

Poglavje 4

Sistemsko upravljanje

To poglavje obravnava najpomembnejše stvari, ki jih morate vedeti o sistemske upravljanju pod Linuxom, in vključuje dovolj podrobnosti, da začnete udobno uporabljati sistem. Da bi bilo to poglavje obvladljivo, pokriva le osnove in izpušča številne pomembne podrobnosti. Knjiga *Linux System Administrator's Guide* (SAG) Larsa Wirzeniusa (glejte dodatek A) govori o občutno več podrobnostih v zvezi s sistemskim upravljanjem. Pomagala vam bo bolje razumeti, kako stvari delujejo skupaj in se medsebojno navezujejo. Vsaj površno preletite SAG, da boste vedeli, kaj vsebuje in kakšne vrste pomoč lahko pričakujete od nje.

4.1 Račun root

Linux loči med različnimi uporabniki. Določeno je, kaj lahko storijo drug drugemu in sistemu. Dovoljenja datotek so urejena tako, da običajni uporabniki ne morejo brisati ali spreminjati datotek v imenikih, kot so `/bin` in `/usr/bin`. Večina uporabnikov ščiti svoje lastne datoteke z ustreznimi dovoljenji, da drugi uporabniki ne morejo dostopati do njih ali jih spreminjati. (Nezaželeno je, da bi lahko kdorkoli bral tuja ljubezenska pisma.) Vsakemu uporabniku je dan **račun**, ki vključuje uporabniško ime in domači imenik. Poleg tega obstajajo posebni, sistemsko definirani računi, ki imajo posebne privilegije. Najpomembnejši od teh je **korenski račun**, ki ga uporablja upravitelj sistema. Po dogovoru je sistemski upravitelj uporabnik, imenovan `root`.

Za uporabnika `root` ni omejitev. Lahko bere, spreminja ali pobriše katerokoli datoteko na sistemu, spreminja dovoljenja in lastništvo katerekoli datoteke in poganja posebne programe, kot so tisti, ki razdelijo trdi pogon ali ustvarijo datotečne sisteme. Osnovna zamisel tega je, da se oseba, ki skrbi za sistem, prijavi v sistem kot `root` ter izvaja opravila, ki jih ne bi mogla izvajati kot običajni uporabnik. Ker lahko `root` naredi karkoli, je mogoče narediti napake s katastrofalnimi posledicami.

Če navaden uporabnik nenamena poskuša pobrisati vse datoteke v imeniku `/etc`, mu sistem tega ne bo dovolil. Če pa isto poskusi `root`, se sistem sploh ne bo pritožil. Ko uporabljate sistem Linux kot `root`, je zelo lahko pokvariti sistem. Najboljši način za preprečitev nesreč je:

- Usedite se na svoje roke, preden pritisnete `Enter` za katerikoli ukaz, ki ni obrnljiv. Če nameravate pravkar počistiti imenik, preberite celoten ukaz še enkrat in se prepričajte, da je vnesen pravilno.

- Za račun root uporabljajte drugačen pozornik. Datoteki `.bashrc` ali `.login` računa root bi morali nastaviti pozornik ukazne lupine na nekaj drugega, kot je standardni pozornik. Mnogo ljudi rezervira znak »#« v pozorniku za račun root in za vse druge račune uporablja v pozorniku znak »\$«.
- Prijavite se kot root le, ko je to absolutno potrebno. Ko ste končali svoje delo kot root, se odjavite. Manj kot uporabljate račun root, manj verjetno boste poškodovali sistem. Manj verjetno boste zamešali privilegije računa root s tistimi običajnega uporabnika.

Predstavljajte si račun root kot poseben čarovniški klobuk, ki vam daje ogromno moč, s katero lahko, z zamahom svoje roke, uničite celotna mesta. Dobra zamisel je, da ste malce previdni pri tem, kar počnete s svojimi rokami. Ker je lahko zamahnuti z vašimi rokami na uničevalen način, kljub čudovitemu občutku ni dobra zamisel, da bi nosili čarovniški klobuk takrat, ko to ni potrebno.

O odgovornostih sistema bomo spregovorili podrobneje od strani 158 naprej.

4.2 Zaganjanje sistema

Nekateri ljudje zaganjajo Linux z diskete, ki vsebuje kopijo jedra za Linux. To jedro ima vkodirano korensko particijo Linuxa, zato ve, kje mora iskati korenski datotečni sistem. Takšne vrste je na primer disketa, ki jo Slackware ustvari med namestitvijo.

Za ustvarjanje vaše lastne zagonske diskete poiščite sliko jedra na vašem trdem disku. Morala bi biti v datoteki `/vmlinuz` ali `/vmlinux`. V nekaterih namestitvah je `/vmlinuz` simbolna povezava na pravo jedro, zato boste morda morali izslediti jedro s sledenjem povezavam.

Ko enkrat veste, kje je jedro, nastavite korensko napravo slike jedra na ime vaše korenske particije Linuxa z ukazom `rdev`. Skladnja za uporabo tega ukaza je

```
rdev ime-jedra korenska-naprava
```

kjer je *ime-jedra* ime slike jedra in *korenska-naprava* ime korenske particije Linuxa. Na primer, za nastavitev korenske naprave v jedru `/vmlinuz` na `/dev/hda2`, uporabite ukaz

```
# rdev /vmlinuz /dev/hda2
```

Ukaz `rdev` lahko nastavlja tudi druge izbire v jedru, kot je privzeti način SVGA za uporabo ob zagonu. Ukaz

```
# rdev -h
```

izpiše sporočilo s pomočjo na zaslon. Ko usposobite korensko napravo (angl. root device), preprosto prekopirajte sliko jedra na disketo. Pred kopiranjem podatkov na katerokoli disketo je vendarle dobra zamisel, da uporabite pripomoček MS-DOS-a `FORMAT.COM` ali program `fdformat` v Linuxu za formatiranje diskete. To zapiše informacije o sektorjih in sledeh, ki so ustrezne za kapaciteto diskete.

Disketni formati in gonilniki naprav zanje so nadalje obravnavani, začenši s stranjo 166.

Datoteke gonilnikov naprav, kot je bilo že omenjeno, počivajo v imeniku `/dev`. Za prepis jedra iz datoteke `/vmlinuz` na disketo v `/dev/fd0` uporabite ukaz

```
# cp /vmlinuz /dev/fd0
```

Ta disketa bi zdaj morala pognati Linux.

4.2.1 Uporaba nalagalnika LILO

LILO je ločen zagonski nalagalnik, ki počiva na vašem trdem disku. Izvede se le, ko zaganja sistem s trdega pogona in lahko samodejno zažene Linux iz slike jedra, shranjene tam.

LILO se lahko uporablja kot prvostopenjski zagonski nalagalnik za precej operacijskih sistemov, kar vam omogoča, da ob zagonu izberete operacijski sistem kot je Linux ali MS-DOS. Z uporabo LILO se zažene privzeti operacijski sistem, razen če med zagonskim zaporedjem pritisnete `[Shift]` ali je podano navodilo za pozornik prompt v datoteki `lilo.conf`. V obeh primerih vam bo priskrbljen zagonski pozornik, kjer vpišete ime operacijskega sistema, ki naj se požene (kot »linux« ali »msdos«). Če v zagonskem pozorniku pritisnete `[Tab]`, se prikaže seznam operacijskih sistemov, ki jih pozna sistem LILO.

Najpreprostejši način za namestitev nalagalnika LILO je ureditev nastavitvene datoteke `/etc/lilo.conf`. Ukaz

```
# /sbin/lilo
```

prepiše spremenjeno nastavitvev `lilo.conf` na zagonski sektor trdega diska in se mora pognati vsakič, ko spremenite `lilo.conf`.

Nastavitvena datoteka za LILO vsebuje »kitico« za vsak operacijski sistem, ki ga želite zagnati. Najboljši način za demonstracijo tega je s primerom. Datoteka `lilo.conf` spodaj je za sistem, ki ima korensko particijo Linuxa na `/dev/hda1` in particijo za MS-DOS na `/dev/hda2`.

```
# Naroči LILO naj spremeni zagonski zapis na /dev/hda (prvi
# trdi pogon ne-SCSI). Če ne zaganjate iz pogona /dev/hda,
# spremenite naslednjo vrstico.
boot = /dev/hda

# Nastavi zdrav grafični način.
vga = normal

# Nastavi premor v mili-sekundah. To je čas, ki ga imate
# za pritisk tipke 'SHIFT', da se prikaže pozornik LILO:,
# če niste določili navodila 'prompt'.
delay = 60

# Ime zagonskega nalagalnika. Ni razloga za spreminjanje
# tega, razen če ste resno hekali po nalagalniku LILO.
install = /boot/boot.b

# To prisili LILO, da vas vpraša, kateri OS bi radi pgnali.
# Tipka 'TAB' v pozorniku LILO: bo izpisala seznam OS-ov,
# dostopnih za zagon, glede na imena, naštetih v spodnjem
# navodilu 'label='.
prompt

# Naj LILO izvede nekatere optimalizacije.
compact

# Kitica za korensko particijo Linuxa na /dev/hda1.
image = /vmlinuz      # Lokacija jedra
```

```

label = linux      # Ime OS-a (za zagonski menu LILA)
root = /dev/hda1   # Lokacija korenske particije
read-only          # Priklopi le za branje

# Kitica za particijo MSDOS-a na /dev/hda2.
other = /dev/hda2  # Lokacija particije
table = /dev/hda   # Lokacija particijske tabele za /dev/hda2
label = msdos      # Ime OS-a (za zagonski menu)

```

Kitica o prvem operacijskem sistemu določa privzeti operacijski sistem, ki naj ga LILO zažene. Vedite tudi, da ni razloga, da bi z `rdev` določili korensko particijo v sliki jedra, če uporabljate vrstico »`root =`«. LILO to nastavi ob zagonu sistema.

Namestitveni program za Microsoft Windows '95 prepiše zagonski upravljalnik LILO. Če boste po namestitvi LILA namestili na vašem sistemu Windows '95, se prepričajte, da boste najprej ustvarili zagonsko disketo (glejte razdelek 4.2). Z zagonsko disketo lahko poženet Linux in ponovno namestite LILO, ko se namestitev Windows '95 konča. To se naredi preprosto kot `root` z vnosom ukaza `/sbin/lilo`, enako kot v zgornjem koraku. Particije z Windows '95 se lahko nastavijo tako, da se zaženejo z nalagalnikom LILO z uporabo enakih vnosov v `lilo.conf` kot tistih za zagon s particije MS-DOS.

Več podatkov o nalagalniku LILO boste našli v *Linux FAQ* (glejte dodatek A), vključno s tem, kako uporabljati LILO za zagon z zagonskim upravljalnikom Boot Manager sistema OS/2.

4.3 Zaustavitev

Zaustavitev sistema Linux je lahko zapletena. Nikoli ne bi smeli preprosto izključiti elektrike ali pritisniti tipke za reset. Jedro shranjuje informacije o branju/pisanju podatkov na disk v vmesnih pomnilnikih (angl. buffers) v RAM-u. Če ponovno zaženete sistem, ne da bi dali jedru priložnost, da zapiše svoje vmesne pomnilnike na disk, lahko pokvarite datotečne sisteme.

Med zaustavitvijo se upoštevajo tudi drugi previdnostni ukrepi. Vsem procesom se pošlje signal, ki jim omogoča mirno smrt (tako da najprej zapišejo in zaprejo vse datoteke, na primer). Datotečni sistemi se zaradi varnosti odklopijo. Če želite, lahko sistem tudi opozori uporabnike, da se sistem ustavlja, in jim da priložnost za odjavo.

Najpreprostejši način za ustavitev sistema je z ukazom `shutdown`. Format tega ukaza je

```
shutdown čas opozorilo
```

Argument `čas` je čas zaustavitve sistema v obliki `uu:mm:ss` in `opozorilo` je sporočilo, ki se pred zaustavitvijo prikaže na vseh uporabniških terminalih. Alternativno lahko določite čas kot »`now`« (zdaj!) za takojšnjo zaustavitev. Izbira `-r` pomeni ukaz `shutdown` ponoven zagon sistema po njegovi zaustavitvi.

Na primer, za ustavitev in ponoven zagon sistema ob 20^h uporabite ukaz

```
# shutdown -r 20:00
```

Za takojšnjo zaustavitev sistema brez opozorilnih sporočil ali čakanja lahko uporabite ukaz `halt`. Ukaz `halt` je uporaben, če ste edini, ki uporablja sistem, in bi radi zaustavili in izključili stroj.

- ◇ Ne izključujte napetosti ali zaganjajte sistema, dokler ne vidite sporočila, da je sistem zaustavljen:

```
The system is halted
```

Zelo je pomembno, da sistem ustavite »na čist način«, z uporabo ukaza shutdown ali halt. Na nekaterih sistemih bo pritisk **Ctrl-Alt-Del** prestrežen in bo pognal shutdown. Na drugih sistemih bo uporaba »triprstnega pozdrava« takoj ponovno zagnala sistem in povzročila katastrofo.

4.4 Datoteka /etc/inittab

Takoj ko se Linux zažene in jedro priklopi korenski datotečni sistem, je prvi program, ki ga sistem izvede, init. Ta program je odgovoren za poganjanje sistemskih začetnih skriptov in spremeni delovanje sistema iz njegovega prvotnega zagonskega stanja v standardno, večuporabniško stanje. Program init tudi požene prijavne ukazne lupine na vseh napravah tty na sistemu in določa druge začetne in ustavitvene postopke.

Ko se zažene, init ostaja tiho v ozadju, pregleduje, in če je potrebno, spreminja tekoče stanje sistema. Obstaja veliko podrobnosti, na katere mora gledati program init. Ta opravila so definirana v datoteki /etc/inittab. Primer datoteke /etc/inittab je prikazan spodaj.

- ◇ Nepravilno spreminjanje datoteke /etc/inittab vam lahko onemogoči prijavo v vaš sistem. Kot minimum, ko spreminjate datoteko /etc/inittab, imejte v roki izvod originala, pravilno datoteko in zagonsko/korensko rešilno disketo, če slučajno naredite napako.

```
#
# inittab      Ta datoteka opisuje, kako naj bi procesi INIT
#              nastavili sistem za delovanje določenega
#              načina delovanja (angl. run-level).
#
# Različica:   @(#)inittab      2.04      17/05/93      MvS
#              2.10      02/10/95      PV
#
# Avtor:       Miquel van Smoorenburg,
#              <miquels@drinkel.nl.mugnet.org>
# Spremenil:   Patrick J. Volkerding,
#              <volkerdi@ftp.cdrom.com>
# Manjše popravke naredil:
#              Robert Kiesling, <kiesling@terracom.net>
#
# Privzeti način delovanja.
id:3:initdefault:

# Sistemska inicializacija (teče, ko se sistem zaganja).
si:S:sysinit:/etc/rc.d/rc.S

# Skript za prekllop v enouporabniški način (način delovanja 1).
su:1S:wait:/etc/rc.d/rc.K

# Skript za prekllop v večuporabniški način.
```

```

rc:23456:wait:/etc/rc.d/rc.M

# Kaj storiti ob Ctrl-Alt-Del
ca::ctrlaltdel:/sbin/shutdown -t5 -rfn now

# Način delovanja 0 zaustavi sistem.
l0:0:wait:/etc/rc.d/rc.0

# Način delovanja 6 ponovno zažene sistem.
l6:6:wait:/etc/rc.d/rc.6

# Kaj storiti, ko zmanjka električne napetosti (izključitev
# sistema v enouporabniški način).
pf::powerfail:/sbin/shutdown -f +5 "ZMANJKUJE NAPETOSTI"

# Če se napetost vrne pred ugasnitvijo (angl. shutdown),
# prekliči tekoče ugašanje.
pg:0123456:powerokwait:/sbin/shutdown -c "NAPETOST SE JE VRNILA"

# Če se napetost vrne v enouporabniškem načinu, se vrni
# v večuporabniški način.
ps:S:powerokwait:/sbin/init 5

# Getty-ji v večuporabniškem načinu na konzolah in zaporednih
# linijah.
#
# POZOR POZOR POZOR - Nastavite to na vaš getty, sicer se ne
# boste mogli prijaviti!!
#
# Pazite: za 'agetty' uporabljate obliko hitrost-linije, linija.
# Za 'getty_ps' uporabljate linija, hitrost-linije in tudi
# uporabite 'gettydefs'
c1:1235:respawn:/sbin/agetty 38400 tty1 linux
c2:1235:respawn:/sbin/agetty 38400 tty2 linux
c3:1235:respawn:/sbin/agetty 38400 tty3 linux
c4:1235:respawn:/sbin/agetty 38400 tty4 linux
c5:1235:respawn:/sbin/agetty 38400 tty5 linux
c6:12345:respawn:/sbin/agetty 38400 tty6 linux

# Zaporedne linije
# s1:12345:respawn:/sbin/agetty -L 9600 ttyS0 vt100
s2:12345:respawn:/sbin/agetty -L 9600 ttyS1 vt100

# Klicne linije
d1:12345:respawn:/sbin/agetty -mt60 38400,19200,9600,2400,1200 ttyS0 vt100
#d2:12345:respawn:/sbin/agetty -mt60 38400,19200,9600,2400,1200 ttyS1 vt100

# Način delovanja 4 je bil včasih sistem le z X-window,
# dokler nismo ugotovili, da vrže init v zanko, ki vseskozi
# vzdržuje vašo povprečno obremenjenost vsaj na 1. Torej je
# zdaj odprta prijavna seja z getty na tty6. Upajmo, da ne

```

```
# bo nihče opazil. ;^)
# Morda tako ali tako ni slabo imeti vsaj eno tekstovno konzolo,
# za vsak slučaj, če se kaj zgodi z X.
x1:4:wait:/etc/rc.d/rc.4

# Konec datoteke /etc/inittab
```

Ob začetku ta /etc/inittab požene šest navideznih konzol, prijavitni pozornik za modem, priključen na /dev/ttyS0, in prijavitni pozornik za znakovni terminal, povezan po zaporedni liniji RS-232 na /dev/ttyS1.

Na kratko, init stopa skozi vrsto **načinov delovanja** (angl. run levels), ki ustrezajo različnim delovnim stanjem sistema. V način delovanja 1 se vstopi takoj, ko se sistem zažene, načina delovanja 2 in 3 sta običajna, večopravilna delovna načina sistema, način delovanja 4 požene X Window System s prikaznim upravljalnikom xdm in način delovanja 6 ponovno zažene sistem. Načini delovanja, povezani z vsakim ukazom, so druge postavke v vsaki vrstici datoteke /etc/inittab.

Na primer, vrstica

```
s2:12345:respawn:/sbin/agetty -L 9600 ttyS1 vt100
```

bo vzdrževala prijavitni pozornik na zaporednem terminalu za načine delovanja 1–5. Postavka »s2« pred prvim dvopičjem je simbolni identifikator, ki ga interno uporablja init. Postavka respawn je ključna beseda programa init, ki se pogosto uporablja v povezavi z zaporednimi terminali. Če po določenem časovnem obdobju program agetty, ki poraja terminalski prijavitni pozornik, ne prejme vhoda s terminala, se čas programu izteče in izvajanje se prekine. Postavka »respawn« pove init-u, naj spet izvede agetty ter tako zagotovi, da je na terminalu vedno prijavitni pozornik, ne glede na to, ali je kdo prijavljen ali ne. Ostali parametri se predajo neposredno programu agetty in mu povedo, naj porodi prijavno ukazno lupino, podatkovno hitrost zaporedne linije, zaporedno napravo in terminalski tip, kot so definirani v datoteki /etc/termcap ali /etc/terminfo.

Program /sbin/agetty ureja mnoge podrobnosti, povezane s terminalskim vhom/izhodom sistema. Obstaja nekaj različic, ki se uporabljajo na sistemih Linux. Te vključujejo mgetty, psgetty ali preprosto getty.

V primeru vrstice v /etc/inittab

```
dl:12345:respawn:/sbin/agetty -mt60 38400,19200,9600,2400,1200 ttyS0 vt100
```

ki dovoljuje uporabnikom prijavo po modemu, priključenem na zaporedno linijo /dev/ttyS0, so parametri za /sbin/agetty enaki »-mt60«, kar dovoljuje sistemu, da se sprehodi skozi vse hitrosti modema, ki jih lahko uporablja klicoči, ko se prijavlja v sistem, in da ustavi /sbin/agetty, če po 60 sekundah še ni povezave. Temu se reče **dogovarjanje** povezave. Podprte hitrosti modema so naštetje tudi v ukazni vrstici, kot tudi zaporedna linija za uporabo in terminalski tip. Seveda, oba modema morata podpirati hitrost prenosa podatkov, za katero se stroja končno dogovorita.

V tem razdelku smo prezrli mnoge pomembne podrobnosti. Opravila, ki jih vzdržuje /etc/inittab, bi lahko sestavljala svojo knjigo. Za nadaljnje informacije so dobre začetne točke strani priročnika o programih init in agetty in spis *Serial HOWTO* Dokumentacijskega projekta za Linux. Dostopen je iz virov, naštetih v dodatku A.

4.5 Upravljanje datotečnih sistemov

Še eno opravilo za upravitelja sistema je skrb za datotečne sisteme. Večino tega sestavljajo periodična preverjanja datotečnih sistemov, da ne vsebujejo napak ali pokvarjenih datotek. Veliko sistemov Linux tudi samodejno preverja datotečne sisteme v času zagona.

4.5.1 Priklop datotečnih sistemov

Preden je datotečni sistem dostopen, mora biti **priklopljen** na imenik. Na primer, če imate datotečni sistem na disketi, ga morate priklopiti pod imenikom, kot je `/mnt`, če želite dostopati do datotek na disketi (glejte stran 169). Po priklopu datotečnega sistema se vse datoteke datotečnega sistema prikažejo v tem imeniku. Po odklopu datotečnega sistema bo imenik (v tem primeru `/mnt`) prazen.

Enako velja tudi za datotečne sisteme na trdem disku. Sistem samodejno priklopi datotečne sisteme na vašem trdem disku v času zagona. Tako imenovani »korenski datotečni sistem« se priklopi na imenik `/`. Če imate poseben datotečni sistem za `/usr`, se priklopi na `/usr`. Če imate le korenski datotečni sistem, bodo vse datoteke (vključno s tistimi iz `/usr`) obstajale na tem datotečnem sistemu.

Ukaza `mount` in `umount` (ne *unmount*) se uporabljata za priklop in odklop datotečnih sistemov. Ukaz

```
mount -av
```

se samodejno požene iz izvedljivega skripta `/etc/rc` v času zagona ali iz skripta `/etc/rc.d/boot` (glejte stran 176) na nekaterih sistemih Linux. Datoteka `/etc/fstab` priskrbi informacije o datotečnih sistemih in točkah priklopa. Primer datoteke `/etc/fstab` je

# naprava	imenik	tip	izbire
/dev/hda2	/	ext2	defaults
/dev/hda3	/usr	ext2	defaults
/dev/hda4	none	swap	sw
/proc	/proc	proc	none

Prvo polje, *naprava*, je ime particije za priklop. Drugo polje je točka priklopa. Tretje polje je tip datotečnega sistema kot `ext2` (za `ext2fs`) ali `minix` (za datotečne sisteme vrste Minix). Tabela 4.1 našteva različne tipe datotečnih sistemov, ki se lahko priklopijo v Linux.¹ Morda vsi ti tipi datotečnih sistemov ne bodo dostopni na vašem sistemu, ker mora biti v jedro prevedena podpora zanje. Glejte stran 170 za informacije o gradnji jedra.

Zadnje polje v datoteki `fstab` so izbire za priklop z `mount`. To je običajno nastavljeno na privzeto vrednost, `defaults`.

Izmenjalne particije so vključene v datoteko `/etc/fstab`. Kot imenik za priklop imajo navedeno vrednost `none`, za tip pa `swap`. Ukaz `swapon -a`, ki se izvede iz `/etc/rc` ali `/etc/init.d/boot`, se uporablja za vklop izmenjevanja na vseh izmenjalnih napravah, ki so našteje v `/etc/fstab`.

Datoteka `/etc/fstab` vsebuje poseben vnos za datotečni sistem `/proc`. Kot je opisano na strani 131, se datotečni sistem `/proc` uporablja za shranjevanje informacije o sistemskih procesih, dostopnem pomnilniku in tako naprej. Če datotečni sistem `/proc` ni priklopljen, ukazi, kot je `ps`, ne bodo delovali.

¹Ta tabela je bila osvežena za jedro 2.0.33.

Datotečni sistem	Ime tipa	Komentar
Second Extended	ext2	Najpogostejši datotečni sistem Linuxa.
Extended	ext	Nasledil ga je ext2.
Minix	minix	Originalni datotečni sistem Minixa; redko uporabljan.
Xia	xia	Kot ext2, a redko uporabljan.
UMSDOS	umsdos	Uporabljan za namestitev Linuxa na particijo MS-DOS.
MS-DOS	msdos	Uporabljan za dostop do datotek MS-DOS-a.
/proc	proc	Ponuja informacije o procesih za ps itd.
ISO 9660	iso9660	Format, ki ga uporablja večina CD-ROM-ov.
Xenix	xenix	Uporabljan za dostop do datotek Xenixa.
System V	sysv	Uporabljan za dostop do datotek na različicah System V za x86.
Coherent	coherent	Uporabljan za dostop do datotek Coherenta.
HPFS	hpfs	Le bralni dostop do particij HPFS (DoubleSpace).

Tabela 4.1: Tipi datotečnih sistemov za Linux

- ◇ Ukaz `mount` lahko uporablja le `root`. To zagotavlja varnost sistema. Saj ne želite, da običajni uporabniki priklaplajo in odklapljajo datotečne sisteme ob vsakem preblisku. Obstaja precej dostopnih programskih paketov, ki dovoljujejo nekorenskim uporabnikom priklapljanje in odklapljanje datotečnih sistemov, posebej disket, ne da bi bila prizadeta varnost sistema.

Ukaz `mount -av` pravzaprav priklopi vse datotečne sisteme, razen korenskega (v zgornji tabeli, `/dev/hda2`). Korenski datotečni sistem samodejno priklopi že jedro v času zagona.

Namesto uporabe `mount -av` lahko priklopite datotečni sistem na roke. Ukaz

```
# mount -t ext2 /dev/hda3 /usr
```

je ekvivalenten priklopu datotečnega sistema z vnosom za `/dev/hda3` v zgornji zgledni datoteki `/etc/fstab`.

4.5.2 Imena gonilnikov naprav

Poleg imen particij, naštetih v datoteki `/etc/fstab`, Linux prepozna številne fiksne in odstranljive naprave. Razvrščene so po tipu, vmesniku in vrstnem redu, kot so nameščene. Na primer, prvi trdi disk vašega sistema, če je to IDE ali starejši trdi disk MFM, je nadzorovan z gonilnikom naprave, na katero kaže `/dev/hda`. Prva particija trdega diska je `/dev/hda1`, druga particija je `/dev/hda2`, tretja particija je `/dev/hda3` in tako naprej. Prva particija drugega pogona IDE je pogosto `/dev/hdb1`, druga particija `/dev/hdb2` in tako dalje. Shema poimenovanja za najbolj pogosto nameščene diske IDE za arhitekturo Intel, stroje z vodili ISA in PCI je podana v tabeli 4.2.

Gonilnik naprave	Disk
<code>/dev/hda</code>	Nadrejeni disk IDE, primarno vodilo IDE.
<code>/dev/hdb</code>	Podrejeni disk IDE, primarno vodilo IDE.
<code>/dev/hdc</code>	Nadrejeni disk IDE, sekundarno vodilo IDE.
<code>/dev/hdd</code>	Podrejeni disk IDE, sekundarno vodilo IDE.

Tabela 4.2: Imena gonilnikov naprav IDE

Ta imena naprav uporabljajo tudi pogoni CD-ROM in tračne enote, ki uporabljajo razširjeni vmesnik pogona IDE/ATAPI.

Mnogi stroji, vključno z osebnimi računalniškimi delovnimi postajami zgornjega razreda in stroji, ki tečejo na procesorju Alpha izdelovalca Digital Equipment Corporation, pa uporabljajo vmesnik SCSI (angl. Small Computer System Interface, sistemski vmesnik za male računalnike). Poimenovalni dogovori za naprave SCSI se zaradi večje prožnosti naslavljanja SCSI nekako razlikujejo od zgornjih. Prvi trdi disk SCSI na sistemu je `/dev/sda`, drugi disk SCSI je `/dev/sdb` in tako naprej. Seznam pogostih naprav SCSI je podan v tabeli 4.3.

Gonilnik naprave	Pogon
<code>/dev/sda</code>	Prvi trdi disk SCSI.
<code>/dev/sdb</code>	Drugi trdi disk SCSI.
<code>/dev/st0</code>	Prvi tračni pogon SCSI.
<code>/dev/st1</code>	Drugi tračni pogon SCSI.
<code>/dev/scd0</code>	Prvi pogon CD-ROM SCSI.
<code>/dev/scd1</code>	Drugi pogon CD-ROM SCSI.

Tabela 4.3: Gonilniki naprav SCSI

Upoštevajte, da se CD-ROM-i in tračni pogoni imenujejo drugače kot trdi diski SCSI. Odstranljivi mediji SCSI, kot je pogon Iomega Zip, upoštevajo poimenovalne dogovore za ne-odstranljive pogone SCSI. Uporaba pogona Zip za izdelovanje varnostnih kopij vaših podatkov je opisana od strani 167 naprej.

Tokovni tračni pogoni, kot so tisti, ki berejo in pišejo magnetne trakove formatov QIC-02, QIC-40 in QIC-80, imajo svoj lastni nabor imen naprav, ki so opisane na strani 168.

Disketni pogoni uporabljajo spet drugo shemo poimenovanja, ki je opisana na strani 166.

4.5.3 Preverjanje datotečnih sistemov

Navadno je dobro vsake toliko časa pregledati vaše datotečne sisteme, ali vsebujejo pokvarjene datoteke. Nekateri sistemi samodejno preverjajo datotečne sisteme v času zagona (z ustreznimi ukazi v `/etc/rc` ali `/etc/init.d/boot`).

Ukaz za preverjanje datotečnega sistema je odvisen od vrste datotečnega sistema. Za datotečne sisteme `ext2fs` (najpogostejše uporabljano vrsto) je ta ukaz `e2fsck`. Na primer, ukaz

```
# e2fsck -av /dev/hda2
```

preveri datotečni sistem `ext2fs` na `/dev/hda2` in samodejno odpravi vsakršne napake.

Navadno je koristno odklopiti datotečni sistem pred njegovim preverjanjem, to je celo potrebno, če ga namerava `e2fsck` popravljati. Ukaz

```
# umount /dev/hda2
```

odklopi datotečni sistem na `/dev/hda2`. Izjema je korenski datotečni sistem, ki ga ni mogoče odklopiti. Za preverjanje korenkega datotečnega sistema, ko je odklopljen, bi morali uporabiti vzdrževalno zagonsko/korensko disketo (glejte stran 178). Datotečnega sistema tudi ne morete odklopiti, če je katerakoli od datotek, ki jih vsebuje, »zasedena« – se pravi,

če jo uporablja katerikoli tekoči proces. Na primer, ne morete odklopiti datotečnega sistema, če je na njem trenutni delovni imenik kateregakoli uporabnika. Namesto tega boste prejeli sporočilo o napaki zasedenosti naprave »Device busy«.

Drugi tipi datotečnih sistemov uporabljajo drugačne oblike ukaza `e2fsck`, kot sta `efscck` in `xfscck`. Na nekaterih sistemih lahko preprosto uporabite ukaz `fsck`, ki samodejno ugotovi tip datotečnega sistema in izvede ustrezeni ukaz.

- ◇ Če `e2fsck` poroča, da je izvedel popravila na priklopljenem datotečnem sistemu, *morate* takoj ponovno zagnati računalnik. Izdati morate ukaz `shutdown -r` s katerim se izvede ponovni zagon. To dovoljuje sistemu, da spet uskladi informacije o datotečnem sistemu, ki ga je `e2fsck` spremenil.

Datotečnega sistema `/proc` ni treba nikoli preveriti na ta način. `/proc` je pomnilniški datotečni sistem in ga neposredno upravlja jedro.

4.6 Uporaba izmenjalne datoteke

Namesto da bi rezervirali posebno particijo za izmenjalni prostor, lahko uporabljate **izmenjalno datoteko** (angl. swap file). Vendar morate namestiti Linux in doseči, da vse deluje, preden ustvarite izmenjalno datoteko.

Ko je Linux nameščen, lahko uporabite naslednje ukaze za ustvarjanje izmenjalne datoteke. Spodnji ukaz ustvari izmenjalno datoteko velikosti 8208 blokov (približno 8 MB).

```
# dd if=/dev/zero of=/swap bs=1024 count=8208
```

Ta ukaz ustvari izmenjalno datoteko `/swap`. Parameter »count=« je velikost izmenjalne datoteke v blokih. Velikost bloka smo določili s parametrom »bs=«.

```
# mkswap /swap 8208
```

Ta ukaz inicializira izmenjevalno datoteko. Spet, nadomestite ime in velikost izmenjevalne datoteke z ustreznimi vrednostmi.

```
# sync
# swapon /swap
```

Zdaj sistem izmenjuje podatke med pomnilnikom in diskom na datoteko `/swap`. Ukaz `sync` zagotavlja, da je datoteka res zapisana na disk.

Precej slaba stran uporabe izmenjalne datoteke je, da se ves dostop do izmenjalne datoteke opravlja preko datotečnega sistema. To pomeni, da si bloki, ki sestavljajo izmenjalno datoteko, morda ne bodo blizu. Izvajanje morda ne bo tako hitro kot pri izmenjalni particiji, kjer bloki vedno ležijo skupaj in se V/I zahteve sklicujejo neposredno na napravo.

Še ena slaba stran velike izmenjalne datoteke je, da je večja nevarnost okvare datotečnega sistema, če gre kaj narobe. Ločevanje običajnih datotečnih sistemov in izmenjalnih particij preprečuje, da bi se to zgodilo.

Izmenjalne datoteke so lahko uporabne, če morate začasno uporabljati več izmenjalnega prostora. Če prevajate ogromen program in bi radi nekako pospešili stvari, lahko ustvarite začasno izmenjalno datoteko in jo uporabljate kot dodatek k običajnemu izmenjevalnemu prostoru.

Za odstranitev izmenjalne datoteke najprej uporabite `swapoff` kot v ukazu

```
# swapoff /swap
```

Potem lahko pobrišete datoteko.

```
# rm /swap
```

Vsaka izmenjevalna datoteka ali particija je lahko velika do 16 megabajtov, vendar lahko na vašem sistemu hkrati uporabljate do 8 izmenjevalnih datotek ali particij.

4.7 Upravljanje z uporabniki

Tudi če ste edini uporabnik na vašem sistemu, je pomembno razumeti vidike upravljanja z uporabniki pod Linuxom. Imeti bi morali vsaj račun zase (drugega kot `root`) za opravljanje večine vašega dela.

- ◇ Vsak uporabnik bi moral imeti svoj lastni račun. Le izjemoma je sprejemljivo, da si več ljudi deli isti račun. Vprašljiva je varnost, saj računi edinstveno istovetijo uporabnike sistema. Imeti morate možnost slediti, kaj kdo počne.

4.7.1 Pojmi upravljanja z uporabniki

Sistem beleži naslednje informacije o vsakem uporabniku:

uporabniško ime

Ta identifikator je edinstven za vsakega uporabnika. Primeri uporabniških imen so `larry`, `karl` in `mdw`. Lahko uporabljate črke in številke, kot tudi `»_«` in `».«` (podčrtaj in piko). Uporabniška imena so navadno omejena na 8 znakov dolžine.

uporabniška identifikacijska številka

Ta številka, okrajšana kot `UID`, je edinstvena za vsakega uporabnika. Sistem običajno beleži uporabnike po njihovih številkah `UID`, ne po uporabniškem imenu.

skupinska identifikacijska številka

Ta številka, okrajšana kot `GID` (iz angl. group ID), določa privzeto skupino uporabnika. V razdelku 3.10 opisujemo skupinska dovoljenja. Vsak uporabnik pripada eni ali več skupinam, kot določi sistemski upravitelj.

geslo

To je uporabniško šifrirano geslo. Za nastavljanje in spremembo uporabniških gesel se uporablja ukaz `passwd` (iz angl. password).

polno ime

Uporabnikovo »pravo ime« ali »polno ime« se shranjuje poleg uporabniškega imena. Na primer, uporabnik `schmoj` je lahko v resničnem življenju »Joe Schmo«.

domači imenik

To je imenik, v katerega je prvotno postavljen uporabnik po prijavi in kjer so shranjene njegove osebne datoteke. Vsakemu uporabniku pripada domači imenik, ki je navadno lociran pod imenikom `/home`.

prijavna ukazna lupina

Ta ukazna lupina se zažene za uporabnika po njegovi prijavi. Primera sta `/bin/bash` in `/bin/tcsh`.

Vse te informacije so shranjene v datoteki `/etc/passwd`. Vsaka vrstica te datoteke ima obliko

```
uporabniško ime:šifrirano geslo:UID:GID:polno ime:domači imenik:prijavna lupina
```

Primer bi lahko bil

```
kiwi:Xv8Q981g71oKK:102:100:Laura Poole:/home/kiwi:/bin/bash
```

V tem primeru je prvo polje, »kiwi«, uporabniško ime.

Naslednje polje, »Xv8Q981g71oKK«, je šifrirano geslo. Gesla se na sistemu ne shranjujejo v obliki, primerni za človeško branje. Geslo se šifrira s seboj kot skrivnim ključem. Z drugimi besedami, če želi kdo odšifrirati geslo, ga mora poznati. Ta oblika šifriranja je zmerno varna.

Nekateri sistemi uporabljajo »senčna gesla« (angl. shadow passwords), pri katerih se informacije o geslu shranjujejo v datoteki `/etc/shadow`. Ker lahko `/etc/passwd` bere ves svet, datoteka `/etc/shadow` priskrbi določeno stopnjo dodatne varnosti, saj so dovoljenja za dostop do nje precej bolj omejena. Senčna gesla ponujajo tudi druge lastnosti, kot je prenehanje veljavnosti gesla po določenem časovnem obdobju.

Tretje polje, »102«, je UID. Ta številka mora biti različna za vsakega uporabnika. Četrto polje, »100«, je GID. Ta uporabnica pripada skupini, oštevilčeni s 100. Informacije o skupinah se shranjujejo v datoteki `/etc/group`. Glejte razdelek 4.7.5 za več informacij.

Peto polje je polno ime uporabnice, »Laura Poole«. Zadnji dve polji sta uporabničin domači imenik (`/home/kiwi`) in njena prijavna ukazna lupina (`/bin/bash`), po vrsti. Ni nujno, da ima uporabničin domači imenik isto ime, kot je uporabniško ime. To le pomaga identificirati imenik.

4.7.2 Dodajanje uporabnikov

Ob dodajanju uporabnikov je treba narediti več korakov. Najprej se uporabniku dodeli zapis v `/etc/passwd` z enkratnim uporabniškim imenom in UID. Določena morajo biti polja o GID, uporabniškem imenu in drugih informacijah. Ustvarjen mora biti uporabnikov domači imenik, dovoljenja za ta imenik pa morajo biti nastavljena tako, da je uporabnik lastnik imenika. V domači imenik morajo biti nameščene inicializacijske datoteke za ukazno lupino in na sistemu morajo biti nastavljene druge datoteke (na primer čakalni imenik za prihajajočo uporabnikovo e-pošto).

Dodajanje uporabnikov na roko ni težko, a če poganjate sistem z mnogimi uporabniki, je zelo lahko kaj pozabiti. Najpreprostejši način za dodajanje uporabnikov je uporaba interaktivnega programa, ki samodejno osveži vse sistemske datoteke. Ime tega programa je `useradd` ali `adduser`, odvisno od nameščenega programja.

Ukaz `adduser` vzame potrebne informacije iz datoteke `/etc/adduser.conf`, ki definira standardno, privzeto nastavitve za vse nove uporabnike.

Tipična datoteka `/etc/adduser.conf` je prikazana spodaj.

```
# /etc/adduser.conf: nastavitve programa 'adduser'.
# Za popolno dokumentacijo glejte adduser(8) in adduser.conf(5).

# Spremenljivka DSHELL določa privzeto prijavno ukazno lupino
# na vašem sistemu.
DSHELL=/bin/bash
```

```

# Spremenljivka DHOME določa imenik, ki vsebuje domače imenike
# uporabnikov.
DHOME=/home

# Če je GROUPTHOMES "yes", potem bodo domači imeniki uporabniki
# ustvarjeni kot /home/ime-skupine/uporabnik.
GROUPTHOMES=no

# Če je LETTERHOMES "yes", potem bodo ustvarjeni domači imeniki
# imeli dodaten imenik - prvo črko uporabniškega imena. Na primer:
# /home/u/uporabnik.
LETTERHOMES=no

# Spremenljivka SKEL določa imenik, ki vsebuje "skeletne"
# uporabniške datoteke; z drugimi besedami, datoteke, kot je
# vzorčni skript .profile, ki se bodo prepisale v domači
# imenik novega uporabnika, ko je ta ustvarjen.
SKEL=/etc/skel

# Nabor UID-ov za dinamično dodeljevanje upravljalne in sistemske
# račune sega od FIRST_SYSTEM_UID do vključno LAST_SYSTEM_UID.
FIRST_SYSTEM_UID=100
LAST_SYSTEM_UID=999

# Nabor UID-ov za dinamično dodeljevanje uporabniške račune
# sega od FIRST_UID do vključno LAST_UID.
FIRST_UID=1000
LAST_UID=29999

# Spremenljivka USERGROUPS je lahko nastavljena na "yes" ali "no".
# Če je "yes", je privzeto vsakemu ustvarjenemu uporabniku dodeljena
# lastna skupina za uporabo in njegov domači imenik bo g+s.
# Če je "no", bo vsak novi uporabnik priključen skupini, katere
# gid je USERS_GID (glejte spodaj).
USERGROUPS=yes

# Če je USERGROUPS "no", potem mora biti USERS_GID GID skupine
# 'users' (ali druge primerne skupine) na vašem sistemu.
USERS_GID=100

# Če je nastavljen QUOTAUSER, bo za novega uporabnika določena
# privzeta kvota z ukazom 'edquota -p QUOTAUSER newuser'
QUOTAUSER=""

```

Poleg definiranja že nastavljenih spremenljivk, ki jih uporablja ukaz `adduser`, `/etc/adduser.conf` določa tudi, kje so shranjene privzete sistemske nastavitvene datoteke za vsakega uporabnika. V tem primeru se nahajajo v imeniku `/etc/skel`, kot definira zgornja vrstica `SKEL=`. Datoteke, ki so postavljene v ta imenik, bo ukaz `adduser` samodejno namestil v domači imenik novega uporabnika, podobno kot sistemske, privzete datoteke `.profile`, `.tcshrc` ali `.bashrc`.

4.7.3 Brisanje uporabnikov

Uporabnike brišemo z ukazom `userdel` ali `deluser`, odvisno od programja, nameščenega na sistem.

Če bi radi začasno »onemogočili« uporabniku, da bi se prijavil v sistem, ne da bi izbrisali njegov račun, preprosto pripnite zvezdico (`*`) v polje za geslo v datoteki `/etc/passwd`. Na primer, sprememba vnosa za kiwi v `/etc/passwd`

```
kiwi:*Xv8Q981g71oKK:102:100:Laura Poole:/home/kiwi:/bin/bash
```

onemogoča kiwi, da bi se prijavila v sistem.

4.7.4 Nastavitev uporabniških atributov

Ko ste ustvarili uporabnika, boste morda želeli zanj spremeniti attribute, kot so domači imenik ali geslo. Najpreprostejši način za to je neposredna sprememba vrednosti v datoteki `/etc/passwd`. Za nastavitev uporabniškega gesla uporabite `passwd`. Ukaz

```
# passwd larry
```

bo spremenil larry-jevo geslo. Le root lahko na tak način spreminja gesla drugih uporabnikov. Navadni uporabniki lahko spreminjajo le svoja lastna gesla.

Na nekaterih sistemih ukaza `chfn` in `chsh` omogočata uporabnikom, da si nastavijo svoje polno ime in attribute prijave lupine. Če ne, mora ta atributa zanje spremeniti sistemski upravitelj.

4.7.5 Skupine

Kot smo že omenili zgoraj, vsak uporabnik pripada eni ali več skupinam. Edini pravi namen povezovanja v skupine se nanaša na dovoljenja za uporabo datotek. Kot se spomnite iz razdelka 3.10, ima vsaka datoteka »skupinsko lastništvo« in nabor skupinskih dovoljenj, ki definirajo, kako lahko uporabniki iz te skupine dostopajo do datoteke.

Obstaja več skupin, ki jih definira sistem, kot so `bin`, `mail` in `sys`. Uporabniki ne bi smeli pripadati nobeni od teh skupin; uporabljajo se za sistemska dovoljenja datotek. Namesto tega bi morali uporabniki pripadati posamezni skupini, kot je `users`. Vzdržujete lahko tudi več skupin za uporabnike, kot `studenti`, `osebje` in `predavatelji`.

Datoteka `/etc/group` vsebuje informacije o skupinah. Oblika vsake vrstice je

```
ime skupine:geslo:GID:drugi člani
```

Nekatere zgledne skupine so lahko:

```
root::0:
users::100:mdw,larry
guest::200:
other::250:kiwi
```

Prva skupina, `root`, je posebna sistemska skupina, rezervirana za račun `root`. Naslednja skupina, `users`, je za običajne uporabnike. Ima GID 100. Uporabnika `mdw` in `larry` sta člana te skupine. Spomnite se, da je v `/etc/passwd` vsakemu uporabniku prirejena privzeta številka GID. Vendar lahko uporabniki pripadajo več kot eni skupini, tako da so njihova

imena našeta v vrsticah drugih skupin v datoteki `/etc/group`. Ukaz `groups` našteva, do katerih skupin imate dostop.

Tretja skupina, `guest`, je za gostujoče uporabnike in `other` za »druge« uporabnike. Uporabnica `kiwi` ima dostop tudi do te skupine, poleg tega je privzeto že članica skupine `users`, saj je njena številka GID enaka 100.

Polje »geslo« v datoteki `/etc/group` se včasih uporablja za nastavljanje gesla za skupinski dostop. To je le redko potrebno. Za zavarovanje uporabnikov, da se ne bi spremenili v privilegirane skupine (z ukazom `newgroup`), nastavite polje z geslom na »*«.

Za dodajanje skupin na vaš sistem lahko uporabljate `addgroup` ali `groupadd`. Navadno je preprosteje, če sami dodate vnose v datoteko `/etc/group`, saj za skupino ni potrebno nobeno drugo nastavljanje. Skupino izbrišete preprosto tako, da izbrišete njen vnos v datoteki `/etc/group`.

4.7.6 Odgovornosti sistemskega upravljanja

Ker ima sistemski upravitelj tako veliko moči in odgovornosti, nekateri uporabniki izkoristijo prvo priložnost, ko se lahko prijavijo kot `root` na sistemu Linux ali drugje, da zlorabijo privilegije korenskega uporabnika. Poznal sem tako imenovane »sistemске upravitelje«, ki so brali pošto drugih uporabnikov, brisali uporabniške datoteke, ne da bi o tem opozorili njihove lastnike in se v splošnem obnašali kot otroci, ko jim je dana tako silna »igrača«.

Ker ima upravitelj takšno moč nad sistemov, je potrebna določena stopnja zrelosti in samonadzora, da uporablja račun `root`, za kar je namenjen – za tek sistema. Obstaja nepisani zakonik časti, ki obstaja med sistemskim upraviteljem in uporabniki na sistemu. Kako bi se počutili, če bi vaš sistemski upravitelj bral vašo e-pismo ali pregledoval vaše datoteke? Še vedno ni vplivnega pravnega precedensa za elektronsko zasebnost na večuporabniških računalniških sistemih. Na sistemih Unix ima uporabnik `root` možnost, da obide vse varnostne in zasebnostne mehanizme sistema. Pomembno je, da sistemski upravitelj s svojimi uporabniki razvije zaupanja vredno zvezo. To želimo še posebej poudariti.

4.7.7 Shajanje z uporabniki

Sistemski upravitelji lahko izberejo dva principa, ko imajo opravka z zlonamernimi uporabniki: lahko so paranoični ali zaupljivi. Paranoični sistemski upravitelj navadno povzroči več škode, kot jo prepreči. Eden mojih priljubljenih izrekov je, »Nikoli ne pripisujte zlobi česarkoli, kar lahko pripišete neumnosti«. Z drugimi besedami, večina uporabnikov nima zmožnosti ali znanja, da bi zares škodovali sistemu. V devetdesetih odstotkih, kadar uporabnik povzroča težave na sistemu (denimo, zapolnjuje uporabniško particijo z velikimi datotekami ali poganja več izvodov velikega programa), se uporabnik preprosto ne zaveda, da ustvarja problem. Izsledil sem uporabnike, ki so povzročali ogromno težav, vendar so preprosto delovali iz nevednosti – ne iz zlobe.

Ko imate opraviti z uporabniki, ki povzročajo potencialne težave, ne bodite obtoževalni. Breme dokaza leži na vas; se pravi, še vedno velja pravilo, da je uporabnik »nedolžen, dokler se ne dokaže, da je kriv«. Najbolje je, da se preprosto pogovorite z uporabnikom in ga povprašate o težavah, namesto da bi se spuščali v spore. Zadnja stvar, ki jo želite, je, da bi se zamerili uporabniku. To bi sprožilo veliko sumov o vas – sistemskem upravitelju – da morda sistema ne poganjate pravilno. Če uporabnik misli, da mu ne zaupate ali da ga ne marate, vas lahko obtoži izbrisa datotek ali kršitve zasebnosti na sistemu. To vsekakor ni položaj, v katerem bi se radi znašli.

Če ugotovite, da uporabnik poskuša »vlamljati« ali drugače namenoma škodovati sistemu, ne vrnite zlobnega obnašanja z lastno zlobo. Namesto tega ga posvarite, vendar bodite prožni. V veliko primerih lahko ujamete uporabnika »na delu« škodovanja sistemu. Posvarite ga. Povejte mu, da naj se to ne zgodi več. Vendar če *ga* spet ujamete, da dela škodo, bodite popolnoma prepričani, da je bilo to namenoma. Ne želim niti začeti opisovati številnih primerov, kjer se je zdelo, da uporabnik povzroča težave, v resnici pa je šlo za nezgodo ali mojo lastno napako.

4.7.8 Postavljanje pravil

Najboljša pot za tek sistema ni z železno roko. Tako se morda priganja vojsko, a Linux ni zasnovan za takšno disciplino. Razumno se zdi zapisati nekaj preprostih in prožnih vodnikov. Manj kot imate pravil, manj verjetno je, da bodo kršena. Tudi če so vaša pravila popolnoma razumna in jasna, jih bodo uporabniki občasno še vedno prekršili, ne da bi to nameravali. To je posebej res za nove uporabnike, ki se uvajajo v sistem. Ni povsem očitno, da ne bi smeli pobrati gigabajta datotek in jih poslati po pošti vsem uporabnikom sistema. Uporabniki potrebujejo pomoč, da razumejo pravila in zakaj so ta tukaj.

Če določite pravila za uporabo vašega sistema, se prepričajte tudi, da je pojasnilo za določen napotek jasno. Če ne, bodo uporabniki iznašli vse vrste ustvarjalnih načinov, da se bodo izognili pravilu in ne bodo vedeli, da ga kršijo.

4.7.9 Kaj vse to pomeni

Ne razlagamo vam do zadnje podrobnosti, kako poganjate vaš sistem. To je odvisno od tega, kako ga uporabljate. Če imate veliko uporabnikov, so stvari zelo drugačne, kot če jih imate le malo ali če ste edini uporabnik na sistemu. Vendar je vedno koristno – v vsakem primeru – da razumete, kaj *zares* pomeni to, da ste sistemski upravitelj.

Če ste sistemski upravitelj, še ne pomeni, da ste čarovnik Linuxa. Obstaja veliko upraviteljev, ki vedo zelo malo o Linuxu. Podobno, veliko »običajnih« uporabnikov ve več o Linuxu kot katerikoli sistemski upravitelj. To, da ste sistemski upravitelj, vam tudi ne dovoljuje, da stresate zlobo na uporabnike. Če sistem dovoljuje upraviteljem brkljanje po uporabniških datotekah, to še ne pomeni, da imajo pravico to tudi početi.

Biti sistemski upravitelj ni velika reč. Ni pomembno, ali je vaš sistem drobcena 386 ali superračunalnik Cray. Poganjanje sistema je enako ne glede na velikost. S poznavanjem korenskega gesla si ne boste prislužili denarja ali slave. Omogočilo vam bo, da vzdržujete sistem in poskrbite, da bo tek. To je to.

4.8 Arhiviranje in komprimiranje datotek

Preden spregovorimo o varnostnih kopijah (angl. backups), moramo uvesti orodja, ki se uporabljajo za arhiviranje datotek na sistemih Unix.

4.8.1 Uporaba pripomočka *tar*

Najpogosteje se uporablja za arhiviranje datotek ukaz *tar*. Njegova ukazna skladnja je

tar *izbire datoteke*

kjer je *izbire* seznam ukazov in izbir za tar in *datoteke* seznam datotek za dodajanje ali izločitev iz arhiva.

Na primer, ukaz

```
# tar cvf rezerva.tar /etc
```

zapakira vse datoteke v imeniku */etc* v arhivsko datoteko *rezerva.tar*. Prvi argument ukaza *tar*, »cvf«, je »ukaz« *tar*-u. Izbira *c* naroči *tar*-u, da naj ustvari novo arhivno datoteko. Izbira *v* prisili *tar*, da uporablja gostobesedni način, izpisujoč vsako ime datoteke, ko se arhivira. Izbira »f« pove *tar*-u, da je naslednji argument *rezerva.tar* ime arhivne datoteke, ki naj se ustvari. Ostanek argumentov ukaza *tar* predstavljajo imena datotek in imenikov, ki naj se dodajo v arhiv.

Ukaz

```
# tar xvf rezerva.tar
```

bo odarhiviral arhivno datoteko *rezerva.tar* v trenutni imenik.

- ◇ Stare datoteke z enakim imenom so prepisane, ko se datoteke odtarajo v obstoječi imenik.

Preden odarhivirate datoteke v formatu za *tar*, je pomembno vedeti, kje bi morale biti datoteke odpakirane. Denimo, da arhivirate naslednje datoteke: */etc/hosts*, */etc/group* in */etc/passwd*. Če uporabite ukaz

```
# tar cvf rezerva.tar /etc/hosts /etc/group /etc/passwd
```

je ime imenika */etc/* dodano pred vsakim imenom datoteke. Za odtaranje datotek na pravilno mesto uporabite

```
# cd /
# tar xvf rezerva.tar
```

saj so datoteke odtarane z imenom poti, shranjenim v arhivni datoteki.

Vendar če arhivirate datoteke z ukazom

```
# cd /etc
# tar cvf rezerva.tar hosts group passwd
```

se ime imenika ne shrani v arhivno datoteko. Zato morate pred odtaranjem datotek narediti »cd /etc«. Kot lahko vidite, način izdelave datoteke *tar* določa veliko razliko, kje jo morate odtarati. Ukaz

```
# tar tvf rezerva.tar
```

se lahko uporablja za prikaz izpisa datotek v arhivu, ne da bi jih odtarali. Vidite lahko imenik, relativno na katerega so datoteke v arhivu shranjene, in odtarate arhiv na pravo mesto.

4.8.2 gzip in compress

Za razliko od arhivnih programov za MS-DOS, *tar* samodejno ne stiska datotek, ko jih arhivira. Če arhivirate dve enomegabajtni datoteki, bo končna datoteka *.tar* velika malo več kot dva megabajta. Ukaz *gzip* stisne podatke v datoteki (ki ni nujno datoteka *.tar*).

Ukaz

```
# gzip -9 rezerva.tar
```

stisne datoteko `rezerva.tar` v datoteko `rezerva.tar.gz`. Izbira `-9` pove `gzip`-u, da naj uporabi največji faktor stiskanja.

Ukaz `gunzip` se lahko uporablja za razširjanje `gzip`ane datoteke. Enako lahko uporabljate tudi `»gzip -d«`.

Pripomoček `gzip` je relativno novo orodje v skupnosti Unixa. Mnogo let se je namesto njega uporabljal ukaz `compress`. Vendar zaradi različnih dejavnikov, vključno s pričkanji o patentu za algoritem stiskanja podatkov programa `compress` in dejstva, da je `gzip` veliko bolj učinkovit, `compress` počasi izginja.

Datoteke, ki jih obdela `compress`, se končajo z `».Z«`. Datoteka `rezerva.tar.Z` je s `compress`om stisnjena različica datoteke `rezerva.tar`, medtem ko je `rezerva.tar.gz`, z `gzip`om stisnjena različica². Ukaz `uncompress` se uporablja za razširitev datoteke, zgoščene s `compress`-om. Je ekvivalenten ukazu `»compress -d«`. Tudi ukaz `gunzip` ve, kako mora obravnavati datoteke, stisnjene s `compress`-om.

4.8.3 Sestavljanje vsega skupaj

Za arhiviranje skupine datotek in stiskanje rezultata uporabite ukaza:

```
# tar cvf rezerva.tar /etc
# gzip -9 rezerva.tar
```

Rezultat je `rezerva.tar.gz`. Za odpakiranje te datoteke uporabite obratna ukaza:

```
# gunzip rezerva.tar.gz
# tar xvf rezerva.tar
```

Vedno se prepričajte, da ste v pravilnem imeniku, preden odpakirate datoteko `tar`.

Z nekaj spretnosti lahko vse to opravite tudi v eni vrstici.

```
# tar cvf - /etc | gzip -9c > rezerva.tar.gz
```

Tukaj pošiljamo datoteko `tar` na `»-«`, kar označuje standardni izhod iz programa `tar`. Ta se po cevi usmeri na standardni vhod za program `gzip`, ki stisne prihajajočo datoteko `tar`. Rezultat se shrani v datoteko `rezerva.tar.gz`. Izbira `-c` naroči `gzip`-u, naj pošlje svoj izhod na standardni izhod, ki ga preusmerimo v `rezerva.tar.gz`.

En sam ukaz za odpakiranje te arhivske datoteke bi bil:

```
# gunzip -c rezerva.tar.gz | tar xvf -
```

Spet, `gunzip` razširi vsebino datoteke `rezerva.tar.gz` in pošlje rezultat, datoteko `tar`, na standardni izhod. Ta se po cevi prenese do `tar`-a, ki prebere `»-«`, tokrat se ta znak nanaša na standardni vhod za `tar`.

Na srečo ukaz `tar` vključuje tudi izbiro `z` za samodejno sprotno zgoščevanje/razširitev datotek z uporabo algoritma zgoščevanja `gzip`.

Ukaz

```
# tar cvfz rezerva.tar.gz /etc
```

je ekvivalenten ukazoma

²Nekaj časa se je za `gzip`ane datoteke uporabljal podaljšek `.z` (mali `»z«`). Dogovorjeni podaljšek za `gzip` je zdaj `.gz`.

```
# tar cvf rezerva.tar /etc
# gzip rezerva.tar
```

Kot lahko uporabite tudi ukaz

```
# tar xvfz rezerva.tar.Z
```

namesto ukazov

```
# uncompress rezerva.tar.Z
# tar xvf rezerva.tar
```

Poglejte strani priročnika za tar in gzip za podrobnejša navodila.

4.9 Uporaba disket in izdelava varnostnih kopij

Diskete se pogosto uporabljajo kot medij za shranjevanje varnostnih kopij. Če na vašem sistemu nimate priključene tračne enote, lahko uporabljate diskete (čeprav so počasnejše in nekoliko manj zanesljive).

Kot je bilo omenjeno, morajo biti diskete formatirane z programom `FORMAT.COM` za MS-DOS ali programom `fdformat` za Linux. Ta zapiše informacije o sektorjih in sledih, ki so primerne, glede na zmogljivost diskete.

Nekaj imen naprav in formatov disket, ki so dostopne v Linuxu, je podanih v tabeli 4.4.

Gonilnik disketne naprave	Format
<code>/dev/fd0d360</code>	Dvojna gostota, 360 KB, 5,25 palcev.
<code>/dev/fd0h1200</code>	Visoka gostota, 1,2 MB, 5,25 palcev.
<code>/dev/fd0h1440</code>	Visoka gostota, 1,44 MB, 3,5 palcev.

Tabela 4.4: Formati disket za Linux

Naprave, ki se začenjajo s `fd0`, so prvi disketni pogon, ki se v MS-DOS-u imenuje `A:`. Imena datotek gonilnikov druge diskovne naprave se začenjajo s `fd1`. V splošnem jedro Linuxa lahko zazna format diskete, ki je že bila formatirana – lahko preprosto uporabite `/dev/fd0` in zaznavanje formata prepustite sistemu. A ko prvič uporabite popolnoma nove, neformatirane diskete, boste morda morali uporabiti določitev gonilnika, če sistem ne more zaznati tipa disket.

Popoln seznam naprav Linuxa z imeni gonilnikov teh naprav je podan v spisu *Linux Allocated Devices* H. Petra Anvina (glejte dodatek A).

Diskete lahko uporabljate tudi za hranjenje posameznih datotečnih sistemov in z `mount` priklopite disketo, da dostopate do podatkov na njej. Glejte razdelek 4.9.4.

4.9.1 Uporaba disket za varnostne kopije

Najpreprostejši način za izdelavo varnostne kopije z uporabo disket je s pripomočkom `tar`. Ukaz

```
# tar cvfzM /dev/fd0 /
```

bo naredil popolno varnostno kopijo vašega sistema z uporabo disketnega pogona `/dev/fd0`. Izbira »M« ukaza `tar` dovoljuje, da se varnostne kopije raztezajo prek več delov; se pravi, ko je ena disketa polna, bo `tar` prosil za naslednjo. Ukaz

```
# tar xvfzM /dev/fd0
```

obnovi celotno varnostno kopijo. Ta metoda se lahko uporablja tudi s tračno enoto, priključeno na vaš sistem. Glejte razdelek 4.9.3.

Obstajajo različni drugi programi za izdelavo večdelnih varnostnih kopij; morda bo priročen program `backflops`, ki ga najdete na `tsx-11.mit.edu`.

Izdelava popolne varnostne kopije sistema z disketami je lahko časovno in zmogljivostno potratna. Mnogi sistemski upravitelji uporabljajo politiko **diferenčnih varnostnih kopij** (angl. incremental backup). Vsak mesec se naredi popolno varnostno kopijo, vsak teden pa se naredi le varnostna kopija tistih datotek, ki so se spremenile v zadnjem tednu. Če sesujete sistem sredi meseca, lahko v tem primeru preprosto obnovite zadnjo popolno mesečno varnostno kopijo in potem po potrebi obnovite zadnje tedenske varnostne kopije.

Ukaz `find` je uporaben za iskanje datotek, ki so bile spremenjene po določenem datumu. Veliko skriptov za upravljanje diferenčnih varnostnih kopij lahko najdete na `metalab.unc.edu`.

4.9.2 Varnostne kopije s pogonom Zip

Izdelava varnostnih kopij na pogon Zip je podobna izdelavi varnostnih kopij na diskete, a ker imajo diski Zip največkrat kapaciteto 98 MB, je mogoče uporabiti en sam priklopljen disk Zip za en sam varnostni arhiv.

Pogoni Zip so dostopni s tremi različnimi strojnimi vmesniki: vmesnikom SCSI, vmesnikom IDE in vmesnikom vzporednih vrat PPA. Podpora za pogon Zip ni vključena kot že prevedena izbira v Linuxu, a se lahko določi ob gradnji prirejenega jedra za vaš sistem. Stran 172 opisuje namestitev gonilnika za napravo Iomega Zip.

Pogona Zip z vmesnikoma SCSI in PPA uporabljata vmesnik SCSI in sledita poimenovalnim dogovorom za druge naprave SCSI, ki so opisani na strani 155.

Diski Zip so navadno vnaprej formatirani z datotečnim sistemom za MS-DOS. Lahko uporabljate bodisi obstoječi datotečni sistem MS-DOS, ki mora biti podprt v vašem jedru Linuxa, bodisi `mke2fs` ali podoben program za zapis datotečnega sistema za Linux na disk.

Disk Zip, ko je priklopljen kot prva naprava SCSI, je `/dev/sda4`.

```
# mount /dev/sda4 /mnt
```

Pogosto je udobno priskrbeti dodatno točko priklopa za datotečne sisteme diska Zip; na primer `/zip`. Naslednji koraki, ki morajo biti izvedeni kot `root`, bodo ustvarili točko priklopa:

```
# mkdir /zip
# chmod 0755 /zip
```

Potem lahko uporabljate imenik `/zip` za priklop datotečnega sistema diska Zip.

Pisanje arhivov na diske Zip je podobno arhiviranju na diskete. Za arhiviranje in stiskanje imenika `/etc` na priklopljen pogon Zip bi bil uporabljen ukaz

```
# tar zcvf /zip/etc.tgz /etc
```

Ta ukaz se lahko izvede iz poljubnega imenika, saj določa absolutna imena poti. Arhivno ime `etc.tgz` je potrebno, če pogon Zip vsebuje datotečni sistem za MS-DOS, saj morajo tedaj imeti vse datoteke, zapisane na disk, imena, ki zadoščajo poimenovanju datotek tipa 8+3 v MS-DOS-u; sicer bo ime datoteke krajšano.

Podobno, izločanje tega arhiva zahteva ukaza

```
# cd /
# tar zxvf /zip/etc.tgz
```

Za ustvarjanje, na primer, datotečnega sistema ext2 na pogonu Zip bi podali ukaz (za nepriklopljeni disk Zip)

```
# mke2fs /dev/sda4
```

S pogonom Zip, priklopljenim na ta način, z datotečnim sistemom ext2, je mogoče napraviti varnostne kopije celotnih datotečnih sistemov z enim samim ukazom.

```
# tar zcvf /zip/local.tar.gz /usr/local
```

Upoštevajte, da je izdelava varnostnih kopij z ukazom tar v veliko primerih še vedno bolj zaželeno kot preprosta izdelava arhivskega izvoda z ukazom cp -a, saj tar ohranja čas spremembe izvornih datotek.

4.9.3 Izdelava varnostnih kopij na tračne enote

Arhiviranje na tračno napravo brez zaustavljanja je podobno izdelavi varnostnih kopij na disketni datotečni sistem, le z drugim gonilnikom naprave. Trakovi so tudi formatirani in obravnavani drugače kot diskete. Nekateri reprezentativni gonilniki tračnih enot za sisteme Linux so izpisani v tabeli 4.5.

Gonilnik tračne naprave	Format
/dev/rft0	Trak QIC-117, previjanje ob zaprtju.
/dev/nrft0	Trak QIC-117, brez previjanja ob zaprtju.
/dev/tpqic11	Trak QIC-11, previjanje ob zaprtju.
/dev/ntpqic11	Trak QIC-11, brez previjanja ob zaprtju.
/dev/qft0	Disketniška tračna enota, previjanje ob zaprtju.
/dev/nqft0	Disketniška tračna enota, brez previjanja ob zaprtju.

Tabela 4.5: Gonilniki tračnih enot

Disketniške tračne enote uporabljajo krmilnik disketnega pogona in jih nadzoruje gonilnik naprave ftape, ki je pokrit spodaj. Namestitev modula za gonilnik naprave ftape je opisana na strani 174. Tračne enote SCSI so našteje v tabeli 4.3.

Za arhiviranje imenika /etc na tračno enoto z uporabo ukaza tar uporabite ukaz

```
# tar cvf /dev/qft0 /etc
```

Podobno za izločitev datotek s traku uporabite ukaza

```
# cd /
# tar xvf /dev/qft0
```

Ti trakovi morajo biti pred uporabo formatirani, tako kot diskete. Gonilnik ftape lahko formatira trakove pod Linuxom. Za formatiranje traku formata QIC-40 uporabite ukaz

```
# ftformat --format-parameter qic40-205ft --mode-auto
--omit-erase --discard-header
```

Druge tračne enote imajo lastno programje za formatiranje. Preverite strojno dokumentacijo za tračno enoto ali dokumentacijo za gonilnik naprave za Linux, ki ji pripada.

Preden se trakovi lahko odstranijo iz enote, morajo biti previti in V/I vmesni pomnilniki zapisani na trak. To je analogno odklopu diskete, preden jo izvržemo, saj tračni gonilnik tudi predpomni podatke. Standardni ukaz Unixa za nadzor operacij tračne enote je `mt`. Vaš sistem morda ne bo ponujal tega ukaza, odvisno od tega, ali ima zmožnosti uporabe tračne enote. Gonilnik `ftape` ima podoben ukaz, `ftmt`, ki se uporablja za nadzor tračnih operacij.

Za previjanje traku, preden ga odstranite, uporabite ukaz

```
# ftmt -f /dev/qft0 rewoffl
```

Seveda nadomestite ustrezní gonilnik tračne enote za svoj sistem.

Priporočljivo je tudi ponovno napenjanje traku, preden pišete nanj, saj so magnetni trakovi občutljivi za natezanje. Ukaz se glasi

```
# ftmt -f /dev/qft0 retension
```

Za branje statusa tračne enote s formatiranim trakom v enoti, izdajte ukaz

```
# ftmt -f /dev/qft0 status
```

4.9.4 Uporaba disket kot datotečnih sistemov

Datotečni sistem na disketi lahko ustvarite, kot bi ga ustvarili na particiji trdega pogona. Na primer,

```
# mke2fs /dev/fd0 1440
```

ustvari datotečni sistem na disketi v `/dev/fd0`. Velikost datotečnega sistema mora ustrezati velikosti diskete. 3,5" diskete visoke gostote so velike 1,44 megabajtov ali 1440 blokov. 5,25" diskete visoke gostote so velike 1200 blokov. Če sistem ne more samodejno zaznati kapacitete diskete, je nujno določiti velikost datotečnega sistema v blokih.

Za dostopanje do diskete morate priklopiti datotečni sistem, ki je na njej. Ukaz

```
# mount /dev/fd0 /mnt
```

bo priklopil disketo v `/dev/fd0` na imenik `/mnt`. Zdaj se bodo vse datoteke na disketi pojavile pod imenikom `/mnt` na vašem pogonu.

Točka priklopa, imenik, kamor priklopite datotečni sistem, mora že obstajati, ko uporabljate ukaz `mount`. Če še ne obstaja, jo ustvarite z ukazom `mkdir`, kot je opisano na strani 167.

Glejte stran 154 za več informacij o datotečnih sistemih, priklopljanju in točkah priklopa.

- ◇ Upoštevajte, da se ves V/I za disketo predpomni enako kot V/I za trdi disk. Če spreminjate podatke na disketi, morda ne boste videli lučke na disketni enoti, dokler jedro ne izprazni svojega V/I vmesnega pomnilnika. Pomembno je, da ne odstranite diskete, dokler je ne odklopite z ukazom

```
# umount /dev/fd0
```

Disket ne smete preprosto zamenjati, kot bi to storili v sistemu MS-DOS. Vsakič, ko zamenjate disketo, uporabite `umount` za prvo disketo in `mount` za naslednjo.

4.10 Nadgradnja in namestitev novega programa

Še ena dolžnost sistemskega upravitelja je nadgradnja in nameščanje nove programske opreme.

Razvoj sistema Linux je hiter. Novo jedro izide vsakih nekaj tednov in drugo programje se ažurira skoraj tako pogosto. Zaradi tega novi uporabniki Linuxa pogosto čutijo potrebo po nenehni nadgradnji njihovih sistemov, da bi sledili hitremu tempu sprememb. To ni potrebno in je izguba časa. Če bi sledili vsem spremembam v svetu Linuxa, bi porabljali ves svoj čas za nadgrajevanje in vam ga ne bi ostalo nič za uporabo sistema.

Nekateri ljudje menijo, da bi morali nadgraditi, ko je narejena nova izdaja distribucije; na primer, ko Slackware izda novo različico. Mnogi uporabniki Linuxa ponovno v celoti namestijo svoje sisteme z vsako najnovejšo izdajo distribucije Slackware.

Najboljši način za nadgradnjo vašega sistema je odvisen od distribucije Linuxa, ki jo imate. Debian, S. u. S. E., Caldera in Red Hat Linux imajo vsi inteligentno programje za upravljanje paketov, ki dovoljuje preproste nadgradnje z nameščanjem novih paketov. Na primer, prevajalnik za C, gcc, se dobi kot vnaprej pripravljen binarni paket. Ko se namesti, so vse datoteke starejših različic prepisane ali odstranjene.

V največji meri brezumno nadgrajevanje, samo da »sledite trendu«, sploh ni pomembno. To ni MS-DOS ali Microsoft Windows. Ni bistvenega razloga za poganjanje novejših različic vsega programja. Če ugotovite, da bi želeli ali potrebovali zmožnosti, ki jih ponuja nova različica, potem nadgradite. Sicer ne nadgradite. Z drugimi besedami, nadgradite le to, kar morate in kadar morate. Ne nadgrajujte zaradi nadgrajevanja. To zapravlja veliko časa in naporov.

4.10.1 Nadgraditev jedra

Nadgraditev jedra je stvar dobivanja izvorne kodo jedra in prevajanja le-te. To je v splošnem neboleč postopek, a lahko zaidete v težave, če poskušate nadgraditi na razvojno jedro ali nadgraditi na novo različico jedra. Ime različice jedra ima dva dela, številko različico jedra in stopnjo popravkov. V času nastanka te knjige je bilo zadnje stabilno jedro različice 2.2.3. 2.2 je tukaj različica jedra in 3 je stopnja popravkov. Liho oštevilčene različice jeder kot 2.3 so razvojna jedra. Izognite se razvojnim jedrom, razen če želite živeti nevarno! Kot splošno pravilo bi morali biti sposobni preproste nadgradnje na drugo stopnjo popravkov, a nadgradnja na novo različico zahteva nadgradnjo sistemskih pripomočkov, ki tesno sodelujejo z jedrom.

Izvorno kodo jedra za Linux lahko poberete s katerega od mest za anonimni FTP Linuxa (glejte stran 249 za seznam). Na metalab.unc.edu, denimo, najdete izvorno kodo jedra v imeniku /pub/Linux/kernel, organizirano v podimenike po številki različice.

Izvorna koda jedra izhaja kot z gzip-om stisnjena datoteka .tar. Na primer, datoteka, ki vsebuje izvorno kodo jedra 2.0.33, je linux-2.0.33.tar.gz.

Izvorna koda jedra se odpakira v imenik /usr/src, tako da ustvari imenik /usr/src/linux. Splošna praksa je, da je /usr/src/linux simbolna povezava na drug imenik, ki vsebuje številko različice kot /usr/src/linux-2.0.33. Tako lahko namestite novo izvorno kodo jedra in jo preizkusite, preden odstranite staro izvorno kodo jedra. Ukazi za ustvarjanje povezave imenika z jedrom so:

```
# cd /usr/src
# mkdir linux-2.0.33
```



```
# rm -r linux
# ln -s linux-2.0.33 linux
# tar xzf linux-2.0.33.tar.gz
```

Ko nadgrajujete na novi nivo popravkov iste različice jedra, lahko datoteke s popravki jedra (angl. kernel patch files) prihranijo čas prenosa, saj je izvorna koda jedra velika okoli 7 MB, ko je že stisnjena z gzip-om. Za nadgradnjo z jedra 2.0.31 na jedro 2.0.33 bi pobrali datoteki s popravki patch-2.0.32.gz in patch-2.0.33.gz, ki ju najdete na istih mestih za FTP kot celotno izvorno kodo jedra. Ko ste položili popravka v imenik /usr/src, ju zaporedno uporabite na jedru, da boste osvežili izvorno kodo. Eden od načinov za to bi bil

```
# cd /usr/src
# gzip -cd patch-2.0.32.gz | patch -p0
# gzip -cd patch-2.0.33.gz | patch -p0
```

Ko je izvorna koda odpakirana in so bili uporabljeni popravki, se boste morali prepričati, da so tri simbolne povezave v /usr/include pravilne za vašo distribucijo jedra. Za ustvarjanje teh povezav uporabite ukaze

```
# cd /usr/include
# rm -rf asm linux scsi
# ln -s /usr/src/linux/include/asm-i386 asm
# ln -s /usr/src/linux/include/linux linux
# ln -s /usr/src/linux/include/scsi scsi
```

Ko ustvarite te povezave, ni razloga, da bi jih ponovno ustvarjali, ko nameščate nove popravke jedra ali novejšo različico jedra. (Glejte razdelek 3.11 za več o simbolnih povezavah.)

Za prevajanje jedra morate imeti na vašem sistemu nameščen prevajalnik za C gcc. Za prevajanje jedra 2.0 je potrebna različica gcc 2.6.3 ali poznejša.

Najprej naredite cd v /usr/src/linux. Ukaz make config vas sprašuje o številnih nastavitvenih izbirah. To je korak, kjer izberete strojno opremo, ki jo bo podpiralo vaše jedro. Največja napaka, ki se ji je treba izogniti, je, da ne vključite podpore za vaš krmilnik trdega diska. Brez pravilne podpore za trdi disk v jedru se sistem ne bo niti zagnal. Če niste prepričani, kaj pomeni izbira jedra, je kratek opis dostopen s pritiskom na in .

V novejših različicah Linuxa sta dostopna tudi zaslonsko usmerjena ekvivalenta ukaza make config. Za zaslonsko različico nastavitvenih izbir za jedro napišite make menuconfig, če pa imate na vašem sistemu nameščen grafični vmesnik X Window System, lahko uporabite tudi make xconfig. Uporaba obeh je precej intuitivna in gotovo ju boste uporabljali raje kot make config.

Nato poženite ukaz make dep za ažuriranje vseh odvisnosti izvorne kode. To je pomemben korak. Ukaz make clean odstrani stare binarne datoteke iz drevesa izvorne kode jedra.

Ukaz make zImage prevede vaše jedro in ga zapiše v datoteko /usr/src/linux/arch/i386/boot/zImage. Jedra za Linux na sistemih Intel so vedno stisnjena. Včasih je jedro, ki ga želite prevesti, preveliko, da bi bilo lahko stisnjeno po metodi, ki jo uporablja make zImage. Jedro, ki je preveliko, bo izšlo iz prevajanja jedra s sporočilom o napaki: Kernel Image Too Large. Če se to zgodi, poskusite ukaz make bzImage. Ta uporablja program za stiskanje, ki podpira tudi večja jedra. Jedro se zapiše v datoteko /usr/src/linux/arch/i386/boot/bzImage.

Ko ste enkrat prevedli jedro, ga morate prekopirati na zagonsko disketo (z ukazom, kot »cp zImage /dev/fd0«) ali pa namestiti sliko tako, da bo LILO lahko zaganjal sistem z vašega trdega pogona. Glejte stran 149 za več informacij.

4.10.2 Dodajanje gonilnika naprave v jedro

Stran 167 opisuje, kako uporabljati pogon Iomega Zip za izdelavo rezerv. Podpora za pogon Iomega Zip, kot za mnoge druge naprave, ni splošno prevedena v zalogo distribucijskih jeder Linuxa – različnost naprav je preprosto prevelika, da bi podpirali vse od njih v uporabnem jedru. Vendar pa je izvorna koda za gonilnik naprave Zip na vzporednih vratih vključena kot del distribucije izvirne kode jedra. Ta razdelek opisuje, kako dodati podporo za pogon Iomega Zip na vzporednih vratih in doseči sobivanje s tiskalnikom, priključenim na druga vzporedna vrata.

Pogoj je, da ste namestili in uspešno zgradili prilagojeno jedro za Linux, kot je opisano v prejšnjem razdelku.

Izbor naprave ppa za pogon Zip kot možnost za jedro zahteva, da odgovorite z Y na primerna vprašanja med korakom make config, ko določate sestavo prilagojenega jedra. Posebej pri napravi ppa morate odgovoriti z »Y« na tri izbire:

```
SCSI support? [Y/n/m] Y
SCSI disk support? [Y/n/m] Y
IOMEGA Parallel Port Zip Drive SCSI support? [Y/n/m] Y
```

Ko ste uspešno pognali make config z vsemi podpornimi izbirami, ki jih želite vključiti v jedro, pognali make dep, make clean in make zImage za gradnjo jedra, morate povedati jedru, kako naj namesti gonilnik. To se naredi z ukazno vrstico v zagonskem nalagalniku LILO. Kot je bilo opisano v razdelku 4.2.1, ima nastavitvena datoteka za LILO /etc/lilo.conf »kitice« za vsak operacijski sistem, ki ga pozna, in tudi navodila za predstavljanje teh izbir uporabniku v času zagona.

Drugo navodilo, ki ga LILO prepozna, je »append=«, ki dovoljuje dodajanje informacij, ki jih potrebujejo različni gonilniki naprav, ob času zagona v ukazno vrstico. V tem primeru gonilnik ppa za Iomega Zip potrebuje neuporabljeno prekinitve (angl. interrupt) in naslov vrat za V/I (angl. I/O port address). To je analogno določanju različnih tiskalniških naprav, kot sta v MS-DOS-u LPT1: in LPT2:.

Na primer, če vaš tiskalnik uporablja šestnajstiški (z osnovo 16) naslov vrat 0x378 (glejte namestitveni priročnik za vašo kartico vzporednih vrat, če ne veste, kakšen je naslov) in je izprašan (angl. polled, se pravi, ne potrebuje linije linije IRQ, pogosta nastavitvev Linuxa), bi postavili naslednjo vrstico v vašo sistemsko datoteko /etc/lilo.conf:

```
append="lp=0x378,0"
```

Velja omeniti, da Linux ob zagonu samodejno prepozna ena vrata /dev/lp, a ob določitvi prirejene nastavitve vrat so potrebna navodila v času zagona.

Število »0« za naslovom vrat pove jedru, naj *ne* uporablja linije IRQ (zahteva prekinitve, angl. interrupt request) za tiskalnik. To je v splošnem sprejemljivo, saj so tiskalniki veliko počasnejši kot procesorji, zato počasnejša metoda dostopa do V/I naprav, znana kot **izpraševanje** (angl. polling), kjer jedro periodično na lastno pest preverja status tiskalnika, še vedno dovoljuje računalniku, da dohaja tiskalnik.

Vendar naprave, ki delujejo na višjih hitrostih, kot so serijske linije in diski, potrebujejo linijo **IRQ** ali linijo **za zahtevo prekinitve**. To je strojni signal, ki ga naprava pošlje

procesorju, kadar naprava potrebuje njegovo pozornost; na primer, če ima naprava podatke, ki čakajo za vhod v procesor. Procesor preneha delati to, kar počne, in obdela zahtevo za prekinitev, ki jo je izdala naprava. Naprava ppa za pogon Zip potrebuje prosto prekinitve, ki mora ustrezati prekinitvi, ki je nastavljena na tiskalniški kartici, na katero povežete pogon Zip. V času tega pisanja gonilnik naprave ppa za Linux ne podpira »verženja« (angl. chaining) naprav na vzporednih vratih, in za napravo ppa za Zip in tiskalnik se morajo uporabljati ločena vrata.

Za ugotovitev, katere prekinitve se že uporabljajo na vašem sistemu, ukaz

```
# cat /proc/interrupt
```

prikaže seznam naprav in linije IRQ, ki jih uporabljajo. Vendar morate biti previdni, da ne uporabljate nobene od samodejno nastavljenih prekinitve serijskih vrat, ki morda niso našteje v datoteki `/proc/interrupt`.

Spis *Serial HOWTO* (iz projekta LDP), dostopen iz virov, naštetih v dodatku A, podrobno opisuje nastavitve serijskih vrat.

- ◇ Preveriti bi morali tudi strojne nastavitve različnih vmesnih kartic na vašem stroju – z odpiranjem ohišja stroja in ogledom nastavitvev skakačev, če je potrebno – da zagotovite, da ne privzemate linije IRQ, ki jo že uporablja druga naprava. Več naprav, ki se bojujejo za prekinitveno linijo, je morda najpomembnejši razlog za nedelujoče sisteme Linux.

Tipična datoteka `/proc/interrupt` je videti nekako takole:

```
0: 6091646 timer
1: 40691 keyboard
2: 0 cascade
4: 284686 + serial
13: 1 math error
14: 192560 + ide0
```

Tukaj je zanimiv prvi stolpec. To so številke linij IRQ, ki se uporabljajo na sistemu. Za gonilnik ppa želimo izbrati linijo, ki *ni* našteja. Pogosto je dobra izbira IRQ 7, saj je redko uporabljan v privzetih sistemskih nastavitvah. Določiti moramo tudi naslov vrat, ki jih bo uporabljala naprava ppa. Ta naslov mora biti fizično nastavljen za kartico vmesnika. Vzpostojnim V/I vratom so dodeljeni posebni naslovi, zato boste morali prebrati dokumentacijo za vašo kartico vzpostojnih vrat. V tem primeru bomo uporabljali V/I naslov vrat 0x278, ki ustreza tiskalniškim vratom LPT2: pod MS-DOS-om. Podporo za druga vzpostojna vrata in prekinitveno linijo vklopimo z naslednjo vrstico v primerni kitici datoteke `/etc/lilo.conf`:

```
append="lp=0x378,0 ppa=0x278,7"
```

Ti stavki so pripeti začetnim parametrom jedra v času zagona. Zagotavljajo, da se katerikoli tiskalnik, priključen na sistem, ne vtika v delovanje pogona Zip. Seveda, če vaš sistem nima nameščenega tiskalnika, se navodilo »lp=« lahko izpusti in bi se tudi moralo.

Ko namestite prirejeno jedro, kot je bilo opisano v razdelku 4.2.1, in preden ponovno zaženete sistem, se prepričajte, da ste že pognali ukaz

```
# /sbin/lilo
```

za namestitev nove nastavitve LILO na zagnoski sektor trdega pogona.

4.10.3 Namestitev modula za gonilnik naprave

Stran 168 opisuje, kako narediti rezervne kopije datotek na tračno enoto. Linux ponuja podporo za različne tračne enote z vmesniki IDE, SCSI in nekaterimi lastniškimi vmesniki. Drug pogosti tip tračnih enot se priključi neposredno na krmilnik disketnega pogona. Linux priskrbi gonilnik naprave ftape kot modul.

V času tega pisanja je najnovejša različica gonilnika ftape 3.04d. Paket lahko pridobite z arhiva za FTP metalab.unc.edu (glejte dodatek B za navodila). Arhivno datoteko za ftape najdete v imeniku `/pub/Linux/kernel/tapes`. Prepričajte se, da boste vzeli najnovejšo različico. V času tega pisanja je to `ftape-3.04d.tar.gz`.

Po odpakiranju arhiva ftape v imenik `/usr/src` bo ukaz `make install` v najvišjem imeniku ftape prevedel module gonilnika in pripomočke za ftape, če je potrebno, in jih namestil. Če izkušate združljivostne težave z izvedljivimi distribucijskimi datotekami za ftape in vašim sistemskim jedrom ali knjižnicami, bo izvajanje ukazov `make clean` in `make install` zagotovilo, da so moduli prevedeni na vašem sistemu.

- ◇ Za uporabo te različice gonilnika ftape morate imeti v jedro prevedeno podporo modulom (angl. module support) kot tudi podporo za jedrni strežnik (angl. kernel daemon) `kerneld`. Vendar *ne* smete vključiti zgrajene kode jedra za ftape kot izbiro jedra, saj novejši modul ftape popolnoma nadomesti to kodo.

Ukaz `make install` tudi namesti module gonilnikov naprav v ustreznih imenikih. Na standardnih sistemih Linux so moduli locirani v imeniku

```
/lib/modules/različica-jedra
```

Če je vaša različica jedra 2.0.30, so moduli na vašem sistemu v imeniku `/lib/modules/2.0.30`. Korak `make install` tudi zagotavlja, da se ti moduli lahko najdejo, saj doda primerne stavke v datoteko `modules.dep`, ki je locirana v najvišjem imeniku datotek za module, v tem primeru `/lib/modules/2.0.30`. Namestitev za ftape dodaja naslednje module na vaš sistem (v tem primeru uporabljamo jedro različice 2.0.30):

```
/lib/modules/2.0.30/misc/ftape.o
/lib/modules/2.0.30/misc/zft-compressor.o
/lib/modules/2.0.30/misc/zftape.o
```

V sistemsko nastavitveno datoteko za module morajo biti dodana tudi navodila za nalaganje modulov. Ta datoteka se na mnogih sistemih imenuje `/etc/conf.modules`. Za samodejno nalaganje modulov za ftape na zahtevo dodajte naslednji vrstici v datoteko `/etc/conf.modules`:

```
alias char-major-27 zftape
pre-install ftape /sbin/swapout 5
```

Prvi stavek naloži vse module, povezane s ftape, če je to potrebno, ko jedro dostopa do naprave z velikim številom (angl. major number) 27 (naprava ftape). Ker podpora za modul `zftape` (ki priskrbi samodejno stiskanje podatkov za tračne enote) potrebuje podporo drugih modulov za ftape, so vsi naloženi ob zahtevi jedra. Druga vrstica določa nalagalne parametre za module. V tem primeru zagotavlja pripomoček `/sbin/swapout`, ki je poleg paketa ftape, da je dostopnega dovolj pomnilnika DMA za delovanje gonilnika ftape.

Za dostop do naprave ftape morate najprej v enoto vstaviti formatirani trak. Navodila za formatiranje trakov in delovanje tračne enote so podana v razdelku 4.9.3.

4.10.4 Nadgradnja knjižnic

Kot smo že omenili, je večina programja na sistemu prevedena za uporabo deljenih knjižnic, ki vsebujejo skupne podprograme, deljene med različnimi programi.

Če vidite sporočilo o nezdružljivi različici knjižnice

```
Incompatible library version
```

ko poskušate pognati program, morate knjižnice nadgraditi na različice, ki jih potrebuje program. Knjižnice so za nazaj združljive. Program, preveden za uporabo starejše različice knjižnic, bi moral delovati tudi z nameščeno novo različico knjižnic. Obratno ni res.

Novejše različice knjižnic lahko najdemo na mestih za FTP datotek za Linux. Na metalab.unc.edu so zbrane v imeniku /pub/Linux/GCC. Datoteke o »izdaji« (angl. release), ki jih najdete tam, bi morale pojasniti, katere datoteke morate vzeti in kako jih namestiti. Na kratko, morali bi vzeti datoteki `image-različica.tar.gz` in `inc-različica.tar.gz`, kjer je *različica* različica knjižnic za namestitev, kot je 4.4.1. To sta z gzip-om stisnjeni datoteki za tar. Datoteka `image` vsebuje slike knjižnic za namestitev v imenika `/lib` in `/usr/lib`. Datoteka `inc` vsebuje glave za namestitev v imenik `/usr/include`.

Datoteka `release-različica.tar.gz` bi morala podrobno pojasniti namestitveni postopek (natančna navodila se razlikujejo z vsako izdajo). V splošnem morate namestiti knjižnične datoteke `.a` in `.so` v imenik `/usr/lib`. To so knjižnice, ki se uporabljajo ob prevajanju.

Dodatno, datoteke `libc.so.različica` s slikami deljenih knjižnic se namestijo v imenik `/lib`. To so slike deljenih knjižnic, ki se naložijo pri poganjanju programov, ki uporabljajo knjižnice. Vsaka knjižnica ima simbolno povezavo in veliko številko različice knjižnice v `/lib`.

Knjižnica `libc` različice 4.4.1 ima veliko številko različice enako 4. Datoteka, ki vsebuje knjižnico, je `libc.so.4.4.1`. V imenik `/lib` je postavljena tudi simbolna povezava `libc.so.4`, ki kaže na knjižnico. Ko nadgrajujete knjižnice, morate spremeniti to simbolno povezavo. Na primer, ko nadgrajujete iz `libc.so.4.4` na `libc.so.4.4.1`, morate spremeniti simbolno povezavo, da kaže na novo različico.

- ◇ Simbolno povezavo morate spremeniti v enem koraku, kot je opisano spodaj. Če pobrišete simbolno povezavo `libc.so.4`, bodo programi, ki so odvisni od te povezave (vključno z osnovnimi pripomočki kot `ls` in `cat`), nehali delovati. Uporabite naslednji ukaz za posodobljenje simbolne povezave `libc.so.4`, da bo kazala na datoteko `libc.so.4.4.1`:

```
# ln -sf /lib/libc.so.4.4.1 /lib/libc.so.4
```

V enakem smislu morate spremeniti tudi simbolno povezavo `libm.so.različica`. Če nadgrajujete na drugačno različico knjižnic, zgoraj vstavite primerna imena datotek. Zapiski ob izdaji knjižnice bi morali pojasnjevati podrobnosti. (Glejte stran 129 za več informacij o simbolnih povezavah.)

4.10.5 Nadgradnja prevajalnika gcc

Prevajalnik `gcc` za C in C++ se uporablja za prevajanje programja na vašem sistemu, najpomembneje za prevajanje jedra. Najnovejša različica prevajalnika `gcc` se nahaja na mestih za FTP Linuxa. Na metalab.unc.edu je locirana v imeniku /pub/Linux/GCC (skupaj s knjižnicami). Tam mora biti tudi datoteka `release` za distribucijo `gcc`, ki podrobno opisuje, katere datoteke morate prenesti in kako jih namestiti. Večina distribucij (RedHat,

Debian itd.) ima različice za nadgradnjo, ki delujejo z njihovim programjem za upravljanje programskih paketov. V splošnem so ti paketi veliko preprostejši za namestitev kot »splošne« distribucije.

4.10.6 Nadgradnja drugega programja

Nadgradnja drugega programja je pogosto le stvar nabiranja ustreznih datotek in njihovega nameščanja. Večina programja za Linux se razširja kot stisnjene datoteke `.tar`, ki vključujejo izvorno kodo, binarne datoteke ali oboje. Če binarne datoteke niso vključene v izdajo, jih boste morali prevesti sami. To pomeni, da boste morali vsaj vpisati make v imeniku z izvorno kodo.

Najpreprostejši način za obveščanje o novih izdajah programja je branje novičarske skupine Useneta `comp.os.linux.announce`. Ko iščete programje na mestu za FTP, je jemanje indeksne datoteke `ls-lR` z mesta za FTP in uporaba pripomočka `grep` za iskanje datotek, ki jih želite, najpreprostejši način za iskanje programja. Če vam je dostopen `archie`, vam je tudi ta lahko v pomoč³. Obstajajo tudi drugi viri Interneta, ki so posvečeni posebej Linuxu. Glejte dodatek A za več podrobnosti.

4.11 Različna opravila

Verjeli ali ne, obstajajo številna oskrbniška opravila za sistemskega upravljalca, ki ne spadajo v nobeno od večjih kategorij.

4.11.1 Sistemske začetne datoteke

Ko se sistem zažene, samodejno izvede številne skripte, še preden se prijavi kak uporabnik. Tukaj je opis, kaj se dogaja.

V času zagona jedro porodi proces `/etc/init`. `init` je program, ki prebere svojo nastavitveno datoteko `/etc/inittab` in rodi druge procese glede na vsebino te datoteke. Eden od pomembnih procesov, ki se zaženejo iz `inittab`, je proces `/etc/getty`, ki se začne na vsaki navidezni konzoli (angl. virtual console, VC). Proces `getty` vzame navidezno konzolo, ki jo uporablja, in začne na njej prijavi proces z ukazom `login`. To vam dovoljuje, da se prijavite na vsako navidezno konzolo. Če `/etc/inittab` ne vsebuje procesa `getty` za določeno navidezno konzolo, se na to navidezno konzolo ne boste mogli prijaviti.

Še en proces, ki se izvede iz `/etc/inittab`, je `/etc/rc`, glavna sistemska inicializacijska datoteka. Ta datoteka je preprost skript ukazne lupine, ki izvaja vse inicializacijske ukaze, potrebne v času zagona, kot so priklop datotečnih sistemov (glejte stran 154) in priprava izmenjalnega prostora. Na nekaterih sistemih `init` izvaja datoteko `/etc/init.d/rc`.

Vaš sistem lahko izvaja tudi druge inicializacijske skripte, na primer `/etc/rc.local`, ki navadno vsebuje pripravljalne ukaze, specifične za vaš lasten sistem, kot je nastavitev gostiteljskega imena (glejte naslednji razdelek). Skript `rc.local` se lahko zažene iz `/etc/rc` ali neposredno iz `/etc/inittab`.

³Če nimate programa `archie`, se lahko priključite s telnet na strežnik za `archie`, kot je `archie.rutgers.edu`, se prijavite kot »archie« in uporabite ukaz »help«.

4.11.2 Nastavitev gostiteljskega imena

V omrežnem okolju se gostiteljsko ime (angl. host name) uporablja za enoznačno določitev določenega stroja, medtem ko pri samostojnem stroju gostiteljsko ime preprosto daje sistemu osebnost in šarm. To je kot poimenovanje domače živali: vedno lahko rečete svojemu psu »Pes«, a je veliko bolj zanimivo, da poimenujete svojega psa z imenom, kot je Piko ali Runo.

Nastavitev gostiteljskega imena vašega sistema zadeva le uporabo ukaza `hostname`. Če ste na omrežju, bo vaše gostiteljsko ime polno gostiteljsko ime vašega stroja, kot je `goober.norelco.com`. Če niste na omrežju nobene vrste, lahko izberete poljubno ime gostitelja in domeno, kot `loomer.vpizza.com`, `shoop.nowhere.edu` ali `floof.org`.

Gostiteljsko ime se mora pojaviti v datoteki `/etc/hosts`, ki priredi naslove za IP vsakemu gostitelju. Tudi, če vaš stroj ni na omrežju, bi morali vključiti vaše lastno gostiteljsko ime v datoteko `/etc/hosts`. Če niste na omrežju TCP/IP in je vaše gostiteljsko ime `floof.org`, preprosto vključite naslednjo vrstico v `/etc/hosts`:

```
127.0.0.1      floof.org localhost
```

To dodeli vaše gostiteljsko ime `floof.org` zaprtozračnemu naslovu (angl. loopback address) `127.0.0.1`. Povratni vmesnik je prisoten, če je stroj priključen na omrežje ali pa ne. Vzdevku `localhost` se vedno priredi ta naslov.

Če ste na omrežju TCP/IP, bi v datoteki `/etc/hosts` morala biti vaš naslov za IP in gostiteljsko ime. Na primer, če je vaše gostiteljsko ime `goober.norelco.com` in vaš naslov za IP `128.253.154.32`, dodajte naslednjo vrstico v `/etc/hosts`:

```
128.253.154.32    goober.norelco.com
```

Za nastavitev vašega gostiteljskega imena preprosto uporabite ukaz `hostname`. Na primer, ukaz

```
# hostname -S goober.norelco.com
```

nastavi gostiteljsko ime na `goober.norelco.com`. V večini primerov se ukaz `hostname` izvede iz ene od sistemskih zagonskih datotek, kot sta `/etc/rc` ali `/etc/rc.local`. Uredite ti dve datoteki in spremenite ukaz `hostname`, ki ga najdete tam, da bo odražal vaše lastno gostiteljsko ime. Po ponovnem zagonu bo sistem uporabljal novo ime.

4.12 Dejanja v sili

V nekaterih primerih se bo sistemski upravitelj soočil s problemom okrevanja od popolne katastrofe, kot je pozaba korenskega gesla ali sesutje datotečnih sistemov. Najboljši nasvet je, *brez panike*. Vsakdo dela neumne napake – to je najboljši način za učenje sistemskega upravljanja: *trnova pot*.

Linux ni nestabilna različica Unixa. Pravzaprav sem imel manj problemov z obešanjem sistema kot pri komercialnih različicah Unixa na veliko platformah.

Prvi korak samostojne odprave problema je ugotovitev, kje problem sploh leži. Brskajte naokoli in opazujte, kako stvari delujejo. Velikokrat sistemski upravitelj obupano prosi za pomoč, še preden sploh ve, v čem je problem. Opazili boste, da je pravzaprav zelo lahko samostojno odpraviti težave. To je pot razsvetlitve in gurujstva.

Obstaja le nekaj primerov, ko je nujna ponovna namestitev sistema od samega začetka. Mnogo novih uporabnikov po nesreči pobriše kakšno nujno potrebno sistemsko datoteko

in takoj posežejo po namestitvenih disketah. To ni dobra ideja. Preden uporabite tako drastične ukrepe, raziščite problem in poprosite druge za pomoč. V veliko primerih lahko obnovite vaš sistem z vzdrževalne diskete.

4.12.1 Okrevanje z vzdrževalne diskete

Eno od nepogrešljivih orodij sistemskega upravitelja je tako imenovana »zagon-ska/korenska disketa« (angl. boot/root floppy disk), disketa, s katere lahko zaženete popoln sistem Linux, neodvisno od vašega trdega pogona. Zagonske/korenske diskete so pravzaprav zelo preproste – na disketi naredite korenski datotečni sistem, prepisete nanj vse potrebne pripomočke, namestite LILO in zagonsko jedro na disketo. Druga tehnika je uporaba ene diskete za jedro in druge za korenski datotečni sistem. V vsakem primeru, rezultat je enak: sistem Linux poganjate popolnoma iz disketne enote.

Kanonični primer zagonske/korenske diskete sta zagonski disketi za Slackware. Ti disketi vsebujeta zagonsko jedro in korenski datotečni sistem, vse na disketi. Njun namen je namestitev distribucije Slackware, a prideta prav tudi ob vzdrževanju sistema.

Drug primer vzdrževalne diskete je zagonska/korenska disketa H. J. Luja, dostopna iz imenika `/pub/Linux/GCC/rootdisk` na `metalab.unc.edu`. Če ste častihlepni, lahko naredite svojo lastno. V večini primerov pa so že narejene zagonske/korenske diskete veliko preprostejše za uporabo in bodo verjetno tudi popolnejše.

Uporaba zagonske/korenske diskete je zelo preprosta. Zaženite disketo na vašem sistemu in se prijavite kot `root` (navadno brez gesla). Za dostop do datotek na trdem pogonu boste morali ročno priklopiti datotečne sisteme. Na primer, ukaz

```
# mount -t ext2 /dev/hda2 /mnt
```

bo priklopil datotečni sistem `ext2fs` na `/dev/hda2` pod `/mnt`. Ne pozabite, da je zdaj imenik `/` sama zagonska/korenska disketa; datotečne sisteme trdega pogona morate priklopiti pod kak imenik, če želite dostopati do datotek. Torej je `/etc/passwd` na vašem trdem pogonu zdaj `/mnt/etc/passwd`, če ste priklopili vaš običajni korenski datotečni sistem na `/mnt`.

4.12.2 Poprava korenskega gesla

Presenetljivo – če pozabite vaše korensko geslo, to sploh ni problem. Zaženite zagonsko/korensko disketo, priklopite korenski datotečni sistem na `/mnt` in izpraznite polje za geslo uporabnika `root` v datoteki `/mnt/etc/passwd`, kot v tem primeru:

```
root::0:0:root:/:/bin/sh
```

Zdaj `root` nima gesla. Ko boste ponovno zagnali sistem s trdega pogona, bi morali biti sposobni prijave kot `root` in ponovne nastavitve gesla z uporabo ukaza `passwd`.

Kaj niste veseli, da ste se naučili uporabe urejevalnika `vi`? Verjetno na vaši zagonski/korenski disketi ne bodo dostopni urejevalniki, kot je Emacs, toda `vi` bi moral biti.

4.12.3 Sesuti datotečni sistemi

Če nekako sesujete datotečne sisteme, lahko poženete pripomoček `e2fsck` ali ustrezno obliko pripomočka `fsck` za vaš tip datotečnega sistema. (Glejte stran 156.) V večini primerov je najvarneje popraviti vse poškodovane podatke na trdem disku z diskete.

Pogost vzrok okvare datotečnega sistema je poškodovan superblok. **Superblok** (angl. super block) je »glava« datotečnega sistema, ki vsebuje informacije o njegovem statusu, velikosti, prostih blokih in tako naprej. Če poškodujete superblok (na primer, z nenamernim pisanjem podatkov neposredno v particijsko tabelo datotečnega sistema), sistem verjetno sploh ne bo prepoznal datotečnega sistema. Poskus priklopa datotečnega sistema bo neuspešen in `e2fsck` ne bo mogel odpraviti problema.

Na srečo datotečni sistem `ext2fs` shranjuje kopije superbloka na mejah »skupin blokov« na pogonu, navadno na vsakih 8 K blokov. Pripomočku `e2fsck` lahko naročite, naj uporabi kopijo superbloka z ukazom, kot je

```
# e2fsck -b 8193 particija
```

kjer je *particija* particija, na kateri počiva datotečni sistem. Izbira `-b 8193` pove `e2fsck`, naj uporablja kopijo superbloka, shranjeno na bloku številka 8193 v datotečnem sistemu.

4.12.4 Obnova izgubljenih datotek

Če pomotoma pobrišete pomembno datoteko na vašem sistemu, ni nobenega načina, da jo »povrnete«. Včasih pa lahko prepisete ustrezne datoteke z diskete na vaš trdi pogon. Če, na primer, zbrisate `/bin/login`, ki vam omogoča prijavo, preprosto zaženite sistem z zagonke/korenske diskete, priklopite korenski datotečni sistem na `/mnt` in uporabite ukaz

```
# cp -a /bin/login /mnt/bin/login
```

Izbira `-a` naroči `cp`, da ohrani dovoljenja datotek, ki jih prepisuje.

Seveda, če datoteke, ki ste jih zbrisali, niso potrebne sistemske datoteke, ki imajo ustrezne na zagonski/korenski disketi, potem nimate sreče. Če redno delate varnostne kopije, jih lahko obnovite iz njih.

4.12.5 Sesute knjižnice

Če pomotoma sesujete vaše knjižnice ali simbolne povezave v imeniku `/lib`, ukazi, ki so odvisni od teh knjižnic, zagotovo ne bodo več delovali (glejte stran 175). Najpreprostejša rešitev je, da zaženete sistem z vaše zagonke/korenske diskete, priklopite vaš korenski datotečni sistem in popravite knjižnice v imeniku `/mnt/lib`. Stran 175 opisuje, kako namestiti sprotne knjižnice (angl. run time libraries) in njihove simbolne povezave.

