije nisu međusobno kompatibilne, pa ako vaš disk podržava samo DVD-R ili samo DVD+R, morate koristiti odgovarajuće medije za snimanje. Postoje i uređaji koji podržavaju oba formata.

Spremanje datoteka na CD ili DVD nije tako jednostavno ni fleksibilno kao spremanje datoteka na tvrdi disk. Mediji na koje se može snimati više puta mogu zaobići neka ograničenja, ali su skuplji i manje kompatibilni. U ovom odjeljku usredotočit ćemo se na spremanje datoteka CD-R medije. Metode za spremanje na DVD su slične.

CD s podacima sastoji se od polja sektora, a svaki sektor ima 2048 bajtova. Poseban sustav datoteka *ISO-9660* koristi se za organiziranje datoteka na CD-u tako da im se može pristupiti na širokom rasponu računala i ostalih uređaja. Noviji linijski CD uređaji također podržavaju medije u ISO-9660 formatu, pa mogu pristupiti glazbenim datotekama u komprimiranim formatima kao što je MP3. DVD-ovi koriste noviji sustav datoteka nazvan Universal Disk Format (UDF).

Da bi snimili podatke na CD ili DVD, većina snimača zahtijeva kontinuirani tok podataka do medija. Ako podaci ne budu dostupni kada ih laser pokuša snimiti, morat će stati, što prekida kontinuitet snimanja. Te metode snimanja CD-ova bile su osmišljene za sporija računala kako bi se maksimalizirala pouzdanost snimaka. Današnja računala, iako brža, još uvijek su suočena s izazovom da osiguraju neprekidan tok podataka modernim, sve bržim, uređajima za snimanje. Mnogi snimači danas podržavaju Buffer Underrun Free tehnologiju koje im omogućava da nastave snimanje čak i ako se međuspremnik za podatke u nekom trenutku isprazni.

Datoteke koje će biti snimljene na medij prvo se sakupljaju u datoteci nazvanoj *ISO datoteka slika* (engl. *ISO image file*) koja ima nastavak imena *.iso* . Ta se datoteka potom izravno prenosi na CD-R medij. Moguće je snimiti datoteke izravno na CD-R bez prethodne izrade .iso datoteke, ali ta metoda povećava rizik od prekidanja toka podataka u nezgodnom trenutku.

Program potreban da bi se snimio CD ili DVD na Linuxu nalazi se u paketu *cdrecord* (obratite pozornost da se tom paketu ime mijenja u *cdrtools*). Ako taj paket nije instaliran na vašem sustavu, trebali biste ga instalirati koristeći metode koje ste već naučili. U Debian odzivniku zadajte naredbu:

```
# apt-get install cdrecord mkisofs
```

Debian 4.0 rasporedio je paket *cdrecord* u *wodim*. Drugi paketi uključuju *dvd+rw-tools* (opisan na *http://www.debianhelp.co.uk/burningdvd.htm*) i *K3b* (*http://www.k3b.org*).

## Pristupanje CD-R uređaju

Linux podržava snimanje na IDE ATAPI CD-R uređajima preko posebnog upravljačkog programa *ide-scsi*. Većina Linux distribucija ima ovaj upravljački program u jezgri. Ako ga vaš sustav nema, , trebat ćete učitati modul upravljačkog programa (instalirati ga ako je potrebno) ili možda ponovno prevesti jezgru.

Upravljački program *ide-scsi* emulira SCSI uređaj za potrebe programa koji su projektirani tako da rade samo sa SCSI uređajima. Ako je *ide-scsi* upravljački progam aktivan, IDE ATAPI CD i DVD uređaji izgledat će kao da su SCSI uređaji.

Sljedeća naredba će popisati SCSI uređaje na sustavu tako da možete pronaći broj emuliranog CD-R pisača. Na popisu će biti i drugi uređaji, uključujući i prave SCSI uređaje ako su priključeni na računalo. Zadajte naredbu kao *root*:

```
# cdrecord -scanbus
```

Rezultat bi mogao izgledati slično ovome:

```
Cdrecord-Clone 2.01 (i686-pc-linux-gnu) Copyright (C) 1995-2004 J&#246;rg Schilling
scsidev: 'ATA'
devname: 'ATA'
scsibus: -2 target: -2 lun: -2
Linux sg driver version: 3.5.27
Using libscg version 'schily-0.8'.
scsibus1:
        1,0,0   100) 'SONY    ' 'CD-RW  CRX195E1 ' 'ZYS5' Removable CD-ROM
        1,1,0   101) 'DVD-16X ' 'DVD-ROM BDV316E ' '0052' Removable CD-ROM
        1,2,0   102) *
        1,3,0   103) *
        1,4,0   104) *
        1,5,0   105) *
        1,6,0   106) *
        1,7,0   107) *
```

Potražite uređaj čiji opis odgovara vašem CD-R pisaču. Ako imate više uređaja, ime proizvođača i model bi trebali pomoći u prepoznavanju odgovarajućeg. U ispisu biste trebali vidjeti barem CD-R ili CD-RW. U ovom primjeru CD pisač je emulirani SCSI uređaj 1,0,0.

Ako *ide-scsi* upravljački program nije instaliran ili nije aktivan, mogli biste dobiti ovakav rezultat:

```
Cdrecord-Clone 2.01 (i686-pc-linux-gnu) Copyright (C) 1995-2004 J&#246;rg Schilling
cdrecord: No such file or directory. Cannot open '/dev/pg*'. Cannot open SCSI driver.
cdrecord: For possible targets try 'cdrecord -scanbus'.
cdrecord: For possible transport specifiers try 'cdrecord dev=help'.
cdrecord:
cdrecord: For more information, install the cdrtools-doc
cdrecord: package and read /usr/share/doc/cdrecord/README.ATAPI.setup .
```

Ako dobijete takav rezultat, trebat ćete aktivirati *ide-scsi* upravljački program prije samog snimanja.

## Zadavanje podrazumijevanih postavki

Možete konfigurirati brojne parametre programa *cdrecord*. Na primjer, možete konfigurirati *cdrecord* tako da prepoznaje imena uređaja za snimanje (da ne morate pamtiti njihove brojeve) i zadati podrazumijevani uređaj. Da biste konfigurirali *cdrecord*, prijavite se (ili koristite `su` da se prebacite) kao *root*. Zatim izradite tekstualnu datoteku:

```
# vi /etc/default/cdrecord
```

Dodat ćemo sljedeće redove teksta u ovu datoteku da bismo odabrali uređaje prikazane u rezultatu naredbe *cdrecord -scanbus*. Morat ćete promijeniti te vrijednosti kako bi se poklapale sa vrijednostima na vašem sustavu. Koristite koja god imena želite umjesto `cd` i `dvd`. Bijeli prostor između polja u svakom redu su tabulatori, ne razmaci:

```
CDR_DEVICE=cd
cd=1,0,0        -1        -1        ""
dvd=1,1,0       -1        -1        ""
```

Ako je Linux jezgra koju koristite inačice 2.6, najvjerojatnije ćete trebati zadati uređaje s prefiksom `ATA:`, zbog redizajna upravljačkog programa. U tom slučaju, konfiguracijska datoteka mogla bi izgledati ovako:

```
CDR_DEVICE=cd
cd=ATA:1,0,0    -1        -1        ""
dvd=ATA:1,1,0   -1        -1        ""
```

Možete zadati i podrazumijevanu brzinu snimanja za svaki uređaj, odmah nakon broja uređaja. `-1` zadaje da treba koristiti podrazumijevanu vrijednost. Sljedeći broj je veličina FIFO međuspremnika. Još jednom, `-1` zadaje podrazumijevanu vrijednost na Linux sustavu. Zadnja stavka u redu omogućuje vam da proslijedite opciju specifičnu za upravljački program. Ostavili smo prazan niz.

Novije inačice programa *cdrecord* podržavaju opciju `driveropts=burnfree` koja štiti od pražnjenja međuspremnika.

## Pripremanje datoteka za snimanje na CD-R

Naredba *mkisofs* stvara datoteku sliku sa ISO datotečnim sustavom. Ona bi trebala sadržavati sve datoteka koje će biti snimljene na CD-R. Za ovu naredbu postoji mnogo opcija ša ćemo ovdje navesti samo one važnije koje ćemo koristiti:

*-J*
: Uključuje Joliet imena za kompatibilnost s Windowsima.

*-r*
: Uključuje Rock Ridge imena za kompatibilnost s Unix/Linux sustavima.

*-v*
: Uključuje doslovni režim za prikazivanje statusa.

*-V* `identifikator`
: Zadaje identifikator volumena s kojim će biti imenovan disk koji se snima.

`-o datoteka`
: Zadaje ime datoteke slike koja se stvara.

Evo primjera naredbe za uključivanje svih datoteka iz zadanog direktorija:

```
# mkisofs -JrvV "disc name" -o backup.iso /home/amy
```

Ova naredba prikazuje brojne poruke na zaslonu. Među njima ćete pronaći i procjenu vremena preostalog do kraja operacije. Ako ne želite gledati poruke izostavite opciju *-v* iz naredbe.

## Snimanje na CD-R

Sad možete snimiti CD-R sa ISO slikom koju ste izradili. Da biste snimili podatke na disk prijavite se (ili koristite naredbu su za prebacivanje) kao *root*. Dopuštenja *root* korisnika su potrebna kako bi program *cdrecord* mogao pristupiti sirovom SCSI sloju, prilagoditi prioritete procesa i spriječiti premještanje međuspremnika za snimanje u datoteku za razmjenu (engl. *swap file*). Snimanje CD-a sastoji se od nekoliko koraka koji strogo slijede jedan iza drugog pa je dobra ideja što manje opterećivati sustav tijekom snimanja kako ništa ne bi poremetilo raspored izvođenja tih koraka.

Ako koristite prepisivi CD medij (CD-RW) u uređaju koji ga može koristiti, trebate izbrisati postojeći sadržaj prije snimanja novog:

```
# cdrecord blank=fast padsize=63s -pad -dao -v -eject
```

Neki CD pisači zahtjevaju izbacivanje medija kako bi se pripremili za sljedeću operaciju. Osim ako vaš uređaju to ne zahtijeva, koristite opciju *-eject* kao u ovom primjeru.

Za snimanje ISO slike koju ste izradili u prethodnom odjeljku, unesite:

```
# cdrecord padsize=63 -pad -dao -v -eject backup.iso
```

Izbjegavajte bilo kakav rad na računalu dok se CD ili DVD snima.

Neki moderni pisači podržavaju posebne tehnologij,e kao što je burnfree, koje pomažu da se izbjegnu problemi kada računalo nije dovoljno brzo. Diskovi snimljeni uz pomoć ovih „popravljača“ možda neće biti kompatibilni s nekim starijim uređajima. Ako se neki snimljeni diskovi ne mogu pročitati, snimajte ih na manjoj brzini. Možete promijeniti brzinu navodeći opciju *speed=*, što je dokumentirano na stranici s priručnikom za *cdrecord*. Smanjivanje brzine snimanja može biti vrlo važno ako se datoteka sa ISO slikom koja se snima na medij nalazi na poslužitelju i pristupate joj preko mreže.

Nekim IDE ATAPI CD čitačima potrebno je popunjavanje (engl. *padding*) kako bi pravilno radili sa operacijama čitanja unaprijed, što Linux i drugi sustavi obično rade. Možda ćete ustanoviti da na novim CD uređajima sve radi i bez popunjavanja, ali kako se ovaj problem javlja za vrijeme čitanja, trebali biste uvijek uključiti popunjavanje kako biste osigurali da će i stariji uređaji moći čitati diskove koje snimite. U suprotnom, mogli biste ustanoviti da privremeno zamjensko računalo ne može pročitati CD sa vrlo važnim podacima.

## Provjera snimka

Nakon što ste snimili CD ili DVD dobro je provjeriti može li se ispravno pročitati. Medij može biti oštećen ili ste slučajno pomaknuli računalo tijekom snimanja, što je pomaknulo lasersku zraku izvan staze. Pravilan način za provjeru snimka je usporediti snimljene sektore sa sektorima na tvrdom disku, ili generirati kontrolne zbrojeve podataka (checksum) iz tih sektora i usporediti ih. Obje metode se moraju koristiti isključivo sa stvarnim sektorima, a ne sa sektorima za popunjavanje. Sljedeća *bash* skripta čini ovu verifikaciju lakom kad je dostupna originalna datoteka ISO slike:

```
#!/bin/bash
if [[ $# -lt 1 ]] ; then
    echo "usage: isomd5 <file_or_device> ..." 1>&2
    exit 1
fi
for name in "$@" ; do
    isoinfo -di "${name}" 1>/dev/null || exit 1
done
for name in "$@" ; do
    count=( $( isoinfo -di "${name}"    \
        | egrep "^Volume size is: " ) )
    count="${count[3]}"
    bsize=( $( isoinfo -di "${name}"    \
        | egrep "^Logical block size is: " ) )
    bsize="${bsize[4]}"
    md5=$( dd                           \
        if="${name}"                    \
        ibs="${bsize}"                  \
        obs=4096 count="${size}"        \
        2>/tmp/isomd5.$$.err            \
        | md5sum )
    if [[ $? != 0 ]] ; then
        cat /tmp/isomd5.$$.err
        rm -f /tmp/isomd5.$$.err
        exit 1
    fi
    rm -f /tmp/isomd5.$$.err
    echo "${md5:0:32}" "" "${name}"
done
```

Ova skripta radi tako što uzima broj sektora koje koristi ISO datotečni sustav datoteke slike. Ona ograničava broj sektora koji se učitavaju u MD5 program za izračunavanje kontrolnog zbroja na točan broj korištenih sektora. Time se izbjegava čitanje sektora za popunjavanje čiji broj može varirati.

Tu skriptu zovemo *isomd5*. Zadajte joj datoteku sa ISO slikom i CD uređaj koji ćete koristiti za čitanje CD-R medija (otvorite ladicu uređaja pa ju ponovno zatvorite ako ste upravo snimili CD). Trebali biste dobiti rezultat sličan ovome:

```
amy@desk12:~$ isomd5 backup.iso /dev/sr0
d41d8cd98f00b204e9800998ecf8427e  backup.iso
d41d8cd98f00b204e9800998ecf8427e  /dev/sr0
amy@desk12:~$
```

Kontrolni zbroj koji daje MD5 program je 32-znamenkasti heksadecimalni dio. Ako on nije isti za ISO slikovnu datoteku i sadržaj CD-R diska, snimak nije dobar.

Snimak s pogreškama se sarkastično naziva „podmetač za šalicu". Možete ga koristiti da zaštitite površinu stola od stvaranja ružnih kružnih tragova čaša i šalica, ali za razliku od pravog podloška, ovaj će se razletiti u gomilu oštrih komadića ako ga stavite u mikrovalnu pećnicu.

Kad zapisivanje na disk zakaže, pokušajte sljedeće:

1. Ponoviti snimanje s drugim praznim medijem.
2. Snimati na manjoj brzini.
3. Koristiti drugu seriju ili drugu vrstu praznih diskova.

Ako diskovi i dalje ne budu dobri, možda vam je CD pisač oštećen.

### Rezervne kopije na DVD-u

Koraci navedeni u ovom odlomku su specifični za snimanje CD medija, ali se i DVD mediji mogu snimati na sličan način, korištenjem istog softvera iz *cdrecord* ili *cdrtools* paketa. Neki DVD mediji – posebice r DVD-RAM mediji s kojima se rijetko susrećemo – mogu raditi slično tvrdim diskovima ali zahtijevaju poseban uređaj koji podržava taj režim rada.

## Izrada rezervnih kopija i arhiviranje na vrpcu s Amandom

Vrpca je još uvijek popularan medij za izradu rezervnih kopija. Amanda – *Advanced Maryland Automated Network Disk Archiver* – je paket otvorenog izvornog koda koji upravlja snimanjem rezervnih kopija na vrpcu. Razvijen na Sveučilištu Maryland, ovaj program dio je mnogih Linux distribucija, uključujući Debian. Njegove značajke uključuju:

- Korištenje tradicionalnih Unix formata za rezervne kopije, kao što su *tar* i *dump*.
- Rad preko lokalne mreže uz spremanje rezervnih kopija podataka na glavni poslužitelj s vrpcama.
- Podršku za izradu rezervnih kopija Windows klijenata preko sustava za dijeljenje datoteka.
- Podršku za standardne uređaje s vrpcom, mjenjače vrpci, *jukebox* uređaje i slagače.
- Mogućnost uravnoteživanja dnevnih rezervnih kopija tijekom višednevnog ciklusa.
- Podršku za inkrementalne rezervne kopije za zapisivanje dnevnih promjena.
- Komprimiranje podataka, bilo na klijentu ili poslužitelju ili preko uređaja s hardverskim komprimiranjem.

- Sprječavanje slučajnog presnimavanja pogrešnog medija.
- Strategiju koja omogućava fazno ili odgođeno snimanja na medije.
- Provjeru identiteta kroz Kerberos ili vlastitu shemu.
- Šifriranje podataka za zaštitu prilikom prijenosa preko mreže.

## Instaliranje Amande

Amanda ima klijentsku i poslužiteljsku komponentu. Klijent se instalira na računalo na kojem se nalaze podaci čiju kopiju treba izraditi. Poslužiteljska komponenta se instalira na računalo koje obavlja izradu kopija i snima podatke na vrpcu.

Zadajte sljedeću naredbu za instaliranje Amande na poslužitelju:

```
# apt-get install amanda-server
```

Zadajte sljedeću naredbu za instaliranje Amande na svaki klijentski Linux stroj:

```
# apt-get install amanda-client
```

Kad instalirate ove pakete, bit će instalirani i ostali potrebni paketi. Ako želite koristiti *amplot* program u Amandi, trebat ćete instalirati i *gnuplot* paket.

Amanda koristi datoteke iz brojnih direktorija. Te se postavke mogu konfigurirati, ali podrazumijevane vrijednosti su:

*/etc/amanda*
: Konfiguracijske datoteke (poslužitelj).

*/root*
: Datoteka */root/.amandahosts*

*/usr/man/man8*
: Stranice s uputama.

*/usr/share/doc/amanda-common*
: Datoteke s dokumentacijom.

*/usr/share/doc/amanda-client*
: Datoteke s dokumentacijom specifičnom za klijente.

*/usr/lib*
: Dijeljene datoteke koje koriste Amandini programi.

*/usr/lib/amanda*
: Pozadinski servisi i unutarnji pomoćni programi.

*/usr/bin*
: Naredbeni programi.

*/var/lib/amanda*
: Status izvođenja, dnevnik i ostale datoteke.

## Konfiguriranje Amande

Datoteka */etc/services* bi već trebala imati onose sa sljedećim imenima i ulazima. Ako ti zapisi ne postoje, uredite datoteku */etc/services* i dodajte ih na kraj. Komentari su opcionalni:

```
/etc/services:
amanda          10080/udp       # amanda servisi za izradu rezervnih kopija
amandaidx       10082/tcp       # amanda servisi za izradu rezervnih kopija
amidxtape       10083/tcp       # amanda servisi za izradu rezervnih kopija
```

Možete urediti i */etc/inetd.conf* datoteku koja bi trebala sadržavati sljedeće unose:

```
/etc/inetd.conf: (for clients)
amanda    dgram  udp wait   backup /usr/sbin/tcpd /usr/lib/amanda/amandad

/etc/inetd.conf: (for server)
amandaidx stream tcp nowait backup /usr/sbin/tcpd /usr/lib/amanda/amindexd
amidxtape stream tcp nowait backup /usr/sbin/tcpd /usr/lib/amanda/amidxtaped
```

Prvi unos nazvan *amanda* potreban je na svim klijentima. Druga dva unosa potrebni su samo na poslužitelju. Ako ti redovi ne postoje, uredite datoteku */etc/inetd.conf* i dodajte ih na kraj.

Amanda koristi slučajne ulaze nakon početne komunikacije. Trebali biste koristiti Amandu preko Interneta samo preko VPN veze. To sprječava nepotrebno otvaranje ulaza prema lokalnoj mreži.

Amanda se izvršava kao korisnik *backup* sa dopuštenjima grupe *disk*. Trebat ćete zadati dopuštenja za sve datoteke čije će rezervne kopije biti izrađene tako da ih Amanda može pročitati.

Amanda poslužitelj treba biti dobro povezan s lokalnom mrežu, sa dovoljno propusnog kapaciteta za količinu podataka koji se prenose. Trebao bi imati vrlo veliki disk, sa dovoljno mjesta da pohrani dvostruko više podataka od najveće količine koja nastane u jednoj sesiji izrade rezervnih kopija. Potreban mu je i brz procesor ako će softverski obavljati komprimiranje podataka.

Amanda podržava višestruke konfiguracije. Svaka konfiguracija se sastoji od skupa od tri datoteke u poddirektoriju direktorija */etc/amanda*:

*amanda.conf*
: Glavna konfiguracijska datoteka. Uređujete je da biste zadali *disklist* (pogledajte sljedeću stavku), uređaj s vrpcom, učestalost izrade kopija, adresu elektronske pošte, formate izvještaja i ogroman broj drugih opcija.

*disklist*
: Ova datoteka zadaje računala i njihove diskove za koje treba izraditi sigurnosne kopije.

*tapelist*
: Ova datoteka sadrži popis aktivnih vrpci, uključujući datume kad je svaka od njih snimljena. Amanda upravlja ovom datotekom, pa ju možete pregledati ali ne biste ju trebali uređivati.

Detaljan opis svih Amandinih opcija zauzeo bi nekoliko stranica, pa ćemo taj posao prepustiti vama. Primjeri ovih datoteka s korisnim komentarima nalaze se u direktoriju */ets/amanda/DailySet1* koji nastaje kada instalirate paket *amanda-server* za Debian. Za više detalja o konfiguracijskim datotekama, pogledajte Amandine stranice s uputama (MAN) ili *http://wiki.zmanda.com*.

Amanda vodi zapisnik za svaku izradu rezervnih kopija koju obavi. Detaljni izvještaji šalju se na e-mail adresu zadanu u opciji `mailto` *amanda.conf* konfiguracijske datoteke. Trebali biste redovito pregledavati izvještaje da biste vidjeli jesu li se dogodile kakve pogreške te koliko je operacija trajala.

### Vraćanje podataka iz rezervnih kopija

Amanda koristi standardne Unix formate za rezervne kopije (*tar* ili *dump*), što zadajete u konfiguracijskoj datoteci. To omogućava vraćanje podataka pohranjenih na vrpcama čak i ako Amanda više ne postoji. To može biti važnosti kod vraćanja datoteka nakon potpunog kvara diska.

Amanda pruža alate za indeksirani oporavak kojima se mogu vratiti samo odabrane datoteke. Postavite `index` na `yes` da bi Amanda izradila potrebni indeks datoteka. Stranica s uputama *amrecover* pruža sve detalje.

## Izrada rezervnih kopija podataka iz MySQL baze

Do sad smo pohranjivali rezervne kopije datoteka i direktorija. Baze podataka imaju posebne zahtjeve s kojima se moramo pozabaviti. U našim primjerima koristimao MySQL, ali ista načela vrijede za PostgreSQL i druge relacijske baze podataka.

Ako vaš MySQL poslužitelj ne treba biti raspoloživ 24 sata dnevno, 7 dana u tjednu, brza i jednostavna procedura za izradu sigurnosnih kopija je:

1. Zaustavite MySQL poslužitelj:

   ```
   # /etc/init.d/mysqld stop
   ```

2. Kopirajte MySQL datoteke s podacima i direktorije. Na primjer, ako je vaš MySQL direktorij */var/lib/mysql* i želite ga spremiti u */tmp/mysql-backup*, zadajte:

   ```
   # cp -r /var/lib/mysql /tmp/mysql-backup
   ```

   Umjesto *cp* možete koristiti *rsync*, *tar*, *gzip* ili druge naredbe spomenute u ovom poglavlju.

3. Ponovno pokrenite poslužitelj:

   ```
   # /etc/init.d/mysqld start
   ```

Izrada rezervne kopije baze podataka je kompliciranija ako ne možete zaustaviti poslužitelj. Ako imate međusobno neovisne MyISAM tablice (bez vanjskih ključeva

ili transakcija), mogli biste svaku posebno zaključati, kopirati datoteke i zatim otključati. Ali, možda imate InnoDB tablice ili bi netko mogao napisati transakciju u kojoj se koristi više tablica. Srećom, postoji nekoliko razumnih nekomercijalnih rješenja, uključujući *mysqlhotcopy*, *mysqlsnapshot*, replikaciju i *mysqldump*.

*mysqlhotcopy* je Perl skripta koja radi sirovu kopiju ISAM ili MyISAM tablica dok je poslužitelj pokrenut. Na stranicama s uputama opisane su mnoge opcij, ali evo ukratko kako da izradite rezervnu kopiju baze podataka *drupal*:

```
# mysqlhotcopy -u korisnik -p lozinka drupal /tmp
Locked 57 tables in 0 seconds.
Flushed tables (`drupal`.`access`, `drupal`.`accesslog`, `drupal`.`aggregator_
category`, `drupal`.`aggregator_category_feed`, `drupal`.`aggregator_category_item`,
`drupal`.`aggregator_feed`, `drupal`.`aggregator_item`, `drupal`.`authmap`, `drupal`.
`blocks`, `drupal`.`book`, `drupal`.`boxes`, `drupal`.`cache`, `drupal`.`client`,
`drupal`.`client_system`, `drupal`.`comments`, `drupal`.`contact`, `drupal`.`file_
revisions`, `drupal`.`files`, `drupal`.`filter_formats`, `drupal`.`filters`,
`drupal`.`flood`, `drupal`.`forum`, `drupal`.`history`, `drupal`.`locales_meta`,
`drupal`.`locales_source`, `drupal`.`locales_target`, `drupal`.`menu`, `drupal`.
`node`, `drupal`.`node_access`, `drupal`.`node_comment_statistics`, `drupal`.`node_
counter`, `drupal`.`node_revisions`, `drupal`.`permission`, `drupal`.`poll`,
`drupal`.`poll_choices`, `drupal`.`poll_votes`, `drupal`.`profile_fields`, `drupal`.
`profile_values`, `drupal`.`role`, `drupal`.`search_dataset`, `drupal`.`search_
index`, `drupal`.`search_total`, `drupal`.`sequences`, `drupal`.`sessions`, `drupal`.
`system`, `drupal`.`term_data`, `drupal`.`term_hierarchy`, `drupal`.`term_node`,
`drupal`.`term_relation`, `drupal`.`term_synonym`, `drupal`.`url_alias`, `drupal`.
`users`, `drupal`.`users_roles`, `drupal`.`variable`, `drupal`.`vocabulary`,
`drupal`.`vocabulary_node_types`, `drupal`.`watchdog`) in 0 seconds.
Copying 171 files...
Copying indices for 0 files...
Unlocked tables.
mysqlhotcopy copied 57 tables (171 files) in 1 second (1 seconds overall).
```

*mysqlsnapshot* se još lakše koristi. Stvara rezervne kopije svih ISAM ili MyISAM tablica na poslužitelju i smješta ih u po jednu *tar* datoteku za svaku bazu podataka:

```
# ./mysqlsnapshot -u korisnik -p lozinka -s /tmp --split -n
checking for binary logging... ok
backing up db drupal... done
backing up db mysql... done
backing up db test... done
snapshot completed in /tmp
```

*mysqlsnapshot* ćete pronaći na *http://jeremy.zavodny.com/mysql/mysqlsnapshot*.

Ako ste postavili MySQL replikaciju da bude dostupna 24x7, možete raditi rezervne kopije sa sporednog poslužitelja koristeći jednu od metoda koje smo upravo opisali. Trebat ćete spremiti i informacije o replikaciji (dnevnike, konfiguracijske datoteke itd.). Za više detalja pogledajte poglavlja 7 i 9 knjige *High Performance MYSQL* koju su napisali Jeremy D. Zawodny i Derek J. Balling (u nakladi O'Reilly Media).

Za dodatnu zaštitu od hardverskih pogrešaka (ali ne i ljudski) zadajte replikaciju i opremite pomoćni (i/ili glavni) poslužitelj RAID1 diskovima.

Mnoge MySQL lokacije sele podatke iz MyISAM u InnoDB tablice kako bi dobile prave transakcije baza podataka i bolje performanse zapisivanja. Autori InnoDB modula nude komercijalni proizvod InnoDB Hot Backup za izradu InnoDB rezervnih kopija "na živo". Možete ga naručiti na adresi *http://www.innodb.com/order.php*.

Zadnja metoda je obično prva spomenuta u većini dokumentacije: *mysqldump*. Umjesto sirove (doslovne) kopije, *mysqldump* stvara ASCII kopiju zadanih baza podataka i tablica. Radi sa svim tipovima MySQL tablica, uključujući i InnoDB. Relativno je spora, a tekstualne datoteke koje stvara su velike premda se prilično dobro komprimiraju. Korisno je s vremena na vrijeme izraditi i tkave kopije jer sadrže jednostavnu skriptu za restauriranje baza podataka i tablica. Možete koristiti programe za uređivanje, *grep* i druge alate za rad s tekstom za pretraživanje ili mijenjanje ovih datoteka.

Da biste zaključali sve tablice i smjestili ih u jednu datoteku, zadajte:

```
# mysqldump -u korisnik -p lozinka -x --all-databases >/tmp/mysql.dump
```

Možete *preusmjeriti* rezultat na *gzip* da biste uštedjeli nešto vremena i prostora na disku:

```
# mysqldump -u korisnik -p lozinka -x --all-databases | gzip >/tmp/mysql.dump.gz
```

Novi alat otvorenog izvornog koda (možete ga besplatno preuzeti s Interneta, ali morate platiti za podršku) *Zmanda Recovery Manager for MySQL* pruža korisno sučelje za mnoge od ovih alternativa. Zmanda Web lokacija (*http://www.zmanda.com/backup-mysql.html*) sadrži više informacija, ali mi ćemo spomenuti samo neke važnije značajke:

- Ima sučelje odzivnika.
- Pravi rezervne kopije lokalnih baza podataka ili udaljenih baza podataka preko SSL-a.
- Šalje status posla izrade rezervnih kopija elektroničkom poštom.
- Radi sa svim tipovima tablica, uključujući InnoDB.
- Ne pruža nikakve nove metode za izradu rezervnih kopija. Umjesto toga, bira između *mysqldump*, *mysqlhotcopy*, MySQL replikacije i LVM snimaka.
- Podržava oporavak/vraćanje sustava u prethodno stanje do određene transakcije ili vremenske točke.

Zmanda pruža *.tar.gz* i *.rpm* datoteke za mnoge Linux distribucije. Upute za instaliranju na Debianu pogledajte na *http://www.howtoforge.com/mysql_zrm_debian_sarge*.

DODATAK

# Primjeri bash skripti

Ovaj dodatak sadrži nekoliko skripti koje vam mogu biti korisne u svakodnevnom radu, ili poslužiti kao modeli za pisanje drugih skripti. Možete ih preuzeti s adrese *http://www.centralsoft.org*.

## Dodavanje korisnika

Ako je protok ljudi u vašoj organizaciji veliki (kao na sveučilištu, gdje novi studenti dolaze jednom ili više puta godišnja) ova skripta će vam pomoći da ih brzo dodate u sustava. Ona čita datoteku s popisom informacija o svakom korisniku i poziva *useradd* s odgovarajućim argumentima (pogledajte poglavlje 8 za više detalja o naredbi *useradd* i njenim inačicama):

```
#!/bin/bash

expiredate=2009-02-18

if [[ -z "$1" ]] ; then
    echo ""
    echo "Please give exactly one file name."
    echo "The file will have one user per line."
    echo "Each line will have:"
    echo "    username"
    echo "    group"
    echo "    personal real name"
    echo ""
    echo "Sample line:
    echo "alfredo marketing Alfredo de Darc"
    exit 1
fi

cat "$1" | while read username groupname realname
do
    # Preskače prazne redove.
    if [[ -z $username || -z $groupname || -z $realname ]]; then
        continue
    fi
```

```
    # Provjerava da li korisnik već postoji.
    # Ako postoji, prijavljuje stanje i preskače tog korisnika.
    result=$( egrep "^$username:" < /etc/passwd )
    if [[ -n "$result" ]] ; then
        echo "User '$username' already exists"
        continue
    fi

    # Provjerava da li grupa već postoji.
    # Ako ne postoji, dodaje grupu.
    result=$( egrep "^$groupname:" < /etc/group )
    if [[ -z "$result" ]] ; then
        groupadd "$groupname"
    fi

    # Dodaje korisnika.
    useradd    -c "$realname" \
        -d "/home/$username"  \
        -e "$expiredate"      \
        -f 365                \
        -g "$groupname"       \
        -m                    \
        -s /bin/bash          \
        "$username"
    if [[ $? == 0 ]]; then
        echo "Successfully added user '$username'."
    else
        echo "Error adding user '$username' (group \
             '$groupname', real name '$realname')"
        exit 1
    fi

done
```

# Generator lozinki

Evo skripte koja generira lozinku zadane duljine sa ASCII znakovima:

```
#!/bin/bash
n="$1"
[[ -n "$n" ]] || n=12
if [[ $n -lt 8 ]]; then
    echo "A password of length $n would be too weak"
    exit 1
fi
p=$( dd if=/dev/urandom bs=512 count=1 2>/dev/null \
    | tr -cd 'a-zA-ZO-9' \
    | cut -c 1-$n )

echo "${p}"
```

Ako vam ova skripta ima smisla, zaslužili ste nagradu*. Dok ste vi vani, mi ćemo pažljivije pogledati pogreške koje postoje u ovom kodu.

Ovaj kôd je tipičan primjer što možete naslijediti od svog prethodnika: nema komentara, imena varijabli su misteriozna i nedostaju još samo magične riječi. Kako želite svijet učiniti boljim mjestom, ima nekoliko stvari koje možete učiniti kada pišete skriptu poput ove.

Najmanje što možete učiniti jeste dodati komentare s opisom koda. Ti komentari bi trebali biti podijeljeni na dva dijela: opći opis u zaglavlju skripte koji opisuje što bi argumenti proslijeđeni skripti trebali zadavati ili koje su podrazumijevane vrijednosti i izravna objašnjenja kompliciranih dijelova na odgovarajućim mjestima u kodu. Ne gubite vrijeme objašnjavajući osnovne naredbe jer ih onaj tko održava skriptu vjerojatno već poznaje. Međutim, tamo gdje upotrijebite egzotičniju inačicu naredbe, trebali biste izravno objasniti njezine učinke i kako ste ih postigli.

Načelno, trebali biste nastojati dokumentirati *rezultate* koje želite dobiti od skupova naredbi i zašto to radite na način koji ste odabrali.

Evo detaljnog objašnjenja koda za generator lozinki kakvo vjerojatno nećete susresti u stvarnom svijetu. Skripta počinje uobičajenim početnim komentarom koji govori sustavu da pokrene *bash* interpretator. Zatim dodjeljujemo niz prvog argumenta varijabli `n` što će biti broj znakova koje treba generirati. Stavljamo ga u navodnike jer bi to mogao biti prazan niz ako se skripta pokrene bez argumenta. Zatim se taj niz testira da bi se utvrdilo je li stvarno *null*. Argument *-n* znači „duljina koja nije nula“, pa je test zapravo *true* ako je niz zadan.

Dvije okomite crte će izvršiti zadatak koji slijedi ako je test neuspješan. To nameće podrazumijevanu duljinu od 12 znakova za lozinku. Sljedeća četiri reda provjeravaju je li zadani broj znakova prekratak. Odlučili smo (na temelju uobičajenih preporuka stručnjaka za sigurnost) da lozinka ne bi trebala biti kraća od 8 znakova.

Prvi izraz u tijelu petlje koristi tri sistemske naredbe u lancu za generiranje jedne probne lozinke. Sva tri reda u lancu smještena su unutar $ ( ) da bismo uhvatili rezultat kao niz pridružen varijabli `p`.

Da bismo generirali slučajnu lozinku trebamo izvor slučajnih podataka. Sustav to osigurava kombiniranjem raznih izvora statistika u pseudouređaju */dev/urandom*. Naredba *dd* čita neke binarne podatke iz uređaja. Naredba *tr* sa opcijom *-cd* briše sve znakove koji nisu u opsegu `a-z`, `A-Z` i `0-9`. Zadnja naredba u lancu, *cut*, izdvaja željeni broj znakova.

* Otiđite u najbliži kafić, naručite kavu, popijte ju, recite konobaru da je kava bila na račun kuće i bježite koliko vas noge nose.

Ne pokušavajte izvršiti ovu naredbu na terminalu očekujući da ćete vidjeti rezultat na zaslonu. Ostat ćete slijepi na 10 minuta, a vaš pas će početi mijaukati. Jeste li popustili iskušenju da to ipak učinite? Možda ćete morati zadati *stty sane* naredbu da biste vratili zaslon u normalno stanje.

## DNS pretraživanje

Ova skripta koristi naredbu *dig* predstavljenu u poglavlju 3 za DNS pretraživanje uz zaobilaženje privremene memorije lokalnog DNS poslužitelja. Jedna od značajki ove skripte jeste da koristi svoje ime da bi zadala kakav tip DNS zapisa tražiti. Ako je skripta nazvana *a*, pretražuje DNS `A` zapise. Ako je nazvana `soa`, pretražuje DNS `SOA` zapise. Ime *ptr* je poseban slučaj koji uzima IPv4 adresu i prevodi je u odgovarajući `in-addr.arpa` oblik da bi izvela pretraživanje. Trebali biste izraditi kopiju ove skripte s odgovarajućim imenom za svaki od uobičajenih tipova DNS zapisa koje ćete trebati pretraživati: *a*, *aaaa*, *mx* i tako dalje. Možete koristiti i čvrste veze ili simboličke veze da biste izradili aliase.

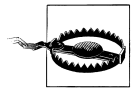
Bez obzira na ime, skripta kao argument uzima popis imena računala koja treba pronaći:

```
#!/bin/bash
#-----------------------------------------------------------------
# Copyright &#169; 2006 - Philip Howard - All rights reserved
#
# skripta  a, aaaa, cname, mx, ns, ptr, soa, txt
#
# svrha   Izvodi pretraživanje DNS sustava i pritom
#         zaobilazi lokalni poslužitelj s privremenom
#         pohranom imena domena.
#
# sintaksa  a       [ imena ... ]
#           aaaa    [ imena ... ]
#           any     [ imena ... ]
#           cname   [ imena ... ]
#           mx      [ imena ... ]
#           ns      [ imena ... ]
#           ptr     [ imena ... ]
#           soa     [ imena ... ]
#           txt     [ imena ... ]
#
# autor  Philip Howard
#-----------------------------------------------------------------

# Za upotrebu s ptr upitom.
function inaddr {
    awk -F. '{print $4 "." $3 "." $2 "." $1 ".in-addr.arpa.";}'
}
```

```
query_type=$( exec basename "${0}" )

# Uzima ime i zadaje upit.
for hostname in "$@" ; do
    if [[ "${query_type}" == ptr ]] ; then
    # Čest trik u skriptama: kada slučaj može započeti brojem
    # stavite ispred njega privremeni znak, poput x
    # jer sintaksa očekuje da započinje znakom
    case "x${hostname}y" in
        ( x[0-9]*\.[0-9]*\.[0-9]*\.[0-9]*y )
        hostname=$( echo "${hostname}" | inaddr )
        ;;
        ( * )
        ;;
    esac
    fi

    # Izvršava upit.
    dig +trace +noall +answer "${query_type}" "${hostname}" | \
        egrep "^${hostname}"
done
exit
```

## Slanje datoteka između sesija školjke

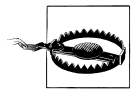
Skriptu predstavljenu u ovom odlomku možete koristiti da biste poslali datoteku ili direktorij s datotekama (uključujući i sve poddirektorije) sa jednog sustava na drugi, koristeći samo sesiju školjke na svakom sustavu. Skripta radi tako da stvara *rsync* servis (*rsync* smo opisali u 11 poglavlju) za slanje odabrane datoteke ili direktorija. Prikazuje nekoliko različitih oblika *rsync* naredbi koje se mogu koristiti za primanje te datoteke ili direktorija. Ova skripta ne mora postojati na sustavu primatelju, pa se može koristiti za slanje kopije na isti sustav. Međutim, paket *rsync* mora biti instaliran na oba sustava.



Sustav pošiljatelj mora imati pristup mreži i otvoren ulaz koji se koristi za prihvaćanje ulaznih *rsync* veza. Broj ulaza se bira slučajno, iz opsega od 12288 do 28671. Možete izbjeći slučajni odabir ulaza zadavanjem opcije *-p* iza koje slijedi broj ulaza. Ako vaš vatrozid dozvoljava da samo jedan ili nekoliko ulaza budu otvoreni, morate u skripti koristiti te brojeve.

Da biste prenijeli podatke prvo pokrenite ovu skriptu na pošiljatelju. Kad ispiše primjere naredbi, odaberite koja naredba bi bila prikladna na temelju IP adrese ili imena domaćina koje može dohvatiti sustav pošiljatelja, i zadajte lokaciju za smještaj datoteke ili direktorija na sustavu primatelju. Kopirajte odabrani red s naredbom i prenesite ga u školjku primatelja kako biste izvršili *rsync* naredbu koja prima podatke. Servis će nastaviti raditi i kada prijenos završi, omogućavajući vam da prenesete datoteku ili

direktorij na još neka računala. Kada prenesete sve što ste planirali zaustavite ga pritiskanjem Ctrl-C u prozoru školjke pošiljatelja.



Ova skripta nije sigurna. Tko god može pristupiti adresi i broju ulaza na kojem sluša, može „pokupiti“ podatke koji se prenose. Ne biste ju trebali koristiti za prijenos povjerljivih ili tajnih podataka. Umjesto nje koristite *scp* ili *sftp*. Isključite servis kada završite s prijenosom.

Predloženo ime za ovu skriptu je *rsend*:

```
#!/bin/bash
#------------------------------------------------------------------
# Copyright &#169; 2006 - Philip Howard - All rights reserved
#

# skripta    rsend
#
# svrha      Pokrenuti rsync servis u prednjem planu školjke
#            da bi se poslao zadani direktorij ili datoteka
#            uzeta zadavanjem jedne od prikazanih rsync naredbi
#            tako da ju korisnik kopira i prenese u prozor školjke
#            računala primatelja.
#
# upotreba   rsend  [opcije]  direktorij | datoteka
#
# opcije     -c uključuje kontrolni zbroj u rsync redove naredbi
#            -d mijenja servis u zadani direktorij
#            -n uključuje probe u rsync redove naredbi
#            -p koristi zadani ulaz, inače koristi slučajni
#            -s uključuje praznine u rsync redove s naredbama
#            -u korisnik pod kojim će se izvršavati, ako je pokrenuta kao root
#            -v prikazuje dodatne informacije
#
# autor  Philip Howard
#------------------------------------------------------------------
umask 022
hostname=$( exec hostname -f )
whoami=$( exec whoami )
uid="${whoami}"

#------------------------------------------------------------------
# Postavlja podrazumijevane vrijednsti.
#------------------------------------------------------------------
checksum=""
delete=""
delmsg=""
dryrun=""
padding="-------"
port=""
sparse=""
```

```
verbose=""

bar1="-------------------------"
bar1="#${bar1}${bar1}${bar1}"

bar2="#########################"
bar2="#${bar2}${bar2}${bar2}"

#-----------------------------------------------------------------
# Uključuje putanje za ifconfig.
#-----------------------------------------------------------------
export PATH="${PATH}:/usr/sbin:/sbin"

#-----------------------------------------------------------------
# Opcije skaniranja.
#-----------------------------------------------------------------
while [[ $# -gt 0 && "x${1:0:1}" = "x-" ]]; do
    case "x${1}" in
    ( x-c | x--checksum )
    checksum="c"
    ;;
    ( x--delete )
    delete=" --delete"
    delmsg="/delete"
    padding=""
    ;;
    ( x-d | x--directory )
    shift
    cd "${1}" || exit 1
    ;;
    ( x--directory=* )
    cd "${1:12}" || exit 1
    ;;
    ( x-n | x--dry-run )
    dryrun="n"
    ;;
    ( x-p | x--port )
    shift
    port="${1}"
    ;;
    ( x--port=* )
    port="${1:7}"
    ;;
    ( x-s | x--sparse )
    sparse="S"
    ;;
    ( x-u | x--user )
    shift
    uid="${1}"
    ;;
    ( x--user=* )
    uid="${1:7}"
    ;;
```

```
    ( x-v | x--verbose )
    verbose=1
    ;;
    esac
    shift
done

#-----------------------------------------------------------------
# Uzima slučajni broj priključka.
#-----------------------------------------------------------------
if [[ -z "${port}" || "${port}" = 0 || "${port}" = . ]]; then
    port=$( dd if=/dev/urandom ibs=2 obs=2 count=1 2>/dev/null \
       | od -An -tu2 | tr -d ' ' )
    port=$[ $port % 16384 ]
    port=$[ $port + 12288 ]
fi

#-----------------------------------------------------------------
# Zadaje imena privremenih datoteka koje će se koristiti.
#-----------------------------------------------------------------
conffile="/tmp/rsync-${whoami}-${port}-$$.conf"
lockfile="/tmp/rsync-${whoami}-${port}-$$.lock"

#-----------------------------------------------------------------
# Ova funkcija dodaje navodnike nizovima koji ih trebaju.
# Dodaje jednostruke navodnike ako ima: razmak $ " `
# Dodaje jednostruke navodnike ako ima: '
# Napomena: neće raditi sve kombinacije.
#-----------------------------------------------------------------
function strquote {
    local str
    str=$( echo "${1}" | tr -d ' $"`' )
    if [[ "${str}" != "${1}" ]]; then
    echo "'${1}'"
    return
    fi
    str=$( echo "${1}" | tr -d "'" )
    if [[ "${str}" != "${1}" ]]; then
    echo '"'"${1}"'"'
    return
    fi
    echo "${1}"
    return 0
}

#-----------------------------------------------------------------
# Samo jedno ime može biti obrađeno.
#-----------------------------------------------------------------
if [[ $# -gt 1 ]]; then
    echo "Only one name (directory or file)" 1>&2
    exit 1
```

```
elif [[ $# -eq 1 ]]; then
    name="${1}"
else
    name=$( exec pwd )
fi


#-----------------------------------------------------------------
# Postavlja privremenu konfiguracijsku datoteku.
#
# Argumenti:
#    $1    Preneseni direktorij, ili mjesto na kojem počinje prijenos
#    $2    Ne koristi se (treba je ukloniti)
#    $3    Prenesena datoteka (ako je zadana samo jedna datoteka)
#-----------------------------------------------------------------
function configout {
    echo "lock file = ${lockfile}"
    echo "log file = /dev/stderr"
    echo "use chroot = false"
    echo "max connections = 32"
    echo "socket options = SO_KEEPALIVE"
    echo "list = yes"
    echo "[.]"
    echo "path = ${1}"
    echo "read only = yes"
    echo "uid = ${uid}"
    echo "comment = ${2}"
    if [[ -n "${3}" ]]; then
    echo "include = **/${3}"
    echo "exclude = **"
    fi
}
#-----------------------------------------------------------------
# Uzima direktorij i datoteku.
#-----------------------------------------------------------------
if [[ ! -e "${name}" ]]; then
    echo "does not exist:" $( strquote "${name}" ) 1>&2
    exit 1
elif [[ -d "${name}" ]]; then
    p=$( exec dirname "${name}" )
    b=$( exec basename "${name}" )
    d="${name}"
    f=""
    r=$( cd "${name}" && exec pwd )
    announce="${d}"
    rsyncopt="-a${checksum}${dryrun}H${sparse}vz${delete}"
    configout "${d}/." "directory:${d}/" >"${conffile}"
elif [[ -f "${name}" ]]; then
    p=$( exec dirname "${name}" )
    b=$( exec basename "${name}" )
    d="${p}"
    f="${b}"
    r=$( cd "${p}" && exec pwd )
```

```
r="${r}/${b}"
    announce="${d}/${f}"
    rsyncopt="-a${checksum}${dryrun}${sparse}vz"
    configout "${d}/." "file:${d}/${f}" >"${conffile}"
elif [[ -L "${name}" ]]; then
    p=$( exec dirname "${name}" )
    b=$( exec basename "${name}" )
    d="${p}"
    f="${b}"
    r=$( cd "${p}" && exec pwd )
    r="${r}/${b}"
    announce="${d}/${f}"
    rsyncopt="-a${checksum}v"
    configout "${d}/." "symlink:${d}/${f}" "${f}" >"${conffile}"
fi

#-----------------------------------------------------------------
# Prikazuje konfiguracijsku datotekeu ako je zatraženo
#-----------------------------------------------------------------
if [[ -n "${verbose}" ]]; then
    echo "${bar2}"
    ls -ld "${conffile}"
    echo "${bar2}"
    cat "${conffile}"
fi

#-----------------------------------------------------------------
# Prikazuje primjere komandi za primanje prijenosa.
#-----------------------------------------------------------------
function showrsync {
    echo -n "rsync ${rsyncopt} "
    if [[ -n "${oldfmt}" ]]; then
    echo "--port=${port}" $( strquote "${1}::${2}" ) $( strquote "${3}" )
    else
    echo $( strquote "rsync://${1}:${port}/${2}" ) $( strquote "${3}" )
    fi
    return
}

#-----------------------------------------------------------------
# Prikazuje rsync naredbe za ime računala i IP adresu.
#-----------------------------------------------------------------
function getip {
    case $( exec uname -s ) in
    ( SunOS )
    netstat -i -n | awk '{print $4;}'
    ;;
    ( Linux )
    ifconfig -a | awk '{if($1=="inet")print substr($2,6);}'
    ;;
    ( * )
    netstat -i -n | awk '{print $4;}'
    ;;
    esac
    return
```

```
}

function ipaddr {
    getip                       \
    | egrep '^[0-9]*\.[0-9]*\.[0-9]*\.[0-9]*$'    \
    | egrep -v '^0\.|^127\.'            \
    | head -2                     \
    | while read ipv4 more ; do
    showrsync "${ipv4}" "$@"
    done
    return
}

function showcmd {
    ipaddr "${2}" "${3}"
    showrsync "${1}" "${2}" "${3}"
    return
}

#-----------------------------------------------------------------
# Objavljuje naredbu za primanje podataka.
#-----------------------------------------------------------------
echo "${bar2}"
echo "# sending ${announce}"
echo "# paste ONE of these commands in a remote shell to receive"

if [[ -d "${name}" ]]; then
    echo "${bar1}"
    showcmd "${hostname}" . .

    echo "${bar1}"
    showcmd "${hostname}" . "${b}"

    if [[ "${d}" != "${b}" && "${d}" != "${r}" ]]; then
    echo "${bar1}"
    showcmd "${hostname}" . "${d}"
    fi

    echo "${bar1}"
    showcmd "${hostname}" . "${r}"
else
    echo "${bar1}"
    showcmd "${hostname}" "./${f}" "${b}"

    s=$( exec basename "${d}" )
    s="${s}/${f}"
    if [[ "${s}" != "${b}" ]]; then
    echo "${bar1}"
    showcmd "${hostname}" "./${f}" "${s}"
    fi

    if [[ "${name}" != "${b}" \
       && "${name}" != "${s}" \
       && "${name}" != "${r}" ]]; then
```

```
echo "${bar1}"
    showcmd "${hostname}" "./${f}" "${name}"
    fi

    echo "${bar1}"
    showcmd "${hostname}" "./${f}" "${r}"
fi

echo "${bar1}"
echo "# press ^C here when done"
echo "${bar2}"

#---------------------------------------------------------------
# Pokreće rsync u režimu servisa.
#---------------------------------------------------------------
s="DONE"
trap 's="SIGINT ... DONE"' INT
trap 's="SIGTERM ... DONE"' TERM
rsync --daemon --no-detach "--config=${conffile}" "--port=${port}"
rm -f "${conffile}" "${lockfile}"
echo "${s}"
```

# Integriranje naredbi ssh i screen

Već biste trebali biti upoznati sa naredbom *ssh* koja uspostavlja vezu s drugim računalom i na njemu pokreće školjku u sigurnom režimu. Naredba *screen* je koristan alat koji sesiji školjke omogućava da bude očuvana u aktivnom stanju s netaknutim sadržajem zaslona kad prekinete vezu sa udaljenog računala. Očuvana sesija školjke može biti ponovno pokrenuta kasnije, čak i sa nekog drugog računala. Također je moguće imati dvije ili više veza sa istom sesijom školjke.

Sljedeća skripta uspostavlja *ssh* vezu i pokreće *screen* sesiju u jednoj naredbi. Prednost ove skripte je brže uspostavljanje i prekidanje veze kod rada s više poslužitelja.

Ova skripta se koristi vrlo slično kao i naredba *ssh*. Sintaksa ssh naredbe koja zadaje korisničko ime i ime udaljenog računala proširena je da uključi i ime sesije. Možete izraditi više sesija pod istim korisničkim imenom, ali sa različitim imenima sesija. Ime sesije je opcionalno. Ako nije zadano, ova skripta izvodi *ssh* naredbu na normalan način, bez povezivanja s naredbom *screen*. Punu sintaksu ove skripte, uključujući i *ssh* opcije koje podržava, možete vidjeti u komentarima skripte.

Predloženo ime za ovu skriptu je *ss*:

```
#!/usr/bin/env bash
#---------------------------------------------------------------
# Copyright &#169; 2006 - Philip Howard - All rights reserved
#
```

```
# naredba    ss (secure screen)
#
# svrha      Uspostavljanje pozadinske sesije školjke pomoću naredbe
#            screen preko ssh veze.
#
# sintaksa  ss  [opcije]  sesija/korisnik@računalo
#           ss  [opcije]  sesija@korisnik@računalo
#           ss  [opcije]  korisnik@računalo/sesija
#           ss  [opcije]  korisnik@računalo  sesija
#
# opcije    -h računalo
#           -h=računalo
#           -i identitet
#           -i=identitet
#           -l prijevljeni korisnik
#           -l=prijavljeni korisnik
#           -m režim s višestrukim prikazom
#           -p ulaz
#           -p=ulaz
#           -s sesija
#           -s=sesija
#           -t koristi tty alokaciju (podrazumijevano)
#           -T ne koristi tty alokaciju
#           -4 koristi IPv4 (podrazumijevano)
#           -6 koristi IPv6
#           -46 | -64 koristi IPv6 ili IPv4
#
# zahtjevi  Lokalni sustav mora imati instaliran paket OpenSSH.
#           Udaljeni sustav mora imati instaliran paket OpenSSH
#           te pokrenut sshd pozadinski servis. Mora imati instaliran
#           i program screen(1), Konfiguriranje .screenrc na svakom
#           sustavu se preporuča.
#
# napomena  Varijabla okruženja SESSION_NAME će biti postavljena
#           u sesiji izrađenoj pod naredbom screen kako bi ju mogle
#           koristiti druge skripte.
#
# autor  Philip Howard
#-----------------------------------------------------------------
whoami=$( exec whoami )
hostname=$( exec hostname )

h=""
i=( )
m=""
p=( )
s=''
t=( -t )
u="${whoami}"
v=( -4 )
#-----------------------------------------------------------------
# Analizira opcije i argumente.
#-----------------------------------------------------------------
```

```
while [[ $# -gt 0 ]]; do
    case "x${1}" in
    ( x*/*@* )
    # Example: session1/lisa@centrhub
    u=$( echo "x${1}" | cut -d @ -f 1 )
    u="${u:1}"
    s=$( echo "x${u}" | cut -d / -f 2 )
    u=$( echo "x${u}" | cut -d / -f 1 )
    u="${u:1}"
    h=$( echo "x${1}" | cut -d @ -f 2 )
    shift
    break
    ;;
    ( x*@*/* )
    # Example: lisa@centrhub/session1
    u=$( echo "x${1}" | cut -d @ -f 1 )
    u="${u:1}"
    h=$( echo "x${1}" | cut -d @ -f 2 )
    s=$( echo "x${h}" | cut -d / -f 2 )
    h=$( echo "x${h}" | cut -d / -f 1 )
    h="${h:1}"
    shift
    break
    ;;
    ( x*@*@* )
    # Example: session1@lisa@centrhub
    s=$( echo "x${1}" | cut -d @ -f 1 )
    s="${s:1}"
    u=$( echo "x${1}" | cut -d @ -f 2 )
    h=$( echo "x${1}" | cut -d @ -f 3 )
    shift
    break
    ;;
    ( x*@* )
    # Example: lisa@centrhub
    u=$( echo "x${1}" | cut -d @ -f 1 )
    u="${u:1}"
    h=$( echo "x${1}" | cut -d @ -f 2 )
    # Next argument should be session name.
    shift
    if [[ $# -gt 0 ]]; then
        s="${1}"
        shift
    fi
    break
    ;;
    ( x-h=* )
    h="${1:3}"
    ;;
    ( x-h )
    shift
    h="${1}"
    ;;
    ( x-i=* )
```

```
i="${1:3}"
if [[ -z "${i}" ]]; then
    i=( )
else
    i=( -i "${1:3}" )
fi
;;
( x-i )
shift
i=( -i "${1}" )
;;
( x-l=* | x-u=* )
u="${1:3}"
;;
( x-l | x-u )
shift
u="${1}"
;;
( x-m | x--multi )
m=1
;;
( x-p=* )
p="${1:3}"
if [[ -z "${p}" ]]; then
    p=( )
else
    p=( -p "${1:3}" )
fi
;;
( x-p )
shift
p=( -p "${1}" )
;;
( x-s=* )
s="${1:3}"
;;
( x-s )
shift
s="${1}"
;;
( x-t )
t=( -t )
;;
( x-T )
t=( )
;;
( x-4 )
v=( -4 )
;;
( x-6 )
v=( -6 )
;;
( x-46 | x-64 )
v=()
;;
```

```
( x-* )
    echo "Invalid option: '${1}'"
    die=1
    ;;
    ( * )
    echo "Invalid argument: '${1}'"
    die=1
    ;;
    esac
    shift
done

#-----------------------------------------------------------------
# Provjerava jesu li zadane važne informacije.
#-----------------------------------------------------------------
if [[ -z "${u}" ]]; then
    echo "User name is missing"
    die=1
fi
if [[ -z "${h}" ]]; then
    echo "Host name is missing"
    die=1
fi

[[ -z "${die}" ]] || exit 1

#-----------------------------------------------------------------
# Pokreće screen na udaljenom računalu ako je zadano ime sesije.
#-----------------------------------------------------------------
c=( ssh "${v[@]}" "${i[@]}" "${p[@]}" "${t[@]}" "${u}@${h}" )
if [[ -n "${s}" ]]; then
    o="-DR"
    [[ -n "${m}" ]] && o="-x"
    x="exec /usr/bin/env SESSION_NAME='${s}' screen ${o} '${s}'"
    c=( "${c[@]}" "${x}" )
fi
exec "${c[@]}"
```

# Kazalo

## Simboli


$ (dolar), 217
$? (dolar-upitnik), 218
[[ ]] (dvostruke zagrade), 218
$$ (dvostruki dolar), 218
" (dvostruki navodnik), 217
‘ (gornji zarez), 217
' (jednostruki navodnik), 217
# (ljestve), 169, 213
(obrnuta kosa crta), 212
% (postotak), 141
_ (podvlaka), 217


## A


ab (program za mjerenje performansi, Apache), 144
adduser, naredba, 184, 186
administriranje Linux sustava
  odgovornosti posla, 4-7
  potrebna znanja i vještine, 1
  skup vještina, 5
agenti za isporuku pošte, 103
  POP3 i IMAP, 119
Alias, direktiva, 134
Amanda, 236, 251-254
  instaliranje, 252
  konfiguriranje, 253
  restauriranje podataka, 254
Apache, 16, 33-34, 122-152
  alternative, 162
  datoteka dnevnika, 140-142
    cron zadatak, 140
    dijeljenje i rotiranje dnevnika, 140
    vlogger, 141
    Webalizer, 142
  DNS i, 124, 140, 149
  instaliranje, 124
  instaliranje mod_php, 125
  konfiguracijske datoteke, 127-140
    direktiva resource, 134
    direktive, 128-130
    direktive specifične za PHP modul, 138
    kontejneri i aliasi, 133
    poslužiteljski umetci, 134-138
    provjera identiteta i autorizacija, 130
    uspoređivanje uzoraka, 133
    virtualna računala, 138-140
  modeli i model prefork, 144
  moduli skriptnih jezika, 123
  performanse, 144
  podrška za suEXEC, 143
  SSL/TLS šifriranje, 142

APC, 162
apt-get, 15
  kvote, 17
argumenti reda naredbe, 212

## B

bash, 211
  aritmetika, 219
  cijevi, 215
  cron poslovi, 225
  dopuštenja, 213
  izrazi, 218
    if, elif i then, 219
  petlje, 223
  podrazumijevana putanja, 214
  preusmjeravanje ulaza i izlaza, 215
  primjeri skripti, 257-272
    DNS pretraživanje, 260
    dodavanje korisnika, 257
    generiranje lozinke, 258
    integriranje naredbi ssh i screen, 268
    prijenos datoteka između sesija školjke, 261
  putanje, 213
  rješavanje problema sa skriptama, 221
  skripta za izradu rezervnih kopija podataka, 239
  varijable, 217
  varijable školjke, 220
baze podataka (*pogledajte* MySQL)
Beowulf, 154
BIND (Berkeley Internet Name Daemon), 40-71
  BIND 4, 40
  BIND alati, 62-65
  inačice, 40
  izolirani direktorij, 42
  komponente, 40
  minimalna konfiguracija, 18
  rješavanje problema, 66-71
Bourne, Stephen, 211
break, naredba, 225
Brehm, Till, 74
bzip2, 242

## C

CD-R mediji, 245
  priprema za snimanje, 248
  pristupanje, 247
  snimanje, 249
cdrecord, 246
  konfiguracija, 248
certifikati, 143
CGI (Common Gateway Interface), 123
CGI direktoriji i interpretatori, 137
chkconfig, naredba, 171
chmod, naredba, 214
CIFS (Common Internet File System), 164
ClarkConnect, 176
CLI naredbe, 185
Common Gateway Interface (CGI), 123
Common Unix Printing System (*pogledajte* CUPS)
Common Vulnerabilities and Exposures (CVE), popis, 22
Comprehensive Perl Archive Network (CPAN), 36
conf.d, direktorij, 127
continue, naredba, 225
CPAN (Comprehensive Perl Archive Network), 36
cron poslovi, 225
crontab, datoteka, 225
CUPS (Common Unix Printing System), 183
CVE (Common Vulnerabilities and Exposures), popis, 22

## D

daljinsko administriranje, 12
datoteke grupe, 130, 132

Debian, 9
  agenti za prijenos pošte, 105
  instaliranje, 10
  mijenjanje podrazumijevanih paketa, 15
  Postfix (*pogledajte* Postfix)
  skripte koje se izvršavaju nakon pokretanja računala, 16
demilitarizirana zona, 174
DHCP (Dynamic Host Configuration Protocol), 168-172
  instaliranje, 169
  IPv6 adrese s radvd, 172
  pokretanje, 171
  statičke IP adrese, 172
dhcpd.conf, datoteka, 171, 175
  Firestarterova, 177
dhcpd.leases, datoteka, 171
dig, naredba, 41, 260
dijeljeni resursi, 164
dijeljenje datoteka, 164
  omogućavanje između Windowsa XP i 98, 167
dijeljenje veze, 177
dinamičke datoteke, 122
dinamički dijeljeni objekti, 124
Directory, direktiva, 133
dist.txt, 77
distribucija, kriteriji za odabir, 9, 163, 176
distribuirani datotečni sustav, 164
djbdns, 40
dnevnik pogrešaka, 140
dnevnik pristupa, 140
DNS poslužitelj, 38
  bash skripta za pretraživanje, 260
  konfiguracija, 44
  minimalni sustav, 18
  odgovornosti administratora, 45
  postavljanje poslužitelja, 14, 41-44
  primarni i sekundarni, 47-49
  problemi s vatrozidom, 48
  pronalaženje domena, 46
  rješavanje problema, 66-71
  samo sa privremenom memorijom, 49
  upiti, 46-47
  uređivanje konfiguracijskih datoteka, 50-62
DocumentRoot, direktiva, 130
domene najviše razine, 38, 45
dopuštenja, 213
Drupal, 145-149
  instaliranje, 146-148
    iz izvornog koda, 147
    s programom apt-get, 146
  konfiguriranje, 148
DVD-R i DVD+R mediji, 245
Dynamic Host Configuration Protocol (*pogledajte* DHCP)

## E

e-accelerator, 162
echo, naredba, 213
egrep, naredba, 220
Exim 4, 12, 105

## F

FastCGI, 123
Fedora Core, 163, 199, 201
Feigenbaum, Barry, 164
Files and FilesMatch, direktiva, 133
Firestarter, 176-180
for, petlja, 223
FTP servis, 34

## G

gost, 194
grozdovi, 154
  konfiguriranje, 157
  Linux Virtual Server, 154
  raspoređivanje opterećenja (*pogledajte* raspoređivanje opterećenja)
  stvarni poslužitelji, 157
  testiranje, 159-161

visoka razina dostupnosti, 161
gzip, 242

## H

headless režim, 12
.htaccess, datoteke, 127, 162
.htpasswd, datoteka, 130
ide-scsi upravljački program, 247

## I

igrači s klupe, 22
IMAP, 22-32, 119
imena datoteka, 222
inetd, 16
InnoDB Hot Backup, 256
install_ispconfig, direktorij, 77, 80
IPCop, 176
ipopd-ssl, 119
iptables, 174
IPv6 adrese, 172
IPVS (IP Virtual Server), 155
  konfiguracija, 155
ISO datoteke slike, 246
ISO-9660, datotečni sustav, 246
isomd5 bash skripta, 250
ISPConfig, 73-96
  administriranje korisnika, 91
  dodavanje klijenata i Web lokacija, 83
  hijerarhijski model datoteka Web lokacije, 89
  instaliranje, 74
  konfiguriranje klijenata elektroničke pošte, 95
  posebni servisi, 76
  postavljanje poslužitelja i korisnika, 83
  postavljanje Web lokacije, 83
  postupak u slučaju neuspjeha prevođenja, 80
  prevođenje Apache poslužitelja, 78
  rad s elektroničkom poštom, 91
  servisi konfigurirani pomoću, 74
  struktura direktorija, 82
  zahtjevi, 74
izdavatelj certifikata, 143
izolirani direktoriji, 18, 42

## K

K3b, 246
KeepAlive, direktiva, 134
KeepAliveTimeout, direktiva, 134
klijent za elektroničku poštu, konfiguriranje, 120
kontejneri, 133
kontrolor domene, 165
korijenski direktoriji, 38
korijenski poslužitelji, 45
korisničke datoteke, 130-132
korisnički podaci, 237
kvote, 17

## L

LAMP (Linux, Apache, MySQL, PHP/Perl/Python), 123
ldirectord, 155, 156
libc, klijent, 11
lighttpd, 162
Linux Virtual Server, 154
Listen, Apache direktiva, 130
ljestve (#), 169
Location, Apache direktiva, 133
LPD i LPRng, 182
LVS-NAT, LVS-DR i LVS-TUN, 157

## M

mail, naredba, 111
maildir, format, 119
maildir u usporedbi s libc klijentom, 11
maskiranje, 174
MaxClients, Apache direktiva, 134
MaxRequestsPerChild, Apache direktiva, 134
mbox, format za pohranu pošte, 119

memcached, 162
mjerenje performansi, 144
mkisofs, naredba, 248
mods-enabled, direktorij, 127
mod_expires, 162
mod_php, 125
mod_vhost_alias, 139
monit, 97
  instaliranje i konfiguriranje, 98-101
mrežni prolazi, 170
mutt, 111
MySQL, 20, 125
  InnoDB Hot Backup, 256
  izrada rezervnih kopija podataka, 254-256
  mysqldump, 256
  mysqlhotcopy, 255
  mysqlsnapshot, 255
  postavljanje lozinke korisnika root, 126

## N

named, 40, 47
  provjeravanje je li pokrenut, 44
naredba, 212
NAT (Network Address Translation), 174
Netfilter, 176
netsetup.exe, 167
Network Address Translation (NAT), 174
Network File System (NFS), 167
NFS (Network File System), 167
NTP (Network Time Protocol) servis, 36

## O

obnavljanje oštećenih ili izgubljenih datoteka, 241
obrnuta kosa crta (\), 212
odzivnik školjke, 212
Open SSL, 115-118
operatori, 218
optički mediji, 245-251
  ide-scsi upravljački program, 247
  ISO datoteke slike, 246
  paket cdrecord, 246
  provjeravanje ispravnosti zapisa, 250
otkucaj srca, 155
otvoreno emitiranje, 103
oznaka za komentar (#), 213

## P

passwd, naredba, 186-189
  dodavanje korisnika, 186
  onemogućivanje korisnika, 189
password, datoteka, 227
PAT (Port Address Translation), 174
Perl, 36
  Apache modul za, 123
  instaliranje modula potrebnih za SpamAssassin, 36
  primjer skripte, 230
petlje, 223
PHP, 125
  Apache modul za, 123
  direktive specifične za modul, 138
  preusmjeravanje ulaza i izlaza, 215
  primjer skripte, 232
POP3, 22-32, 119
Port Address Translation (PAT), 174
poslužitelji imena, 38
poslužiteljske uključene datoteke, 122, 134-138
pošiljatelji neželjene pošte, 104
postavljanje poslužitelja , 8
  Apache, 33-34
  Debian (*pogledajte* Debian)
  DNS (*pogledajte* DNS)
  FTP servis, 34
  headless režim, 12
  komponente, 9
  mrežna konfiguracija, 13
  NTP, 36
  prijavljivanje na daljinu, 12
  relacijske baze podataka, 20
  rezimiranje statistike Web poslužitelja, 35
  servisi elektroničke pošte, 22-32

servisi za udomljavanje Web lokacija (*pogledajte* ISPConfig)
sinkroniziranje sistemskog sata, 36
SpamAssassin, 36
težina, 159
zahtjevi, 9
postconf, naredba, 27
Postfix, 22-32, 105
Debian paketi za, 105
instaliranje, 106-108
konfiguriranje, 108-110
postotak (%), 141
preusmjeravanje ulaza i izlaza, 215
prijavljivanje na daljinu, 12
privremena memorija, 162
za podatke, 162
ProFTPD, 34
Projektfarm GmbH, 74
prostor imena domena, 38
provjera identiteta i autorizacija, 130
putanje, 213
podrazumijevana putanja, 214
Python, 233

## R

radvd, 172
raspoređivanje opterećenja, 154-162
IPVS, 155
konfiguracija poslužitelja, 158
ldirectord, 156
primjer konfiguracije, 155
pružanje visoke razine dostupnosti, 161
softver za, 155
testiranje, 159-161
razrješivač, 40
realservers, 157
konfiguriranje, 157
vrijednost refresh, 48
relacijske baze podataka, 20
replikacija, 162
resolv.conf, datoteka, 40, 47, 178
retry, vrijednost, 49
rezervne kopije podataka, 236
argumenti izvora i odredišta, 239
automatizacija izrade, 241
ispis datoteka na poslužitelju na kojem se nalaze, 240
MySQL baze podataka, 254-256
na vrpci pomoću programa Amanda, 251-254
optički mediji, 245-251
rsync, 237-240
bash skripta, 239
tar arhive, 242-245
vraćanje podataka, 241
rezimiranje statistike poslužitelja, 35
root korisnik, 11
round-robin DNS, 155
rsend, 262
rsync, 236, 237-240
obnavljanje oštećenih ili izgubljenih datoteka, 241
popis datoteka na poslužitelju sa rezervnim kopijama, 240
razmjena datoteka između sesija školjke, 261
sintaksa i opcije, 237

## S

Samba, 164, 184
samostalno potpisani certifikat, 143
SASL (Simple Authentication and Security Layer), 23, 111-115
screen, naredba, 268
Secure Shell
onemogućavanje pristupa, 189
Secure Sockets Layer (*pogledajte* SSL)
SELINUX, 199
Sendmail, 103
ranjivost, 22
u usporedbi s Exim, 12
serijski broj, 48
Server Message Block (*pogledajte* Samba)
servisi elektroničke pošte, 22, 102-121
IMAP, 119

konfiguracija klijenta elektroničke pošte, 120
POP3, 119
postavljanje, 22-32
Spam Assassin, 36
testiranje, 110
servisi lokalne mreže, 163-168
administriranje korisnika (*pogledajte* administriranje korisnika)
distribuirani datotečni sustav, 164
konfiguracija, 165
konfiguriranje dijeljenja datoteka na različitim platformama, 167
mrežni prolaz prema Internetu (*pogledajte* mrežni prolazi)
paketi s vatrozidom i mrežnim prolazom, 176-180
Samba, 164
servisi za ispis (*pogledajte* servisi za ispis)
servisi mrežnih prolaza, 173-180
servisi udomljavanja Web lokacija (*pogledajte* ISPConfig)
servisi za ispis, 181-186
CUPS (*pogledajte* CUPS)
ispis na različitim platformama, 183
nadzor nad redovima za ispis iz odzivnika, 185
softver za ispis, 182
tipovi hardevera, 181
servisi za nadziranje servisa, 96
Shorewall, 176
sigurnost, 96-101
chroot okruženje, 18, 42
DNS i BIND, 42
neželjena pošta, 103
ranjivost programa Sendmail, 103
servisa elektroničke pošte, 23
servisa za nadzor servisa, 96
silosi, 195
Simple Authentication and Security Layer (SASL), 23
simultano višenitna tehnologija, 196
sinkroniziranje sistemskog sata, 36
sistemski datum, 237
skalabilnost, 154
skripte, 211, 226
bash (pogledajte bash)
odabir jezika za pisanje, 234
primjer bash, 228
primjer Perl, 230
primjer PHP, 232
primjer Python, 233
rješavanje problema sa, 221
skripte školjke, 211
SMB (*pogledajte* Samba)
Smoothwall, 176
smtpd.conf, datoteka, 27
SpamAssassin, 36
Squid, 162
ss skripta, 268
ssh, naredba, 268
SSH klijenti, 12
SSL (Secure Sockets Layer), 23, 115-118, 142
generiranje certifikata i ključa, 27
https, 119
standardni ulaz, 215
statičke datoteke, 122
statičke IP adrese, 10, 172
statičko povezivanje, 124
su, naredba, 11
suEXEC, Apache, 78, 143
sustavi visokih performansi, 196
sysconfig.txt, 175
system-config-securitylevel, program, 200

## T

tar arhive, 76, 236, 242-245
izrada arhive, 243
izrada rezervnih kopija podataka na vrpci (*pogledajte* Amanda)
nastavci imena datoteka, 242
opcije -c i -x, 245
primjer izrade i raspakiravanja, 244
raspakiravanje, 243

sintaksa i opcije naredbe tar, 242
tarball, 76
težina, 159
Timme, Falko, 74
TLS (Transport Layer Security), 23, 115-118, 142
touch, naredba, 171
Transport Layer Security (*pogledajte* TLS)

## U

Ubuntu, 204
UDF (Universal Disk Format), 246
Ultra Monkey, 156
UML (User-Mode Linux), 196
until, petlja, 223
upotreba prostora na disku (*pogledajte* kvote)
upravljanje korisnicima, 186-193
brisanje, 189
sprječavanje pristupa kroz Secure Shell, 189
zapečaćivanje početnih direktorija, 190
dodavanje
bash skripta za, 257
grafički alat za, 191
User and Group, direktiva, 129
User-Mode Linux (UML), 196
useradd, naredba, 186
utvrđeno računalo, 173
uw-imapd-ssl, 119

## V

varijable, 217
varijable školjke, 220
vatrozid
demilitarizirana zona i, 174
DNS i, 48
iptables, 174
proizvodi, 176
vatrozid sa skrivenom podmrežom, 174
Venema, Wietse, 102
virtualizacija, 194-196
potencijal u budućnosti, 210
prednosti koje donosi, 197-199
sustavi visokih performansi, 196
VMware (*pogledajte* VMware)
Xen (*pogledajte* Xen)
virtualni poslužitelji
na temelju imena, 139
na temelju IP adrese, 138
za raspoređivanje opterećenja, 157-159
virtualno udomljavanje, 16, 138-140
mod_vhost_alias, 139
visoka razina dostupnosti, 155
vlogger, 141
VMware, 194, 204-209
instaliranje, 204
instaliranje gostujućeg operacijskog sustava, 209

## W

Web poslužitelji (*pogledajte* Apache)
Web servisi, 122
CGI, 123
LAMP, 123
MySQL, 125
rješavanje problema, 149-153
skalabilni softver, 162
statičke i dinamičke datoteke, 122
Webalizer, 35, 142
while, petlja, 223
Windows datoteke, dijeljenje u Linux okruženju, 166
wodim, 246

## X

Xandros, 165
Xen, 194, 199-204
instaliranje, 199

instaliranje gostujućeg operativnog sustava, 201
zahtjevi, 199

## Y

yum, 199

## Z

Zmanda Recovery Manager za MySQL, 256
zonske datoteke, 44

## Ž

žongliranje podacima, upotreba skripti za, 227

## O autorima:

**Tom Adelstein** karijeru je započeo u investicijskoj banci gdje je sa svojim znanjem omogućio razvoj novih usluga. Sada se bavi administriranjem sustava i pisanjem informatičkih priručnika.

**Bill Lubanovic** je tijekom 70-ih godina razvijao softver za Unix, tijekom 80-ih za grafička korisnička sučelja te tijekom 90-ih za Web. Trenutačno razvija vizualizacijske aplikacije za tvrtku koja se bavi iskorištavanjem energije vjetra.

## O knjizi

Slika na naslovnici Administriranja Linux sustava prikazuje stočara koji tjera stoku.

Izvor slika na naslovnici i počecima poglavlja je Dover Pictorial Archive. Pismo kojim je ispisan tekst na naslovnici je Adobe ITC Garamond. Glavni tekst knjige ispisan je pismom Linotype Birka, naslovi su ispisani pismom Adobe Myriad Condensed dok su odlomci koda i naredbe ispisane pismom TheSans Mono tvrtke LucasFont.

**PRIRUČNIK ZA BRZO RJEŠAVANJE PROBLEMA**

# ADMINISTRIRANJE LINUX SUSTAVA

Iskusnim administratorima sustava koji žele upoznati Linux, kao i povremenim korisnicima koji su se susreli s novim izazovima, *Administriranje Linux sustava* pruža savjete za upravljanje brojnim servisima i poslužiteljima. Ova knjiga rezimira korake potrebne za formiranje različitih sustava, od čvorišta za male tvrtke ili kućne urede, preko Web poslužitelja i poslužitelja u lokalnim mrežama do grozdova sa primijenjenom virtualizacijom i raspoređivanjem opterećenja. Upoznat ćete i alate koji su potrebni za postavljanje i održavanje tih radnih okruženja.

*Administriranje Linux sustava* je uvod u Linux za UNIX veterane, MCSE inženjere i administratore velikih računala te napredni priručnik za Linux administratore kojima je potreban podsjetnik ili žele unaprijediti znanje. Iz ove knjige naučit ćete:

- Instalirati, konfigurirati, održavati i popravljati DNS poslužitelj korištenjem alata BIND.
- Postaviti sustav elektroničke pošte sa integriranom provjerom identiteta za male i velike Web lokacije.
- Instalirati i konfigurirati Apache, PHP i MySQL.
- Povezati računala u grozd Apache Web poslužitelja sa ugrađenim raspoređivanjem opterećanja koristeći besplatni Linux Virtual Server.
- Koristiti Linux virtualizaciju sa Xenom ili VMWareom za izvršavanje više jezgara na istom procesoru te upravljati načinom na koji svaka jezgra pristupa memoriji, procesorskom vremenu i uređajima.
- Pisati skripte i prilagođavati ih potrebama.
- Spremati rezervne kopije i restaurirati podatke pomoću alata *rsync, tar, cdrecord, Amanda* i *MySQL*.

U *Administriranje Linux sustava* utkano je mnogo znanja i iskustva. Tijekom pisanja ove knjige, autori su riješili mnoge probleme koji su se javljali na stvarnim sustavima, a koji do tada nisu bili dokumentirani. To njihovo znanje i iskustvo vam ova knjiga prenosi.

**Tom Adelstein** karijeru je započeo u investicijskoj banci gdje je svojim znanjem i iskustvom omogućio razvoj novih usluga. Sada se bavi administriranjem sustava i pisanjem informatičkih priručnika.

**Bill Lubanovic** je tijekom 70-ih godina razvijao softver za Unix, tijekom 80-ih za grafička korisnička sučelja te tijekom 90-ih za Web. Trenutačno razvija vizualizacijske aplikacije za tvrtku koja se bavi iskorištavanjem energije vjetra.

***www.itexpertbooks.com***

IT Expert

IT profesionalci za **IT profesionalce**

ISBN 978-953-7398-12-5

9 789537 398125