



**P A M A T I**

**Alan Hicks  
Chris Lumens  
David Cantrell  
Logan Johnson**

**Tulkojums: Alvils Bērziņš  
Rīga, 2006**

Autortiesības © 1998, 1999, 2000, 2001, 2002, 2003, 2004, 2005 Slackware Linux, Inc.

*Slackware Linux* ir Patrika Volkerdinga un Slackware Linux, Inc. reģistrēts zīmols.

*Linux* ir Linus Torvalda reģistrēts zīmols

Visi citi grāmatā minētie zīmoli pieder/ir reģistrēti to attiecīgajiem īpašniekiem.

## Saturs

<b>Tulkotāja priekšvārds.....</b>	<b>8</b>
<b>Priekšvārds.....</b>	<b>10</b>
Mērķauditorija.....	10
Īss satura pārskats.....	10
Šajā grāmatā lietotie apzīmējumi.....	11
Tipogrāfiskie apzīmējumi.....	11
Lietotāja ievadītā informācija.....	11
Piemēri.....	12
Atzinības.....	12
<b>1.nodaļa. Ievads Slackware Linux.....</b>	<b>13</b>
1.1. Kas ir Linux?.....	13
Daži vārdi par GNU.....	13
1.2. Kas ir Slackware?.....	13
1.3. Atvērtais pirmkods un brīvā programmatūra.....	14
<b>2.nodaļa. Kur saņemt palīdzību?.....</b>	<b>16</b>
2.1. Sistēmas palīdzība.....	16
man.....	16
/usr/doc direktorija.....	17
HOWTO un mini-HOWTO dokumentācija.....	17
2.2. Tiešsaistes palīdzība.....	18
Oficiālā mājaslapa un palīdzības forumi.....	18
E-pasta atbalsts.....	18
Slackware Linux projekta e-pasta vēstkopas.....	18
Neoficiālās mājaslapas un palīdzības forumi.....	19
<b>3.nodaļa. Instalēšana.....</b>	<b>20</b>
3.1. Apgādājamies ar Slackware.....	20
Oficiālie diski un iepakojuma komplekts.....	20
Lejupielāde Internet tīklā.....	20
3.2. Sistēmas prasības.....	21
Pakotņu sērijas.....	21
Instalēšanas metodes.....	22
Ielādes diski.....	23
Saknes diski.....	23
Papilddisks.....	23
Disku izveide.....	23
3.3. Cietā diska sadaļu izveide.....	24
3.4. Uzstādīšanas programma setup.....	25
HELP (palīdzība).....	26
KEYMAP (tastatūras izkārtojums).....	27
ADDSWAP (maiņvietas aktivizēšana).....	28
TARGET (mērķa sadaļas).....	29
SOURCE (instalēšanas datu nesējs).....	29
SELECT (pakotņu izvēle).....	30
INSTALL (instalēšana).....	31
CONFIGURE (konfigurācija).....	33
Modema konfigurācija.....	34
Laika zona.....	35
Peles konfigurācija.....	36
Datora pulkstenis.....	37
Šrifti (fonti).....	38
LILO.....	39
Tīkls.....	39

X logu pārvaldnieks.....	40
<b>4.nodaļa. Sistēmas konfigurēšana.....</b>	<b>41</b>
4.1. Sistēmas apskats.....	41
Failsistēmas izkārtojums.....	41
Failu meklēšana.....	42
/etc/rc.d direktorija.....	43
Sistēmas startēšana.....	44
Darbīmeņu inicializācijas skripti.....	44
4.2. Kodola izvēle.....	45
/kernels direktorija Slackware CD-ROM diskā.....	46
Kodola kompilēšana no pirmkoda.....	46
Kodola moduļu izmantošana.....	48
<b>5.nodaļa. Tīkla konfigurēšana.....</b>	<b>49</b>
5.1. Ievads: netconfig ir jūsu draugs.....	49
5.2. Tīkla aparatūras konfigurēšana.....	49
Kodola moduļu ielāde.....	49
LAN (10/100/1000Base-T un Base-2) kartes.....	50
Modemi.....	50
PCMCIA.....	50
5.3. TCP/IP konfigurēšana.....	50
DHCP.....	51
Statiska IP adrese.....	52
/etc/rc.d/rc.inet1.conf.....	52
/etc/resolv.conf.....	52
/etc/hosts.....	53
5.4. PPP.....	53
pppsetup.....	53
5.5. Bezvadu tīkls.....	54
Aparatūras atbalsts.....	54
Kartes konfigurēšana.....	54
Tīkla konfigurēšana.....	55
5.6. Tīkla failsistēmas.....	55
SMB/Samba/CIFS.....	56
Network File System (NFS).....	57
<b>6.nodaļa. X logu sistēma.....</b>	<b>59</b>
6.1. xorgconfig.....	59
6.2. xorgsetup.....	63
6.3. xinitrc.....	63
6.4. xwmconfig.....	64
6.5. xdm.....	66
<b>7.nodaļa. Sistēmas ielāde.....</b>	<b>68</b>
7.1. LILO.....	68
Root.....	69
Floppy.....	69
MBR.....	69
7.2. LOADLIN.....	70
7.3. Duālā ielāde.....	71
Windows.....	71
Linux.....	74
<b>8.nodaļa. Čaula.....</b>	<b>75</b>
8.1. Lietotāji.....	75
Autorizēšanās.....	75
root: superlietotājs.....	75
8.2. Komandrinda.....	76
Programmu darbināšana.....	76
Aizstājējzīmju salīdzināšana.....	76

8.3. Bourne Again Shell (bash).....	77
Vides mainīgie.....	77
Rindiņas aizpilde ar Tab.....	78
8.4. Virtuālie termināļi.....	79
screen.....	79
<b>9.nodaļa. Failsistēmas struktūra.....</b>	<b>81</b>
9.1. Īpašumtiesības.....	81
9.2. Tiesības.....	81
9.3. Saites.....	83
9.4. Iekārtu montēšana.....	84
fstab.....	84
mount un umount.....	85
9.5. NFS montēšana.....	85
<b>10.nodaļa. Darbs ar failiem un direktorijām.....</b>	<b>86</b>
10.1. Navigācija: ls, cd un pwd.....	86
ls.....	86
cd.....	87
pwd.....	87
10.2. Peidžeri: more, less un most.....	88
more.....	88
less.....	88
most.....	88
10.3. Vienkārša izvade: cat un echo.....	88
cat.....	88
echo.....	89
10.4. Radīšana: touch un mkdir.....	89
touch.....	89
mkdir.....	89
10.5. Kopēšana un pārvietošana.....	90
cp.....	90
mv.....	90
10.6. Dzēšana: rm un rmdir.....	90
rm.....	90
rmdir.....	91
10.7. Failu aizstājvārdu izveide ar ln.....	91
<b>11.nodaļa. Procesu vadība.....</b>	<b>92</b>
11.1. Procesu darbināšana fonā.....	92
11.2. Procesu darbināšana priekšplānā.....	92
11.3. ps.....	93
11.4. kill.....	95
11.5. top.....	96
<b>12.nodaļa. Sistēmas administrēšanas pamati.....</b>	<b>99</b>
12.1. Lietotāji un grupas.....	99
Iekļautie rīki.....	99
Paroļu maiņa.....	101
Lietotāja informācijas maiņa.....	102
12.2. Lietotāji un grupas - grūtais ceļš.....	102
12.3. Sistēmas sagatavošana izslēgšanai.....	104
<b>13.nodaļa. Svarīgākās tīkla komandas.....</b>	<b>106</b>
13.1. ping.....	106
13.2. traceroute.....	106
13.3. DNS rīki.....	106
host.....	107
nslookup.....	107
dig.....	107
13.4. finger.....	108

13.5. telnet.....	109
Cits telnet lietojums.....	109
13.6. Drošā čaula - SSH.....	110
13.7. E-pasts.....	110
pine.....	110
elm.....	111
mutt.....	112
nail.....	113
13.8. Pārlūki.....	114
lynx.....	114
links.....	114
wget.....	115
13.9. FTP klienti.....	116
ftp.....	116
ncftp.....	117
13.10. Tērzēšana ar citiem.....	118
wall.....	118
talk.....	118
ytalk.....	119
<b>14.nodaļa. Drošība.....</b>	<b>121</b>
14.1. Servisu aizliegšana.....	121
Ar inetd startētie servisi.....	121
Ar inicializācijas skriptiem startētie servisi.....	121
14.2. Resursdatora piekļuves kontrole.....	122
iptables.....	122
tcpwrappers.....	123
14.3. Jauninājumu uzturēšana.....	123
Vēstkopa slackware-security.....	123
Direktorija /patches.....	124
<b>15.nodaļa. Failu arhivēšana.....</b>	<b>125</b>
15.1. gzip.....	125
15.2. bzip2.....	125
15.3. tar.....	125
15.4. zip.....	127
<b>16.nodaļa. vi.....</b>	<b>128</b>
16.1. vi startēšana.....	128
16.2. Režīmi.....	129
Komandu režīms.....	129
Ievietošanas režīms.....	130
16.3. Failu atvēršana.....	130
16.4. Failu saglabāšana.....	131
16.5. vi aizvēršana.....	131
16.6. vi konfigurēšana.....	131
16.7. vi komandas.....	132
Kursora pārvietošana.....	132
Labošana.....	132
Meklēšana.....	133
Saglabāšana un aizvēršana.....	133
<b>17.nodaļa. Emacs.....</b>	<b>134</b>
17.1. Emacs startēšana.....	134
Komandas.....	135
17.2. Buferi.....	135
17.3. Režīmi.....	136
Failu atvēršana.....	136
17.4. Rediģēšanas pamati.....	137
Svarīgākās Emacs rediģēšanas komandas.....	137

17.5. Failu saglabāšana.....	138
17.6. Emacs aizvēršana.....	138
<b>18.nodaļa. Slackware pakotņu vadība.....</b>	<b>139</b>
18.1. Pakotņu formāta apskats.....	139
18.2. Pakotņu rīki.....	139
pkgtool.....	140
installpkg.....	141
removepkg.....	141
upgradepkg.....	142
rpm2tgz rpm2targz.....	142
18.3. Pakotņu izveide.....	142
explodepkg.....	143
makepkg.....	143
SlackBuild skripti.....	143
18.4. Birkas un birku failu izveide.....	143
<b>19.nodaļa. ZipSlack.....</b>	<b>145</b>
19.1. Kas ir ZipSlack?.....	145
Priekšrocības.....	145
Trūkumi.....	145
19.2. Apgādājamies ar ZipSlack.....	145
Instalēšana.....	145
19.3. ZipSlack ielāde.....	146
<b>Pielikums A. GNU vispārējā publiskā licence.....</b>	<b>147</b>

## Tulkotāja priekšvārds

Deviņdesmito gadu otrajā pusē ievēroju kādu tendenci - gandrīz visi man zināmie datorentuziasti, sistēmadministratori, programmētāji un vienkāršie lietotāji regulāri veltīja sulīgus lamuvārdus firmai *Microsoft* un tās operētājsistēmai *Microsoft Windows*, vienlaikus turpinot izmantot (bieži vien pirātiskas) *Windows* kopijas. Tas likās nedaudz dīvaini, tomēr alternatīvas jau it kā nebija.

1998.gada nogalē kādā žurnālā izlasīju par operētājsistēmu *Linux*, kura esot stabila, brīva no vīrusiem un, pats galvenais - bezmaksas. Daudz nedomājot, devos uz tirgu "Latgale" (tautā saukts par *Latgalīti*) un iegādājos CD-ROM disku ar lepnu uzrakstu "*RedHat Linux 5.2*". Devos mājās un ķēros pie sava datora cietā diska (tolaik 1,7 GiB) daļišanas, pēc kā sāku *Linux* instalēšanu. Liels bija mans pārsteigums, ieraugot teksta režīma instalācijas programmu un vēl lielāks tas kļuva, kad RedHat instalēšanas programma vēlējās, lai es tai pastāstu, kāds ir manas videokartes *RAMDAC* ātrums. ;) Pēc vairāk vai mazāk veiksmīgas instalēšanas ieraudzīju uzrakstu

```
localhost login:
```

Tā kā piekļuves Internet tīklam man tolaik nebija, nācās palikt bešā, jo nezināju, kā autorizēties sistēmā... :( Kad 1999.gada sākumā tiku pie iezvanpieejas *Internet* pieslēguma, noskaidroju, ka lietotājsvārds ir `root`. Kopumā mani šī sistēma neapmierināja (galu galā, tas bija 1998.gads, kad *Linux* vēl tiešām nebija izmantojama darbstacijās) un ar to viss pagaidām beidzās.

Nākamā saskarsme ar *Linux* bija 2000.gadā, kad iegādājos RedHat *Linux 6.2*. Šajā distributīvā bija pieejama, šķiet, *KDE1* vide, pēc nelieliem pūliņiem pat izdevās instalēt Tildes (jā, jā!) izlaisto *Linux* latviešu valodas pakotni. Izdevās iedarbināt un konfigurēt gandrīz visu datora aparatūru, izņemot *Winmodem* iekārtu.

2001.gadā uzstādīju vairākus *Linux* failserverus. Pārsvarā tie tika darbināti morāli novecojušos datoros ar procesoru 80486. Pamazām parādījās izpratne par *Linux*, tā failsistēmu un uzbūvi, un šī sistēma vairs nešķīta grūta un nepieejama.

Kopš 2003.gada *Linux* izmantoju kā primāro darbstācijas operētājsistēmu - un nesūdzos. Arī šīs grāmatas tulkojums ir rakstīts datorā ar *Linux*. Darbstacijām izmantoju *Mandriva Linux*, jo tas ir viegli instalējams un konfigurējams (lai arī nav tik pielāgojams, kā *Slackware*) un to droši var instalēt arī draugiem un paziņām, nebaidoties, ka tiem radīsies sevišķas problēmas ar sistēmas konfigurēšanu. 2003.gada vasarā nolēmu izmēģināt *Slackware Linux*. Un tad sākās...

Kopš tā laika esmu instalējis ap 70 *Linux* serverus, un gandrīz visos darbojas *Slackware Linux*. Serveru funkcionalitāte ir visdažādākā - sākot ar Internet starpniekserveriem un uguns sienām, turpinot ar tīmekļa vietņu serveriem un e-pasta serveriem, un beidzot ar sarežģītām produkcijas sistēmām, kuras savienotas ar *IBM FastT900* disku masīviem, izmantojot divus neatkarīgus automātiski rezervējamus optiskos pieslēgumus. *Slackware* vienmēr ir izcēlusies ar lielisku konfigurējamību un, pats galvenais - tajā nav nekā lieka - lieliska serveru operētājsistēmas īpašība, jo sistēmai, manuprāt, jāsaturs tikai to funkcionalitāti, kas nepieciešama "īsto" programmu/informācijas sistēmu darbināšanai. Arī darbstacijās *Slackware* sevi ir parādījusi kā pielāgojamu un universālu sistēmu (protams, atšķirībā no *Mandriva Linux*, ir jāzina, ko vēlas panākt un kā to panākt).

Esmu mēģinājis par šo sistēmu ieinteresēt arī daudzus savus draugus. Daļai *Slackware* ir iepatīcies, tomēr daudziem traucē valodas barjera. Jā, tā tas diemžēl ir - lai ne vien lasītu, bet arī izprastu tehnisko literatūru, ir nepieciešamas labas angļu (vai citas) valodas zināšanas. Latviešu valodā diemžēl šāda veida literatūra ir tulkota un rakstīta maz un nepilnīgi - parasti tas tā, jo iespējamai grāmatai būtu pārāk mazs tirgus. Arī ar brīvām licencēm pieejamām grāmatām (kā šai) tulkojumu ir maz un tie ir ļoti nepilnīgi - protams, daudziem, kas uzņemas tulkošanas darbu, ir maz laika; citi labāk pasēž krogā pie alus kausa un palamā *Microsoft*, citas tautas (piemēram, krievus) vai kāmīšus un jūtas lieli patrioti. Lai kaut nedaudz labotu šo samilzušo problēmu - tehniskās literatūras trūkumu latviešu valodā (pamēģiniet kādā grāmatnīcā palūgt šādu literatūru - jums piedāvās plānas burtničas "Word ikvienam", "Excel ikvienam" vai ko tādu) - un vienlaikus popularizētu man mīļo *Slackware Linux* operētājsistēmu, nolēmu pārtulkot šo grāmatu - *Slackware Linux bībeli*. Pie tam nevis vienkārši iztulkot dažas sadaļas, bet tulkot no sākuma līdz beigām.



Šo priekšvārdu rakstu kā pēdējo - grāmatu šobrīd jau esmu iztulkojis, tādēļ lūdzu mīlēt un žēlot. ;) Tulkojuma apjoms ir liels un tajā noteikti ir ielavījušās kļūdas. Gan pozitīvo, gan negatīvo kritiku lūdzu sūtīt man uz e-pastu [alvilzb@parks.lv](mailto:alvilzb@parks.lv). Nekonstruktīvu kritiku ("šitā nav labi, raksti savādāk", "kāpēc tā, vajag tā un tā", "vajag vairāk par to un šito") neizskatīšu, toties būšu pateicīgs par norādēm uz kļūdām un tulkojuma neprecizitātēm.

Un vēl - pateicības. Vispirms vēlos pateikties saviem vecākiem, ka viņi man bērnībā un agrā jaunībā (pirms 12 gadiem) neliedza to prieku - iestāties datorpulciņā. Bez tā es noteikti šobrīd nebūtu *Linux* lietotājs, administrators un programmētājs. Vēlos pateikties arī savai draudzenei, kura izprata manu vēlmi iztulkot šo grāmatu un (pārāk) neprotēja, kad pie portatīvā datora sēdēju reizēm pat līdz 2-3 naktīm. Paldies arī maniem draugiem (īpaši Matīsam Sīlim) par morālo atbalstu. Protams - arī biedrībai "Latvietis" (<http://www.latvietis.com>). Nebūtu šīs organizācijas - nebūtu arī daudzu paziņu, kuri iedvesmoja mani ķerties pie šī darba. Mūsu - latviešu - ir pārāk maz, lai mēs dalītos labajos un sliktajos. Ja katrs darīs savu darbu, mēs kopā spēsim nodrošināt latviešu tautas izdzīvošanu. Viens no šādiem darbiem ir latviešu valodas klātbūtnes nodrošināšana informācijas tehnoloģijās. Un beidzot - paldies visiem tiem, kam neesmu pateicies :)

Šo un arī citus manis rakstītos/tulkotos materiālus iespējams lejuplādēt manā mājaslapā - <http://home.parks.lv/alvilsb>. Šīs grāmatas licence norādīta pielikumā A - lūdzu to ievērot. Ja kāds vēlas iegūt rediģējamu šīs grāmatas tekstu (*OpenDocument* formātā) - rakstiet man: [alvilzb@parks.lv](mailto:alvilzb@parks.lv).

Ak jā - grāmatas pielikumā A publicētās GNU Vispārējās publiskās licences teksta tulkojumu ņēmu no Latvijas Atvērtā koda asociācijas mājaslapas - <http://www.laka.lv>. Paldies arī šai organizācijai par ieguldīto darbu *Linux* un brīvās programmatūras popularizēšanā Latvijā.

## Priekšvārds

### Mērķauditorija

*Slackware Linux* operētājsistēma ir jaudīga darba platforma datoriem, kas veidoti uz *Intel* procesoru bāzes. Tā ir veidota kā stabila, droša un funkcionāla vide gan jau morāli novecojušiem datoriem, gan lieljaudas serveriem, gan jaudīgām darbstacijām.

Šī grāmata ir ieskats *Slackware Linux* operētājsistēmā, parādot šīs sistēmas iespējas un sniedzot pamatzināšanas, kas ļaus lasītājam uzsākt darbu ar *Slackware Linux*.

Cerams, ka, kļūstot par pieredzējušu *Slackware Linux* lietotāju, šī grāmata lasītājam kļūs par noderīgu rokasgrāmatu, kā arī noderēs, lai palīdzētu saviem draugiem, kad tie taujās par lielisko *Slackware Linux* operētājsistēmu, kuru lasītājs izmantos.

Lai arī šī grāmata nav gluži aizraujošs romāns, mēs centāties to padarīt pēc iespējas interesantāku, lai veiksmes gadījumā to varētu izmantot kā scenāriju kādai Holivudas filmai. :) Tomēr galvenais - ceram, ka lasītājs gūs noderīgu pieredzi un zināšanas, izlasījis mūsu grāmatu.

Un tā - izrāde sākas...

### Īss satura pārskats

#### 1.nodaļa. Ievads Slackware Linux

Ievadmateriāls par *Linux*, *Slackware*, kā arī Atvērtā pirmkoda un Brīvās programmatūras kustībām.

#### 2.nodaļa. Kur saņemt palīdzību?

Informācija par palīdzības resursiem - gan tiem, kas pieejami pašā *Slackware Linux* operētājsistēmā, gan *Internet* tīklā.

#### 3.nodaļa. Sistēmas instalēšana

Soli pa solim aprakstīts sistēmas instalācijas process ar ekrānattēliem.

#### 4.nodaļa. Sistēmas konfigurēšana

Informācija par svarīgākajiem konfigurācijas failiem, kā arī par *Linux* kodola pārkompilēšanu.

#### 5.nodaļa. Tīkla konfigurēšana

Kā pieslēgt *Slackware Linux* datoru tīklam? Apraksts par TCP/IP, PPP/īezvanpieejas, bezvadu tīkla u.c. konfigurēšanu.

#### 6.nodaļa. X logu sistēma

Apraksts par grafiskās vides - X logu sistēmas - konfigurēšanu *Slackware Linux* vidē.

#### 7.nodaļa. Sistēmas ielāde

Detalizēti aprakstīts *Slackware Linux* ielādes process, kā sistēmas duālās ielādes iespēja kopā ar *Microsoft Windows*.

#### 8.nodaļa. Čaula

Jaudīgās *Linux* komandrindas saskarnes apraksts.

#### 9.nodaļa. Failsistēmas struktūra

*Slackware Linux* failsistēmas struktūra - arī informācija par failu lietošanas tiesībām, īpašniekiem un saitēm.

#### 10.nodaļa. Darbs ar failiem un direktorijām

Komandas, kuras izmanto, lai veiktu dažādas manipulācijas ar failiem un direktoriņām no komandrindas saskarnes.

### **11.nodaļa. Procesu vadība**

Jaudīgās *Linux* procesu vadības komandas, kuras vada daudzas vienlaikus strādājošās programmas.

### **12.nodaļa. Sistēmas administrēšanas pamati**

Svarīgākās sistēmas administrēšanas darbības - lietotāju pievienošana un dzēšana, sistēmas korekta sagatavošana izslēgšanai u.c.

### **13.nodaļa. Svarīgākās tīkla komandas**

*Slackware Linux* iekļauto tīkla klientu programmu kolekcijas apraksts.

### **14.nodaļa. Drošība**

Apraksti dažādiem rīkiem, kuri ir pieejami, lai palīdzētu uzturēt *Slackware Linux* sistēmu drošu - *iptables*, *tcpwrappers* u.c.

### **15.nodaļa. Failu arhivēšana**

Dažādu *Linux* vidē pieejamu failu arhivēšanas un kompresēšanas rīku apskats.

### **16.nodaļa. vi**

Jaudīgais tekstu redaktors *vi*.

### **17.nodaļa. Emacs**

Jaudīgais tekstu redaktors *Emacs*.

### **18.nodaļa. Slackware pakotņu vadība**

*Slackware Linux* pakotņu vadības rīki, kā arī jaunu pakotņu un t.s. birku failu radīšanas process.

### **19.nodaļa. ZipSlack**

*ZipSlack* - *Slackware Linux* versija, kuru iespējams izmantot *Windows* vidē bez instalēšanas.

Pielikums A. GNU vispārējā publiskā licence.

## ***Šajā grāmatā lietotie apzīmējumi***

Lai nodrošinātu vienotu un viegli lasāmu teksta stilu, šajā grāmatā tiek lietoti vairāki apzīmējumi.

### **Tipogrāfiskie apzīmējumi**

*Kursīvs*

*Kursīvs* tiek izmantots komandu, izcēlumu un pirmo reizi minētu tehnisko terminu izcelšanai.

**Vienplatuma burti**

Ar **vienplatuma burtiem** tiek apzīmēti kļūdu paziņojumi, komandas, vides mainīgie, porti, hostu nosaukumi, lietotājevārdi, grupu nosaukumi, iekārtu nosaukumi, mainīgie un programmu pirmkoda fragmenti.

**Treknraksts**

Treknraksts izmantots piemēros lietotāja ievadītās informācijas apzīmēšanai.

### **Lietotāja ievadītā informācija**

Tastatūras taustiņu apzīmējumi norādīti **treknrakstā**, lai tie izceltos no pārējā teksta. Taustiņu

kombinācijas, kuras jāievada, vienlaicīgi nospiežot vairākus taustiņus, tiek apzīmētas ar + simbolu starp taustiņu apzīmējumiem. Piemēram,

### **Ctrl+Alt+Del**

nozīmē, ka vienlaicīgi jānospiež taustiņus **Ctrl**, **Alt** un **Del**.

Taustiņus, kurus jāievada secīgi, apzīmē ar komatu starp taustiņu vai to kombināciju apzīmējumiem. Piemēram,

### **Ctrl+X, Ctrl+S**

nozīmē, ka vispirms vienlaicīgi jānospiež taustiņus **Ctrl** un **X**, un pēc tam - **Ctrl** un **S**.

## **Piemēri**

Piemēri, kuri sākas ar **E:\>**, apzīmē *MS-DOS* komandu. Ja vien nav īpaši norādīts, šīs komandas iespējams izpildīt "Command Prompt" logā mūsdienu *Microsoft Windows* operētājsistēmās.

```
E:\> rawrite a: bare.i
```

Piemēri, kuri sākas ar #, apzīmē komandu, kuru jāizpilda *Slackware Linux* vidē kā superlietotājam. Lai to veiktu, iespējams pieslēgties sistēmai kā **root** lietotājam, vai pieslēgties ar parasta lietotāja tiesībām un izmantot **su (1)**, lai piekļūtu superlietotāja tiesībām.

```
# dd if=bare.i of=/dev/fd0
```

Piemēri, kuri sākas ar %, apzīmē komandu, kuru iespējams izpildīt ar parasta lietotāja tiesībām. Ja vien nav īpaši norādīts, vides mainīgo uzstādīšanai un citu komandu izpildei tiek izmantota *C-shell* sintakse.

## **Atzinības**

Šis projekts ir daudzu cilvēku mēnešiem ilga darba apkopojums. Es nebūtu varējis radīt šo darbu vakuumā. Par nesavtīgo ieguldījumu vēlos pateikties daudziem: Keith Keller par viņa bezvadu tīkla aprakstu, Joost Kremers par lielo darbu, ar vienu roku rakstot **emacs** sadaļu, Simon Williams par drošības sadaļu, Jurger Phillipaerts par galvenajām tīkla komandām, Cibao Cu Ali G Colibri par iedvesmu. Neskaitāmi citi ir sūtījuši ieteikumus un labojumus. Ļoti nepilnīgs šo cilvēku saraksts: Jacob Anhoej, John Yast, Sally Welch, Morgan Landry un Charlie Law. Es vēlētos arī pateikties Keith Keller par šī projekta e-pasta vēstkopas uzturēšanu, kā arī Carl Inglis par sākotnējo mājaslapas uzturēšanu. Visbeidzot gribu pateikties Patrikam Volkerdingam par *Slackware Linux* un David Cantrell, Logan Johnson un Chris Lumens par grāmatas *Slackware Linux pamati* pirmo redakciju. Bez viņu sākotnējā darba šis projekts neeksistētu. Ļoti daudzi, kas vairāk vai mazāk ir devuši savu ieguldījumu šajā projektā, diemžēl nav šeit pieminēti. Ceru, ka viņi man piedos par manu vājo atmiņu.

**Alan Hicks, 2005.gada maijs**

## 1.nodaļa. Ievads Slackware Linux

### 1.1. Kas ir Linux?

1991.gadā somu students Linus Torvalds uzsāka savu personīgo projektu *Linux* - operētājsistēmas kodolu, jo viņš vēlējās savā personālajā datorā darbināt *Unix* tipa operētājsistēmu, bet nevēlējās iztērēt būtisku naudas summu. Papildus tam viņš vēlējās sīki izprast tolaik moderno *Intel 80386* procesoru. *Linux* tika publicēts kā bezmaksas programmatūra, lai jebkurš varētu to izpētīt un uzlabot, izmantojot GNU Vispārējo publisko licenci (sk. sadaļu 1.3 un Pielikumu A). Mūsdienās *Linux* ir kļuvis par vērā ņemamu operētājsistēmu tirgus pārstāvi. Tā ir pieejama daudzām sistēmu arhitektūrām - *HP/Compaq Alpha*, *Sun SPARC* un *UltraSPARC*, kā arī *Motorola PowerPC*. Simtiem, ja ne tūkstošiem programmētāju visā pasaulē izstrādā *Linux*. Šis kodols spēj darbināt tādas programmas, kā *Sendmail*, *Apache* un *BIND*, kas ir populārākā *Internet* serveru programmatūra. Ir svarīgi atcerēties, ka termins *Linux* patiesībā apzīmē tikai sistēmas kodolu. Šis kodols ir operētājsistēmas sastāvdaļa, kas vada datora procesoru, atmiņu, cietos diskus un perifērijas iekārtas. Patiesībā tas ir arī viss, ko dara *Linux* - tas vada datora darbu un seko līdž tam, lai visas programmas "uzvestos" pieklājīgi. Dažādi uzņēmumi un privātpersonas gatavo pakotnes, kuras ietver kodolu un dažādas programmas, kas kopā veido operētājsistēmu. Katru šādu pakotni sauc par *Linux* distribūciju.

### Daži vārdi par GNU

*Linux* kodols sākās kā Linus Torvalds soloprojekts, tomēr, kā reiz teicis Īzaks Ņūtons: "Ja es esmu redzējis tālāk kā citi, tad tādēļ, ka stāvēju uz milžu pleciem". Kad Linus Torvalds uzsāka savu projektu, Brīvās programmatūras fonds (Free Software Foundation) jau bija radījuši ideju par kopīgas izstrādes programmatūru. Savu sasniegumu viņi nosauca par *GNU* - rekursīvu akronīmu, kas nozīmē "*GNU's Not Unix*" jeb "*GNU nav Unix*". *Linux* kodols tika izstrādāts ar GNU programmatūru, un GNU programmatūra jau kopš *Linux* pirmsākumiem darbojās ar šo kodolu. Arī mūsdienās daudzas *GNU* sastāvdaļas, no *gcc* līdz *gnutar*, joprojām veido gandrīz jebkura *Linux* distributīva pamatu. Šī iemesla dēļ daudzi Brīvās programmatūras fonda atbalstītāji uzskata, ka viņu darbam jādod vismaz tikpat atzinības, kā *Linux* kodolam, un iesaka, ka visus *Linux* distributīvus patiesībā jāsauc par *GNU/Linux* distributīviem.

Šī tēma ir bijusi daudzu asu diskusiju avots, un par to plašāks ir bijis tikai senais "*vi pret emacs*" *svētais karš*. Šis grāmatas mērķis ir nevis uzkurināt naida ugunis šajā karstajā diskusijā, bet *Linux* jaunpieņacējiem apskaidrot terminoloģiju. Redzot terminu *GNU/Linux*, ir skaidrs, ka domāts ir *Linux* distributīvs. Redzot terminu *Linux*, var būt domāts gan *Linux* distributīvs, gan tikai kodols - sarežģīti... Bieži vien terminu *GNU/Linux* nelieto tā sarežģītības dēļ.

### 1.2. Kas ir Slackware?

*Slackware Linux* projekts, kuru 1992.gada beigās uzsāka Patriks Volkerdings, un kuru pirmo reizi publicēja 1993.gada 17.jūlijā, kļuva par pirmo populāro *Linux* distributīvu. Volkerdings pirmo reizi iepazinās ar *Linux*, kad viņam kāda projekta realizēšanai bija nepieciešams lēts *LISP* valodas interpretators. Tolaik viens no dažiem pieejamajiem distributīviem bija *Soft Landing Systems* izstrādātais *SLS Linux*. Volkerdings nolēma izmantot *SLS Linux*, vienlaicīgi labojot tajā atrastās kļūdas. Vēlāk viņš nolēma apvienot visus izstrādātos kļūdu labojumus savā distributīvā, kuru varētu izmantot viņš un viņa draugi. Šis privātais distributīvs ātri kļuva populārs, tādēļ Volkerdings nolēma to nosaukt par *Slackware* un padarīt to publiski pieejamu. Pie reizes Patriks pievienoja *Slackware* jaunas īpašības - uz izvēlnēm balsītu lietotājam draudzīgu instalācijas programmu, kā arī pakotņu vadības koncepciju, kas ļāva lietotājiem vienkārši un ērti pievienot, dzēst un atjaunināt programmatūras pakotnes savās sistēmās.

Ir daudz iemeslu, kādēļ *Slackware Linux* ir vecākais joprojām dzīvais *Linux* distributīvs. *Slackware Linux* nav mēģinājums emulēt *Windows* vidi - tas ir pēc iespējas tuvs *Unix* tipa sistēmām. Tas nav

mēģinājums aizsegt sistēmas procesus ar skaistām, grafiskām konfigurācijas programmām. Tā vietā pilna kontrole tiek dota lietotājam, ļaujot tam precīzi redzēt un noteikt, kas notiek sistēmā. *Slackware* izstrāde netiek steidzināta ar termiņiem - katra jauna versija tiek publicēta tad, kad tā ir pabeigta.

*Slackware* ir paredzēts cilvēkiem, kuriem patīk mācīties un likt savai sistēmai darīt tieši to, ko viņi vēlas. *Slackware* stabilitāte un vienkāršība ir iemesls, kādēļ to turpinās lietot vēl daudzus gadus. *Slackware* šobrīd ir pazīstama kā nopietna serveru un darbstaciju platforma. Ir iespējams atrast *Slackware* darbstacijas, kuras darbina gandrīz jebkura paveida logu vadības sistēmu vai darbvirsma vidi, vai pat iztiek bez tās. *Slackware* serverus izmanto uzņēmumi viskritiskāko sistēmu darbināšanai. *Slackware* lietotāji ir starp visapmierinātākajiem *Linux* lietotājiem. Protams, ka mēs to apgalvojam :^)

### **1.3. Atvērtais pirmkods un brīvā programmatūra**

*Linux* komūnā darbojas divas galvenās ideoloģiskās kustības. Brīvās programmatūras kustība (nedaudz vēlāk aplūkosim to tuvāk) cīnās par visas programmatūras atbrīvošanu un intelektuālā īpašuma ierobežojumiem, piemēram, programmatūras patentiem. Šīs kustības atbalstītāji uzskata, ka šie ierobežojumi traucē programmatūras attīstībai un darbojas pret kopējo sabiedrības labumu. Atvērtā pirmkoda kustība - darbojas, lai sasniegtu līdzīgus mērķus, tomēr pievēršas tiem pragmatiskāk. Šīs kustības piekritēji kā argumentus izmanto nevis morāles un ētikas principus, bet gan ekonomiskos un tehniskos ieguvumus, kas rodas, padarot programmatūras pirmkodu brīvi pieejamu.

Otrā spektra galā ir grupas, kuras vēlas paturēt kontroli pār savu programmatūru.

Brīvās programmatūras kustību vada Brīvās programmatūras fonds - *GNU* projekta ietvaros izveidota organizācija. Brīvā programmatūra ir vairāk kā ideoloģija. Bieži vien tiek lietots teiciens "brīva kā runa, nevis par brīvu kā alus". Patiesībā brīvā programmatūra ir mēģinājums garantēt zināmas tiesības gan lietotājiem, gan izstrādātājiem. Šīs brīvības ietver tiesības izmantot programmatūru jebkuram mērķim, pētīt un modificēt programmas pirmkodu, kā arī izplatīt jebkuras veiktās modifikācijas. Lai garantētu šīs brīvības, ir radīta *GNU* Vispārējā publiskā licence (*GNU* General Public License, *GPL*). *GPL* būtībā nosaka, ka jebkuram, kurš izplata kompilētu programmu, kas licencēta ar *GPL*, ir jānodrošina programmatūras saņēmēju arī ar programmatūras pirmkodu, un jebkuram ir tiesības mainīt šo programmatūru, kamēr vien visas pirmkoda izmaiņas arī tiek publicētas. Tas garantē, ka programmu, kura vienreiz tiek "atvērta" sabiedrībai, vairs nav iespējams "aizvērt", izņemot gadījumu, kad to pieļauj visu programmatūras fragmentu (arī modifikāciju) visi autori. Lielākā daļa *Linux* programmu ir licencētas ar *GPL*.

Svarīgi, ka *GPL* nekas nav teikts par programmatūras cenu. Jocīgi, bet par brīvo programmatūru ir atļauts pieprasīt un saņemt naudu. Tas, ka programmatūra ir "brīva", nozīmē, to, ka, saņemot šo programmatūru, lietotājs saņem arī visas *GPL* noteiktās brīvības. Tas nenozīmē, ka programmatūrai obligāti jābūt "par brīvu". Tomēr, ja kāds jums ir pārdevis vai pat iedevis programmatūru, kas licencēta ar *GPL*, viņam obligāti jānodrošina jūs ar tās pirmkodu.

Vēl kāda populāra licence ir *BSD* licence. Atšķirībā no *GPL*, *BSD* licence neprasa, lai programmatūras saņēmējs tiktu nodrošināts ar programmatūras pirmkodu. Programmatūru, kas publicēta ar *BSD* licenci, ir iespējams izplatīt gan pirmkoda, gan kompilētā (binārā) veidā, ja vien tiek ievēroti daži noteikumi. Piemēram, oriģinālās programmatūras autora vārdu nedrīkst izmantot programmatūras reklamēšanai. Tāpat programmatūras autors nav atbildīgs ne par kādām problēmām/bojājumiem/zaudējumiem, kas var rasties, izmantojot šo programmatūru (jāpiebilst, ka šāds pat noteikums ir spēkā ne vien *BSD* licencei, bet arī *GPL* un praktiski visām komercprogrammatūras licencēm - tulk.piez.). Lielākā daļa *Slackware Linux* iekļautās programmatūras ir publicēta ar *BSD* licenci.

Nedaudz jaunākās Atvērtā pirmkoda kustības priekšgalā ir organizācija "Atvērtā pirmkoda iniciatīva" ("Open Source Initiative", *OSI*). Šīs organizācijas galvenais darbības veids ir atvērtā pirmkoda programmatūras popularizēšana, pie tam neatkarīgi no licences. Svarīgākais, lai jebkurai programmatūrai bez izpildāma (kompilēta) moduļa būtu pieejams arī pirmkods.

*OSI* būtība ir "pierunāt" pēc iespējas vairāk uzņēmumu atbalstīt atvērtā pirmkoda programmatūru, atļaujot tiem pašiem izstrādāt atvērtā pirmkoda licences un tās sertificējot. Daudzi uzņēmumi vēlas

publicēt savu produktu pirmkodu, bet nevēlas to veikt ar *GPL* licenci. Tā kā nav iespējams veikt būtiskas izmaiņas *GPL* licencē, uzņēmumiem tiek piedāvāts izstrādāt savu licenci un sertificēt to šajā organizācijā.

Lai arī Brīvās programmatūras fonds un Atvērtā pirmkoda iniciatīva darbojas, palīdzot viena otrai, tās tomēr nav viens un tas pats. Brīvās programmatūras fonds izmanto īpašu licenci un piedāvā ar šo licenci publicētu programmatūru. Atvērtā pirmkoda iniciatīva atbalsta visas atvērtā pirmkoda licences, arī Brīvās programmatūras fonda izstrādātās. Abu kustību argumentu pamatojumi reizēm sašķeļ šīs kustības, tomēr fakts, ka divas ideoloģiski atšķirīgas grupas darbojas, lai sasniegtu vienu mērķi, patiesībā uzlabo abu grupu darbības rezultātus.

## 2.nodaļa. Kur saņemt palīdzību?

Bieži gadās, ka nepieciešama palīdzība specifiskai sistēmas komandai, programmatūras uzstādīšanai, vai aparatūras konfigurēšanai. Varbūt jūs vienkārši vēlaties labāk izprast kādu komandu, vai aplūkot, kādas pielietojuma papildiespējas tā piedāvā. Par laimi, ir daudz veidu, kā iespējams saņemt nepieciešamo palīdzību. Instalējot *Slackware*, ir pieejama izvēle instalēt pakotnes no "F" sērijas, t.i. - *Bieži uzdotie jautājumi* (FAQ) un *HOWTO dokumenti* (HOWTOs). Papildus tam, programmas tiek komplektētas ar palīdzību par pieejamajiem iestatījumiem, konfigurācijas failiem un lietošanu.

### 2.1. Sistēmas palīdzība

#### man

Kā jau iespējams iedomāties, šo dokumentu skaits var ātri pieaugt, kļūstot pārāk liels pat pieredzējušam lietotājam. Šī iemesla dēļ dokumentācija ir sadalīta numurētās sadaļās. Šī sistēma ir darbojusies jau ilgu laiku - pietiekoši ilgu, lai bieži vien nāktos saskarties ar komandām, programmām un pat programmu bibliotēku funkcijām, kuru nosaukumam pievienots attiecīgais man sadaļas numurs. Piemēram, jūs varat saskarties ar atsaukumi uz `man(1)`. Numurs norāda, ka man dokumentācija atrodas sadaļā ar numuru 1 (lietotāja komandas). Attiecīgi ir iespējams norādīt, ka vēlaties aplūkot man dokumentāciju no sadaļas ar numuru 1, izmantojot komandu `man 1 man`. Iespēja norādīt man dokumentu sadaļu var būt noderīga, ja dažādās sadaļās atrodas dokumentācija dažādām komandām ar vienādiem nosaukumiem.

Komanda `man` (saīsinājums no angļu valodas *manual* - rokasgrāmata) *Unix* un *Linux* operētājsistēmās ir sistēmas tiešsaistes dokumentācijas tradicionālā forma. `man` palīdzības dokumenti jeb *man pages* ir īpaši formatēti teksta faili, kuri tiek sarakstīti lielākajai daļai sistēmas komandu un tiek izplatīti kopā ar attiecīgo programmatūru. Izpildot sistēmas komandu `man somecommand`, tiks attēlots `man` dokuments (bet protams) norādītajai komandai. Šajā piemērā tā būtu iedomāta programma `somecommand`.

#### man dokumentu sadaļas

Sadaļa	Saturs
Sadaļa 1	lietotāja komandas (ievads)
Sadaļa 2	sistēmas izsaukumi
Sadaļa 3	C bibliotēku izsaukumi
Sadaļa 4	iekārtas (piemēram, <code>hd</code> , <code>sd</code> )
Sadaļa 5	failu formāti un protokoli (piemēram, <code>wtmp</code> , <code>/etc/passwd</code> , <code>nfs</code> )
Sadaļa 6	spēles (ievads)
Sadaļa 7	standarti, makropakotnes u.c. (piemēram, <code>nroff</code> , <code>ascii</code> )
Sadaļa 8	sistēmas administrēšana (ievads)

Vienkāršākai dokumentācijas atrašanai papildus `man(1)` ir pieejamas arī komandas `whatis(1)` un `apropos(1)`.

Komanda `whatis` izvada ļoti īsu sistēmas komandu aprakstu - līdzīgi, kā kabatas rokasgrāmata.



Piemēram,

```
% whatis whatis
whatis (1) - search the whatis database for complete words
```

Komandu `apropos` izmanto, lai meklētu `man` dokumentu, kurš satur norādīto atslēgvārdu. Piemēram,

```
% apropos wav
cdda2wav (1) - a sampling utility that dumps CD audio data into wav sound files
netwave_cs (4) - Xircom Creditcard Netwave device driver
oggdec (1) - simple decoder, Ogg Vorbis file to PCM audio file (WAV or RAW)
wavelan (4) - AT&T CIS WaveLAN ISA device driver
wavelan_cs (4) - AT&T CIS WaveLAN PCMCIA device driver
wvlan_cs (4) - Lucent WaveLAN/IEEE 802.11 device driver
```

Vairāk informācijas par šīm komandām iespējams uzzināt, izlasot atbilstošo `man` dokumentāciju. ;)

## /usr/doc direktorija

Lielākās daļas mūsu kompilēto pakotņu pirmkoda arhīvos atrodama dažāda dokumentācija: *README* faili, lietotāja instrukcijas, licenču faili u.t.t. Jebkura dokumentācija, kura iekļauta pirmkoda arhīvos, tiek instalēta jūsu sistēmas `/usr/doc` direktorijā. Katrai programmai atbilstošā dokumentācija atradīsies direktorijā

```
/usr/doc/$program-$version
```

kur `$program` ir jūs interesējošās programmas nosaukums un `$version` ir (acīmredzami ;P) attiecīgās jūsu sistēmā instalētās pakotnes versija.

Piemēram, ja vēlaties aplūkot komandas `man (1)` dokumentāciju, jums jāizpilda komandu

```
% cd /usr/doc/man-$version
```

Ja atbilstošo `man` dokumentu lasīšana jums nesniedz pietiekoši informācijas, vai arī nesatur to īpašo detaļu, ko jūs meklējat, jūsu nākamajai meklēšanas vietai būtu jābūt `/usr/doc` direktorijai.

## HOWTO un mini-HOWTO dokumentācija

Tieši Atvērtā pirmkoda sabiedrības gars ir tas, kas novedis mūs pie *HOWTO/mini-HOWTO* kolekcijas. Šie dokumenti satur informāciju, ko izsaka to nosaukums angļu valodā - "*kā darbojas*" / "*kā veikt*" - tie ir dokumenti un rokasgrāmatas, kas satur instrukcijas, kā veikt dažādas lietas. Ja sistēmā ir uzstādīta *HOWTO* kolekcija, tad *HOWTO* dokumenti tiks instalēti direktorijā `/usr/doc/Linux-HOWTOs` un *mini-HOWTO* dokumenti - direktorijā `/usr/doc/Linux-mini-HOWTOs`.

Šajā pašā sistēmas pakotnē ir iekļauti arī dokumenti ar nosaukumu *FAQ*, kas angļu valodā nozīmē

**Frequently**

**Asked**

**Questions**

jeb latviski -

**Bieži**

**Uzdotie**

**Jautājumi**

Šie dokumenti ir sastādīti jautājumu un atbilžu formā par (pārsteigums!) bieži uzdotiem jautājumiem.

FAQ dokumenti bieži var būt ļoti noderīga vieta, kur lūkoties, ja jūs meklējat informāciju, kā vienkārši un ātri atrisināt kādu problēmu savā *Linux* sistēmā. Ja jūs instalēsiet *FAQ* pakotni sistēmas uzstādīšanas laikā, tie tiks instalēti direktoriņā `/usr/doc/Linux-FAQs`.

Šos failus ir vērts izlasīt arī tad, ja jūs neesat īsti pārliecināts, kā veikt to vai citu darbu. Tajos atrodama informācija par apbrīnojami daudz dažādām *Linux* niansēm, pie tam visbiežāk tie ir arī pārsteidzoši detalizēti. Laba lieta!

## 2.2. Tiešsaistes palīdzība

Papildus dokumentācijai, kura ir iekļauta *Slackware Linux* operētājsistēmas komplektā un ir instalējama datorā, ir pieejams arī plašs klāsts tiešsaistes dokumentācijas *Internet* tīklā.

### Oficiālā mājaslapa un palīdzības forumi

Oficiālā *Slackware Linux* mājaslapa reizēm satur nedaudz novecojušu informāciju, tomēr vienmēr ir atrodama informācija par jaunākajām *Slackware* versijām. Savulaik lapā eksistēja arī aktīvs palīdzības forums, pirms to apsēda *trollu*, problēmu radītāju un īdētāju ordas. Foruma uzturēšana kļuva pārāk komplicēta, tādēļ Patriks to slēdza. Vecā foruma arhīvs ir pieejams tiešsaistē mājaslapā <http://www.userlocal.com/phorum/>.

Pēc tam, kad oficiālajā mājaslapā <http://www.slackware.com> forums tika slēgts, parādījās vairākas citas lapas, kurās piedāvāja *Slackware* forumus. Pēc ilgām pārdomām Patriks atzina <http://www.linuxquestions.org> par oficiālo *Slackware Linux* forumu.

### E-pasta atbalsts

Katram, kurš iegādājas oficiālo *Slackware* CD komplektu, ir tiesības par brīvu saņemt izstrādātāju instalācijas atbalstu, izmantojot e-pastu. Vienlaikus lūdzam ņemt vērā, ka mēs, *Slackware* izstrādātāji (un arī lielākā daļa pieredzējušo lietotāju), piederam pie "vecās skolas". Tas nozīmē, ka mēs labprātāk palīdzam tiem, kuriem pašiem ir interese kaut ko apgūt, un vislabprātāk palīdzam tiem mācību procesā. Protams, mēs vienmēr mēģināsim darīt visu, ko varēsim, lai palīdzētu jebkuram, kurš nosūtījis mums e-pastu ar jautājumiem. Tomēr, pirms sūtīt mums e-pastu, lūdzam vispirms rūpīgi izlasīt dokumentāciju un mājaslapu (īpaši *FAQ* un varbūt dažus no zemāk minētajiem forumiem). Tā jūs varēsiet iegūt ātrāku atbildi un, jo mazāk mums būs e-pasta, jo ātrāk mēs varēsim palīdzēt tiem, kam tas patiešām ir nepieciešams.

Tehniskā atbalsta e-pasta adrese ir [support@slackware.com](mailto:support@slackware.com). Citas e-pasta adreses un saziņas informācija ir publicēta *Slackware Linux* mājaslapā.

### Slackware Linux projekta e-pasta vēstkopas

Mums ir vairākas e-pasta vēstkopas, kas pieejamas gan biļetena, gan vienkāršā veidā. Lūdzam iepazīties ar instrukcijām, kā uz tām parakstīties.

Lai parakstītos uz e-pasta vēstkopu, nosūtiet e-pastu uz adresi

[majordomo@slackware.com](mailto:majordomo@slackware.com)

E-pasta vēstules tekstā jābūt frāzei "**subscribe** (**vēstkopas nosaukums**)". Pieejamo vēstkopu nosaukumi ir norādīti zemāk.

E-pasta vēstkopu arhīvi ir pieejami *Slackware* projekta mājaslapā:

<http://www.slackware.com/lists/archive/>

**slackware-announce**

**slackware-announce** vēstkopā tiek izplatīti paziņojumi par jaunām versijām, būtiskiem jauninājumiem un cita vispārīga informācija.

#### **slackware-security**

**slackware-security** vēstkopa ir izveidota paziņojumiem par dažādiem drošības jautājumiem. Jebkādu *Slackware Linux* drošības caurumu atklāšanas gadījumos šajā vēstkopā nekavējoties tiek publicēta atbilstošā informācija.

Šīs vēstkopas ir pieejamas arī biļetena veidā. Tas nozīmē, ka jūs saņemsit vienu lielu ziņojumu, nevis daudzus mazus ziņojumus dienā. Tā kā *Slackware* vēstkopās to lasītāji informāciju izsūtīt nevar, tad vēstkopu datu plūsma ir samērā maza. Līdz ar to daudziem lasītājiem biļetena versija var likties neinteresanta. Tomēr, ja vēlaties, tā ir pieejama, attiecīgi parakstoties uz vēstkopām **slackware-announce-digest** un **slackware-security-digest**.

## **Neoficiālās mājaslapas un palīdzības forumi**

### **Mājaslapas**

**Google** (<http://www.google.com>)

Meklēšanas dzinēju kung-fu melnā josta. Izmantojiet vienmēr, kad vēlaties atrast informāciju par jūs interesējošo tematu.

**Google:Linux** (<http://www.google.com/linux>)

Google meklēšana par Linux tēmu

**Google:BSD** (<http://www.google.com/bsd>)

Google meklēšana par *BSD* tēmu. *Slackware* ir veidots tā, lai būtu tik tuva standarta *Unix* tipa sistēmai, ka, meklējot informāciju par *BSD*, gandrīz vienmēr iespējams atrast informāciju, kas gandrīz 100% attiecas arī uz *Slackware*. Bieži vien *BSD* meklējumi atklāj vairāk tehnisku informāciju, nekā uz *PR* orientētie *Linux* meklējumi.

**Google:Groups** (<http://groups.google.com>)

Meklējiet desmitiem gadus vecus *Usenet* ierakstus savām gudrības pērlēm.

**<http://userlocal.com>**

Virtuāls dārgums - plaša zināšanu bāze, labi padomi, pieredzes apmaiņa un interesanti raksti. Bieži vien šī ir pirmā vieta, kur varēsiet atrast informāciju par jaunām iestrādņēm *Slackware Linux*.

### **Palīdzības forumi**

**[linuxquestions.org](http://linuxquestions.org)**

Oficiāli atzīts tīmekļa forums *Slackware* lietotājiem

**[linuxiso.org](http://linuxiso.org) Slackware forums**

"Vieta, kur ielādēt *Linux* un saņemt palīdzību par to."

***Usenet* grupas (NNTP)**

*Usenet* jau daudzus gadus ir bijis datorfanātiķu pulcēšanās un savstarpējās palīdzības vieta. Ir maz *Usenet* vēstkopu, kas veltītas *Slackware Linux*, tomēr tajās pulcējas daudz zinošu cilvēku.

**`alt.os.linux.slackware`**

**`alt.os.linux.slackware`** ir viena no visaktīvākajām vietām, kur meklēt tehnisku palīdzību par *Slackware* problēmām. Kā jebkurā *Usenet* vēstkopā, daži neizpalīdzīgi biedri ("*trolli*") var sabojāt visu iespaidu, nepārtraukti *gudrējoties*, tomēr, apgūstot prasmes ignorēt šos *trollus* un identificēt tos, kuri patiešām palīdzēs, ļaus iegūt maksimālu labumu no šī resursa.

## 3.nodaļa. Instalēšana

Pirms iespējams uzsākt *Slackware Linux* lietošanu, to vispirms ir jāiegādājas vai jāielādē, un jāinstalē. *Slackware* saņemšana ir ļoti vienkārša - to ir iespējams iegādāties vai bez maksas ielādēt *Internet* tīklā. Instalēšana arī ir vienkārša, ja vien jums ir kādas priekšzināšanas par savu datoru un vēlēšanās iemācīties vairāk. Instalēšanas programma pati par sevi ir izveidota tā, ka instalēšanas process ir sadalīts vairākās daļās, kuras jāizpilda soli pa solim, tādēļ sistēmas instalēšanu ir iespējams veikt samērā ātri. Patiesībā pieredzējuša lietotāja rokās *Slackware Linux* instalēšanas laiki ir vieni no mazākajiem dažādo *Linux* distributīvu vidū.

### 3.1. Apgādājamies ar *Slackware*

#### Oficiālie diski un iepakojuma komplekts

Slackware Linux, Inc. piedāvā iegādāties oficiālos *Slackware Linux* CD-ROM diskus. CD-ROM disku komplektā ietilpst 4 diski. Pirmais disks satur visu nepieciešamo vienkārša servera instalēšanai, kā arī X logu sistēmu. Otrais disks ir "dzīvais" jeb "live" CD - ielādējams disks, kas ielādē sistēmu operatīvajā atmiņā bez instalēšanas un ļauj jums aplūkot *Slackware Linux* darbībā vai veikt datu vai datora atjaunošanu. Šis CD-ROM satur arī papildus pakotnes - piemēram, KDE un GNOME grafiskās vides. Bez tam, **extra** direktoriņā ir iekļautas arī dažas papildus pakotnes. Trešais un ceturtais CD-ROM satur visu *Slackware* izmantoto programmu pirmkodu, kā arī šīs grāmatas angļu versijas oriģinālo tekstu.

Ir iespējams arī iegādāties iepakotu komplektu, kas satur minētos 4 diskus un šīs grāmatas angļu valodas versijas drukātu eksemplāru, kā arī daudz dažādus *Slackware* suvenīriņus, kas ļaus jums palielīties par savu ieguvumu. Ir pieejama arī CD disku abonēšana.

*Slackware* produkciju ir iespējams iegādāties *Internet* tīklā *Slackware* internetveikalā:

<http://store.slackware.com>

Pasūtījumus ir iespējams arī iesniegt pa e-pastu.

#### Slackware Linux, Inc. saziņas informācija

Metode	Saziņas informācija
Tālrunis	1-(925) 674-0783
Internets	<a href="http://store.slackware.com">http://store.slackware.com</a>
E-pasts	<a href="mailto:orders@slackware.com">orders@slackware.com</a>
Pasts	1164 Claremont Drive, Brentwood, CA 94513

#### Lejupielāde *Internet* tīklā

*Slackware Linux* ir par brīvu pieejams *Internet* tīklā. Tādā gadījumā jūs vienalga varat mums sūtīt e-pasta vēstules ar saviem jautājumiem, tomēr augstāka prioritāte tiks piešķirta tiem, kuri būs iegādājušies oficiālo CD komplektu. Pirms sūtīt mums e-pastu, lūdzu izlasiet 2.nodaļu.

Oficiālās *Slackware Linux* mājaslapas adrese ir

<http://www.slackware.com/>

Primārās FTP vietnes adrese ir

<ftp://ftp.slackware.com/pub/slackware/>

Lūdzu ņemiet vērā, ka mūsu FTP vietnei, lai arī tā ir brīvi pieejama, ir ierobežots tīkla plūsmas ātrums. Lūdzam apsvērt iespēju *Slackware* lejupielādei izmantot jums fiziski tuvāko spoguļserveri. Nepilnīgs spoguļserveru saraksts pieejams mūsu mājaslapā

<http://www.slackware.com/getslack/>

## 3.2. Sistēmas prasības

Vienkāršas *Slackware* sistēmas prasības ir vismaz šādas:

### Sistēmas prasības

Aparatūra	Prasības
Procesors	Pentium vai jaunāks
RAM	32MiB
Diska vieta	1GiB
Ārējais datu nesējs	4x CD-ROM

Ja jums ir ielādējams CD-ROM, tad, ticamākais, jums nebūs nepieciešama diskešu iekārta. Protams, ja jums nav CD-ROM iekārtas, tad jums noteikti vajadzēs diskešu iekārta, lai veiktu instalēšanu no tīkla. NFS instalēšanai ir nepieciešama tīkla karte. Vairāk informācijas sadaļā par *NFS*.

Diska vietas prasība ir samērā āķīga. 1GiB ieteikums lielākajā daļā gadījumu ir pietiekošs, tomēr pilnai sistēmas instalācijai jums būs nepieciešami aptuveni 2GiB, kā arī papildus diska vieta jūsu personīgajiem failiem. Tiesa, lielākā daļa lietotāju neveic pilnu sistēmas instalāciju. Patiesībā daudzi darbina *Slackware*, aizņemot tik vien, kā 100MiB diska vietas.

*Slackware* ir iespējams instalēt sistēmās ar mazāku RAM apjomu, mazākas ietilpības diskos un vājākiem procesoriem, tomēr šāda nestandarta instalēšana var prasīt nelielu *roku smērēšanu*. Ja nu jūs esat gatavs nedaudz pasvīst, aplūkojiet failu `LOWMEM.TXT` distribūcijas kokā. Tas satur dažus derīgus padomus.

### Pakotņu sērijas

Vienkāršības nolūkā *Slackware Linux* vēsturiski ir sadalīts programmatūras sērijās. Kādreiz tos sauca par "disku komplektiem", jo tie tika izveidoti instalēšanai no disketēm. Mūsdienās sērijas tiek izmantotas programmatūras sadalīšanai kategorijās un instalēšana no disketēm vairs nav iespējama.

Zemāk redzamajā tabulā sniegti īsi komentāri katrai programmatūras sērijai.

### Programmatūras sērijas

Sērija	Saturs
A	Bāzes sistēma. Satur pietiekoši daudz programmatūras, lai iegūtu strādājošu sistēmu ar tekstu redaktoru un vienkāršu komunikāciju programmu.
AP	Dažādas aplikācijas, kurām nav nepieciešama X logu sistēma.
D	Programmatūras izstrādes rīki. Kompilatori, atklūdotāji, interpretatori un man dokumenti.
E	<i>GNU Emacs</i> tekstu redaktors.

Sērija	Saturs
F	FAQ, HOWTO un cita dokumentācija.
GNOME	GNOME darbvirsma vide
K	Linux kodola pirmkods
KDE	K Desktop Environment darbvirsma vide. X logu sistēmas vide, kurai ir daudz kas kopīgs ar MacOS un Microsoft Windows. KDE pamats - Qt bibliotēka - arī ir iekļauta šajā sērijā.
KDEI	KDE internacionalizācijas pakotnes.
L	Bibliotēkas. Dažādas dinamiskās bibliotēkas, kuras nepieciešamas daudzām citām programmām.
N	Tīkla programmas. Dēmoni, e-pasta programmas, telnet, vēstkopu lasītāji u.c.
T	teTeX dokumentu formatēšanas sistēma
TCL	Tool Command Language. Tk, TclX un TkDesk pakotnes.
X	X logu sistēma - Slackware Linux operētājsistēmas grafiskā vide.
XAP	X logu sistēmas aplikācijas, kuras nav standarta darbvirsma vides sastāvdaļas, piemēram, Ghostscript un Netscape.
Y	BSD teksta režīma spēles.

## Instalēšanas metodes

### Diskete

Lai gan kādreiz bija iespējams visu *Slackware Linux* sistēmu instalēt no disketēm, arvien lielākais programmatūras pakotņu (patiesībā atsevišķu programmu) apjoms ir piespiedis mūs atstāt šo instalēšanas iespēju. Vēl *Slackware 7.1* versijā bija iespējams daļēji instalēt sistēmu no disketēm. Bija iespējams gandrīz pilnībā instalēt A un N sērijas, lai iegūtu pamatsistēmu, no kuras instalēt visu pārējo. Ja jūs vēlaties instalēt sistēmu no disketēm (tipiski gadījumā ar novecojušu aparatūru), ieteicams atrast citu instalēšanas veidu vai instalēt vecāku *Slackware* versiju. *Slackware 4.0* šī iemesla dēļ vēl ir ļoti populāra, līdzīgi, kā 7.0.

Lūdzam ņemt vērā, ka disketes joprojām ir nepieciešamas CD-ROM instalēšanai, ja jums nav ielādējams CD-ROM, kā arī NFS instalēšanai.

### CD-ROM

Ja jums ir ielādējama CD, kas pieejama oficiālajā Slackware Linux, Inc. publicētajā disku komplektā (lūdzu skatīt sadaļu "Apgādājamies ar Slackware"), CD instalēšana būs vienkāršāka. Ja jums šāda CD nav, jums būs jāielādējas no disketēm. No disketēm būs jāielādējas arī tad, ja jums ir specifiska aparatūra, kas sarežģī komplektā iekļautā *Linux* kodola izmantošanu. Tādā gadījumā būs jāgatavo un jāizmanto specifiskas disketes.

*Slackware Linux 8.1* versijā ielādējamo CD izveidei tiek izmantota jauna metode, kura pārāk labi nedarbojas ar dažām pusjukušām BIOS mikroshēmām (jāpiemin arī, ka lielākā daļa *Linux* CD cieš no šīs pašas problēmas). Ja gadījusies šāda situācija, iesakām ielādēties no disketes.

Zemāk sniegta informācija par to, kā izvēlēties un izveidot ielādējamās disketes, ja tas būtu nepieciešams.

## NFS

NFS (Network File System jeb Tīkla failsistēma) ir veids, kā padarīt failsistēmas pieejamas attālinātiem datoriem. NFS instalēšana ļauj instalēt *Slackware* no cita datora jūsu tīklā. Dators, no kura notiek instalēšana, ir jākonfigurē tā, lai eksportētu *Slackware* distribuīva koku datoram, kurā tiks instalēta sistēma. Protams, šim nolūkam nepieciešamas NFS zināšanas, kuras aplūkotas sadaļā 5.6.

Ir iespējams veikt NFS instalēšanu, izmantojot tādas metodes kā PLIP (ar paralēlo portu), SLIP un PPP (izmantojot modemu). Ja vien iespējams, mēs tomēr iesakām izmantot tīkla karti. Galu galā, operētājsistēmas instalēšana, izmantojot paralēlo portu, ir ļoti, ļoti lēns process.

## Ielādes disks

Ielādes (boot) disks ir diskete, no kuras tiek ielādēta sistēma, lai uzsāktu instalēšanas procesu. Tā satur kompresētu (saspiestu) kodolu, kurš tiek izmantots aparatūras vadībai instalēšanas laikā. Tas nozīmē, ka šī diskete ir obligāti nepieciešama, ja vien jūs sistēmas ielādi neveicat no CD diska, kā aprakstīts sadaļā *CD-ROM*. Ielādes disku attēlfaili ir novietoti distribūcijas koka `bootdisks/` direktorijā.

*Slackware* ielādes disku ir vairāk, kā jūs varat iedomāties (tas ir - 16). Pilns ielādes disku saraksts ar katra aprakstu ir pieejams distribūcijas kokā failā `bootdisks/README.TXT`. Tiesa, lielākajā daļā gadījumu ir iespējams izmantot ielādes diskus `bare.i` (IDE iekārtām) vai `scsi.s` (SCSI iekārtām).

Nedaudz tālāk šajā grāmatā jūs atradīsiet instrukcijas ielādes disku izveidei, izmantojot disku attēlfailus.

Pēc ielādes jums tiks piedāvāts ievietot saknes (`root`) disku. Mēs iesakām vispirms izveidot tikai ielādes disku un pamēģināt ielādēt sistēmu no tā.

## Saknes diski

Saknes (`root`) diski ir disketes, kas satur *Slackware* uzstādīšanas programmu, kā arī failsistēmu, kura tiek izmantota instalēšanas laikā. Arī tie ir nepieciešama sekmīgai sistēmas instalēšanai. Saknes disku attēlfaili atrodas distribūcijas koka `rootdisks/` direktorijā. Jums būs jāizveido divus saknes diskus, izmantojot attēlfailus `install.1` un `install.2`. Papildus tam, `rootdisks/` direktorijā jūs atradīsiet diskattēlus `network.dsk`, `pcmcia.dsk`, `rescue.dsk` un `sbootmgr.dsk`.

## Papilddisks

Papilddisks ir nepieciešams, lai veiktu NFS instalēšanu vai, lai instalētu *Slackware* datorā ar PCMCIA iekārtām. Papilddisku attēlfaili atrodas distribūcijas koka `rootdisks/` direktorijā failos `network.dsk` un `pcmcia.dsk`. Nesen šiem papilddiskiem ir pievienoti arī papilddiski `rescue.dsk` un `sbootmgr.dsk`. Fails `rescue.dsk` satur sistēmas atjaunošanas disketes attēlfailu. Sistēmas atjaunošanas diskete ir neliels saknes disks, kura saturs tiek ielādēts datora operatīvajā atmiņā un darbojas 4 MB RAM diskā. Tas satur dažas nepieciešamākās tīkla utilitprogrammas un `vi` tekstu redaktoru, kas ļauj veikt nepieciešamos labojumus sabojātās sistēmās. `sbootmgr.dsk` satur alternatīvu disku ielādes programmu. Jūs varat ielādēt sistēmu no šī diska, ja jūsu dators nevēlas ielādēties no *Slackware* instalācijas CD. Jums tiks piedāvāta iespēja izvēlēties, no kuras iekārtas ielādēt sistēmu, tādējādi rodot ērtu risinājumu BIOS kļūdu apiešanai.

Par nepieciešamību izmantot papilddiskus pēc ielādes jūs informēs saknes disks.

## Disku izveide

Kad jūs esat izvēlējis ielādes diska attēlfailu, to ir jāieraksta disketē. Šis process var atšķirties dažādām operētājsistēmām. Ja jūsu datorā darbojas *Linux* (vai praktiski jebkura cita *Unix* tipa operētājsistēma), jums būs jāizmanto komanda `dd (1)`. Pieņemot, ka izvēlēts diska attēlfails `bare.i` un jūsu datora diskešu iekārta ir `/dev/fd0`, jāizpilda šādu komandu disketes izveidei:

```
% dd if=bare.i of=/dev/fd0
```

Ja jūs darbināt kādu no *Microsoft* operētājsistēmām, jums būs jāizmanto distribūcijas kokā iekļauto programmu **RAWRITE.EXE**. Tā atrodas tajā pat direktoriņā, kur disku attēlfaili. Pieņemot, ka izvēlētais attēlfails ir **bare.i** un diskešu iekārta ir **A:**, jāatver DOS komandrindu un jāizpilda komandu

```
C:\> rawrite a: bare.i
```

### 3.3. Cietā diska sadaļu izveide

Ja jums ir izdevies veiksmīgi ielādēties no jūsu izvēlēta datu nesēja (ielādes disketes vai CD-ROM - tulk.piez.), nākamais veicamais darbs ir cietā diska sadaļu izveide. Diska sadaļa ir vieta cietajā diskā, kurā tiks izveidota *Linux* failsistēma un kurā pēc tam tiks instalēta *Slackware*. Mēs iesakām izveidot vismaz divas sadaļas - vienu jūsu saknes failsistēmai (/) un otru maiņvietai (**swap**).

Kad saknes disks (vai ielādes CD-ROM - tulk.piez.) ir beidzis ielādi, jums tiks piedāvāts ievadīt lietotājevārdu. Ievadiet lietotājevārdu **root** (bez paroles). Komandrindā iedarbiniet vai nu **cfdisk(8)** vai **fdisk(8)** programmu. **cfdisk** piedāvā lietotājam draudzīgāku interfeisu, kā parastā **fdisk** programma, tomēr tai trūkst dažas iespējas. Zemāk mēs aplūkosim **fdisk** programmu.

Jāsāk ar **fdisk** programmas iedarbināšanu jūsu cietajam diskam. *Linux* sistēmās cietajiem diskam nav iekārtu burtu apzīmējumi - tie tiek apzīmēti ar failiem. Pirmais IDE disks (*primary master*) ir **/dev/hda**, *primary slave* ir **/dev/hdb** utt. SCSI diskam ir līdzīgi apzīmējumi, taču tie ir formā **/dev/sdX**. jums būs jāiedarbina **fdisk** un jānorāda tam jūsu cietā diska fails:

```
# fdisk /dev/hda
```

Kā jebkura laba *Unix* programma, **fdisk** piedāvā jums komandrindu (jūs jau cerējāt redzēt skaitu izvēlni, ko?). Pirmais, ko jāizdara, ir esošo diska sadaļu pārbaude. To ir iespējams veikt, ievadot komandu **p**:

```
Command (m for help): p
```

Šī komanda attēlos dažādu informāciju par esošajām diska sadaļām. Lielākā daļa tomēr izvēlas neizmantotu disku, kurā instalēt *Slackware* un nodzēš visas esošās sadaļas, lai atbrīvotu vietu *Linux* sadaļām.

**Brīdinājums: IR ĻOTI SVARĪGI PIRMS ŠĪS DARBĪBAS IZVEIDOT REZERVES KOPIJU VISAI SVARĪGAJAI INFORMĀCIJAI, KAS IZVIETOTA SADAĻĀ, KURU VĒLATIES DZĒST.**

Neeksistē vienkārša metode, kas ļautu atjaunot izdzēstu sadaļu, tādēļ, pirms veikt kādas darbības ar diska sadaļām, vienmēr izveidojiet datu rezerves kopiju.

Sadaļu informācijas tabulā jāparādās sadaļas numuram, sadaļas apjomam un tipam. Tabulā tiek parādīta arī cita informācija, tomēr par to pagaidām varat neuztraukties. Izdzēsīsīm visas šīs diska sadaļas, lai atbrīvotu vietu *Linux* sadaļām. Šim nolūkam jāizmanto komandu **d** (*delete*):

```
Command (m for help): d
```

```
Partition number (1-4): 1
```

Šo darbību jāatkārto katrai sadaļai. Kad tās visas ir izdzēstas, mēs esam sagatavojušies jaunai *Linux* sadaļu izveidei. Iepriekš mēs nolēmām izveidot vienu sadaļu saknes failsistēmai un vienu - maiņvietai. Svarīgi atgādināt, ka *Unix* sadaļu veidošanas sistēmas ir iemesls daudziem strīdiem, un lielākā daļa pieredzējušo lietotāju jums ieteiks katrs savu metodi diska dalīšanai. Sākotnēji izveidosim sadaļu / un sadaļu maiņvietai. Ar laiku jūs izveidosit metodi, kas ļaus jums izveidot sadaļas tā, kā jums tas patiks.

Es izmantoju divas pamata disku dalīšanas shēmas. Pirmo izmantoju darbstaģijām. Es izveidoju 4 sadaļas - /, /home, /usr/local un maiņvietu. Tas ļauj man pārinstalēt vai atjaunināt visu manu sistēmu, kas izvietota /, nedzēšot manus datu failus /home direktoriņā vai manas paškompilētās programmas /usr/local direktoriņā. Serveru gadījumā es bieži vien aizstāju /usr/local sadaļu ar /var. Daudz dažādu serveru glabā informāciju tieši šajā sadaļā un tās atdalīšana no / dod zināmu ātrdarbības pieaugumu. Tomēr pagaidām paliksīm pie divām sadaļām: / un maiņvietas.



Tagad izveidosim sadaļas, izmantojot komandu `n`:

```
Command (m for help): n
Command action
  e extended
  p primary partition (1-4)
p
Partition number (1-4): 1
First cylinder (0-1060, default 0): 0
Last cylinder or +size or +sizeM or +sizeK (0-1060, default 1060): +64M
```

Jāpārlicinās, ka mēs izveidojam tikai primārās sadaļas. Pirmā sadaļa būs maiņvietas sadaļa. Vispirms mēs norādām `fdisk`, ka vēlamies izveidot sadaļu 1 kā primāro sadaļu. Tās sākums būs cietā diska cilindrā 0 (t.i., diska sākumā - tulk. piez.). Kā sadaļas beigas mēs norādām +64M. Tas nozīmē, ka maiņvietas sadaļas apjoms būs 64 MiB. Patiesībā maiņvietas sadaļas ieteicamais apjoms ir atkarīgs no jūsu datora RAM apjoma. Parasti maiņvietas apjomu nosaka divas reizes lielāku par RAM apjomu. Tad noteiksim, ka primārā sadaļa 2, kas sāksies pirmajā pieejamajā cilindrā, aizpildīs atlikušo pieejamo diska vietu:

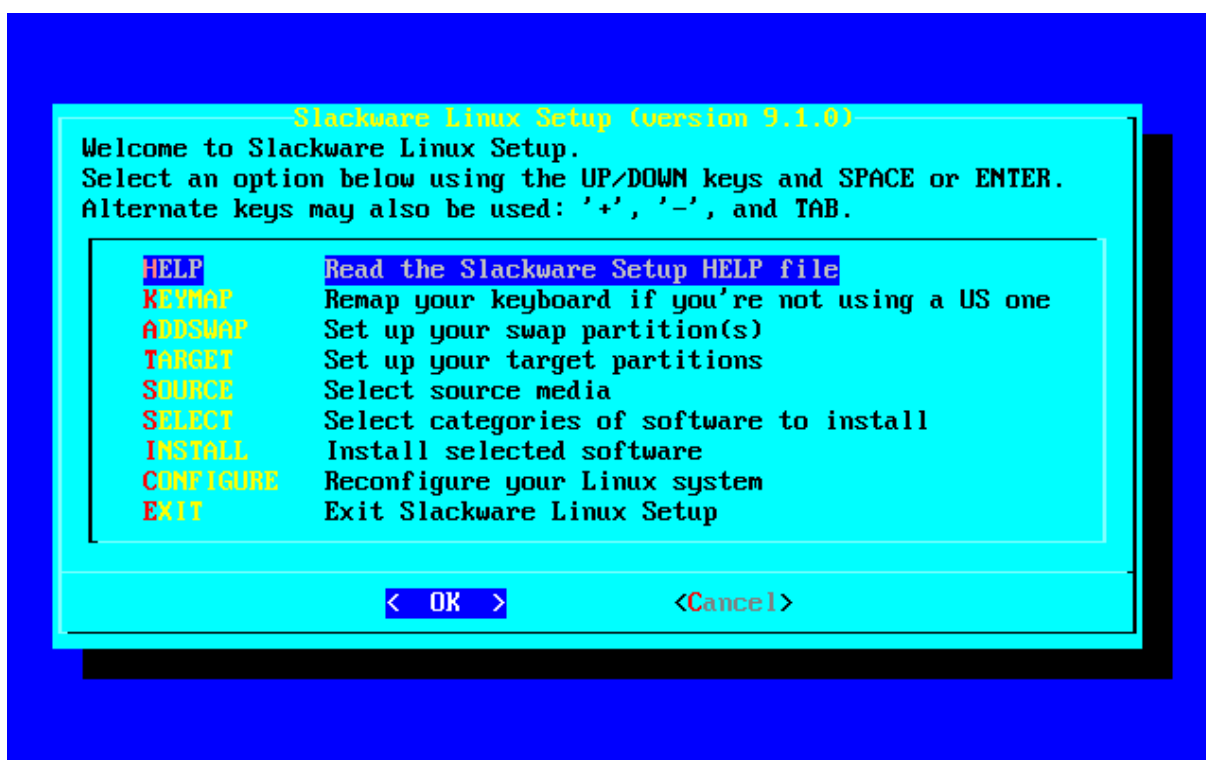
```
Command (m for help): n
Command action
  e extended
  p primary partition (1-4)
p
Partition number (1-4): 2
First cylinder (124-1060, default 124): 124
Last cylinder or +size or +sizeM or +sizeK (124-1060, default 1060): 1060
```

Tā - esam gandrīz galā. Vēl tikai jānomaina pirmās sadaļas tipu uz 82 (*Linux swap*). Ievadiet komandu `t`, lai nomainītu tipu, izvēlieties sadaļu 1, un ievadiet 82. Pirms ierakstīt izmaiņas diskā, aplūkojiet vēlreiz jaunizveidoto sadaļu tabulu, izmantojot komandu `p`. Ja viss izskatās labi, ievadiet `w`, lai ierakstītu izmaiņas diskā un izietu no `fdisk`.

### 3.4. Uzstādīšanas programma setup

Kad izveidotas cietā diska sadaļas, ir veikti visi priekšdarbi, un ir iespējams instalēt *Slackware*. Nākamais instalēšanas procesa solis ir programmas `setup(8)` iedarbināšana. Lai to veiktu, komandrindā vienkārši ievadiet komandu `setup`. `setup` ir uz izvēlnēm balstīta sistēma, kas ļauj instalēt *Slackware* pakotnes un veikt sistēmas sākotnējo konfigurēšanu.

## Slackware Linux setup programma



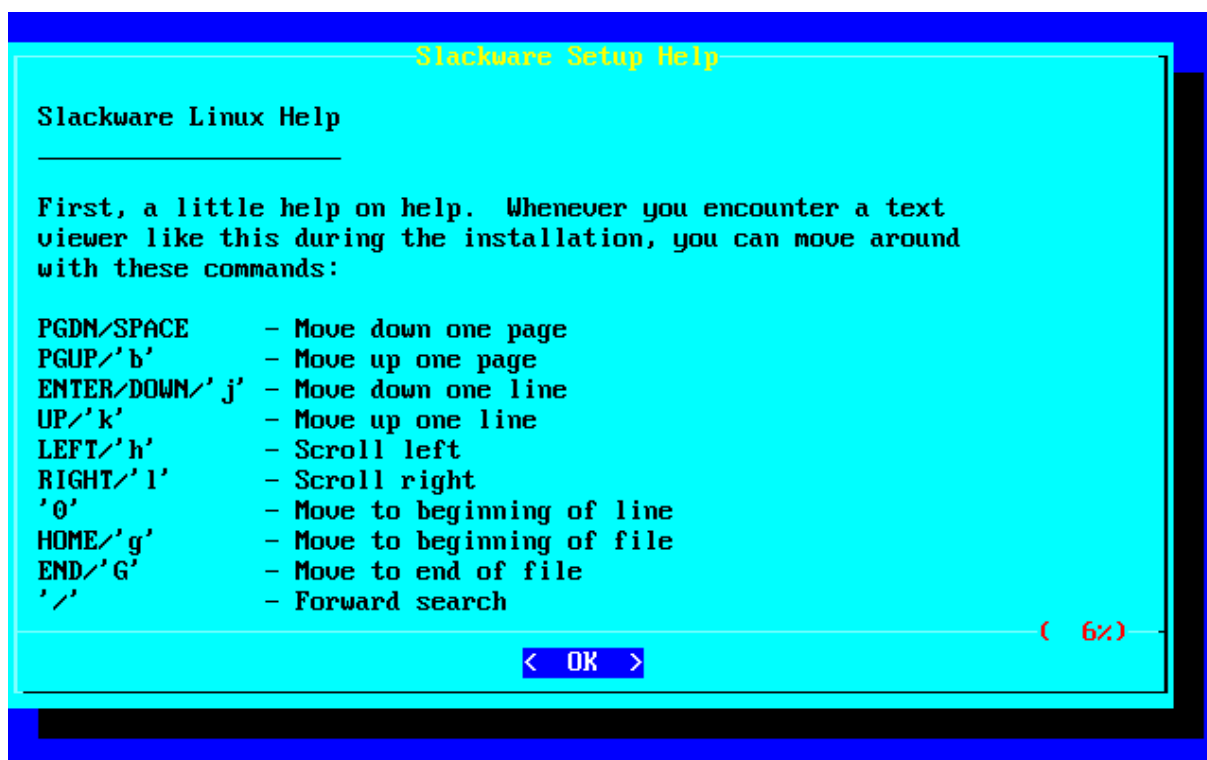
Uzstādīšanas process notiek, secīgi izpildot katru `setup` programmas izvēlnē norādīto darbību. Protams, ir iespējams šīs darbības izpildīt gandrīz jebkurā secībā, tomēr pastāv zināma iespēja, ka tas var īsti nedarboties. Izvēlnes punkti tiek izvēlēti, izmantojot tastatūras taustiņus "bultiņa uz augšu" un "bultiņa uz leju". Ekrānpogas "OK" un "Cancel" tiek izvēlētas, izmantojot taustiņus "bultiņa pa labi" un "bultiņa pa kreisi". Papildus tam, katram punktam ir piekārtots atbilstošs tastatūras taustiņš, kurš ir izgaismots punkta nosaukumā. Izvēlnes punkti, kuri ir pārslēdzami (apzīmēti ar [x]), tiek pārslēgti, izmantojot atstarpes taustiņu.

Protams, viss šis process ir izskaidrots `setup` sadaļā "help", tomēr mēs vēlamies sniegt mūsu lasītājiem viņu samaksātajam atbilstošu vērtību (bez tam, sadaļa "help" ir angļu valodā - tulk. piez.).

### HELP (palīdzība)

Ja šī ir pirmā reize, kad jūs instalējat *Slackware*, jūs varētu vēlēties aplūkot palīdzības ekrānu. Tajā atrodams katras `setup` sadaļas apraksts (līdzīgs, kā šeit) un instrukcijas visas pārējās instalācijas programmas navigācijai.

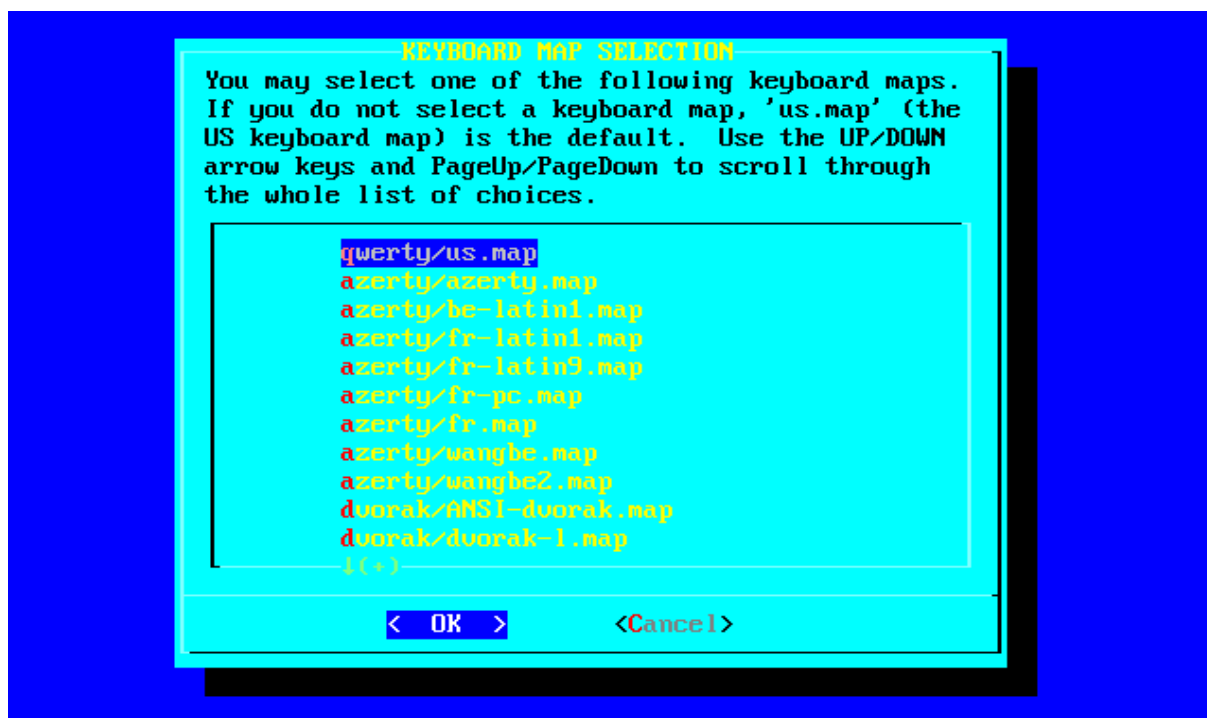
## HELP ekrāns



## KEYMAP (tastatūras izkārtojums)

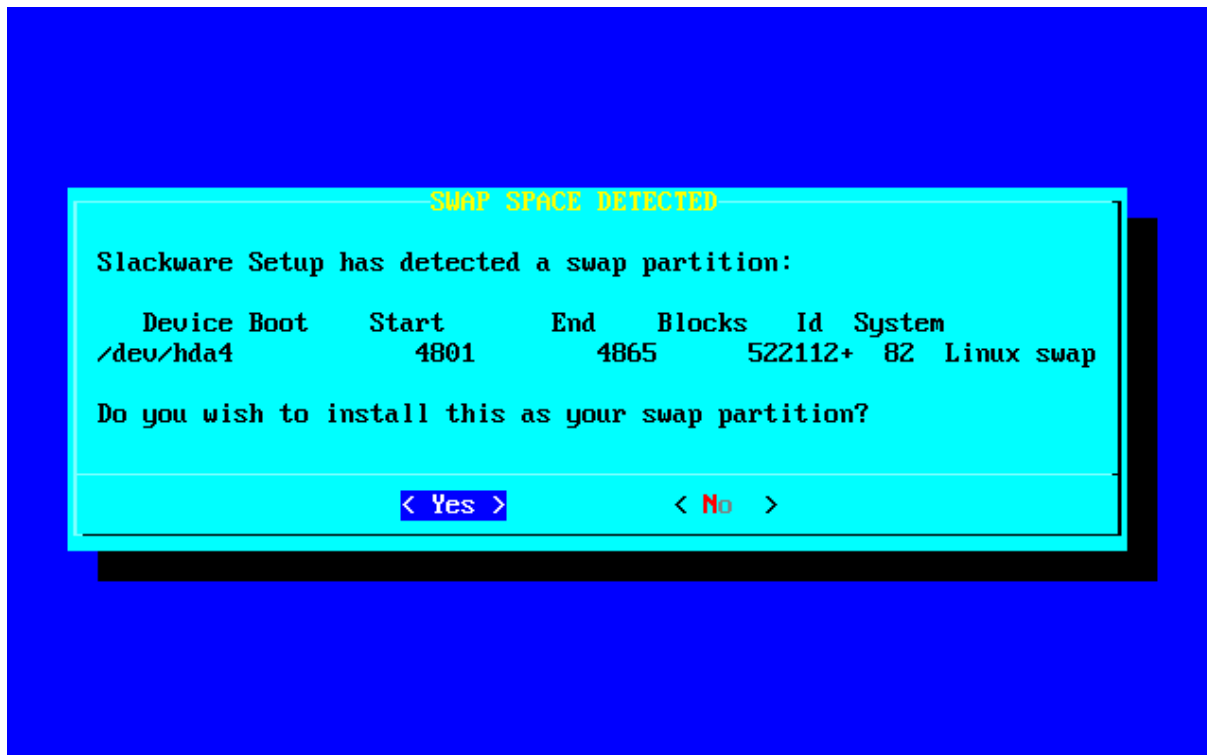
Ja jums nepieciešams tastatūras izkārtojums, kas atšķiras no ASV QWERTY izkārtojuma, jūs varētu vēlēties aplūkot šo sadaļu. Tajā tiek piedāvāti vairāki alternatīvi tastatūras izkārtojumi.

## KEYMAP ekrāns



## ADDSWAP (maiņvietas aktivizēšana)

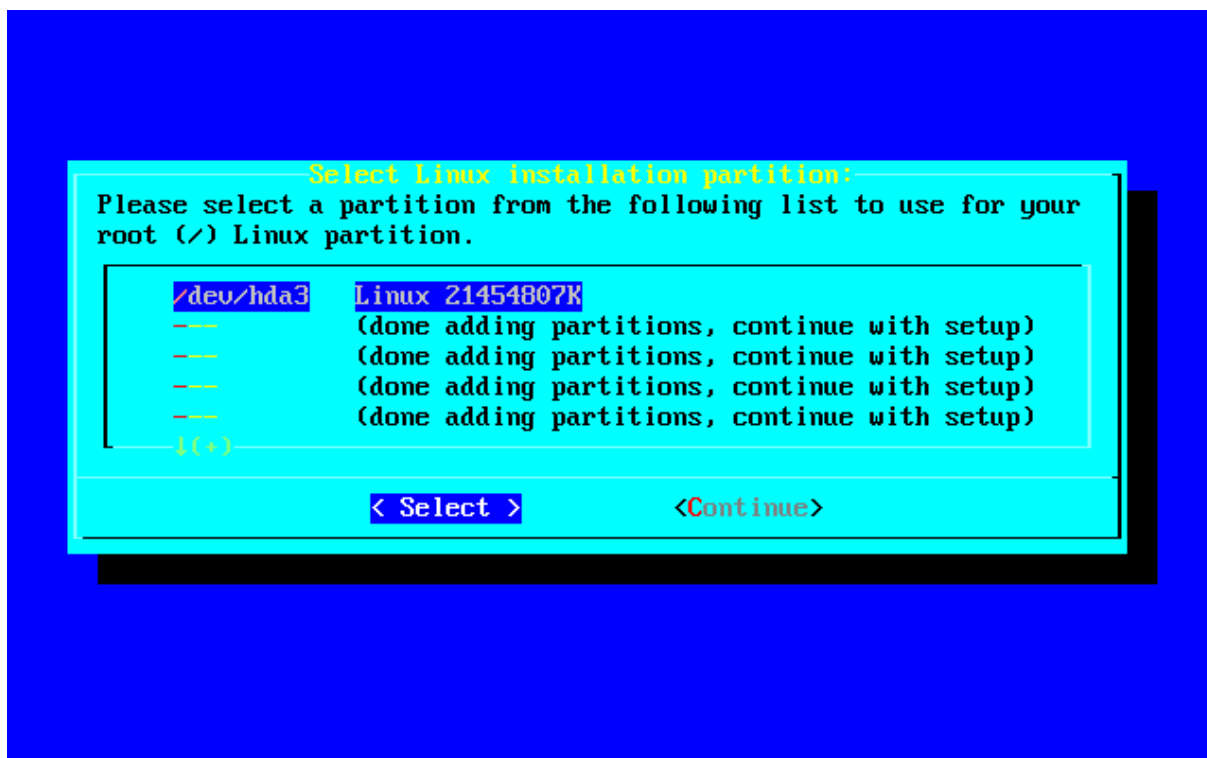
Noteikta maiņvietas sadaļa



Ja jūs izveidojāt maiņvietas sadaļu (sk. sadaļu 3.3), šī `setup` sadaļa ļaus jums to aktivizēt. Tā automātiski noteiks un attēlos jūsu cietajā diskā pieejamās maiņvietas sadaļas un ļaus izvēlēties vienu, kuru formatēt un aktivizēt.

## TARGET (mērķa sadaļas)

### Mērķa sadaļu izvēles ekrāns



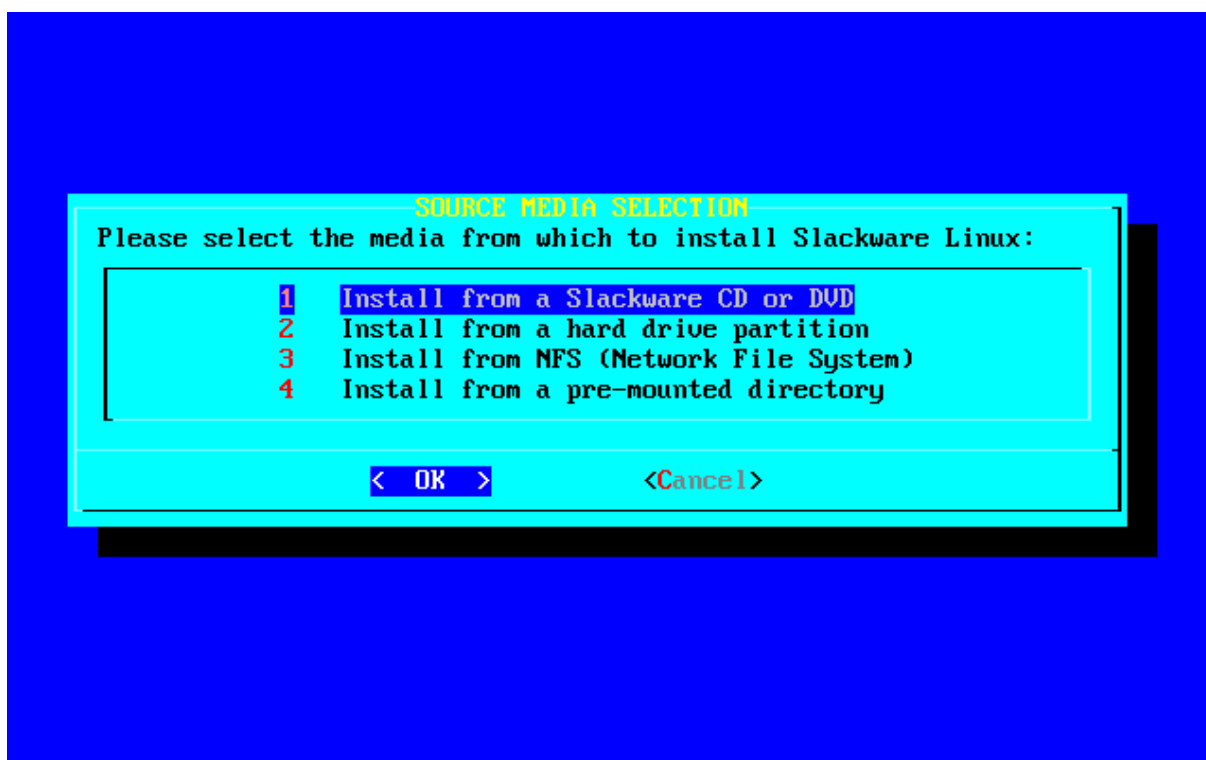
Mērķa sadaļu izvēles ekrāns ir vieta, kurā jūs varat formatēt un aktivizēt pārējās (tās, kuras nav maiņvietas) sadaļas. Sarakstā tiks parādītas visas pieejamās sadaļas. Katrai sadaļai būs iespējams izvēlēties, vai to formatēt. Atkarībā no izvēlēta kodola, ir iespējams izvēlēties starp `reiserfs` (pēc noklusējuma), `ext3`, `ext2`, `jfs` un `xfs` failsistēmām. Lielākā daļa izmanto `reiserfs` vai `ext3`. Tuvākajā nākotnē dienas gaismu varētu ieraudzīt arī `reiserfs4` atbalsts.

Pirmais, ko šajā sadaļā jāveic, ir saknes failsistēmas (/) sadaļas izvēle. Pēc tam atlikušās sadaļas ir iespējams piesaistīt jūsu izvēlētajām failsistēmām. Piemēram, jūs varētu vēlēties trešo sadaļu (pieņemsim, ka tāda ir izveidota un tās fails ir `/dev/hda3`) izmantot kā jūsu mājas (`/home`) failsistēmu. Šis ir tikai piemērs - piesaistiet failsistēmas tā, kā uzskatāt par nepieciešamu.

## SOURCE (instalēšanas datu nesējs)

Šajā sadaļā jūs varat izvēlēties datu nesēju, no kura notiks *Slackware* instalēšana. Šobrīd ir iespējams izvēlēties CD-ROM, NFS, vai iepriekš piemontētu direktoriju.

## Datu nesēja izvēles ekrāns



CD-ROM izvēle ļauj instalēt sistēmu no CD diska. jums tiks piedāvāta papildizvēlne, kas ļaus izvēlēties starp automātisku CD-ROM iekārtas atrašanu un sarakstu, no kura izvēlēties CD-ROM iekārtas tipu. Lūdzam pārliecināties, ka *Slackware* CD disks ir ievietots CD-ROM iekārtā, pirms ļauj uzstādīšanas programmai to meklēt.

NFS izvēle pieprasa ievadīt informāciju par jūsu tīkla konfigurāciju un NFS servera adresi. NFS serverim jābūt uzstādītam un jādarbojas. Papildus atgādinām, ka nedrīkst izmantot hostu nosaukumus - servera un jūsu datora norādīšanai jāizmanto to IP adreses. Protams, jāizmanto arī `network.dsk` papilddisku, lai uzstādīšanas procesam pievienotu tīkla atbalstu.

Iepriekš piemontēta direktorija pieļauj vispielāgojamāko instalēšanu. Šo metodi iespējams izmantot, lai uzstādītu *Slackware* no *Jaz* diskkiem, piemontētām NFS failsistēmām (izmantojot *PLIP*), kā arī FAT failsistēmām. Piemontējiet failsistēmu jūsu izvēlētā vietā pirms `setup` izpildes un norādiet to šeit.

## SELECT (pakotņu izvēle)

Šajā uzstādīšanas posmā iespējams izvēlēties uzstādāmās pakotņu sērijas. Šīs sērijas ir aprakstītas sadaļā 3.2. Lūdzam atcerēties, ka, lai iegūtu strādājošu pamatsistēmu, obligāti ir jāinstalē tikai A sērijas. Visas pārējās var instalēt pēc izvēles.

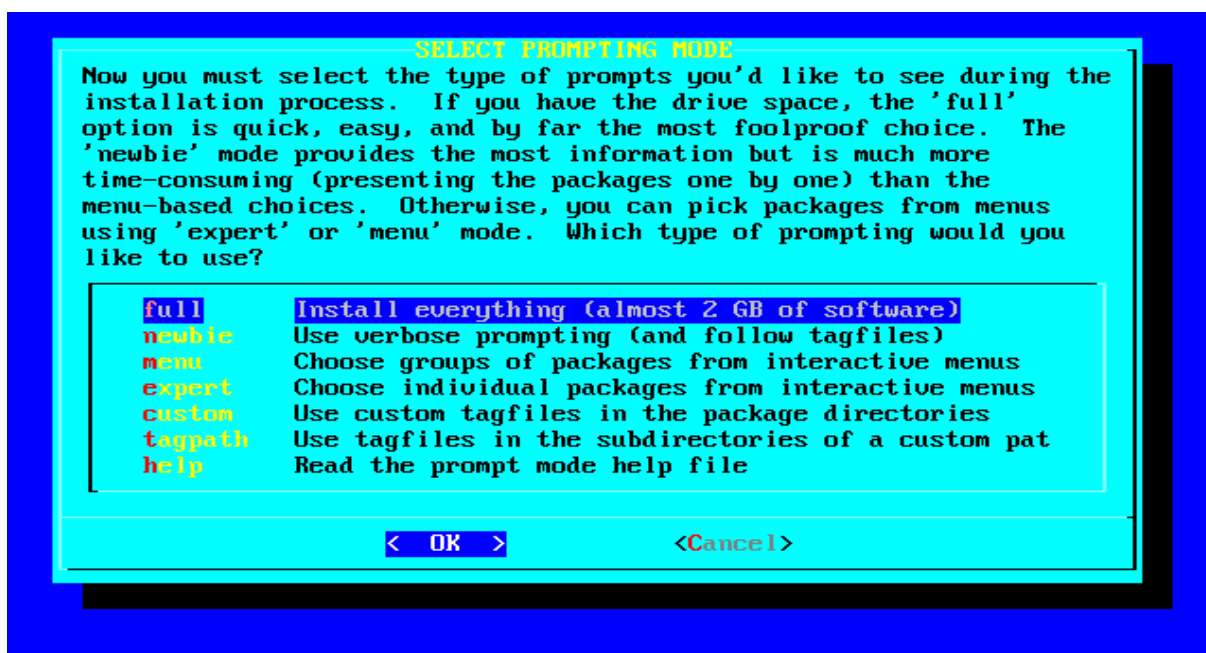
### Pakotņu sēriju izvēles ekrāns



### INSTALL (instalēšana)

Pieņemot, ka veiksmīgi izpildītas *TARGET*, *SOURCE* un *SELECT* sadaļas, šī sadaļa ļaus jums izvēlēties atsevišķas pakotnes no jūsu izvēlētajām pakotņu sērijām. Ja iepriekšējās sadaļas nav veiksmīgi izpildītas, *INSTALL* sadaļa izvadīs paziņojumu, liekot jums atgriezties pie iepriekšējo sadaļu izpildes. Šī sadaļa ļauj izvēlēties vienu no sešām instalēšanas metodēm: *full*, *newbie*, *menu*, *expert*, *custom* un *tag path*.

## Instalēšanas metodes izvēles ekrāns



Izvēloties metodi *full*, tiks instalētas visas pakotnes no izvēlētajām sērijām. Šī ir visvienkāršākā instalēšanas metode, jo nav jāpieņem detalizētus lēmumus par atsevišķām pakotnēm. Protams, šī metode prasīs arī visvairāk diska vietas.

Nākamā metode ir *newbie*. Šī metode uzstāda visas obligāti nepieciešamās katras sērijas pakotnes. Atlikušajām pakotnēm tiks piedāvāta izvēle, kurā iespējams izvēlēties starp *Yes*, *No*, vai *Skip*. *Yes* uzstādīs pakotni, *No* - izlaidīs, bet *Skip* pāries pie nākamās sērijas. Papildus tam ir iespējams iepazīties ar katras pakotnes aprakstu un apjomu - tas palīdzēs pieņemt lēmumu. Šo metodi mēs iesakām jauniem lietotājiem, jo tā nodrošina, ka tiek uzstādītas visas nepieciešamās pakotnes. Šīs metodes trūkums ir tās lēnā izpilde, jo par katru neobligāto pakotni ir jāpieņem lēmumu.

*menu* ir ātrāka, uzlabota *newbie* metodes versija. Katrai sērijai tiek attēlota izvēlne, kurā iespējams izvēlēties neobligātās pakotnes. Obligātās pakotnes netiek attēlotas.

Pieredzējušiem lietotājiem uzstādīšanas programma piedāvā metodi *expert*. Tā nodrošina pilnu kontroli pār instalētajām pakotnēm. Ir iespējams izvēlēties neinstalēt arī obligāti instalējamās pakotnes, tādējādi iegūstot nestrādājošu sistēmu. No otras puses, ir iespējams precīzi kontrolēt, kas tiek instalēts. Vienkārši jāizvēlas pakotnes no katras izvēlētas sērijas. Šī metode nav ieteicama jauniem lietotājiem, jo viegli iespējams kļūdīties.

Metodes *custom* un *tag path* arī ir piemērotas pieredzējušiem lietotājiem. Šīs metodes ļauj instalēt sistēmu pēc iepriekš sastādītiem birku failiem, kas izveidoti distribūcijas kokā. Šīs metodes ir ērtas, lai ātri uzstādītu sistēmu daudzos datoros. Vairāk informācijas par birku failiem un to lietojumu lūdzam skatīt sadaļā 18.4.

Tālākais atkarīgs no izvēlētas instalēšanas metodes. Ja izvēlēta *expert* vai *menu*, parādīsies izvēlne, kurā būs iespējams izvēlēties instalējamās pakotnes. Ja izvēlēts *full*, pakotnes tiks nekavējoties instalētas. Ja izvēlēts *newbie*, pakotņu instalēšana notiks līdz pirmajai neobligātajai pakotnei.

Lūdzam atcerēties, ka instalēšanas procesā iespējama diska pārpilde. Ja jūs izvēlējāties pārāk daudz pakotnes, kuru kopapjoms ir lielāks par cietā diska brīvās vietas apjomu, būs problēmas. Drošākā metode ir izvēlēties tikai nepieciešamo programmatūru un pārējo pievienot vēlāk, ja nepieciešams. To ir iespējams veikt, izmantojot *Slackware* pakotņu vadības rīkus. Vairāk informācijas meklējiet sadaļā 18.



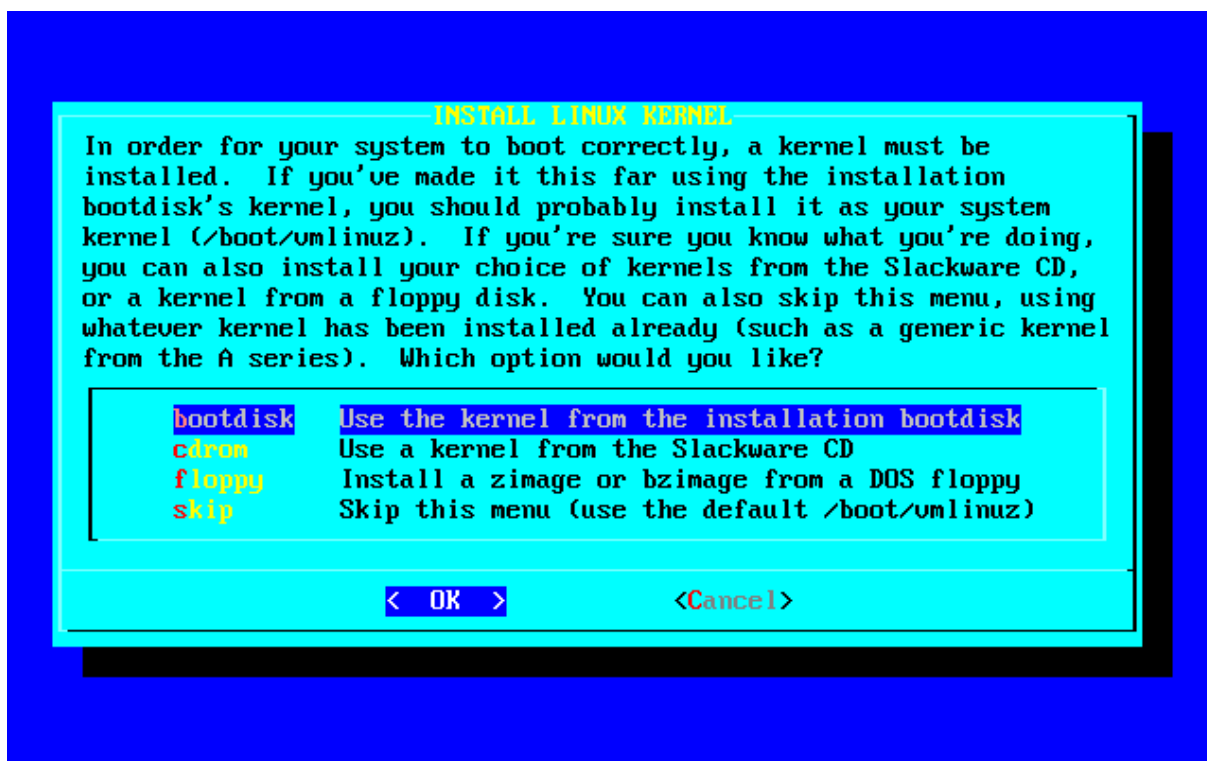
## CONFIGURE (konfigurācija)

Tā kā pakotnes ir uzstādītas, konfigurācijas sadaļā iespējams veikt sistēmas pamatkonfigurēšanu. Piedāvātās konfigurācijas iespējas ir lielā mērā atkarīgas no instalētajām pakotnēm, tomēr vienmēr būs pieejamas vismaz šādas:

### Kodola izvēle

Jums vienmēr būs jāizvēlas sistēmas kodolu. jūs varat uzstādīt kodolu no instalēšanai izmantotā ielādes diska, no Slackware CD-ROM, vai no iepriekš sagatavotas disketes. Ir iespējams izvēlēties izlaist šo posmu. Tādā gadījumā tiks uzstādīts noklusētais kodols.

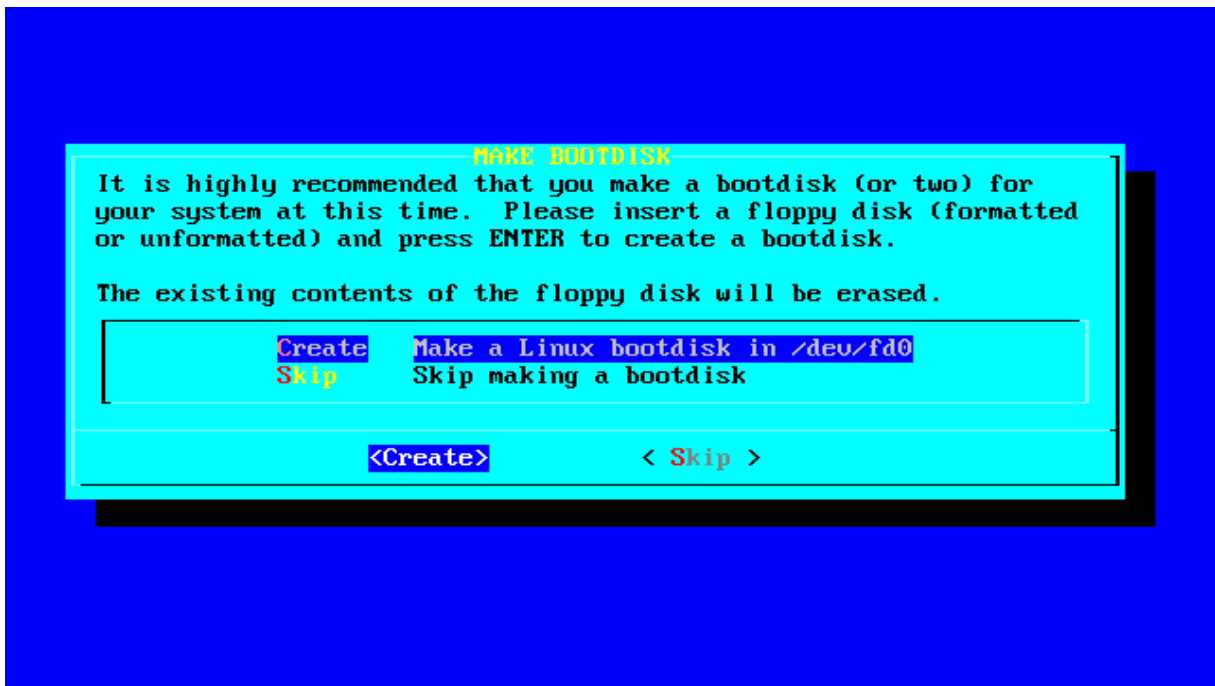
### Kodola izvēles ekrāns



### Ielādes diska izveide

Ielādes diska izveide, šķiet, ir laba doma gadījumam, ja nākotnē rodas problēmas ar sistēmas ielādi. Tiek piedāvāta iespēja noformatēt disketi un izveidot vienu no diviem ielādes disku tipiem. Pirmais, *simple* (vienkāršais - ej nu saziņi...), vienkārši ieraksta kodolu disketē. Uzlabots (un arī ieteicamais) variants ir *lilo*, kas izveidos LILO ielādes disku. Vairāk informācijas par *LILO* meklējiet sadaļā 7.1. Protams, ir iespējams arī izvēlēties punktu *continue* (turpināt) - tad ielādes disks netiks sagatavots.

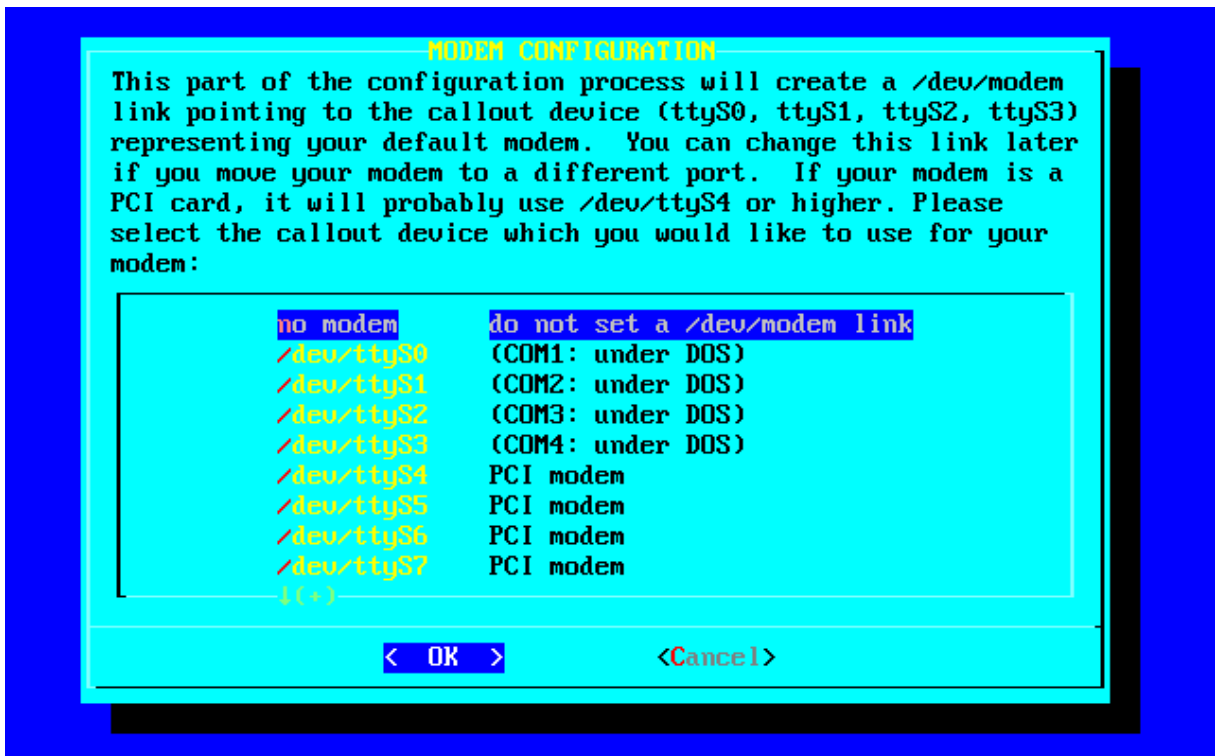
### Ielādes diska izveides ekrāns



### Modema konfigurācija

Jums tiks uzdoti jautājumi par modema konfigurāciju - vai jums vispār ir modems un, ja ir, tad pie kura jūsu datora seriālā porta tas ir pieslēgts.

### Modema konfigurācijas ekrāns



Nākamās konfigurācijas sadaļas var arī neparādīties - viss atkarīgs no tā, kādas pakotnes ir uzstādītas.

## Laika zona

Šeit viss ir samērā vienkārši - jums jānorāda savu laika joslu. Ja jūs atrodaties Zulu laika joslā, jums nav paveicies. Ļoti garais ģeogrāfisko vietu saraksts ir sakārtots alfabētiskā secībā, un Zulu ir saraksta pašās beigās...

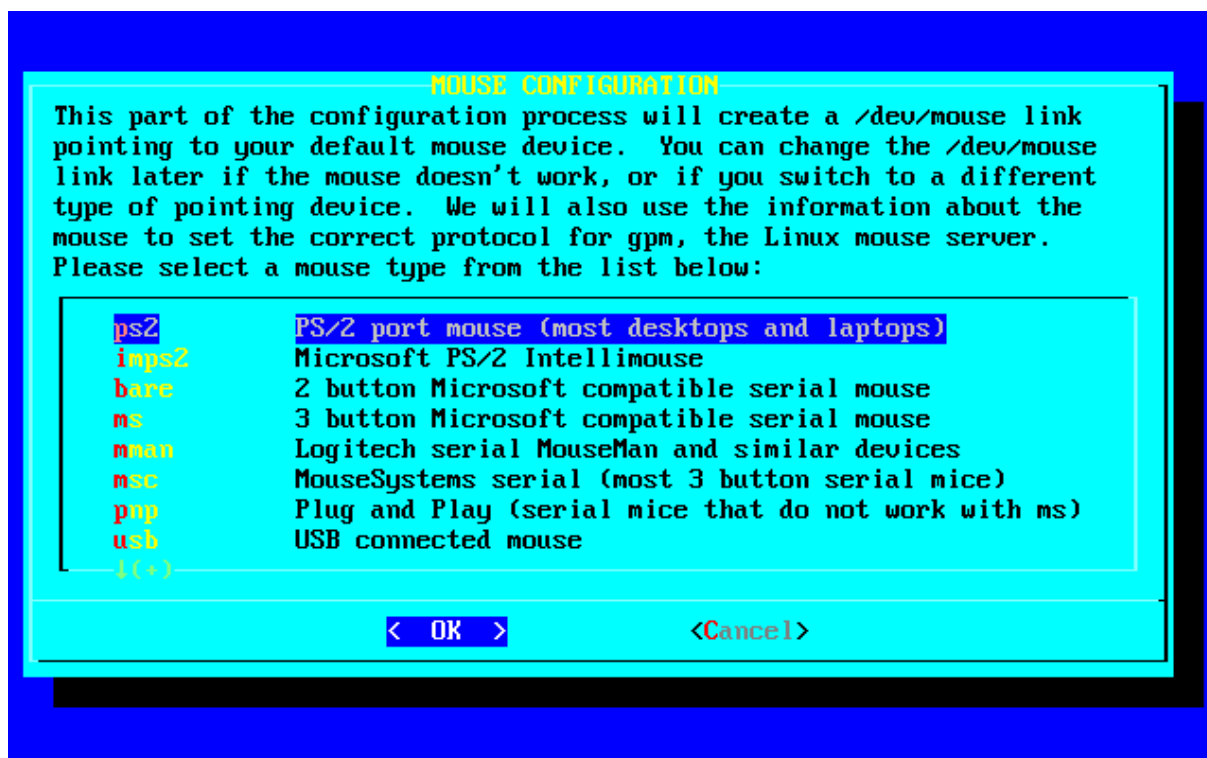
Laika zonas konfigurācijas ekrāns



## Peles konfigurācija

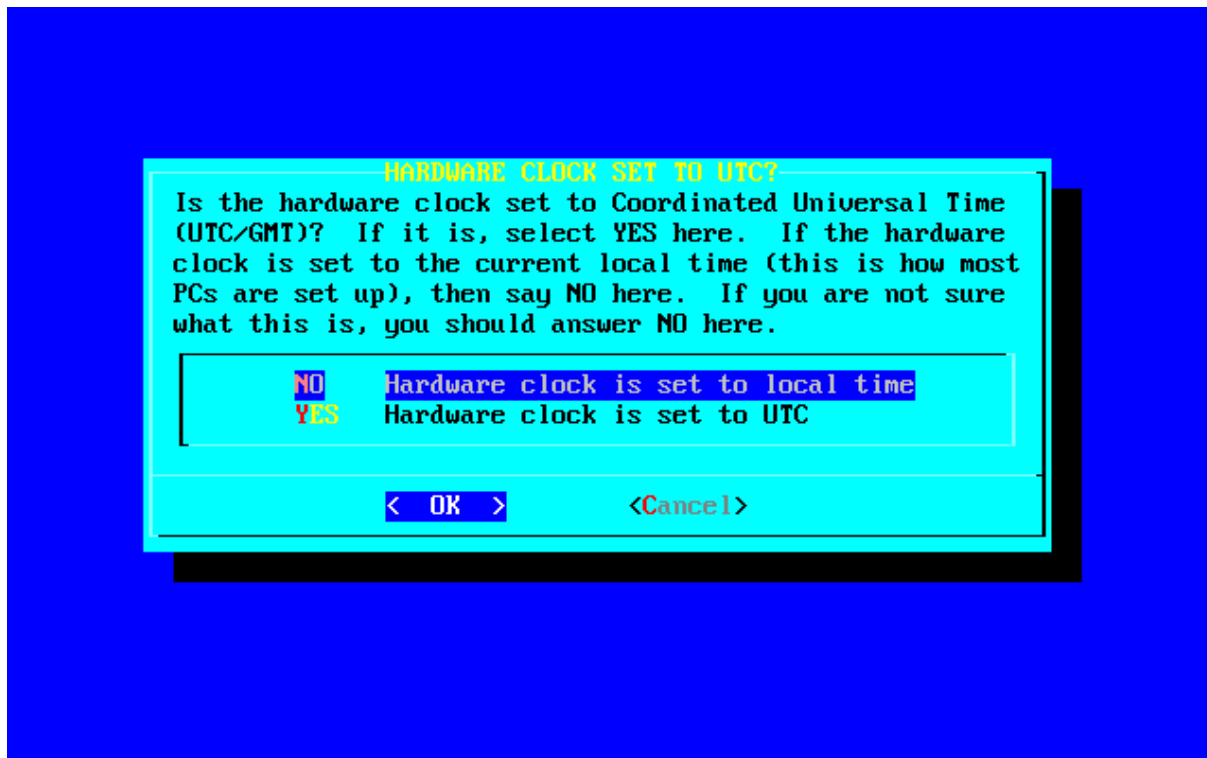
Šī sadaļa vienkārši pajautās jums, vai jūsu datoram ir pieslēgta pele un vai jūs vēlaties, lai, sistēmai startējoties, tiktu ielādēts arī `gpm` (8) - peles atbalsts konsolei.

### Peles konfigurācijas ekrāns



## Datora pulkstenis

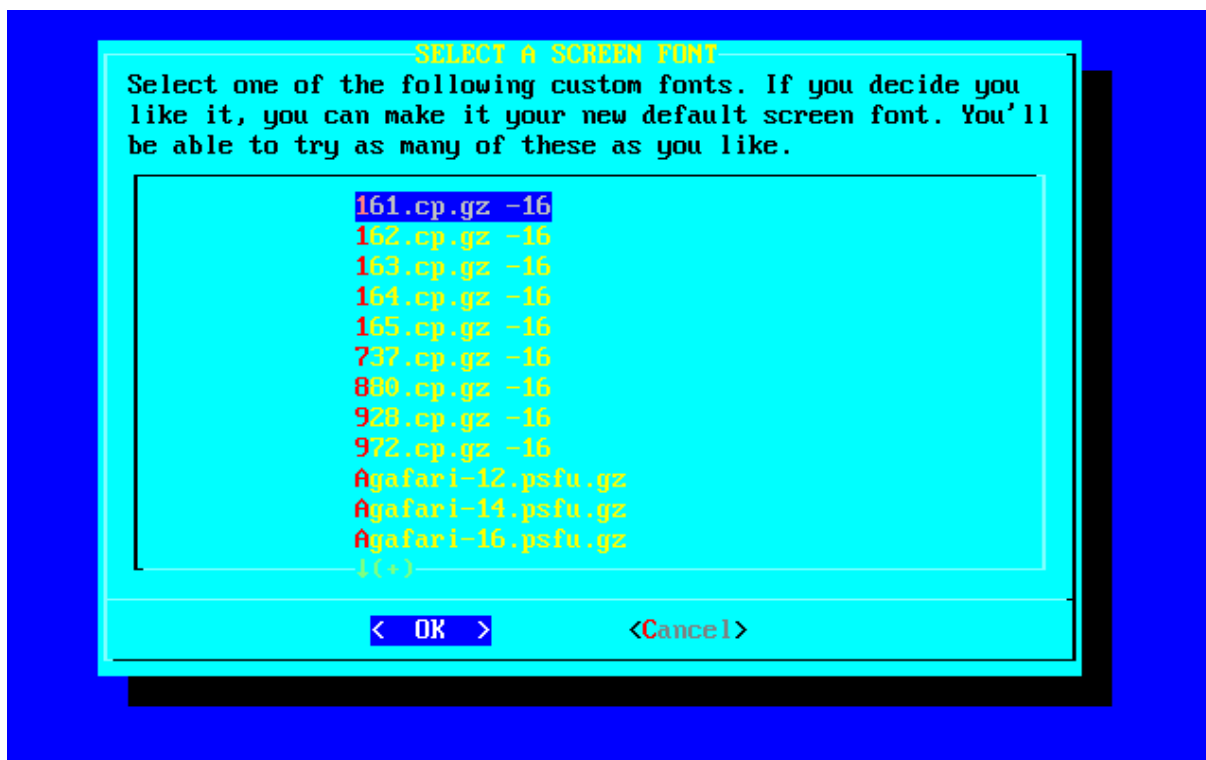
Šajā apakšsadaļā jums uzdos jautājumu, vai jūsu datora pulkstenis ir uzstādīts uz Koordinēto universālo laiku (UTC vai GMT). Lielākajai daļai datoru tā nav, tādēļ arī jūs, šķiet, varat atbildēt "NO" (nē).



## Šrifti (fonti)

Šajā sadaļā jūs varat izvēlēties jums patīkamāko konsoles teksta režīma šriftu.

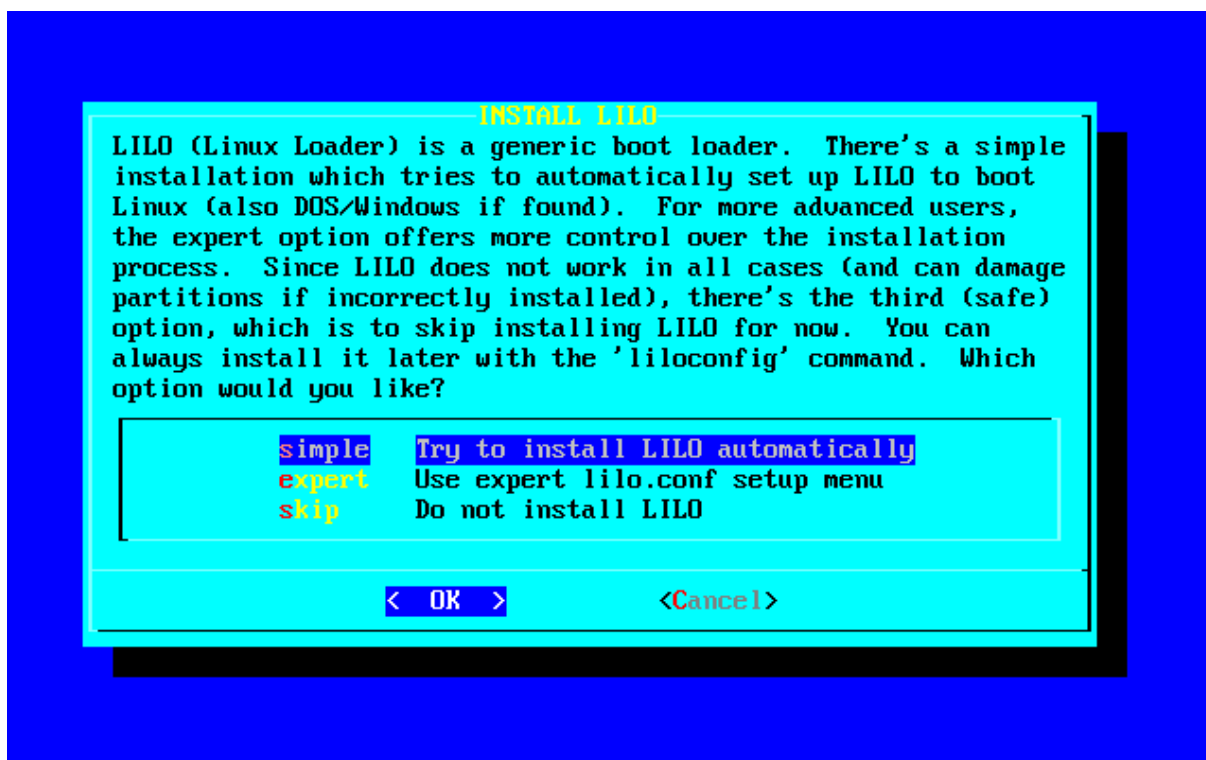
### Šriftu izvēles ekrāns



## LILO

Šajā sadaļā iespējams norādīt, kā sistēmā jāuzstāda *LILO* (Linux LOader - Linux ielādes programma; lūdzu sk. sadaļu 7.1).

### LILO instalēšanas ekrāns



Ja *Slackware* būs vienīgā datorā izmantotā operētājsistēma, instalēšanas metodei *simple* būtu lieliski jādarbojas. Ja jūs veidojat duālas ielādes sistēmu, ieteicams izmantot *expert* metodi - skatiet sadaļu 7.3 informācijai par duālo ielādi. Trešo iespēju - *skip* jeb izlaist - nebūtu jāizvēlas, ja nu vienīgi jūs patiešām zināt, ko darāt, un jums ir labs iemesls, kādēļ neuzstādīt LILO. Ja LILO tiek uzstādīts ar *expert* metodi, ir iespējams izvēlēties, kur instalēt LILO. LILO var uzstādīt cietā diska *MBR* (*Master Boot Record*), *Linux* saknes sadaļas superblokā vai disketē.

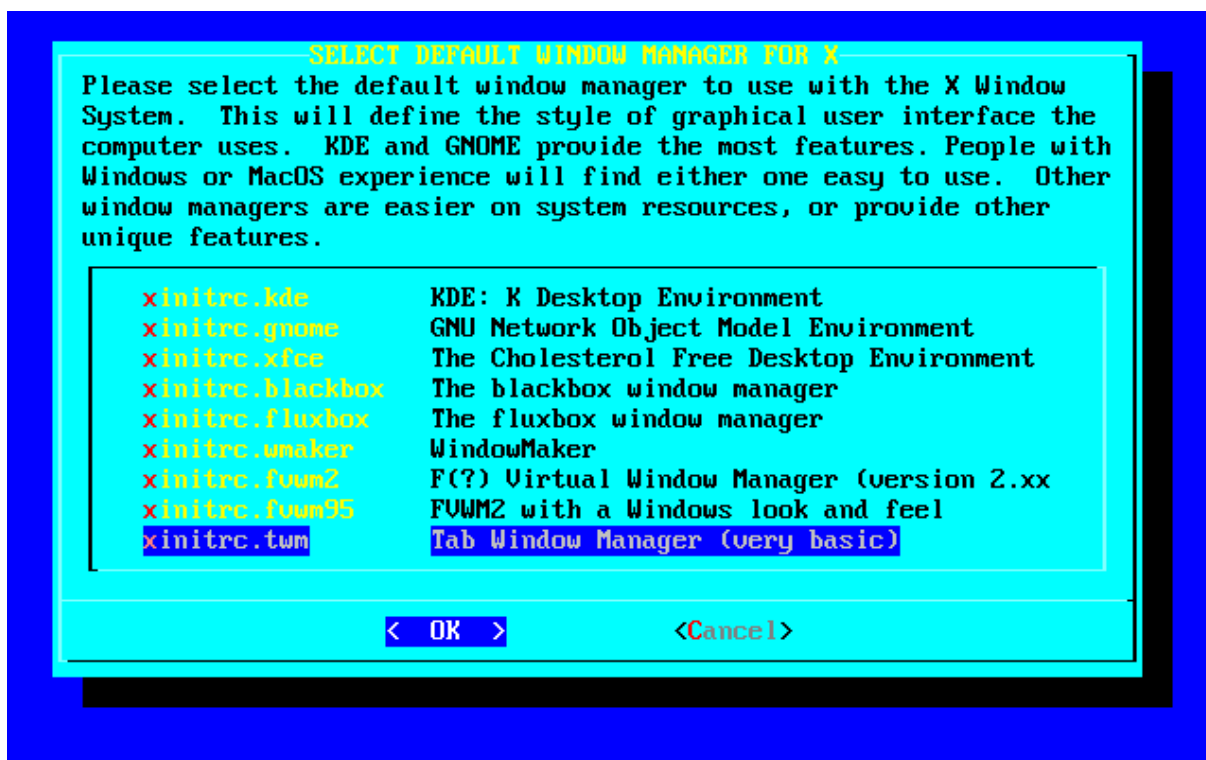
## Tīkls

Tīkla konfigurācijas sadaļa patiesībā ir *netconfig* programma. Lūdzu sk. sadaļu 5.1, lai atrastu vairāk informācijas.

## X logu pārvaldnieks

Šī sadaļa ļaus jums izvēlēties noklusēto X logu pārvaldnieku. Vairāk informācija par X logu sistēmu un logu pārvaldniekiem atrodama sadaļā 6.

### Noklusētā logu pārvaldnieka izvēles ekrāns



Neatkarīgi no tā, kādas pakotnes ir instalētas, pēdējais, ko konfigurācijas sadaļa piedāvās, ir `root` paroles uzstādīšana. Drošības nolūkiem to ir vēlams veikt, tomēr, līdzīgi kā viss, ko var veikt ar *Slackware Linux*, arī šī darbība ir jūsu ziņā.



## 4.nodaļa. Sistēmas konfigurēšana

Pirms konfigurēt sarežģītākās sistēmas daļas, ir vērtīgi iepazīties ar sistēmas organizāciju un to, kādas komandas iespējams izmantot failu un programmu meklēšanai. Nav slikti arī uzzināt, kā nokompilēt *Linux* kodolu un ko jādara, lai to paveiktu. Šī nodaļa ļaus jums iepazīties ar *Slackware Linux* konfigurēšanu un konfigurācijas failiem. Pēc tam var konfigurēt sarežģītākās sistēmas sastāvdaļas.

### 4.1. Sistēmas apskats

Pirms mēģināt izprast dažādās konfigurēšanas nianšes, ir svarīgi saprast, no kā sastāv *Linux* sistēma. *Linux* sistēma būtiski atšķiras no *DOS*, *Windows* vai *Macintosh* sistēmām (izņemot, programs, uz *Unix* balstīto *MacOS X*), tomēr šī sadaļa iepazīstinās jūs ar *Linux* uzbūvi, lai pēcāk jums būtu iespējams viegli konfigurēt un pielāgot savu sistēmu.

### Failsistēmas izkārtojums

Pirmā pamanāmā atšķirība starp *Slackware Linux* un *DOS* vai *Windows* sistēmu ir failsistēma. Iesācējiem - dažādu sadaļu un iekārtu apzīmēšanai netiek izmantoti iekārtas apzīmējoši burti. *Linux* sistēmā ir tikai viena galvenā (saknes) direktorijs. Varam pieņemt, ka tā ir kas līdzīgs *c:* diskam *DOS* vai *Windows* sistēmā. Katra sistēmas sadaļa tiek **piemontēta** kādā no galvenās direktorijas apakšdirektorijām - kaut kas līdzīgs (gandrīz) bezgalīgi paplašināmam cietajam diskam.

Galveno direktoriju sauc par saknes (`root`) direktoriju, un to apzīmē ar slīpsvītru (`/`). Šī ideja var šķist dīvaina, bet patiesībā tā atvieglo dzīvi, ja vēlamies paplašināt pieejamo diska vietu. Pieņemsim, ka mums beigusies diska vieta iekārtā, kurā atrodas direktorijs `/home`. Lielākā daļa instalē *Slackware* ar vienu lielu saknes disku. Tātad, tā kā sadaļu ir iespējams piemontēt jebkurā direktorijā, tad varam vienkārši doties uz veikalu, iegādāties jaunu cieto disku un to piemontēt direktorijā `/home`. Esam ieguvuši papildus diska vietu, pie tam nenodarbojoties ar datu pārvietošanu.

Zemāk atrodami *Slackware* failsistēmas galveno direktoriju apraksti.

#### `bin`

Šeit tiek glabātas svarīgākās lietotāja programmas - minimālais komplekts, kas nepieciešams sistēmas lietošanai. Piemēram, šajā direktorijā atrodas tādas programmas, kā komandrinda un failsistēmas komandas (`ls`, `cp` utt.). `/bin` direktorijs pēc sistēmas instalēšanas parasti vairs netiek mainīts un, ja arī tiek, tad to veic mūsu sagatavotās sistēmas atjauninājuma pakotnes.

#### `boot`

*Linux Loader* (LILO) izmantotie faili. Arī šī direktorijs pēc sistēmas instalēšanas tiek mainīts minimāli. Kopš *Slackware 8.1* versijas šajā direktorijā tiek glabāts arī sistēmas kodols. Iepriekš kodols tika glabāts `/`, tomēr mūsdienās kodols un ar to saistītie faili tiek glabāti šajā direktorijā, lai vienkāršotu duālo ielādi.

#### `dev`

*Linux* sistēmā viss tiek uzskatīts par failiem, pat aparatūra, kā seriālie porti, cietie diski un skaneri. Lai piekļūtu šīm iekārtām, nepieciešams īpašs fails, ko sauc par **iekārtas mezglu**. Visi iekārtu mezgli atrodas direktorijā `/dev`, pie tam ne tikai *Linux*, bet arī daudzās citās *Unix* tipa operētājsistēmās.

#### `etc`

Šī direktorijs satur sistēmas konfigurācijas failus - sākot ar *X* logu sistēmas konfigurācijas failiem un lietotāju datu bāzi, un beidzot ar sistēmas startēšanas skriptiem. Sistēmas administrators ar laiku kļūst visnotaļ zinošs par šo direktoriju.

#### `home`

*Linux* ir daudzlietotāju operētājsistēma. Katram sistēmas lietotājam tiek piešķirts lietotāja konts un

direktorija personīgajiem failiem. Šo direktoriju sauc par lietotāja **mājas** (*home*) direktoriju. `/home` direktorija ir noklusētā vieta lietotāju mājas direktorijām.

#### **lib**

Šeit tiek glabātas sistēmas pamatdarba nodrošināšanai nepieciešamās sistēmas bibliotēkas. Starp citu, šeit tiek glabāta arī C bibliotēka, dinamiskais ielādētājs, `ncurses` bibliotēka un kodola moduļi.

#### **mnt**

Šī direktorija satur montēšanas punktus darbam ar papildus cietajiem diskkiem vai noņemamajām ārējās atmiņas iekārtām - arī CD-ROM un diskešu iekārtu.

#### **opt**

Papildus programmatūras pakotnes. Galvenā `/opt` ideja ir, ka katra programmatūras pakotne tiek instalēta direktorijā `/opt/pakotnes-nosaukums`, kas vēlāk atvieglo tās dzēšanu. Arī dažas *Slackware* pakotnes tiek izvietotas šajā direktorijā (piemēram KDE direktorijā `/opt/kde`), tomēr jūs varat šai direktorijai brīvi pievienot jebkuru programmatūru.

#### **proc**

Šī ir īpaša direktorija. Patiesībā tā nav pat īsti failsistēmas daļa, bet virtuāla failsistēma, kas nodrošina piekļuvi kodola informācijai. Izmantojot `/proc` direktorijas failus, kodols ļauj jums aplūkot dažādu informāciju. Bez tam, izmantojot dažus no šiem failiem, ir iespējams arī nosūtīt informāciju kodolam. Piemēram, pamēģiniet izpildīt komandu `cat /proc/cpuinfo`.

#### **root**

*Linux* sistēmās administrators ir pazīstams kā `root`. `root` mājas direktorija atrodas `/root` direktorijā, nevis `/home/root`. Iemesls ir ļoti vienkāršs. Kas notiktu, ja `/home` būtu izvietota atsevišķā sadaļā un to nebūtu iespējams piemontēt? `root` lietotājs noteikti gribētu autorizēties sistēmā, lai novērstu problēmu. Ja viņa mājas direktorija atrastos bojātā failsistēmā, būtu problēmas ar autorizēšanos.

#### **sbin**

Šajā direktorijā atrodas sistēmas startēšanās laikā izmantotās svarīgās programmas. Tās izpilda `root` lietotājs. Pārējie lietotāji diez vai izpildīs programmas, kas atrodas šajā direktorijā.

#### **tmp**

Pagaidu datu glabāšanas vieta. Šajā direktorijā var lasīt un rakstīt visi lietotāji.

#### **usr**

Šī ir *Linux* sistēmas lielā direktorija. Šajā direktorijā principā atrodas lielākā daļa no visa, kas uzstādīts sistēmā - programmas, dokumentācija, kodola pirmkods un X logu sistēma. Visticamākais, ka tieši šajā direktorijā jūs uzstādīsiet jaunas programmas.

#### **var**

Šeit atrodas sistēmas žurnālfaili, kešdati un programmu atslēgfaili. Šī ir direktorija datiem, kas bieži mainās.

Tagad jums būtu jābūt nojausmai, kādu informāciju satur katra failsistēmas direktorija. Detalizēta informācija par failsistēmu ir pieejama `hier(7)` man dokumentā. Nākamajā sadaļā jūs atradīsiet informāciju par to, kā vienkāršā veidā atrast specifiskus failus, lai to nebūtu jādara manuāli.

## **Failu meklēšana**

Lai gan šobrīd jūs zināt, ko satur katra no galvenajām direktorijām, tas tomēr maz palīdz, lai atrastu konkrētas lietas. Es domāju - protams, lai atrastu failu, ir iespējams aplūkot katru direktoriju, tomēr eksistē arī citi veidi. Šeit aplūkotas četras galvenās *Slackware Linux* pieejamās failu meklēšanas komandas.

## which

Pirmā ir komanda **which**(1). **which** parasti izmanto, lai ātri atrastu programmu. Tā vienkārši aplūko jūsu **PATH** norādītās direktorijas un izvada pirmo atrasto rezultātu un ceļu uz to. Piemēram:

```
% which bash
/bin/bash
```

Redzam, ka **bash** atrodas **/bin** direktorijā. Šīs komandas funkcionalitāte ir ļoti ierobežota, jo tā meklē informāciju tikai tajās direktorijās, kas norādītas **PATH**.

## whereis

**whereis**(1) komanda darbojas līdzīgi, kā **which**, tomēr tā ļauj meklēt arī *man* dokumentus un pirmkoda failus. Liekot **whereis** meklēt **bash**, rezultātam būtu jābūt:

```
% whereis bash
bash: /bin/bash /usr/bin/bash /usr/man/man1/bash.1.gz
```

Šī komanda mums norādīja ne tikai to, kurā direktorijā izvietota atbilstošā programma, bet arī to, kur tiek glabāta tiešsaistes dokumentācija. Neraugoties uz to, šī komanda arī ir ierobežota. Ko darīt, ja jūs vēlējaties meklēt specifisku konfigurācijas failu? **which** un **whereis** tam nevar izmantot.

## find

**find**(1) komanda ļauj lietotājam meklēt informāciju failsistēmā, piedāvājot bagātīgu meklēšanas nosacījumu kopumu. Lietotāji var norādīt meklējamo informāciju ar failu nosaukumu aizstājējzīmēm, failu izmaiņu vai radīšanas laiku diapazoniem, kā arī citām papildus īpašībām. Piemēram, lai meklētu sistēmas noklusēto **xinitrc** failu, var izmantot šādu komandu:

```
find / -name xinitrc
/var/X11R6/lib/xinit/xinitrc
```

Tiesa, **find** meklēšana var prasīt samērā daudz laika, jo tā aplūko katru apakšdirektoriju visā direktoriju kokā. Bez tam, ja šo komandu izpilda ar parasta lietotāja tiesībām, tā izvadīs lietotāja tiesību kļūdu paziņojumus tām direktorijām, kuras var aplūkot tikai **root** lietotājs. Tomēr **find** atrada mūsu failu, tādēļ ir jau labi. Ja nu vienīgi tas varētu darboties nedaudz ātrāk...

## slocate

**slocate**(1) komanda meklē informāciju visā failsistēmā, līdzīgi, kā **find** komanda, taču izmantojot datu bāzi, nevis failsistēmu. Datu bāze tiek atjaunināta katru rītu, tādēļ jums vienmēr būs vairāk vai mazāk svaigs failu saraksts. Jūs varat arī manuāli izpildīt **updatedb**(1), lai atjauninātu **slocate** datu bāzi (pirms izpildīt **updatedb** manuāli, vispirms jāizpilda komanda **su**, lai nomainītu lietotāja tiesības uz **root**). Piemērs **slocate** izmantošanai:

```
% slocate xinitrc # mums nav jābūt root
/var/X11R6/lib/xinit/xinitrc
/var/X11R6/lib/xinit/xinitrc.fvwm2
/var/X11R6/lib/xinit/xinitrc.openwin
/var/X11R6/lib/xinit/xinitrc.twm
```

Mēs atradām pat vairāk, kā meklējām, pie tam ļoti ātri. Ar šīm komandām jums savā *Linux* sistēmā būtu jāvar atrast jebko.

## /etc/rc.d direktorija

Direktorijā **/etc/rc.d** tiek glabāti sistēmas inicializācijas faili. *Slackware* izmanto *BSD* stila failu izvietojumu, atšķirībā no citiem *Linux* distribuīviem, kas izmanto *System V* inicializācijas skriptus, kuri padara konfigurācijas izmaiņu veikšanu daudz sarežģītāku bez īpašu programmu izmantošanas. *BSD* inicializācijas skriptos katram sistēmas darblīmenim (*runlevel*) ir izveidots viens **rc** fails, atšķirībā no *System V*, kur katram darblīmenim ir izveidota atsevišķa direktorija ar daudziem inicializācijas

skriptiem. Tā tiek iegūta organizēta, viegli uzturama struktūra.

Slackware inicializācijas faili ir sadalīti vairākās kategorijās - sistēmas startēšanas, darblīmeņu, tīkla inicializācijas un *System V* savietojamības skriptos. Tradicionāli visu pārējo ieskaitīsim citā kategorijā.

## Sistēmas startēšana

Pirmā programma, kas tiek startēta Slackware sistēmā bez Linux kodola, ir `init(8)`. Šī programma nolasa failu `/etc/inittab(5)` un pēc tā nosaka, kā jādarbina sistēma. Tad `init` izpilda `/etc/rc.d/rc.S` skriptu, kas sagatavo sistēmu pārslēgšanai uz izvēlēto darblīmeni. `rc.S` norādītās darbības veic virtuālās atmiņas (maiņvietas) aktivizēšanu, piemontē failsistēmas, attīra noteiktas žurnālfailu direktorijas, inicializē *Plug and Play* iekārtas, ielādē kodola moduļus, nokonfigurē *PCMCIA* iekārtas un seriālos portus un izpilda *System V* inicializācijas skriptus (ja tādi ir). Acīmredzami `rc.S` ir apjomīgs skripts, tomēr lūk, daži `/etc/rc.d` skripti, ko `rc.S` izsauc, lai izpildītu savu uzdevumu:

`rc.S`

Šis ir sistēmas inicializācijas skripts.

`rc.modules`

Ielādē kodola moduļus, piemēram, tīkla kartes, *PPP* u.c. atbalstu. Ja šis skripts atrod `rc.netdevice` skriptu, tad arī tas tiek izpildīts.

`rc.pcmcia`

Meklē un konfigurē *PCMCIA* iekārtas, kuras varētu būt pieslēgtas jūsu datoram. Šī iespēja ir noderīga klēpj datoru īpašniekiem, kuriem, iespējams, ir *PCMCIA* modems vai tīkla karte.

`rc.serial`

Konfigurē seriālos portus, izpildot atbilstošās `setserial` komandas.

`rc.sysvinit`

Meklē atbilstošā darblīmeņa *System V* inicializācijas skriptus un izpilda tos. Šī iespēja ir aplūkota zemāk.

## Darblīmeņu inicializācijas skripti

Kad sistēmas inicializācija ir pabeigta, `init` pāriet pie darblīmeņu inicializācijas. Darblīmenis nosaka režīmu, kādā darbosies jūsu dators. Izklusās lieki? Nepavisam. Darblīmenis nosaka to, vai jūsu dators atbalstīs daudzlietotāju režīmu vai nē, vai nepieciešams iedarbināt tīkla servisu, kā arī to, vai jūs izmantosiet X logu sistēmu vai `agetty(8)` autorizācijas nodrošināšanai. Zemāk aprakstītie faili nosaka dažādos Slackware Linux darblīmeņus.

`rc.0`

Apturēt sistēmu (darblīmenis 0). Pēc noklusējuma šī ir simbolsaite uz `rc.6`.

`rc.4`

Daudzlietotāju startēšana (darblīmenis 4) X logu sistēmā ar KDM, GDM vai XDM kā autorizācijas pārvaldnieku.

`rc.6`

Pārstartēt sistēmu (darblīmenis 6).

`rc.K`

Vienlietotāja režīms (darblīmenis 1).

`rc.M`

Daudzlietotāju režīms (darblīmeņi 2 un 3), tomēr ar standarta teksta režīma autorizāciju. *Slackware* sistēmā šis ir noklusētais darblīmenis.

### Tīkla inicializācija

Darblīmeņi 2, 3 un 4 iedarbinās tīkla servisu. Par tīkla inicializāciju atbild šādi faili:

`rc.inet1`

Šo failu izveido `netconfig` komanda, un tas veic tīkla interfeisa konfigurēšanu.

`rc.inet2`

Tiek izpildīts pēc `rc.inet1`. Startē tīkla pamatservisu.

`rc.atalk`

Startē AppleTalk servisu.

`rc.httpd`

Startē *Apache* tīmekļa serveri. Līdzīgi, kā dažus citus `rc` skriptus, arī šo var izmantot servisa apturēšanai vai pārstartēšanai. `rc.httpd` atbalsta argumentus `stop` (apturēt), `start` (startēt) un `restart` (pārstartēt).

`rc.news`

Startē jaunumu vēstkopu serveri.

### Savietojamība ar System V

Savietojamība ar System V `init` sistēmu tika ieviesta *Slackware 7.0* versijā. Šo sistēmu *BSD* sistēmas vietā izmanto daudzi citi *Linux* distribuīvi. Šajā sistēmā jebkuram darblīmenim atbilst vesela direktorija ar `init` skriptiem, kamēr *BSD* sistēmā katram darblīmenim atbilst viens `init` skripts.

`rc.sysvinit` skripts meklē visus *System V* inicializācijas skriptus, kuri atrodas `/etc/rc.d` direktorijā un izpilda tos, ja tie atbilst darblīmenim. Tas noder, izmantojot atsevišķas komerciālas programmatūras pakotnes, kas instalē *System V* inicializācijas skriptus.

### Citi faili

Zemāk aprakstītie faili ir "pārējie" sistēmas inicializācijas skripti. Pārsvārā tie tiek izpildīti no kāda augstākminētā skripta, tādēļ atliek tikai tos atbilstoši rediģēt.

`rc.gpm`

Startē vispārējas nozīmes peles servisu. Tie ļauj jums kopēt un ielīmēt informāciju *Linux* konsolē. Reizēm, lietojot `gpm` X logu sistēmā, tas var radīt problēmas ar peli. Ja šādas problēmas rodas, šim failam jānoņem izpildes tiesības un jāaptur `gpm` serveri.

`rc.font`

Ielādē pielāgotus konsoles šriftus (fontus).

`rc.local`

Satur jebkuras specifiskas sistēmas startēšanas komandas. Tūlīt pēc instalēšanas šis fails ir tukšs un rezervēts sistēmas administratora vajadzībām. Šis skripts tiek izpildīts pēc tam, kad izpildījušies visi citi inicializācijas skripti.

Lai atļautu skripta izpildi, viss, ko jāizdara, ir jāpievieno tam izpildes tiesības, izmantojot `chmod` komandu. Lai aizliegtu skripta izpildi, izpildes tiesības jānoņem. Vairāk informācijas par `chmod` komandu skatiet sadaļā 9.2.

## 4.2. Kodola izvēle

Kodols ir operētājsistēmas daļa, kas nodrošina piekļuvi pie aparatūras, procesu vadību un vispārēju

sistēmas kontroli. Kodols satur atbalstu jūsu aparatūrai, tādēļ pareiza kodola izvēle ir svarīgs sistēmas instalēšanas procesa solis.

Slackware piedāvā vairāk kā duci jau nokompilētu kodolu, no kuriem iespējams izvēlēties piemērotāko. Katram kodolam ir piekompilēts standarta dziņu un papilddziņu komplekts. Jebkurā gadījumā jāpārlicinās, ka jūsu kodols atbalsta jūsu aparatūru.

## /kernels direktorija Slackware CD-ROM diskā

Kompilētie *Slackware Linux* kodoli ir atrodami *Slackware* CD-ROM vai FTP servera galvenās *Slackware* direktorijas `/kernels` apakšdirektorijā. Pieejamie kodoli mainās ar jaunu versiju iznākšanu, tādēļ šajā direktorijā iekļautā dokumentācija vienmēr ir primārais informācijas avots. `/kernels` direktorijā ir izveidotas apakšdirektorijas katram pieejamajam kodolam. Apakšdirektorijām ir tādi pat nosaukumi, kā atbilstošajam ielādes diskam. Katrā apakšdirektorijā izvietoti šādi faili:

Fails	Nozīme
System.map	Kodola sistēmas karte
bzImage	Kodola attēlfails
config	Kodola pirmkoda konfigurācijas fails

Lai izmantotu kodolu, nokopējiet failus `System.map` un `config` jūsu `/boot` direktorijā, kā arī failu `bzImage` uz `/boot/vmlinuz`. Tad izpildiet `/sbin/lilo(8)`, lai instalētu LILO ar informāciju par jauno kodolu. Pēc tam pārstartējiet sistēmu. Būtībā tas ir viss, ko jāveic jauna kodola instalēšanai.

Kodoli, kuru nosaukumi beidzas ar `.i`, ir IDE kodoli - bez papildmoduļiem tie neuztur SCSI. Kodoli, kuru nosaukumi beidzas ar `.s`, ir SCSI kodoli - tie satur gan IDE, gan SCSI atbalstu.

## Kodola kompilēšana no pirmkoda

Jauni lietotāji bieži uzdod jautājumu: "Vai man būtu jākompilē savas sistēmas kodolu pašam?". Atbilde noteikti ir "varbūt". Ir situācijas, kad ir nepieciešams kompilēt kodolu specifiskai sistēmai, tomēr lielākā daļa lietotāju pilnvērtīgai aparatūras darbināšanai var izmantot kādu no jau nokompilētajiem kodoliem un ielādējamos kodola moduļus. Jums būs jākompilē kodolu, ja vēlēsities atjaunināt kodola versiju uz tādu, kas netiek piedāvāta *Slackware* komplektācijā, vai, ja esat papildinājis kodola pirmkodu, lai atbalstītu kādu iekārtu, kuras atbalsts nav iekļauts standarta kodola pirmkodā. Kodolu kompilēt vēlēsities arī jebkurš, kurš izmanto datoru ar SMP (vairākiem procesoriem - tulk. piez.). Daudzi lietotāji arī pamanīs, ka paškompilēts kodols darbojas daudz ātrāk, jo, kompilējot kodolu, to ir iespējams optimizēt tieši jūsu datora procesoram.

Kodola kompilēšana nemaz nav tik sarežģīta. Pirmais, ko jāveic - jāpārlicinās, ka jūsu datorā ir instalēts kodola pirmkods - pakotnes no `κ` sērijas. Tāpat jāpārlicinās, ka ir instalētas `D` sērijas pakotnes - C kompilators, *GNU make* un *GNU binutils*. Patiesībā, par sliktu nenāktu instalēt visu `D` sēriju, ja tiek plānots izstrādāt vai kompilēt programmatūru. Ir iespējams arī lejupielādēt jaunāko kodola pirmkodu no <http://www.kernel.org/mirrors>.

## Linux kodola versijas 2.4.x kompilēšana

```
% su -  
Password:  
# cd /usr/src/linux
```

Pirmais solis ir kodolu pārveidot tā sākumstadijā. Šim nolūkam jāievada komandu (ņemiet vērā, ka pirms šīs darbības jūs varētu vēlēties saglabāt `.config` failu, ko šī komanda to izdzēsīs bez brīdinājuma):

```
# make mrproper
```

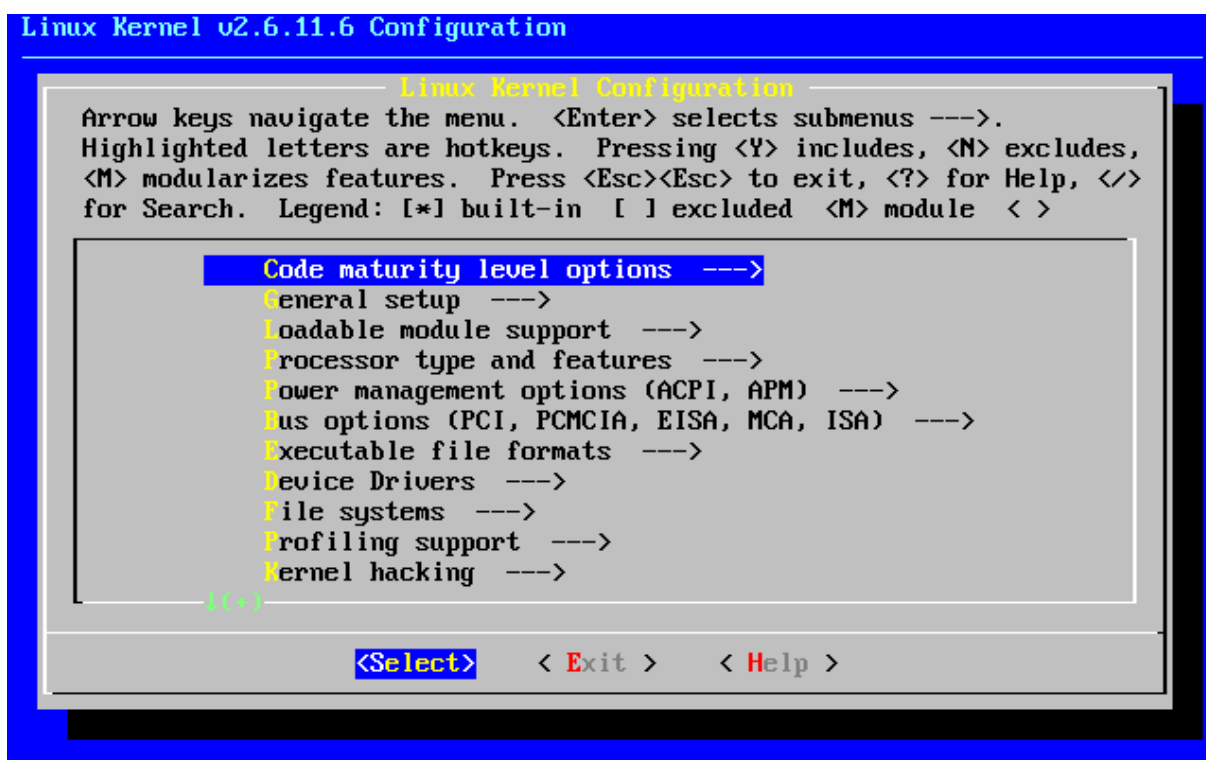
Tagad ir iespējams konfigurēt kodolu jūsu sistēmai. Šobrīd kodols piedāvā trīs veidus, kā to veikt. Pirmais veids ir sākotnējā teksta režīmā jautājumu un atbilžu sistēma. Tā uzdod daudz jautājumus un no tiem izveido konfigurācijas failu. Šīs metodes trūkums ir tāds, ka, ja jūs kaut ko sajaucat un ievadat nepareizi, visu procesu jāsāk no sākuma, tādēļ lielākā daļa dod priekšroku uz izvēlnēm balstītajai sistēmai. Visbeidzot, eksistē arī kodola konfigurācijas rīks X logu videi. Izvēlieties, kuru vēlaties izmantot, un ievadiet atbilstošu komandu:

```
# make config (teksta režīma jautājumu un atbilžu sistēma)
```

```
# make menuconfig (teksta režīma izvēlnu sistēma)
```

```
# make xconfig (X logu sistēma - šo iespējams darbināt tikai strādājošā X logu vidē)
```

## Linux kodola konfigurācijas rīka ekrāns



Jauniem lietotājiem, ticamākais, visvienkāršākā liksies `menuconfig` konfigurēšanas sistēma. Šajā rīkā katrai kodola konfigurācijas daļai ir izveidoti atbilstoši palīdzības ekrāni. Kad kodola konfigurēšana pabeigta, aizveriet konfigurācijas programmu. Tā sagatavos nepieciešamos konfigurācijas failus. Tagad mēs varam sagatavot pirmkoda koku kompilēšanai:

```
# make dep
```

```
# make clean
```

Nākamais solis ir kodola kompilēšana. Vispirms mēģiniet ievadīt komandu

```
# make bzImage
```

Šis solis var prasīt zināmu laiku - viss atkarīgs no jūsu datora procesora ātruma. Kompilēšanas procesa laikā uz ekrāna tiks izvadīti visi kompilatora paziņojumi. Kad kodola attēls ir nokompilēts, jūs varētu vēlēties kompilēt arī tās kodola daļas, kuras konfigurācijas programmā jūs atzīmējāt kā modulāras:

```
# make modules
```

Kad šis process ir beidzies, mēs varam instalēt kodolu un visus nokompilētos moduļus. Lai instalētu

kodolu Slackware sistēmā, jāievada šādas komandas:

```
# mv /boot/vmlinuz /boot/vmlinuz.old
# cat arch/i386/boot/bzImage > /boot/vmlinuz
# mv /boot/System.map /boot/System.map.old
# cp System.map /boot/System.map
# make modules_install
```

Noteikti veiciet izmaiņas arī `/etc/lilo.conf` failā un pievienojiet tajā sadaļu, kas ļaus jums ielādēt veco kodolu, ja jaunais nedarbojas. Pēc tam izpildiet `/sbin/lilo`, lai instalētu jaunu ielādes sektoru. Tad pārstartējiet datoru ar jauno kodolu.

### Linux kodola versijas 2.6.x kompilēšana

2.6 kodola kompilēšana tikai nedaudz atšķiras no 2.4 vai 2.2 kodola kompilēšanas, tomēr ir svarīgi šīs atšķirības izprast. Vairs nav nepieciešams izpildīt `make dep` un `make clean`. Arī kodola kompilēšanas procesā vairs netiek izvadīti visi kompilatora paziņojumi. Tas nozīmē, ka kompilēšanas process ir vienkāršāk izprotams, tomēr tam ir daži trūkumi. Ja jums rodas problēmas ar kodola kompilēšanu, ir silti ieteicams ieslēgt visu kompilatora paziņojumu izvadi. To veic, vienkārši pievienojot `v=1` komandrindas beigās. Tas ļaus jums redzēt vairāk informācijas, kas varētu noderēt kodola izstrādātājiem, lai novērstu problēmu.

```
# make bzImage V=1
```

### Kodola moduļu izmantošana

Kodola moduļi patiesībā ir vienkārši cits nosaukums iekārtu draiveriem jeb dziņiem. Tos ir iespējams ievietot jeb startēt strādājošā kodolā, tā paplašinot jūsu sistēmas kodola atbalstīto aparatūras klāstu bez nepieciešamības izvēlēties citu kodolu vai pat kompilēt to no pirmkoda.

Moduļus var ielādēt un izlādēt jebkurā brīdī - pat, ja sistēma darbojas. Tas ļauj sistēmu administratoriem viegli atjaunināt specifiskus draiverus - var nokompilēt jaunu moduli, izlādēt veco, un ielādēt jauno - bez sistēmas pārstartēšanas.

Moduļi atrodas direktoriņā `/lib/modules/kodola_versija`. Tos var ielādēt sistēmas ielādes laikā, izmantojot failu `rc.modules`. Šis fails ir lieliski komentēts un piedāvā aplūkot piemērus galvenajām aparatūras sastāvdaļām. Lai aplūkotu šobrīd ielādēto moduļu sarakstu, izmantojiet komandu `lsmod(1)`:

```
# lsmod
Module                Size  Used by
parport_pc            7220  0
parport               7844  0 [parport_pc]
```

Redzams, ka šajā situācijā ir ielādēts tikai paralēlā porta atbalsta modulis. Lai izlādētu moduli no atmiņas, izmantojiet komandu `rmmmod(1)`. Moduļus iespējams ielādēt ar `modprobe(1)` vai `insmod(1)` komandām. `modprobe` parasti ir drošāka, jo automātiski ielādēs arī tos moduļus, no kuriem ir atkarīgs jūsu ielādējamais modulis.

Lielākajai daļai lietotāju nekad nav jānodarbojas ar moduļu manuālu ielādi un izlādi. Viņi izmanto kodola autolādētāju. Pēc noklusējuma Slackware iekļauj `kmod` atbalstu savos kodolos. `kmod` ir kodola iespēja, kas ļauj tam automātiski ielādēt moduļus tad, kad tie ir nepieciešami. Vairāk informācijas par `kmod` un to, kā tas tiek konfigurēts, var atrast failā `/usr/src/linux/Documentation/kmod.txt`, kas pieejams sistēmā, ja ir uzstādīta kodola pirmkoda pakotne, vai lejupielādēts kodola pirmkods no <http://kernel.org>.

Vairāk informāciju iespējams atrast arī *man* dokumentos katrai minētajai komandai, kā arī failā `rc.modules`.



## 5.nodaļa. Tīkla konfigurēšana

### 5.1. Ievads: netconfig ir jūsu draugs

Instalējot *Slackware*, uzstādīšanas programma iedarbināja arī `netconfig` programmu. `netconfig` veica šādas darbības:

- Uzdeva jums jautājumus par jūsu datora hosta un domēna vārdu.
- Sniedza īsus skaidrojumus par dažādām adresācijas shēmām, paskaidroja, kad tās jāizmanto un lūdza jums norādīt, kuru IP adresācijas shēmu izmantot tīkla kartes konfigurēšanai:
  - statisku IP adresi;
  - DHCP;
  - atgriezenisko cilpu.
- Tad tā piedāvāja noteikt tīkla karti.

`netconfig`, ja vien jūs to atļausiet, parasti veic aptuveni 80% no jūsu LAN tīkla pieslēguma konfigurācijas. Tomēr es iesaku aplūkot jūsu konfigurācijas failu šādu iemeslu dēļ:

1. Nekad neuzticieties uzstādīšanas programmas spējām korekti konfigurēt jūsu datoru. Ja jūs izmantojat uzstādīšanas programmu, jums tomēr pašam jāpārbauda konfigurāciju.
2. Ja jūs joprojām apgūstat *Slackware* un *Linux* sistēmas pārvaldību, strādājošas konfigurācijas aplūkošana var būt noderīga. Vismaz jūs uzzināsiet, kā izskatās korekta konfigurācija. Tas ļaus jums apgūt nākotnes konfigurācijas problēmu risināšanu.

### 5.2. Tīkla aparatūras konfigurēšana

Kad pieņemts lēmums pieslēgt jūsu *Slackware* datoru kādam datoru tīklam, pirmais, kas nepieciešams, ir ar *Linux* savietojama tīkla karte. Jums būs nedaudz jāpiestrādā, lai pārliecinātos, vai karte ir patiešām savietojama ar *Linux* (lūdzu skatīt *Linux* dokumentācijas projektu un/vai kodola dokumentāciju informācijai par jūsu izvēlētas tīkla kartes savietojamības statusu). Kopumā jūs būsiet patīkami pārsteigts par mūsdienu kodolu atbalstīto tīkla karšu klāstu. Neraugoties uz to, pirms tīkla kartes iegādes es tomēr iesaku aplūkot kādu no dažādajiem *Linux* savietojamības sarakstiem (piemēram, *GNU/Linux* iesācēju grupas aparatūras savietojamības norādes vai *Linux* dokumentācijas projekta aparatūras *HOWTO*), kas ir pieejami *Internet* tīklā. Nedaudz papildus laika, kas patērēts izpētei, var aiztaupīt jums dienas vai pat nedēļas, mēģinot iedarbināt karti, kas patiesībā nemaz nav savietojama ar *Linux*.

Aplūkojot *Internet* tīklā pieejamos *Linux* aparatūras savietojamības sarakstus, ir laba doma noskaidrot, kuru kodola moduli būs jāizmanto jūsu izvēlētas tīkla kartes atbalstam.

#### Kodola moduļu ielāde

Kodola moduļi, kuru ielādei jānotiek sistēmas ielādes laikā, tiek ielādēti no `rc.modules` faila `/etc/rc.d` direktorijā, vai izmantojot kodola autoielādi, ko startē `/etc/rc.d/rc.hotplug`. Noklusētajā `rc.modules` failā ir iekļauta tīkla iekārtu atbalsta sadaļa. Atverot `rc.modules` tekstu redaktorā un meklējot sadaļu *Network device*, jūs pamanīsiet, ka šis skripts vispirms meklē izpildāmu failu `rc.netdevice` direktorijā `/etc/rc.d`. Šis skripts tiek radīts, ja `setup` veiksmīgi atpazīst jūsu tīkla karti instalēšanas laikā.

Zem pārbaudes bloka ir tīkla iekārtu un `modprobe` rindiņu saraksts, kur visas rindiņas ir aizkomentētas. Atrodiet savu iekārtu un atkomentējiet atbilstošo `modprobe` rindiņu, tad saglabājiet

failu. Izpildot `rc.modules` ar `root` tiesībām, būtu veiksmīgi jāielādējas jūsu tīkla iekārtas draiverim (kā arī citiem atkomentētajiem moduļiem). Jāpiebilst, ka dažiem moduļiem (piemēram, `ne2000`) ir nepieciešami papildparametri, tādēļ pārlicinieties, ka jūs izvēlaties pareizo rindiņu.

## LAN (10/100/1000Base-T un Base-2) kartes

Šī sadaļa satur konfigurācijas rindiņas visām PCI un ISA tīkla kartēm. Šo iekārtu draiveri, kā aprakstīts iepriekšējā sadaļā, pieejami kā ielādējami kodola moduļi. `/sbin/netconfig` būtu bijis jāatrod jūsu tīkla karti un veiksmīgi to jāuzstāda `rc.netdevice` failā. Ja tas nav noticis, ticamākais, ka izvēlētais modulis neatbilst jūsu tīkla kartei (nereti pat viena ražotāja viena modeļa, bet dažādu paaudžu tīkla kartēm ir nepieciešami dažādi moduļi). Ja jūs esat pārlicināts, ka izvēlētais modulis ir pareizs, tad nākamais, ko jāpārbauda, ir attiecīgā moduļa dokumentācija, lai noskaidrotu, vai moduļa darbināšanai nav nepieciešami specifiski parametri.

## Modemi

Līdzīgi, kā LAN kartes, arī modemiem var būt dažādi pieslēgumi datoram. Vēl nesen lielākā daļa modemu bija 8 vai 16 bitu ISA kartes. Pateicoties *Intel* un mātesplašu ražotāju pūliņiem izskaust ISA kopnes izmantošanu, lielākā daļa mūsdienu modemu ir vai nu ārējie modemi, kas tiek pieslēgti pie datora, izmantojot seriālo vai USB portu, vai iekšējie PCI modemi. Ja jūs vēlaties, lai jūsu modems darbotos *Linux* sistēmā, ir ļoti svarīgi izziņāt, kādu modema tipu iegādāties, īpaši iegādājoties PCI modemu. Lielākā daļa PCI modemu, kas mūsdienās pieejami veikalos, ir *WinModem* tipa modemi. *WinModem* ir lēti modemi, kuriem ir samazināts detaļu skaits, lielu daļu darba veicot ar datora centrālā procesora palīdzību, izmantojot *Windows* operētājsistēmu modema draiverus. Tas nozīmē, ka tie nav pieslēgti datoram, izmantojot standarta seriālo saskarni, ar kuru vēlēties strādāt `pppd`, kas jūs liksiet tam izmantot iezvanpieeju.

Ja jūs vēlaties būt pilnīgi pārlicināts, ka jūsu iegādātais modems darbosies *Linux* operētājsistēmā, iegādājieties ārējo modemu, kuru jāpieslēdz pie datora, izmantojot standarta seriālo portu. Šie modemi darbojas labāk un tos būs vienkāršāk instalēt un uzturēt, lai arī tiem ir nepieciešams ārējais barošanas avots un tie maksā vairāk.

Ir vairākas *Internet* lapas, kurās pieejami draiveri un atbalsts *WinModem* tipa iekārtu konfigurēšanai. Daži lietotāji ir ziņojuši par veiksmīgu šā tipa iekārtu konfigurēšanu un draiveru instalēšanu - izmantojot *Lucent*, *Conexant* un *Rockwell* shēmu modemus. Tā kā šīm iekārtām nepieciešamā programmatūra nav iekļauta *Slackware* distributīvā, mēs tuvāk tās neaplūkosim.

## PCMCIA

*Slackware Linux* instalēšanas procesā jums ir iespēja instalēt **PCMCIA** pakotni (A sērijā). Šī pakotne satur programmas un konfigurācijas failus, kas nepieciešami, lai strādātu ar **PCMCIA** kartēm *Slackware* vidē. Ir svarīgi atcerēties, ka **PCMCIA** pakotne satur tikai pamatprogrammatūru, kas nepieciešama darbam ar **PCMCIA** kartēm. Tā NEINSTALĒ draiverus vai moduļus. Pieejamie draiveri un moduļi atradīsies `/lib/modules/'uname -r'/pcmcia` direktorijā. Jums var nākties nedaudz paeksperimentēt, lai atrastu moduli, kas darbosies ar jūsu tīkla karti.

Jums būs jāredīgē `/etc/pcmcia/network.opts` (Ethernet kartei) vai `/etc/pcmcia/wireless.opts` (bezvadu kartei) faili. Līdzīgi kā lielākā daļa *Slackware* konfigurācijas failu, tie satur labu dokumentāciju, un nepieciešamo modifikāciju apzināšanai būtu jābūt vienkāršai.

## 5.3. TCP/IP konfigurēšana

Un tā - jūsu tīkla karte ir fiziski uzstādīta datorā un atbilstošie kodola moduļi ir ielādēti. Lai arī jūsu dators joprojām nevar sazināties ar citiem tīkla datoriem, tomēr jūs varat aplūkot informāciju par tīkla

iekārtu, izmantojot komandu `ifconfig -a`.

```
# ifconfig -a
eth0 Link encap:Ethernet HWaddr 00:A0:CC:3C:60:A4
UP BROADCAST NOTRAILERS RUNNING MULTICAST MTU:1500 Metric:1
RX packets:110081 errors:1 dropped:0 overruns:0 frame:0
TX packets:84931 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:100
RX bytes:114824506 (109.5 Mb) TX bytes:9337924 (8.9 Mb)
Interrupt:5 Base address:0x8400

lo Link encap:Local Loopback
inet addr:127.0.0.1 Mask 255.0.0.0
UP LOOPBACK RUNNING MTU:16436 Metric:1
RX packets:2234 errors:0 dropped:0 overruns:0 frame:0
TX packets:2234 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:0
RX bytes:168758 (164.8 Kb) TX bytes:168758 (164.8 Kb)
```

Ja jūs ievadījāt `/sbin/ifconfig` bez `-a`, tad jūs neredzēsiet `eth0` saskarni, jo jūsu tīkla kartei vēl nav derīgas IP adreses un maršrutēšanas informācijas.

Lai arī eksistē daudz dažādu veidu, kā konfigurēt un sadalīt tīklu, tos visus iespējams sadalīt divās grupās: statiskie un dinamiskie. Statiskie tīkli tiek veidoti tā, ka katram mezglam (datoržargons, kas apzīmē iekārtu ar IP adresi) vienmēr ir viena un tā pati IP adrese. Dinamiskie tīkli tiek veidoti tā, ka mezglu IP adreses kontrolē viens resursdators, ko sauc par DHCP serveri.

## DHCP

DHCP (Dynamic Host Configuration Protocol - Dinamiskais hostu konfigurācijas protokols) ir veids, kā datoram tā ielādes laikā iespējams piešķirt IP adresi. Kad DHCP klients veic ielādi, tas lokālajā tīklā izsūta pieprasījumu DHCP serverim, lai tas piešķir IP adresi. DHCP servera rīcībā ir pieejamo adrešu fonds (diapazons). Serveris atbildēs šim pieprasījumam ar IP adresi no fonda, ko nosūtīs kopā ar nomas laiku. Kad nomas laiks noteiktajai IP adresei būs beidzies, klientam atkārtoti jāsaazinās serveri un jāatkārto procedūru.

Kad adrese saņemta, klients konfigurēs atbilstošo tīkla interfeisu ar šo adresi. Patiesībā ir vēl kāds triks, ko izmanto DHCP klienti, lai mēģinātu saglabāt piešķirto IP adresi. Tie atcerēsies, kāda IP adrese tiem ir bijusi piešķirta iepriekšējā nomas periodā un lūgs DHCP serverim, lai tas piešķir šo pašu adresi. Ja tas būs iespējams, serveris tā arī darīs, bet ja nē - tiks piešķirta jauna adrese. Līdz ar to adreses pieprasīšana un saņemšana izskatās apmēram šādi:

```
Klients: Vai šajā tīklā ir kāds DHCP serveris?
Serveris: Jā, ir. Es tas esmu.
Klients: Man vajag IP adresi.
Serveris: Vari ņemt 192.168.10.10 uz 19200 sekundēm.
Klients: Paldies.
...
Klients: Vai šajā tīklā ir kāds DHCP serveris?
Serveris: Jā, ir. Es tas esmu.
Klients: Man vajag IP adresi. Pēdējoreiz, kad mēs plāpājām, tu man iedevi
192.168.10.10. Vai es to varu dabūt vēlreiz?
Serveris: Jā, protams! (vai arī - Nē, nevari - ņem 192.168.10.12!)
```

Linux vidē DHCP klients ir `/sbin/dhpcpd`. Ja jūs ielādēsiet `/etc/rc.d/rc.inet1` savā iecienītajā

tekstu redaktorā, jūs redzēsiet, ka `/sbin/dhpcd` tiek izsaukts aptuveni skripta izpildes vidū. Tas liks notikt dialogam, kas līdzīgs augstāk minētajam. `dhpcd` arī skaita laiku, kas atlicis līdz adreses nomas laika beigām un automātiski sazinās ar DHCP serveri, nosūtot tam pieprasījumu atjaunot nomas laiku, kad tas ir nepieciešams. DHCP servera izsūtītājā informācijā ir arī papilddati, piemēram, par to, kādu NTP (laika) serveri izmantot, kāda ir maršrutēšanas informācija utt.

Uzstādīt DHCP *Slackware* sistēmā ir vienkārši. Vienkārši izpildiet `netconfig` un izvēlieties DHCP, kad tas tiek piedāvāts. Ja jums ir vairāk, kā viena tīkla karte un jūs nevēlaties, lai `eth0` tiktu konfigurēts ar DHCP, vienkārši izlabojiet `/etc/rc.d/rc.inet1.conf` failu un jūsu izvēlētajai kartei uzstādiat attiecīgo parametru uz "YES".

## Statiska IP adrese

Statiskās IP adreses mainās tikai tad, ja tās tiek manuāli mainītas. Tās tiek lietotas gadījumos, kad administrators nevēlas, lai tās mainītos, piemēram, lokālā tīkla iekšējiem serveriem, jebkuram serverim, kas pieslēgts Internet tīklam, kā arī tīkla maršrutētājiem. Ar statisku IP adresāciju adrese tiek piešķirta un tāda tiek atstāta, lai citi datori, kuriem ir zināma šī adrese, pēc tās vienmēr varētu sazināties ar to.

### `/etc/rc.d/rc.inet1.conf`

Ja jūs plānojat piešķirt statisku IP adresi savai jaunajai *Slackware* sistēmai, jūs to varat veikt ar `netconfig` programmu, vai arī izlabot `/etc/rc.d/rc.inet1.conf` failu. Šajā failā jūs atradīsiet rindiņas:

```
# Primary network interface card (eth0)
IPADDR[0]=" "
NETMASK[0]=" "
USE_DHCP[0]=" "
DHCP_HOSTNAME[0]=" "
```

Faila beigās atrodama rindiņa

```
GATEWAY=" "
```

Šajā gadījumā mūsu uzdevums ir ierakstīt pareizo informāciju pēdējās. Šos mainīgos tīkla karšu konfigurēšanai izsauc `/etc/rc.d/rc.inet1` skripts sistēmas ielādes laikā. Vienkārši ievadiet pareizo IP adresācijas informāciju katrai tīkla kartei, vai arī nomainiet `USE_DHCP` vērtību uz "YES". *Slackware* startēs tīkla kartes ar šo konfigurāciju tādā secībā, kādā tās tiek atrastas.

Mainīgais `GATEWAY` uzstāda noklusētā maršrutētāja adresi. Visai datu plūsmai starp jūsu datoru un citiem datoriem *Internet* tīklā jānotiek, izmantojot šo adresi, ja vien nav norādīta cita maršrutēšanas informācija. Ja jūs izmantojat DHCP, parasti šeit neko nebūs jāievada, jo DHCP serveris norādīs, kuru vārteju izmantot.

### `/etc/resolv.conf`

Labi, tagad jūsu datoram ir IP adrese, ir arī noklusētā vārteja, varbūt pat jums ir desmit miljoni dolāru (iedodiet mums arī drusciņ), bet kāds no tā visa labums, ja nav iespējams noteikt IP adreses pēc hostu vārdiem? Neviens taču nevēlas savā *Internet* pārlūkprogrammā ievadīt `72.9.234.112`, lai piekļūtu `www.slackbook.org`. Galu galā, kurš bez lapas autoriem varētu atcerēties šo IP adresi? Mums jāuzstāda DNS, bet kā? Šeit nāk talkā fails `/etc/resolv.conf`.

Pastāv iespēja, ka `/etc/resolv.conf` jau ir ievadītas nepieciešamās vērtības. Ja jūs izmantojat DHCP, tad DHCP serverim būtu jāatjaunina šo failu (patiesībā jāsniedz `dhpcd` informāciju, ko šeit ierakstīt). Ja jūs tomēr atjauninat DNS sarakstu manuāli, jums šo failu jārediģē. Piemērs:

```
# cat /etc/resolv.conf
```

```
nameserver 192.168.1.254
search lizella.net
```

Pirmā rindiņa ir vienkārša. `nameserver` direktīva nosaka, kādus DNS serverus izmantot. Šīs vienmēr ir IP adreses, kuras var norādīt tik, cik vēlaties. *Slackware* ar prieku pārbaudīs visas pēc kārtas, kamēr kāda spēš noteikt nepieciešamo adresi.

Otrā rindiņa ir interesantāka. `search` direktīva mums sniedz domēnu nosaukumu sarakstu, ko ņemt vērā, veicot DNS pieprasījumu. Tas ļauj sazināties ar datoru, izmantojot tikai pirmo tā *FQDN* (Fully Qualified Domain Name - pilns domēna nosaukums) daļu. Piemēram, ja `search` būtu norādīts `slackware.com`, tad `http://store.slackware.com` būtu pieejams arī pēc adreses `http://store`.

```
# ping -c 1 store
PING store.slackware.com (69.50.233.153) 56 data bytes
64 bytes from 69.50.233.154 : icmp_seq=0 ttl=64 time=0.251ms
1 packets transmitted, 1 packets received, 0% packet loss
round-trip min/avg/max = 0.251/0.251/0.251 ms
```

## /etc/hosts

Tagad, kad DNS darbojas nevainojami, iedomāsimies - ko darīt, ja vēlamies apiet DNS serveri, vai pievienot DNS ierakstu datoram, kura adrese nav norādīta DNS? *Slackware* komplektā ir iekļauts visu mīlētais `/etc/hosts` fails, kurš satur lokālu DNS vārdu sarakstu un IP adreses, kurām atbilst šie vārdi.

```
# cat /etc/hosts
127.0.0.1        localhost      localhost.localdomain
192.168.1.101   redtail
172.14.66.32    foobar.slackware.com
```

Šeit redzams, ka `localhost` adrese ir `127.0.0.1` (vienmēr rezervēta `localhost`), `redtail` var piekļūt pēc adreses `192.168.1.101` un `foobar.slackware.com` ir `172.14.66.32`.

## 5.4. PPP

Daudzi pieslēgumam *Internet* tīklam joprojām izmanto iezvanpieeju. Vispopulārākā metode ir PPP, lai gan reizēm joprojām tiek izmantota arī SLIP. Jūsu sistēmas konfigurēšana darbam ar PPP ir samērā vienkārša. *Slackware* sistēmā ir iekļauti daži rīki, kas atvieglos konfigurēšanu.

### pppsetup

*Slackware* ir iekļauta programma `pppsetup`, kas konfigurē jūsu sistēmu darbam ar iezvanpieeju. Tās saskarne ir līdzīga `netconfig` programmai. Lai izpildītu šo programmu, pārliecinieties, ka esat autorizējies kā `root` lietotājs. Tad ievadiet komandu `pppsetup`.

Programma izvadīs vairākus jautājumus, uz kuriem jums būs jāsniedz atbildes - piemēram, par modema iekārtu, modema inicializācijas rindu un *Internet* pakalpojumu sniedzēja tālruna numuru. Dažiem jautājumiem tiks piedāvātas arī noklusētās atbildes, kuras lielākajā daļā gadījumu ir pieņemamas.

Pēc programmas izpildes tā būs radījusi divas citas programmas - `ppp-go` un `ppp-off`. Tās jāizmanto, lai uzsāktu un izbeigtu PPP pieslēgumu. Šīs programmas ir izvietotas `/usr/sbin` direktoriņā un to izpildei nepieciešamas `root` lietotāja tiesības.

Lielākajai daļai lietotāju ar `pppsetup` izpildi būs pietiekoši, tomēr var būt situācijas, kad jūs varētu vēlēt papildināt dažus PPP servisa konfigurācijas parametrus. Visa konfigurācijas informācija tiek

glabāta direktorijā `/etc/ppp`. Šeit ir saraksts ar direktorijā izvietotajiem failiem un to nozīmi:

`ip-down` - šo skriptu `pppd` izpilda tad, kad PPP pieslēgums tiek izbeigts.

`ip-up` - šo skriptu `pppd` izpilda tad, kad nodibināts veiksmīgs PPP pieslēgums.

`options` - vispārējie `pppd` konfigurācijas iestatījumi.

`options.demand` - vispārējie `pppd` konfigurācijas iestatījumi, kad `pppd` tiek darbināts pieprasījuma režīmā.

`pppscript` - komandas, kas tiek nodotas modemam.

`pppsetup.txt` - `pppsetup` izpildes žurnālfails.

Piezīme: lielākā daļa šo failu vairs nebūs šajā direktorijā pēc `pppsetup` izpildes.

## 5.5. Bezvadu tīkls

Lai gan bezvadu tīkli datoru pasaulē ir salīdzinoši jauns ievedums, tomēr tie kļuvuši ļoti populāri, jo arvien vairāk cilvēku iegādājas klēpj datorus un vēlas tīklu bez nepieciešamības krāmēties ar veco, labo vītā pāra kabeli. Diemžēl bezvadu tīklu atbalsts *Linux* sistēmā joprojām nav tik labs kā parasto.

802.11 bezvadu tīkla kartes konfigurēšanas procesā ir trīs pamatsolji:

1. Aparatūras atbalsts bezvadu kartei.
2. Kartes konfigurēšana pieslēguma bezvadu piekļuves punktam.
3. Tīkla konfigurēšana.

### Aparatūras atbalsts

Aparatūras atbalsts bezvadu tīkla kartei tiek nodrošināts, izmantojot kodolu - vai nu ar moduļa palīdzību, vai ar kodolā iebūvētu draiveri. Vispār jaunākās *Ethernet* kartes tiek atbalstītas ar kodola moduļiem, tādēļ jums jānosaka attiecīgo kodola moduli un jāielādē to, izmantojot `/etc/rc.d/rc.modules.netconfig` var nenoteikt jūsu tīkla karti, tādēļ jums, ticamākais, to būs jānosaka pašiem. Aplūkojiet [http://www.hp1.hp.com/personal/Jean\\_Tourrilhes/Linux/mājaslapu](http://www.hp1.hp.com/personal/Jean_Tourrilhes/Linux/mājaslapu). Tajā jūs atradīsit informāciju par kodola moduļiem dažādām bezvadu kartēm.

### Kartes konfigurēšana

Lielāko daļu darba veic komanda `iwconfig`, tādēļ vispirms izlasiet `iwconfig man` dokumentu.

Vispirms jums jākonfigurē bezvadu piekļuves punktu. Dažādi bezvadu piekļuves punkti savstarpēji atšķiras to terminoloģijā un konfigurēšanas procesā, tādēļ jums var gadīties pielāgoties, lai iedarbinātu savu aparatūru. Vispārīgā gadījumā, jums nepieciešama vismaz šāda informācija:

- domēna ID, vai tīkla nosaukums (`iwconfig` to sauc par ESSID)
- WAP izmantotais kanāls
- datu šifrēšanas iestatījumi, arī visas izmantotās atslēgas (ieteicams heksadecimālā formā)

**Brīdinājums:** piezīme par WEP. WEP ir samērā kļūdainis, tomēr tas ir daudz labāks par neko. Ja jūs vēlaties panākt lielāku bezvadu tīkla drošību, jums jāapsver iespēju izmantot VPN vai IPSec. Šīs iespējas nav aplūkotas šajā grāmatā. Ir iespējams arī konfigurēt WAP tā, ka tas neizziņo savu domēna ID/ESSID. Šis sadaļas uzdevums nav aplūkot dažādās bezvadu datu šifrēšanas iespējas, tomēr vienkāršs *Google* meklējums jums sniegs vairāk atbilžu, kā jūs jebkad esat vēlējies zināt.

Kad sagatavota visa nepieciešamā informācija un ar `modprobe` palīdzību veiksmīgi ielādēts bezvadu

kartes atbalsta modulis, jūs varat rediģēt failu `rc.wireless.conf` un pievienot savus iestatījumus. `rc.wireless.conf` šobrīd ir samērā nesakārtots, tomēr patiesībā mazākais, ko jums jāveic, ir sadaļas "generic" modificēšana, iestatot savu **ESSID** un **KEY**, kā arī **CHANNEL**, ja tas nepieciešams. Vispirms mēģiniet neiestatīt **CHANNEL** - ja tas darbojas - lieliski, ja nē - iestatiet **CHANNEL**. Ja jūtaties pietiekoši drošs, jūs varat modificēt failu tā, ka tiek iestatīti tikai nepieciešamie mainīgie. `rc.wireless.conf` mainīgo nosaukumi atbilst `iwconfig` parametriem - tos nolasa `rc.wireless` un izmanto atbilstošajās `iwconfig` komandās.

Ja jums ir atslēga heksadecimālā formā, tad tas ir lieliski, jo jūs varat būt samērā drošs, ka jūsu WAP un `iwconfig` to sapratīs. Ja jums ir vienkārši teksta rinda, jūs nevarat būt pārliecināts, kā jūsu WAP to pārveidos heksadecimālā rindā, tādēļ var gadīties, ka nākas izmēģināt dažādus atslēgu konfigurācijas variantus (vai visvienkāršākais - konvertēt jūsu WAP atslēgu heksadecimālā formā).

Kad nepieciešamās izmaiņas `rc.wireless.conf` failā ir veiktas, kā **root** izpildiet `rc.wireless`. Pēc tam izpildiet `rc.inet1`. Tagad pārbaudiet savu bezvadu tīkla pieslēgumu ar standarta testēšanas rīkiem, kā `ping`. Ja jums ir arī parasta tīkla karte, jūs varētu vēlēties izmantot `ifconfig`, lai to atslēgtu - tā jūs varēsīt būt pārliecināts, ka testi notiek bez traucējumiem. Jūs varētu vēlēties arī pārbaudīt izmaiņas pēc sistēmas pārstartēšanas.

Tagad, kad jums ir skaidrs, kā labot `/etc/rc.d/rc.wireless` jūsu noklusētajam tīklam, aplūkosim `iwconfig` un to, kā tā darbojas. Tas jums parādīs ātrāko metodi, kā konfigurēt bezvadu pieslēgumu tajās reizēs, kad jūs atradīsities *Internet* kafejnīcā, bārā vai kur citur, kur pieejams bezvadu tīkls un jūs vēlēšities pārlūkot *Internet* tīklu.

Pirmais solis ir norādīt jūsu bezvadu kartei, kuram tīklam pievienoties. Pārliecinieties, ka jūs aizstājat `eth0` ar jūsu bezvadu kartes tīkla interfeisa nosaukumu un nomainiet `mynetwork` ar jūsu izmantot ESSID. Un - jā, jā, mēs jau zinām, ka jūs esat pietiekoši gudrs, lai to vienmēr atcerētos. Pēc tam, ja nepieciešams, jums jānorāda tīkla kriptēšanas atslēgu. Beigās norādiet izmantojamo kanālu.

```
# iwconfig eth0 essid "mynetwork"
# iwconfig eth0 key XXXXX XXXXXXXXXXXXX XXXXXXXXXXXXX
# iwconfig eth0 channel n
```

Ar to būtu jāprietiek, lai jūsu bezvadu tīkla karte būtu nokonfigurēta.

## Tīkla konfigurēšana

Tīkla konfigurēšana tiek veikta tieši tāpat, kā parastajiem tīkliem. Vienkārši izlasiet šīs nodaļas iepriekšējās sadaļas.

## 5.6. Tīkla failsistēmas

Šajā brīdī jūsu rīcībā ir strādājošs TCP/IP pieslēgums. Jums būtu jāvar izmantot ehotestēšanu (`ping`), lai sazinātos ar citiem datoriem jūsu iekšējā tīklā un, ja ir konfigurēta atbilstoša vārteja, arī ehotestēt *Internet* serverus. Kā zināms, datortīklu būtība ir piekļuve informācijai. Lai gan daži varētu pieslēgt datoru tīklam tikai tādēļ, lai būtu interesanti, lielākā daļa vēlas izmantot failu un printeru koplietošanas iespējas. Viņi vēlas piekļūt dokumentiem *Internet* tīklā vai spēlēt tiešsaistes spēles. Strādājošs TCP/IP pieslēgums ir svarīgs posms ceļā uz šīm iespējām, tomēr TCP/IP pieslēgums pats par sevi nodrošina tikai rudimentāru tīkla funkcionalitāti. Lai izmantotu failu koplietošanu, mums tos būs jāpārviesto šurp vai turp, izmantojot FTP vai SCP. Mēs nevaram pārlūkot mūsu *Slackware* datora failus, izmantojot *Network Neighborhood* vai *My Network Places* ikoniņas *Windows* datoros. Mēs vēlamies arī piekļūt failiem citos *Unix* datoros.

Ideālā gadījumā mēs vēlamies izmantot tīkla failsistēmu, lai piekļūtu failiem citos datoros. Programmām, kas vēlas izmantot citos datoros pieejamu informācijai, nav jāzina, kur tā tiek glabāta - tām tikai jāzina, ka tā eksistē un kā tai piekļūt. Tālāk par visu parūpēsies operētājsistēma. Divas visbiežāk izmantotās tīkla failsistēmas ir SMB (to nodrošina programmatūras pakotne *Samba*) un

NFS.

## SMB/Samba/CIFS

SMB (Server Message Block - servera ziņojuma bloks) ir IBM *LANManager* izmantotā *NetBIOS* protokola pēctecis. NetBIOS un tā pēcteču (NetBEUI, SMB un CIFS) attīstībā vienmēr ir bijusi ieinteresēta arī firma *Microsoft*.

*Samba* projekts pastāv kopš 1991.gada, kad to izveidoja, lai savienotu NetBIOS izmantojošu IBM PC datoru ar *Unix* serveri. Mūsdienās SMB ir svarīgākā failu un printeru tīkla koplietošanas metode praktiski visai civilizētajai pasaulei, jo to atbalsta arī *Windows*.

*Samba* konfigurācijas fails ir `/etc/samba/smb.conf` - viens no visu laiku vislabāk komentētajiem un dokumentētajiem konfigurācijas failiem. Tajā konfigurēti arī piemēri koplietošanas resurspunktiem, kurus jūs varat modificēt, lai pielāgotu savām vajadzībām. Pat, ja jums nepieciešama pilnīgāka kontrole pār šo procesu, jūs varat aplūkot lielisko `smb.conf man` dokumentu. Tā kā *Samba* ir tik labi dokumentēta manis minētajos dokumentos, mēs neveiksim dokumentācijas pārrakstīšanu, tomēr aplūsim pašus pamatus.

`smb.conf` ir sadalīts vairākās sadaļās - viena sadaļa katram resurspunktam, kā arī globālā sadaļā, kurā tiek norādīti parametri, kas tiks izmantoti visur. Daži iestatījumi ir derīgi tikai globālajā sadaļā, daži - tikai ārpus tās. Atcerieties, ka globālās sadaļas iestatījumus var lokāli pārkonfigurēt jebkurā citā sadaļā. Vairāk informāciju meklējiet atbilstošajā `man` dokumentā.

Visticamākais, jūs vēlēsit rediģēt savu `smb.conf` failu, lai atainotu jūsu LAN tīkla iestatījumus. Es iesaku modificēt šīs rindiņas:

```
[global]
# workgroup = NT-Domain-Name or Workgroup-Name, eg: LINUX2
workgroup = MYGROUP
Šeit nomainiet darbgrupas nosaukumu uz jūsu izmantoto.
# server string is the equivalent of the NT Description field
server string = Samba Server
```

Šis būs nosaukums, ar kādu jūsu *Slackware* dators parādīsies *Network Neighborhood* (vai *My Network Places*) mapē.

```
# Security mode. Most people will want user level security. See
# security_level.txt for details. NOTE: To get the behaviour of
# Samba-1.9.18, you'll need to use "security = share".
security = user
```

Jūs gandrīz noteikti vēlaties ieviest lietotāja (`user`) līmeņa drošību savā *Slackware* sistēmā.

```
# You may wish to use password encryption. Please read
# ENCRYPTION.txt, Win95.txt and WinNT.txt in the Samba
# documentation.
# Do not enable this option unless you have read those documents
encrypt passwords = yes
```

Ja paroli šifrēšana nav ieslēgta, jūs nevarēsiet izmantot *Samba* ar NT4.0, Win2k, WinXP un Win2003. Iepriekšējām *Windows* operētājsistēmas paroli šifrēšana nebija obligāta.

SMB ir autentificēts protokols, kas nozīmē, ka jums šī servisa izmantošanai ir jānorāda korektu lietotājvārdu un paroli. Lai norādītu *Samba* serverim, kādi lietotājvārdi un paroles ir derīgas, jāizmanto `smbpasswd` komandu. `smbpasswd` jānorāda pāris komandrindas parametru, lai norādītu, vai pievienot parastos, vai sistēmas lietotājus. SMB pieprasa, lai datoru NETBIOS nosaukumi tiktu pievienoti kā sistēmas lietotāji, tā ierobežojot to, no kuriem datoriem iespējams autentificēties.



### Lietotāja pievienošana /etc/samba/private/smbpasswd failam

```
# smbpasswd -a lietotajs
```

### Datora nosaukuma pievienošana /etc/samba/private/smbpasswd failam

```
# smbpasswd -a -m dators
```

Ir svarīgi, ka norādītajam lietotājam vai datora nosaukumam jau jāeksistē /etc/passwd failā. Jūs to varat panākt, izmantojot `adduser` komandu. Atcerieties, ka, izmantojot `adduser` komandu, lai pievienotu datora nosaukumu, nosaukumam jāpievieno dolāra simbols (\$). To NEDRĪKST darīt ar `smbpasswd`. `smbpasswd` pats pievieno dolāra simbolu datora nosaukuma beigās. Nekorekti nosaucot sistēmas lietotāju datora nosaukumam, radīsies kļūda, mēģinot pievienot datora nosaukumu *Samba* serverim.

```
# adduser dators$
```

## Network File System (NFS)

NFS (jeb Network File System - tīkla failsistēma) sākotnēji tika veidota firmā *Sun* tās *Unix* tipa operētājsistēmai *Solaris*. Lai gan NFS ir daudz vienkāršāk uzstādīt un konfigurēt, tomēr tā ir daudz nedrošāka. Galvenais, kas padara NFS nedrošu, ir fakts, ka ir iespējams ļoti vienkārši viltot lietotāja un grupas identifikatorus, lai piekļūtu failiem. Tas tādēļ, ka NFS nav lietotāju autentifikācijas mehānisms. Nākotnes NFS protokola versijas plāno uzlabot drošību, tomēr tās šīs grāmatas rakstīšanas brīdī nav plaši ieviestas.

NFS konfigurāciju nosaka fails /etc/exports. Ielādējot noklusēto /etc/exports failu tekstu redaktorā, jūs redzēsiet tukšu failu ar divu rindiņu komentāru faila sākumā. Katrai direktorijai, ko vēlamies eksportēt jeb padarīt pieejamu citiem tīkla datoriem, mums jāpievieno atbilstošu rindiņu `export` failam, norādot tos klienta datorus, kuriem ir tiesības piekļūt šai direktorijai. Piemēram, ja mēs vēlamies, lai direktorija /home/foo būtu pieejama datoram `Bar`, mums jāpievieno rindiņu

```
/home/foo Bar(rw)
```

Cits piemērs - no `exports` faila `man` dokumenta:

```
# sample /etc/exports file
/          master(rw) trusty(rw,no_root_squash)
/projects  proj*.local.domain(rw)
/usr       *.local.domain(ro) @trusted(rw)
/home/joe  pc001(rw,all_squash,anonuid=150,anongid=100)
/pub       (ro,insecure,all_squash)
```

Kā redzams, ir pieejami daudz konfigurācijas iestatījumi, tomēr lielākajai daļai būtu jābūt skaidrai no šī piemēra.

NFS darbojas, pieņemot, ka noteiktajam viena datora lietotājam tā identifikators ir identisks visos tīkla datoros. Kad notiek mēģinājums lasīt vai rakstīt no NFS klienta uz NFS serveri, daļa no lasīšanas/rakstīšanas pieprasījuma ir lietotāja UID. Šis UID tiek apstrādāts tieši tāpat, it kā pieprasījums būtu radies lokāli. Kā redzams, ja kāds brīvi norādītu UID, piekļūstot datiem attālinātā datorā, notiktu *Sliktas Lietas (tm)*. Lai kaut nedaudz novērstu *Slikto Lietu* rašanos, visas direktorijas tiek montētas ar iestatījumu `root_squash`. Šis iestatījums nosaka, ka jebkuram lietotājam, kurš paziņos, ka ir `root`, tā UID tiks nomainīts uz citu, tā novēršot `root` piekļuvi eksportētajai direktorijai. `root_squash` tiek ieslēgts pēc noklusējuma, tomēr autori iesaka to vienlaigā papildus norādīt /etc/exports failā.

Ir iespējams arī eksportēt direktoriju tieši no komandrindas, izmantojot komandu `exportfs`:

```
# exportfs -o rw,no_root_squash Bar:/home/foo
```

Šī rindiņa eksportē /home/foo direktoriju un padara to pieejamu datoram `Bar` lasīšanas un rakstīšanas režīmā. Papildus tam, NFS serveris neiedarbina `root_squash`, kas nozīmē, ka jebkurš

**Bar** lietotājs ar UID 0 (**root** UID) varēs piekļūt failiem ar tādām pat privilēģijām, kā **root** lietotājs uz servera. Šīs rindiņas sintakse izskatās divaini (parasti, kad direktorija tiek norādīta formā **dators:/direktorija/fails**, jūs patiesībā norādat failu norādītā datora noteiktajā direktorijā).

Vairāk informācijas iespējams atrast **exports** faila **man** dokumentā.

## 6.nodaļa. X logu sistēma

Sākot ar *Slackware* versiju 10.0, X logu vidi distribuīvā nodrošina X.org. X logu sistēma nodrošina grafisko lietotāja saskarni, pie tam, atšķirībā no *Windows* vai *MaxOS*, veic to neatkarīgi no operētājsistēmas.

X logu sistēma sastāv no daudzām programmām, kuras darbojas lietojumprogrammu telpā. Divi galvenie sistēmas komponenti ir serveris un logu pārvaldnieks. Serveris nodrošina zemā līmeņa funkcionalitāti sadarbībai ar videoaparāturu, tādēļ tas ir atkarīgs no sistēmas. Logu pārvaldnieks sadarbojas ar serveri un nodrošina lietotāja saskarni. Šādas metodes ieguvums ir iespēja veidot dažādas lietotāja saskarnes, vienkārši nomainot logu pārvaldnieku.

X logu sistēmas konfigurēšana var būt sarežģīta, jo PC arhitektūrai eksistē liels skaits dažādu videokaršu ar dažādām programmēšanas saskarnēm. Par laimi, lielākā daļa mūsdienu videokaršu uztur pamatstandartu, sauktu par VESA, un, ja jūsu datora videokarte ir viena no tām, jūs varat iedarbināt X logu vidi, vienkārši izmantojot komandu `startx`.

Ja tas nedarbojas, vai ja jūs vēlaties izmantot sava datora videokartes augstas veiktspējas iespējas, kā aparatiskā attēla zīmēšanas paātrināšana vai 3D aparatūras atbalsts, jums nāksies pārkonfigurēt X vidi.

Lai konfigurētu X, jums būs jāizveido `/etc/x11/xorg.conf` failu. Šis fails satur daudz informācijas par jūsu videoaparāturu, peli un monitoru. Tas ir ļoti sarežģīts konfigurācijas fails, tomēr par laimi eksistē vairākas programmas, kas palīdzēs jums to izveidot. Mēs aplūkosim dažas no tām.

### 6.1. *xorgconfig*

Šis ir vienkāršs, uz izvēlnēm balstīts, *Slackware* instalēšanas programmai līdzīgs rīks. Tas pieprasa X serverim noteikt videokarti un sagatavo vispiemērotāko sākotnējo konfigurācijas failu, kādu iespējams sagatavot ar saņemto informāciju. Izveidotais `/etc/x11/xorg.conf` fails ir labs sākums lielākajai daļai sistēmu (patiesībā tam būtu jādarbojas bez izmaiņām).

Šis rīks paredzēts pieredzējušiem sistēmu administratoriem. Aplūkosim vienkāršu scenāriju darbam ar `xorgconfig`. Vispirms startējiet programmu:

```
# xorgconfig
```

Parādīsies ekrāns ar informāciju par `xorgconfig`. Lai turpinātu, nospiediet tastatūras taustiņu **ENTER**. `xorgconfig` pārjautās, vai jums ir pareizi iestatīts `PATH` mainīgais. Tam būtu jābūt pareizam, tādēļ šeit nospiediet tastatūras taustiņu **ENTER**.

```
First specify a mouse protocol type. Choose one from the following list:

1. Auto
2. SysMouse
3. MouseSystems
4. PS/2
5. Microsoft
6. Busmouse
7. IMPS/2
8. ExplorerPS/2
9. GlidePointPS/2
10. MouseManPlusPS/2
11. NetMousePS/2
12. NetScrollPS/2
13. ThinkingMousePS/2
14. AceCad

The recommended protocol is Auto. If you have a very old mouse
or don't want OS support or auto detection, and you have a two-button
or three-button serial mouse, it is most likely of type Microsoft.

Enter a protocol number:
```

Nākamajā izvēlnē izvēlieties jūsu datoram pieslēgto peli. Ja jūs sarakstā neatrodas jūsu datoram pieslēgto seriālo peli, izvēlieties *Microsoft* protokolu - tas ir visbiežāk sastopamais un, ticamākais, darbosies korekti. Pēc tam `xorgconfig` uzdos jums jautājumus par `ChordMiddle` un `Emulate3Buttons` izmantošanu. Šos iestatījumu aprakstus iespējams izlasīt ekrānā. Izmantojiet šos iestatījumus tad, ja jūsu peles vidējā poga nedarbojas X logu vidē, vai, ja jūsu datora pelei ir tikai divas pogas (`Emulate3Buttons` ļauj simulēt trešo pogu, vienlaicīgi nospiežot abas pārējās pogas). Tad ievadiet jūsu peles iekārtas faila nosaukumu. Parasti darbosies noklusētā vērtība - `/dev/mouse`, jo tā tika uzstādīta jau *Slackware* instalēšanas laikā. Ja jūs darbināt `GPM` (*Linux* peles serveri) *repeater* režīmā, jūs varat uzstādīt peles tipu uz `/dev/gpmdata`, lai ļautu X saņemt informāciju par peli, izmantojot `GPM`. Dažos gadījumos (īpaši, izmantojot kopnes peles) šī metode varētu darboties labāk, tomēr lielākajai daļai lietotāju nevajadzētu izmantot šo metodi.

`xorgconfig` arī vēlēsies precizēt informāciju par īpašām tastatūras taustiņu piesaistēm. Ja jums tās ir nepieciešamas, atbildiet "y". Lielākā daļa lietotāju var droši atbildēt "n".

```
You must indicate the horizontal sync range of your monitor. You can either
select one of the predefined ranges below that correspond to industry-
standard monitor types, or give a specific range.
```

```
It is VERY IMPORTANT that you do not specify a monitor type with a horizontal
sync range that is beyond the capabilities of your monitor. If in doubt,
choose a conservative setting.
```

```
hsync in kHz: monitor type with characteristic modes
1 31.5; Standard VGA, 640x480 @ 60 Hz
2 31.5 - 35.1; Super VGA, 800x600 @ 56 Hz
3 31.5, 35.5; 8514 Compatible, 1024x768 @ 87 Hz interlaced (no 800x600)
4 31.5, 35.15, 35.5; Super VGA, 1024x768 @ 87 Hz interlaced, 800x600 @ 56 Hz
5 31.5 - 37.9; Extended Super VGA, 800x600 @ 60 Hz, 640x480 @ 72 Hz
6 31.5 - 48.5; Non-Interlaced SUGA, 1024x768 @ 60 Hz, 800x600 @ 72 Hz
7 31.5 - 57.0; High Frequency SUGA, 1024x768 @ 70 Hz
8 31.5 - 64.3; Monitor that can do 1280x1024 @ 60 Hz
9 31.5 - 79.0; Monitor that can do 1280x1024 @ 74 Hz
10 31.5 - 82.0; Monitor that can do 1280x1024 @ 76 Hz
11 Enter your own horizontal sync range
```

```
Enter your choice (1-11):
```

Nākamajā sadaļā ievadiet sava monitora sinhronizācijas diapazonu. Lai uzsāktu monitora konfigurēšanu, nospiediet **ENTER**. Jūs redzēsiet monitoru tipu sarakstu - izvēlieties vienu no tiem, uzmanoties no monitora specifikāciju pārsniegšanas - tas var sabojāt jūsu aparatūru.

```
You must indicate the vertical sync range of your monitor. You can either
select one of the predefined ranges below that correspond to industry-
standard monitor types, or give a specific range. For interlaced modes,
the number that counts is the high one (e.g. 87 Hz rather than 43 Hz).
```

- 1 50-70
- 2 50-90
- 3 50-100
- 4 40-150
- 5 Enter your own vertical sync range

```
Enter your choice: █
```

Norādiet sava monitora vertikālās sinhronizācijas diapazonu (to iespējams atrast monitora rokasgrāmatā). `xorgconfig` pieprasīs ievadīt rindiņas, kas identificēs monitora tipu `xorg.conf` failā. Ievadiet jebko, ko vien vēlaties (izņemot tukšu rindiņu, protams).

- ```
0 * Generic VESA compatible -
1 * Generic VGA compatible -
2 * Unsupported VGA compatible -
3 ** 3DLabs, TI (generic) [glint] -
4 ** 3Dfx (generic) [tdfx] -
5 ** ATI (generic) [ati] -
6 ** ATI Radeon (generic) [radeon] -
7 ** ATI Rage 128 based (generic) [r128] -
8 ** Alliance Pro Motion (generic) [apm] -
9 ** Ark Logic (generic) [ark] -
10 ** Chips and Technologies (generic) [chips] -
11 ** Cirrus Logic (generic) [cirrus] -
12 ** Cyrix MediaGX (generic) [cyrix] -
13 ** DEC TGA (generic) [tga] -
14 ** Intel i740 (generic) [i740] -
15 ** Intel i810 (generic) [i810] -
16 ** Linux framebuffer (generic) [fbdev] -
17 ** Matrox Graphics (generic) [mga] -
```

```
Enter a number to choose the corresponding card definition.
Press enter for the next page, q to continue configuration.
```

Tagad jums ir iespēja aplūkot videokaršu tipu datu bāzi. Jūs noteikti vēlēsit to veikt, tādēļ atbildiet "y" un izvēlieties karti no piedāvātā saraksta. Ja jūs neredzat precīzu jūsu datora videokartes

nosaukumu, mēģiniet izvēlēties citu, kas izmanto to pašu mikroshēmu komplektu - tam būtu jādarbojas.

Pēc tam informējiet `xorgconfig` par jūsu datora videokartes videoatmiņas apjomu. `xorgconfig` arī vēlēšies saņemt aprakstošu tekstu par jūsu videokarti. Ja vēlaties, šajās trīs rindiņās varat ievadīt videokartes aprakstu.

Jums uzdos jautājumu, kādas ekrāna izšķirtspējas izmantot. Piedāvāto noklusēto vērtību izmantošana sākumam noteikti ir pietiekoša. Vēlāk būs iespējams rediģēt `/etc/X11/xorg.conf` failu, lai pārkārtotu režīmus tā, ka 1024x768 (vai jebkurš cits, ko vēlaties) būtu noklusētais režīms.

Šajā vietā `xorgconfig` programma uzdos jums jautājumu, vai vēlaties saglabāt izveidoto konfigurācijas failu. Atbildiet `yes`, un X konfigurācijas fails tiks saglabāts, tā pabeidzot konfigurēšanas procesu. Tagad, izmantojot komandu `startx`, ir iespējams startēt X logu sistēmu.

## 6.2. xorgsetup

Otrs veids, kā konfigurēt X logu sistēmu, ir `xorgsetup` izmantošana. Tā ir automātiska konfigurācijas programma, kas iekļauta *Slackware* distributīvā.

Lai izpildītu `xorgsetup`, autorizējieties kā `root` un ievadiet komandu:

```
# xorgsetup
```

Ja jūsu sistēmā jau ir izveidots fails `/etc/X11/xorg.conf` (tas tādēļ, ka jūs jau esat konfigurējis X), jums tiks piedāvāts izveidot esošā konfigurācijas faila rezerves kopiju. Oriģinālais fails tiks pārsaukts par `/etc/X11/xorg.conf.backup`.

## 6.3. xinitrc

`xinit(1)` ir programma, kas startē X logu sistēmu. Tā kā to izsauc `startx(1)`, jūs to varētu nepamanīt (un, domājams, jums nemaz tas nav vajadzīgs). Toties šīs programmas konfigurācijas fails nosaka, kuras programmas (arī logu pārvaldnieku) startēt. `xinit` vispirms pārbauda jūsu mājas direktoriju un meklē tajā failu `.xinitrc`. Ja šāds fails ir atrodams, tas tiek izpildīts - pretējā gadījumā tiek izpildīts `/var/X11R6/lib/xinit/xinitrc` (sistēmas noklusētais fails). Piemērs vienkāršam `xinitrc` failam:

```
#!/bin/sh
# $XConsortium: xinitrc.cpp,v 1.4 91/08/22 11:41:34 rws Exp $

userresources=$HOME/.Xresources
usermodmap=$HOME/.Xmodmap
sysresources=/usr/X11R6/lib/X11/xinit/.Xresources
sysmodmap=/usr/X11R6/lib/X11/xinit/.Xmodmap

# merge in defaults and keymaps

if [ -f $sysresources ]; then
    xrdp -merge $sysresources
fi

if [ -f $sysmodmap ]; then
    xmodmap $sysmodmap
fi
```

```

if [ -f $userresources ]; then
    xrdb -merge $userresources
fi

if [ -f $usermodmap ]; then
    xmodmap $usermodmap
fi

# start some nice programs

twm &
xclock -geometry 50x50-1+1 &
xterm -geometry 80x50+494+51 &
xterm -geometry 80x20+494-0 &
exec xterm -geometry 80x66+0+0 -name login

```

Šie visi `if` bloki nepieciešami, lai saliedētu dažādos citos failos iekļautos konfigurācijas iestatījumus. Interesantākā faila daļa ir beigās, kur tiek startētas dažādas programmas. X sesija sāksies ar logu pārvaldnieku `twm(1)`, pulksteni un trim termināļiem. Ievērojiet komandu `exec` pirms pēdējā `xterm`. Tā aizstāj tekošo komandrindas čaulu (to, kura darbona šo `xinitrc` skriptu) ar `xterm`. Tas nozīmē, ka, lietotājam aizverot šo `xterm`, X sesija beigs darbu.

Lai pielāgotu savu X startēšanos, nokopējiet noklusēto `/var/X11R6/lib/xinit/xinitrc` uz `~/xinitrc` un modificējiet to, aizstājot šīs rindiņas ar to, ko vēlaties. Mana (autora - tulk. piez.) `xinitrc` faila beigas izskatās šādi:

```

# Start the window manager
exec startkde

```

Ievērojiet, ka `/var/X11R6/lib/xinit` direktoriņā atrodas vairāki `xinitrc.*` faili, kas atbilst dažādiem logu pārvaldniekiem un grafiskajām vidēm. jūs varat izmantot jebkuru no tiem.

## 6.4. xwmconfig

*Unix* tipa sistēmas gadiem ilgi tika izmantotas praktiski tikai kā serveru operētājsistēmas, izņemot vienīgi augstas klases profesionālās darbstacijas. Ar *Unix* tipa sistēmām strādāja tikai izredzētie, un lietotāja saskarne atspoguļoja šo faktu. Grafiskie interfeisi bija primitīvi, veidoti, lai darbinātu dažas grafiskās programmas, kurām gluži vienkārši bija jābūt grafiskām - kā CAD programmas un attēlu apstrādes sistēmas. Lielākā daļa failu un sistēmas vadības notika, izmantojot komandrindu. Dažādi programmatūras izstrādātāji (*Sun Microsystems*, *Silicon Graphics* u.c.) centās izveidot darbstacijas ar vienotu, ērtu lietotāja saskarni, tomēr dažādie interfeisa rīki un bibliotēkas radīja priekšstatu, ka *Unix* darbvirsma vide nekad nebūs vienota. Par ritjosla divās dažādās programmās varēja izskatīties dažādi. Izvēlnes varēja parādīties dažādās vietās. Programmām bija dažādas ekrānpogas un izvēles rutiņas. Krāsas mainījās ļoti plaši un bieži vien bija iekļautas atbilstošās bibliotēkas pirmkodā - t.i., nebija maināmas. Tomēr jāatzīst, ka, kamēr šīs sistēmas lietoja pārsvarā profesionāļi, tas neradīja sevišķas problēmas.

Parādoties brīvām *Unix* tipa operētājsistēmām un pieaugot grafisko programmu skaitam un dažādībai, X logu sistēmai pēdējo gadu laikā ir radušies daudz jauni lietotāji. Lielākā daļa lietotāju, protams, ir pieraduši pie *Microsoft Windows* vai *Apple MacOS* piedāvātā vienotā saskarnes stila; šāda vienotības trūkums X programmās kļuva par traucēkli to plašākai ieviešanai. Kā risinājums šai problēmai ir radušies divi atvērtā pirmkoda projekti - **K Darbvirsma vide** (K Desktop Environment, **KDE**) un **GNU Tīkla objektu modeļa vide** (GNU Network Object Model Environment, **GNOME**). Katrai no šīm vidēm ir izveidotas daudzas programmas, sākot ar rīkjoslām un failu pārvaldniekiem, un beizot ar spēlēm un



biroja pakotnēm - un tās visas ir izstrādātas, izmantojot vienu un to pašu grafiskā interfeisa rīku un ir savstarpēji cieši integrētas, lai sniegtu lietotājam vienotu darbvirsma vidi.

Vispārīgā gadījumā atšķirības starp KDE un GNOME ir samērā smalkas. Katra no tām izskatās atšķirīgi no otras, jo katra izmanto atšķirīgu grafiskās saskarnes rīku komplektu. KDE ir veidots, izmantojot *AS Trolltech* izstrādāto *Qt* bibliotēku, kamēr GNOME izmanto *GTK*, rīku komplektu, kas sākotnēji tika izveidots GNU Attēlu manipulācijas programmai (GNU Image Manipulation Program, GIMP). Kā atsevišķiem projektiem, KDE un GNOME katram ir savi dizaineri un programmētāji ar dažādiem programmēšanas stiliem un filozofiju. Jebkurā gadījumā, rezultāts pamatā ir viens - vienota, cieši saistīta darbvirsma vide un programmu kolekcija. KDE un GNOME funkcionalitāte, lietojamība un skaistums spēj konkurēt ar jebkuru citu operētājsistēmu.

Visjaukākais, protams, ir tas, ka šīs lieliskās darbvirsma vides ir brīvas. Tas nozīmē, jūs varat izmantot jebkuru no tām, vai pat abas vienlaikus. Izvēle ir jūsu.

Papildus KDE un GNOME darbvirsām *Slackware* ir iekļauti arī daudzi logu pārvaldnieki. Daži no tiem ir izveidoti, lai emulētu citas operētājsistēmas, daži - pielāgojamībai, pārējie - ātrdarbībai. To ir diezgan daudz. Protams, jūs varat instalēt tik daudz logu pārvaldnieku, cik vēlaties, paspēlējies ar katru no tiem, un izvēlēties - kurš no tiem jums patīk vislabāk.

Lai padarītu darbvirsma izvēli pēc iespējas vienkāršāku, *Slackware* komplektā ir iekļauta programma `xwmconfig`, kuru izmanto, lai izvēlētos darbvirsma vai logu pārvaldnieku. To izpilda šādi:

```
% xwmconfig
```



Jums tiks piedāvāts saraksts ar visām darbvirsma vidēm un logu pārvaldniekiem, kādi uzstādīti jūsu sistēmā. Vienkārši izvēlieties vēlamu no saraksta. Šo programmu jāizpilda katram sistēmas lietotājam, jo dažādi lietotāji var izmantot dažādas darbvirsma, un ne jau katrs vēlēties izmantot to, kas izvēlēta, sistēmu instalējot.

Kad izvēle veikta, startējiet X logu sistēmu.

## 6.5. xdm

Linux kļūstot arvien noderīgākai kā galddatoru operētājsistēmai, daudzi lietotāji uzskata par lietderīgu to konfigurēt tā, lai tūlīt pēc sistēmas ielādes automātiski tiktu startēta arī grafiskā vide. Lai to panāktu, jums jānorāda *Slackware*, ka tai jāstartē X logu sistēma un jānorāda grafisko autorizācijas pārvaldnieku. Slackware ir iekļauti trīs grafiskie autorizācijas pārvaldnieki - **x<sub>dm</sub>**(1), **k<sub>dm</sub>** un **g<sub>dm</sub>**(1).

**x<sub>dm</sub>** ir grafiskais autorizācijas pārvaldnieks, kas iekļauts X.org sistēmā. Tas veic savu darbu, tomēr nav tik iespējām bagāts, kā alternatīvas. **k<sub>dm</sub>** ir autorizācijas pārvaldnieks, kas iekļauts KDE komplektā, bet **g<sub>dm</sub>** - GNOME komplektā. Jebkurš no tiem ļaus jums autorizēties sistēmā un izvēlēties darbvirsmas vidi.

Diemžēl *Slackware* komplektā nav iekļauta programma kā **x<sub>wm</sub>config**, kas ļautu jums izvēlēties grafisko autorizācijas pārvaldnieku, tādēļ, ja sistēmā uzstādīti visi trīs, jums var nākties nedaudz palabot konfigurācijas failus. Tomēr vispirms noskaidrosim, kā ielādēt sistēmu grafiskajā vidē.

Lai startētu X logu sistēmu pie sistēmas ielādes, jums jāielādējas darblīmenī 4. Darblīmeņi ir veids, kā likt **init**(8) veikt tādas vai citādas darbības sistēmas ielādes laikā. Mums nepieciešamo norādīsim, modificējot **init** konfigurācijas failu **/etc/inittab**.

```
# These are the default runlevels in Slackware
# 0 - halt
# 1 - single user mode
# 2 - unused (but configured the same as runlevel 3)
# 3 - multiuser mode (default Slackware runlevel)
# 4 - X11 with KDM/GDM/XDM (session managers)
# 5 - unused (but configured the same as runlevel 3)
# 6 - reboot

# Default runlevel. (Do not set to 0 or 6)
id:3:initdefault:
```

Lai liktu Slackware ielādēties grafiskajā vidē, vienkārši nomainīsim 3 uz 4.

```
# Default runlevel. (Do not set to 0 or 6)
id:4:initdefault:
```

Tagad Slackware ielādēsies darblīmenī 4 un izpildīs **/etc/rc.d/rc.4**. Šis fails startēs X un izsauks izvēlēto grafisko autorizācijas pārvaldnieku. Tātad - kā izvēlēties pārvaldnieku? Ir vairāki veidi, kā to veikt, un es izskaidrošu tos pēc tam, kad būsīm aplūkojuši **rc.4**.

```
# Try to use GNOME's gdm session manager:
if [ -x /usr/bin/gdm ]; then
    exec /usr/bin/gdm -nodaemon
fi

# Not there? OK, try to use KDE's kdm session manager:
if [ -x /opt/kde/bin/kdm ]; then
    exec /opt/kde/bin/kdm -nodaemon
fi

# If all you have is XDM, I guess it will have to do:
if [ -x /usr/X11R6/bin/xdm ]; then
    exec /usr/X11R6/bin/xdm -nodaemon
fi
```

Kā redzams, `rc.4` vispirms pārbauda, vai ir pieejams un izpildāms `gdm`. Ja tā ir, tas tiek izpildīts. Sarakstā nākamais ir `kdm`, un, visbeidzot - `xdm`. Viens no veidiem, kā izvēlēties autorizācijas pārvaldnieku, ir vienkārši nodzēst liekos, izmantojot `removepkg`. Vairāk informācijas par `removepkg` atradīsiet 18.nodaļā.

Bez tam, ir iespējams vienkārši noņemt izpildes tiesības failiem, kurus jūs nevēlaties izmantot. `chmod` ir aplūkots 9.nodaļā.

```
# chmod -x /usr/bin/gdm
```

Beidzot, ir iespējams vienkārši aizkomentēt rindiņas, kas attiecas uz tiem autorizācijas pārvaldniekiem, kurus jūs nevēlaties izmantot.

```
# Try to use GNOME's gdm session manager:
# if [ -x /usr/bin/gdm ]; then
#   exec /usr/bin/gdm -nodaemon
# fi

# Not there? OK, try to use KDE's kdm session manager:
if [ -x /opt/kde/bin/kdm ]; then
  exec /opt/kde/bin/kdm -nodaemon
fi

# If all you have is XDM, I guess it will have to do:
if [ -x /usr/X11R6/bin/xdm ]; then
  exec /usr/X11R6/bin/xdm -nodaemon
fi
```

Jebkuras rindiņas, kuras sākas ar `#`, tiek uzskatītas par komentāriem un komandrindas čaula tās neapstrādā. Tādēļ, ja arī `gdm` ir instalēts un izpildāms, komandrinda (šajā gadījumā `bash`) to nemaz nemēģinās izpildīt.

## 7.nodaļa. Sistēmas ielāde

Dažkārt *Linux* sistēmas ielāde var būt vienkārša, tomēr dažkārt - sarežģīta. Daudzi lietotāji vienkārši uzstāda *Slackware* savā datorā un ar to viss beidzas - viņi ieslēdz datoru un tas ir gatavs lietošanai. Reizēm pat vienkārša datora ielāde var būt nepatīkama. Lielākajai daļai lietotāju vislabāk darbosies LILO, kurš kopā ar *Loadlin* ir pieejams *Slackware* komplektā kā viens no sistēmas ielādes rīkiem. LILO var ielādēt sistēmu no diska sadaļas, no diska galvenā ielādes sektora (MBR - Master Boot Record), vai no disketes, kas padara šo rīku ļoti jaudīgu. *Loadlin* darbojas no DOS komandrindas, izlādējot DOS no datora atmiņas un tā vietā ielādējot *Linux*.

Vēl kāds populārs rīks *Linux* ielādei ir GRUB. GRUB nav iekļauts *Slackware* distributīvā un netiek oficiāli atbalstīts. Kad jāizvēlas distributīvā iekļaujamās komponentas, *Slackware* pieturas pie pārbaudītām vērtībām, tādēļ, lai arī GRUB darbojas labi un tam pat ir iespējas, kādas nav LILO, LILO lieliski veic savu darbu un to ir pierādījis laika gaitā. GRUB, kas ir jaunāks ielādes rīks, vēl nav gluži sasniedzis šo briedumu un, tā kā tas nav iekļauts *Slackware*, tad mēs to šeit neapspriedīsim. Ja jūs vēlaties izmantot GRUB (ticamākais, tas ticis iekļauts citā *Linux OS* un jūs vēlaties to izmantot sistēmas duālai ielādei), lūdzam izlasīt GRUB dokumentāciju.

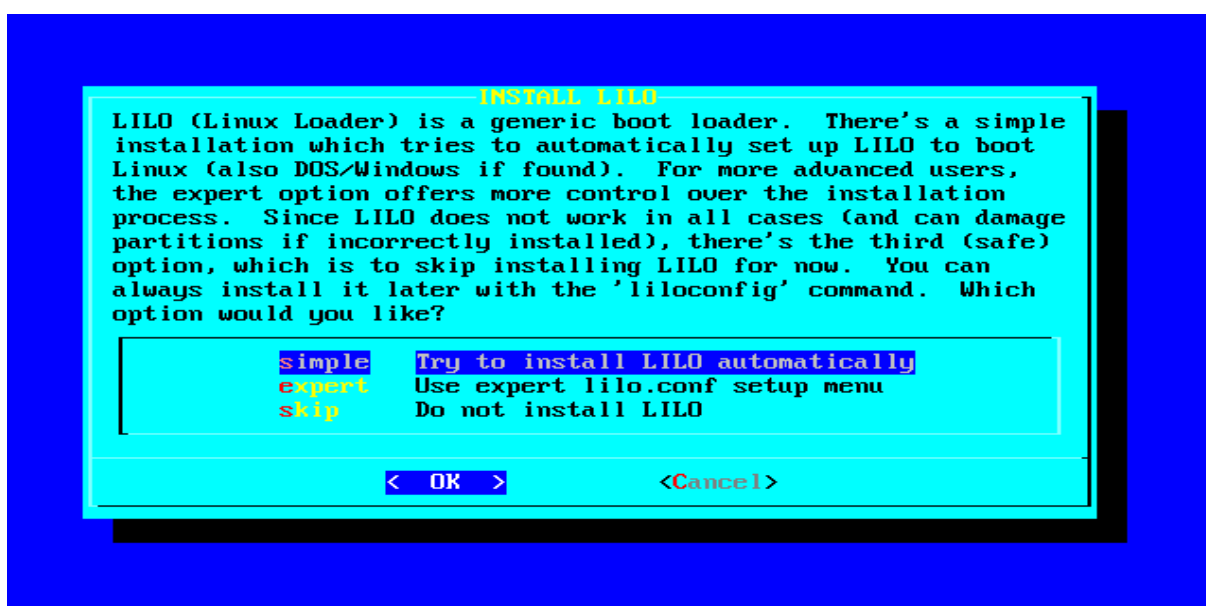
### 7.1. LILO

*Linux Loader* jeb LILO ir vispopulārākais *Linux* sistēmu ielādes rīks. Tas ir samērā plaši pielāgojams un konfigurējams un to var viegli izmantot citu operētājsistēmu ielādei.

*Slackware Linux* ir iekļauts uz izvēlnēm balstīts LILO konfigurācijas rīks - `liloconfig`. Šī programma tiek izpildīta jau sistēmas uzstādīšanas procesā, tomēr jūs varat to izpildīt arī vēlāk, komandrindā ievadot komandu `liloconfig`.

LILO savus iestatījumus nolasa no faila `/etc/lilo.conf(5)`. Tas netiek lasīts katru reizi, kad tiek veikta sistēmas ielāde, bet gan katru reizi, kad LILO tiek instalēts. Katru reizi, kad tiek veiktas konfigurācijas izmaiņas, LILO ir atkārtoti jāinstalē ielādes sektorā. Daudzas LILO darbības kļūdas rodas, veicot izmaiņas `lilo.conf` failā, bet neizpildot `lilo`, lai instalētu šīs izmaiņas. `liloconfig` jums palīdzēt izveidot konfigurācijas failu, ar kura palīdzību iespējams LILO instalēt jūsu sistēmā. Ja jūs labāk izvēlaties manuāli rediģēt failu `/etc/lilo.conf`, tad LILO pārinstalēšana notiek, ar `root` tiesībām komandrindā izpildot `/sbin/lilo`.

Kad `liloconfig` tiek izpildīts pirmo reizi, tas izskatīsies šādi:



Ja šī ir pirmā reize, kad uzstādat LILO, jums būtu jāizvēlas `simple`. Ja jums ir pieredze LILO konfigurēšanā, `expert` varētu šķist ātrāks. `simple` izvēle sāks LILO konfigurēšanu.

Ja jūsu kodols kompilēts ar ekrānbufera (*frame buffer*) atbalstu, `liloconfig` uzdos jums jautājumu par vēlamu izšķirtspēju. Šī izšķirtspēja tiks izmantota arī XFree86 ekrānbufera serverī. Ja jūs nevēlaties darbināt komandrindu īpašā videorežīmā, izvēlieties `normal` - tas ļaus jums izmantot standarta 80x25 teksta režīmu.

Nākamā LILO konfigurēšanas procesa daļa ir instalēšanas mērķa izvēle. Šķiet, šis ir vissvarīgākais solis. Šajā sarakstā izskaidroti dažādie instalēšanas mērķi:

## Root

Šī iespēja instalēs LILO jūsu Linux `root` sadaļas sākumā. Šī ir visdrošākā metode, ja jūsu datorā ir instalētas arī citas operētājsistēmas. Tā nodrošina, ka netiek pārrakstīta neviena cita ielādes programmatūra. Šai metodei ir arī trūkums - LILO spēs ielādēties no šīs sadaļas tikai tad, ja tā ir pirmā sadaļa, tādēļ daudzi izvēlas izveidot ļoti mazu `/boot` sadaļu kā pirmo disku. Tas vienmēr ļauj kodolu un LILO instalēt diska sākumā, kur LILO var tos atrast. Iepriekšējām LILO versijā bija arī bēdīgi slavenais trūkums, kas bija zināms kā "1024 cilindru ierobežojums". LILO nespēja ielādēt kodolus no sadaļām, kas izvietotas tālāk par 1024 cilindru. Jaunākajās LILO versijās šī problēma ir novērsta.

## Floppy

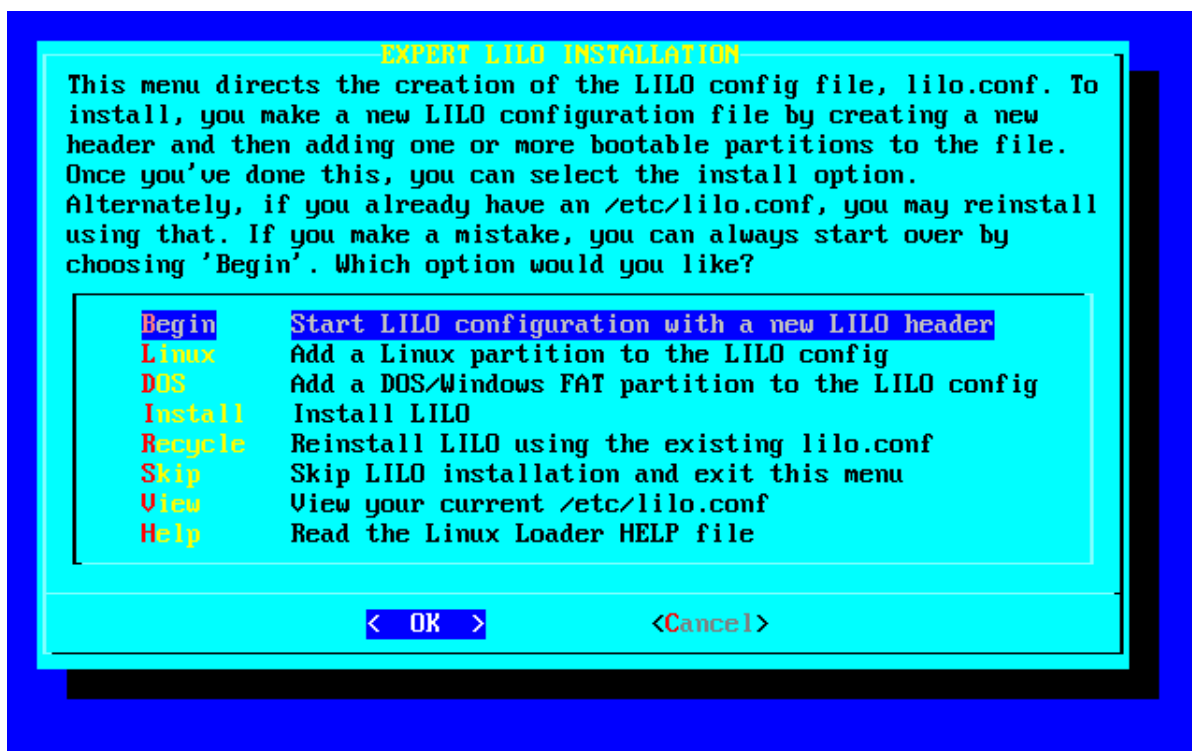
Šī metode ir pat drošāka par iepriekšējo. Tā sagatavo ielādes disketi, ko var izmantot Linux sistēmas ielādei. Tā nodrošina, ka ielādes rīks neatrodas cietajā diskā, tādēļ šo disketi jāizmanto tikai, lai ielādētu *Slackware*. Šīs metodes trūkumi ir acīmredzami. Disketes ir trauslas un viegli sabojājamas. Bez tam ielādes rīks nav integrēts datorā. Ja jūs pazaudējat ielādes disketi, jums jāpagatavo citu, lai ielādētu savu datoru.

## MBR

Jūs vēlēsit izmantot šo metodi, ja *Slackware* ir vienīgā operētājsistēma datorā, vai arī, ja vēlaties izmantot LILO, lai izvēlētos starp dažādām uzstādītajām operētājsistēmām. Šī ir vispiemērotākā LILO instalēšanas metode un darbosies gandrīz jebkurā datorā.

**Brīdinājums: šī metode pārrakstīs jebkuru citu MBR instalēto ielādes programmatūru.**

Pēc instalēšanas vietas izvēles `liloconfig` ierakstīs konfigurācijas failu un instalēs LILO. Tas arī viss. Ja jūs izvēlēsit eksperta režīmu (`Expert`), parādīsies īpaša izvēlne, kas ļaus jums pielāgot `/etc/lilo.conf` failu, pievienot ielādes izvēlei citas operētājsistēmas, kā arī norādīt, vai LILO nepieciešams nodot sistēmas kodolam īpašus ielādes parametrus. Eksperta izvēlne izskatās šādi:



Lai arī kāda nebūtu jūsu sistēmas konfigurācija, ielādes rīka uzstādīšana ir vienkārša. Tādu to padara liloconfig.

## 7.2. LOADLIN

Otra *Slackware* iekļautā ielādes iespēja ir LOADLIN. LOADLIN ir DOS izpildāmfails, ko iespējams izmantot, lai ielādētu *Linux* no strādājošas DOS sistēmas. Tam nepieciešams, lai *Linux* kodols atrastos DOS sadaļā, lai LOADLIN varētu to ielādēt un korekti startēt sistēmu.

Instalēšanas procesā LOADLIN tiks ierakstīts *root* mājas direktorijā kā *.ZIP* fails. LOADLIN nav izveidots automātisks uzstādīšanas process. jums būs jānokopē *Linux* kodolu (parasti /boot/vmlinuz) un LOADLIN failu no *root* mājas direktorijas uz DOS sadaļu.

LOADLIN ir noderīgs, ja jūs vēlaties izveidot ielādes izvēlni savā DOS sadaļā. To iespējams pievienot *AUTOEXEC.BAT* failam, kurā varētu izvēlēties *Linux* vai DOS. *Linux* izvēle izpildītu LOADLIN, tā ielādējot jūsu *Slackware* sistēmu. Šāds *AUTOEXEC.BAT* fails *Windows 95* vidē nodrošinās pietiekošu ielādes izvēlni:

```
@ECHO OFF
SET PROMPT=$P$G
SET PATH=C:\WINDOWS;C:\WINDOWS\COMMAND;C:\
CLS
ECHO Please Select Your Operating System:
ECHO.
ECHO [1] Slackware Linux
ECHO [2] Windows 95
ECHO.
CHOICE /C:12 "Selection? -> "
IF ERRORLEVEL 2 GOTO WIN
IF ERRORLEVEL 1 GOTO LINUX
```

```
:WIN
CLS
ECHO Starting Windows 95...
WIN
GOTO END
:LINUX
ECHO Starting Slackware Linux...
CD \LINUX
LOADLIN C:\LINUX\VMLINUX ROOT=<root partition device> RO
GOTO END
:END
```

Jūs vēlēsities norādīt savu `root` sadaļu kā Linux iekārtas nosaukumu, piemēram, `/dev/hda2` vai ko līdzīgu. Jūs vienmēr varat izmantot `LOADLIN` komandrindā. To izmanto līdzīgi, kā iepriekšminētajā piemērā. `LOADLIN` dokumentācijā iekļauti daudzi izmantošanas piemēri.

### 7.3. Duālā ielāde

Daudzi lietotāji vēlas noskaņot savus datorus tā, lai tie varētu ielādēt gan *Slackware Linux*, gan arī kādu citu operētājsistēmu. Ja jums rodas problēmas, mēs esam aprakstījuši vairākus tipiskus duālās ielādes scenārijus.

## Windows

Datora uzstādīšana ar *MS Windows* un *Linux*, šķiet, ir visbiežāk sastopamais duālās ielādes scenārijs. Ir daudzi veidi, kā šo scenāriju realizēt, tomēr šajā sadaļā aplūkoti divi.

Bieži vien, sagatavojot duālās ielādes sistēmu, lietotājs sagatavo lielisku plānu, apzinoties, kam kur ir jāatrodas, tomēr sajauc instalēšanas secību. Ir ļoti svarīgi saprast, ka duālās ielādes gadījumā operētājsistēmas ir jāinstalē pareizā secībā. *Linux* vienmēr piedāvā izvēlēties, kas un vai tiek rakstīts *MBR*, tādēļ *Linux* ir ieteicams instalēt kā pēdējo. *Windows* vienmēr jāuzstāda kā pirmo, jo tā vienmēr ierakstīs savu ielādes programmu *MBR*, pārrakstot visu, ko *Linux* tur varētu būt ierakstījis.

### LILO izmantošana

Vairums lietotāju izvēlei starp *Linux* un *Windows* vēlēties izmantot LILO. Kā jau rakstīts iepriekš, vispirms jāuzstāda *Windows*, tad *Linux*.

Pieņemsim, ka jums ir 40 GiB IDE disks kā vienīgais sistēmas disks. Pieņemsim arī, ka jūs vēlaties atvēlēt pusi no šīs ietilpības *Windows* un otru pusi - *Linux*. Tas nozīmē, ka radīsies problēma, veicot *Linux* ielādi.

```
20 GB    Windows boot (C:)
 1 GB    Linux root (/)
19 GB    Linux /usr (/usr)
```

Jūs arī varētu vēlēties rezervēt atbilstošu vietu *Linux* maiņatmiņas sadaļai. Ir nerakstīts likums - izmantot apjomu, kas divas reizes lielāks par jūsu datora RAM - 64 MB sistēma izmantotu 128 MB maiņatmiņas utt. Pareiza maiņatmiņas apjoma izvēle ir iemesls daudziem ideoloģiskajiem kariem *IRC* un *Usenet*. Patiesībā nav tāda īsti "pareiza" veida, tomēr pieturēšanās pie iepriekšminētā likuma vismaz nebūs nepareiza.

Kad sadaļas sagatavotas, jūs varat uzsākt *Windows* instalēšanu. Kad *Windows* ir uzstādīta un darbojas, var instalēt *Linux*. Īpašu uzmanību jāpievērš LILO instalēšanai. Jūs vēlēsities izmantot eksperta režīmu.

Sāciet ar jaunu LILO konfigurāciju. Jūs vēlēsities to instalēt *MBR*, lai to varētu izmantot izvēlei starp divām operētājsistēmām. Izvēlnē pievienojiet savu *Linux* sadaļu, pēc tam *Windows* (vai *DOS*) sadaļu.

Kad tas padarīts, varat instalēt LILO.

Pārstartējiet datoru. LILO būtu jāielādējas un jāpiedāvā izvēlni, ļaujot izvēlēties starp instalētajām operētājsistēmām. Izvēlieties ielādējamās OS nosaukumu (šos nosaukumus jūs ievadījāt, konfigurējot LILO).

LILO ir visnotaļ konfigurējams ielādes rīks. Tas nav vienkārši ierobežots ar *Linux* vai DOS ielādi. Tas var ielādēt gandrīz jebko. Vairāk informācija atrodama `man` dokumentos par `lilo(8)` un `lilo.conf(5)`.

Ko darīt, ja LILO nedarbojas? Ir gadījumi, kuros LILO vienkārši nedarbosies un konkrētā datora. Par laimi, ir vēl kāds veids, kā nodrošināt Linux un Windows duālo ielādi.

### LOADLIN izmantošana

Šo metodi var izmantot, ja LILO nedarbojas, vai, ja jūs nevēlaties uzstādīt LILO. Šī metode ir ideāla arī tiem lietotājiem, kas regulāri pārinstalē *Windows*. Katru reizi, pārinstalējot *Windows*, tas pārrakstīs MBR, tā iznīcinot jebkuru LILO instalāciju. Ar LOADLIN šādas problēmas neradīsies. Lielākais trūkums ir, ka šajā situācijā tikai LOADLIN spēs ielādēt *Linux*.

Ar LOADLIN jūs varat instalēt operētājsistēmas tādā secībā, kā vēlaties. Uzmanieties no datu ierakstes MBR - to jūs nedrīkstat veikt. LOADIN ir atkarīgs no *Windows* sadaļas ielādes spējas, tādēļ *Slackware* instalēšanas laikā izvairieties no LILO instalēšanas.

Kad operētājsistēmas uzstādītas, nokopējiet failu `loadlinX.zip` (kur `X` ir versijas numurs, piemēram `16a`) no `root` mājas direktorijas uz jūsu *Windows* sadaļu. Turpat nokopējiet arī kodola attēlfailu. Tam jums būs nepieciešama *Linux* sistēma. Piemērs:

```
# mkdir /win
# mount -t vfat /dev/hda1 /win
# mkdir /win/linux
# cd /root
# cp loadlin* /win/linux
# cp /boot/vmlinuz /win/linux
# cd /win/linux
# unzip loadlin16a.zip
```

Šīs komandas jūs *Windows* sadaļā (pieņemot, ka tā ir `/dev/hda1`) izveidos `C:\LINUX` direktoriju un tajā nokopēs visu nepieciešamo LOADLIN ielādei. Kad tas būs paveikts, jums būs jāpārstartē *Windows*, lai izveidotu ielādes izvēlni.

Kad *Windows* ielādēts, atveriet DOS komandrindu. Vispirms mums jāpārlicinās, ka sistēma nestartē grafisko saskarni.

```
C:\> cd \
C:\> attrib -r -a -s -h MSDOS.SYS
C:\> edit MSDOS.SYS
```

Šim failam pievienojiet rindiņu

```
BootGUI=0
```

Tagad saglabājiet failu un aizveriet redaktoru. Pēc tam atveriet redaktorā failu `c:\AUTOEXEC.BAT`, lai pievienotu ielādes izvēlni. Šeit norādīts piemērs ielādes izvēlnes blokam:

```
cls
echo System Boot Menu
echo.
echo 1 - Linux
echo 2 - Windows
echo.
```



```
choice /c:12 "Selection? -> "  
if errorlevel 2 goto WIN  
if errorlevel 1 goto LINUX  
:LINUX  
cls  
echo "Starting Linux..."  
cd \linux  
loadlin c:\linux\vmlinux root=/dev/hda2 ro  
goto END  
:WIN  
cls  
echo "Starting Windows..."  
win  
goto END  
:END
```

Svarīgākā ir rindiņa, kas izsauc LOADLIN. Mēs tai norādām, kādu kodolu ielādēt, *Linux root* sadaļu un to, ka vēlamies, lai tā sākotnēji tiktu montēta tikai lasīšanas režīmā.

Rīki šīm divām metodēm ir iekļauti *Slackware Linux*. Ir pieejami arī daudzi citi ielādes rīki, tomēr šie darbosies lielākajā daļā duālās ielādes gadījumā.

### Novocojis Windows NT paņēmieni

Šī ir visretāk sastopamā duālās ielādes situācija. Senajās dienās LILO nespēja ielādēt *Windows NT*, pieprasot *Linux* lietotājiem īpaši pielāgot *NTLDR*, kas radīja vairāk problēmu, kā duālā ielāde ar *Windows 9x* un *Linux*. Lūdzu saprotiet, ka šīs instrukcijas ir novecojušas. LILO jau daudzus gadus ir spējīgs ielādēt *Windows NT/2000/XP/2003*. Tiesa, ja jūs izmantojat novecojušu datoru, tad šī metode var jums noderēt.

1. Instalējiet Windows NT
2. Instalējiet *Linux*, pārlicinoties, ka LILO tiek instalēts *Linux* sadaļas superblokā
3. Pirmos 512 baitus no *Linux root* sadaļas saglabājat *Windows NT* sadaļā
4. Modificējiet *C:\BOOT.INI*, lai pievienotu iespēju ielādēt *Linux*

*Windows NT*, līdzīgi, kā *Linux*, instalēšanai vajadzētu būt vienkāršai. No šī brīža viss kļūst nedaudz sarežģītāks, tomēr *paņemt* pirmos 512 baitus no *Linux* sadaļas ir vienkāršāk, kā izklausās. Šim nolūkam jums būs jāizmanto *Linux*. Pieņemot, ka *Linux* sadaļa ir */dev/hda2*, ievadiet komandu

```
# dd if=/dev/hda2 of=/tmp/bootsect.lnx bs=1 count=512
```

Tas arī viss. Tagad nokopējiet *bootsect.lnx* uz *Windows NT* sadaļu. Šeit var rasties cita problēma. *Linux* nav pieejams stabils rakstīšanas režīma atbalsts NTFS failsistēmai. Ja jūs instalējat *Windows NT* un formatējat savu iekārtu kā NTFS, jums būs jānokopē šo failu FAT disketē un jānolasa to *Windows NT* vidē. Ja jūs formatējat *Windows NT* iekārtu kā FAT, jūs to varat vienkārši piemontēt *Linux* vidē un nokopēt failu. Jebkurā gadījumā jums jānokopē failu */tmp/bootsect.lnx* no *Linux* sistēmas uz *C:\BOOTSECT.LNX* *Windows NT* iekārtā.

Pēdējais solis ir atbilstošas izvēlnes pievienošana. *Windows NT* atveriet komandrindu.

```
C:\WINNT> cd \  
C:\> attrib -r -a -s -h boot.ini  
C:\> edit boot.ini
```

Faila beigās pievienojiet rindiņu

```
C:\bootsect.lnx="Slackware Linux"
```

Saglabājiet izmaiņas un aizveriet redaktoru. Kad pārstartēsiet *Windows NT*, jums ielādes izvēlnē būs pieejama *Linux* ielādes iespēja. Tas izvēle ielādēs *Linux*.

## **Linux**

Jā, arī tā notiek. Šis noteikts ir visvienkāršākais duālās ielādes scenārijs. Jūs vienkārši izmantojiet LILO un pievienojiet papildus rindiņas `/etc/lilo.conf` failā. Tas ir viss, kas šeit jādara.

## 8.nodaļa. Čaula

Grafiskajā vidē lietotāja saskarni nodrošina programma, kas izveido logus, ritjoslas, izvēlnes u.c. Komandrindas vidē lietotāja saskarni nodrošina čaula, kas interpretē komandas un vispār padara sistēmu lietojamu. Tūlīt pēc autorizēšanās (tā aplūkota šajā nodaļā) lietotājs nokļūst čaulā un tam ir atļauts veikt visas tam nepieciešamās (un atļautās) darbības. Šī nodaļa ir ievads čaulā, precīzāk - *Linux* lietotāju vidū vispopulārākajā čaulā - *Bourne Again Shell* (**bash**). Sīkākai informācijai par jebko, kas pieminēts šajā nodaļā aplūkojiet **bash(1)** man dokumentu.

### 8.1. Lietotāji

#### Autorizēšanās

Nu tad jūs esat ielādējuši sistēmu un raugāties uz kaut ko šādu:

```
Welcome to Linux 2.4.18
Last login: Wed Jan  1 15:59:14 -0500 2005 on tty6.
darkstar login:
```

Hmm... neviens neko nav teicis par **login**. Un kas ir **darkstar**? Neuztraucieties - ticamākais, ka jums neizdevās nejauši iedarbināt hiperkosmisko savienojumu ar Impērijas mākslīgo pavadoni (bez tam, es baidos, ka šobrīd *Linux* kodols vēl īsti labi neztur hiperkosmiskā savienojuma protokolu - varbūt 2.8 kodols to beidzot uzturēs). **darkstar** ir vienkārši nosaukums vienam no mūsu datoriem, un tā nosaukumu mēs šeit sniedzam kā piemēru. Ja jūs sistēmas instalēšanas laikā norādījāt sava datora nosaukumu, tad patiesībā jūs šeit redzēsiet to.

Par **login**... (angļu val. *login* - autorizācija - tulk. piez.). Ja šī ir pirmā reize, kad izmantojat savu jauno sistēmu, jūs vēlēsities autorizēties kā **root**. Jums pieprasīs ievadīt arī paroli. Ja jūs tādu norādījāt instalēšanas procesā, tad tā ir tā pati. Ja nē, vienkārši nospiediet **Enter**. Tas arī viss - jūs esat autorizējies!

#### root: superlietotājs

Nu labi, tad KAS ir **root**? Un ko tas dara ar savu lietotāja kontu jūsu sistēmā?

*Unix* un tai līdzīgu operētājsistēmu (arī *Linux*) pasaulē sistēmā ir lietotāji un vēl arī lietotāji. Šajā jautājumā mēs iedziļināsimies vēlāk, tomēr šobrīd svarīgākais ir saprast, ka **root** ir lietotājs, kas ir augstāks jeb privilģētāks par visiem citiem lietotājiem. **root** ir visspēcīgāks un visu zinošs, un **nobody** nevar tam neko padarīt. Tam tas vienkārši nav atļauts. **root** ir tas, ko mēs saucam par superlietotāju, un tā ir pareizi. Pats jaukākais - **root** esat jūs.

Nav slikti, ko?

Ja neesat īsti pārliecināts - jā, tas ir lieliski! Vienīgais knifņš ir tas, ka **root** ir atļauts arī sabojāt jebko, ko tas vēlas. Jūs varētu vēlēties izlaist nākamās sadaļas līdz 12.1 un uzzināt par lietotāja pievienošanu, tad autorizēties kā parasts lietotājs un strādāt ar tādām tiesībām. Tradicionāli ir pieņemts, ka vislabāk par superlietotāju ir kļūt tikai tad, kad tas ir absolūti nepieciešams, lai maksimāli samazinātu iespēju nejauši kaut ko sabojāt.

Starp citu, ja jūs nolemjat, ka vēlaties kļūt par **root**, kamēr esat autorizējies kā cits lietotājs - nav problēmu. Vienkārši izmantojiet **su(1)** komandu. Jums tiks pieprasīts ievadīt **root** paroli un tad jūs iegūsit **root** tiesības līdz brīdim, kamēr izpildīsiet **exit** vai **logout**. Izmantojot **su**, jūs varat iegūt arī jebkura cita lietotāja tiesības, ja vien zināt viņa paroli: **su logan**, piemēram, ļautu jums iegūt manas lietotāja tiesības.

Piezīme: **root** ir atļauts iegūt jebkura lietotāja tiesības bez paroles ievades.

## 8.2. Komandrinda

### Programmu darbināšana

Ir grūti kaut ko panākt, nedarbinot kādu programmu - jūs varētu kaut ko piesegt ar savu datoru, vai aizlikt to priekšā vēja virinātām durvīm; daži datori darbojoties izdvesīs visskaistākās skaņas, tomēr tas nebūs īsti *tas*. Un, domājams, mēs visi varam piekrist tam, ka iespēja tikt aizliktam priekšā durvīm īsti nebūs tas iemesls, kas personālos datorus padarījis tik populārus.

Tātad, atcerieties, kā gandrīz viss *Linux* sistēmā ir fails? Patiesībā tas attiecas arī uz programmām. Jebkura komanda, ko jūs izpildat (un kas nav iebūvēta čaulā), ir ierakstīta kādā failā. Jūs izpildat programmu, norādot pilnu tās faila ceļu un nosaukumu.

Piemēram, atcerieties `su` komandu no iepriekšējās sadaļas? Īstenībā tā atrodas `/bin` direktorijā: `/bin/su` to izpildīs tikpat veiksmīgi.

Kādēļ veiksmīgi darbojas arī vienkārša `su` ievade? Galu galā, jūs nenorādījāt, ka tā atrodas direktorijā `/bin`. Tā tik pat labi varētu atrasties arī direktorijā `/usr/local/share`, ne tā? Kā sistēma to zināja? Atbilde uz šo jautājumu slēpjas vides mainīgajā `PATH`. Lielākā daļa čaulu atbalsta `PATH` vai vismaz kaut ko ļoti līdzīgu. Šis mainīgais satur sarakstu ar direktorijām, kuras jāaplūko, lai atrastu programmas, kuras jūs mēģināt darbināt. Kad jūs izpildījāt komandu `su`, jūsu čaula aplūkoja direktoriju sarakstu, katrā meklējot izpildāmu failu ar nosaukumu `su`. Tā izpildīja pirmo, ko atrada. Šāda meklēšana notiek katru reizi, kad jūs izpildāt programmu, nenorādot pilnu ceļu uz tās failu; ja jūs saņemat kļūdas paziņojumu "`Command not found`" ("`Komanda nav atrasta`"), tas nozīmē tikai to, ka programma, kuru jūs mēģinājāt izpildīt, nav atrodama direktorijās, kuras norādītas vides mainīgajā `PATH` (protams, šis paziņojums parādītos arī tad, ja programma neeksistē nemaz). Vides mainīgos mēs sīkāk aplūkosim sadaļā 8.3.

Atcerieties arī, ka "." ir saīsināts apzīmējums tekošajai direktorijai, tādēļ, ja jūsu tekošā direktorija būtu bijusi `/bin`, `./su` būtu apzīmējis pilnu ceļu uz `su` komandas failu.

### Aizstājējzīmju salīdzināšana

Gandrīz jebkura čaula dažus simbolus atpazīst kā aizstājējus vai saīsinājumus, kas nozīmē jebko, kas tiek ievietots to vietā. Šādus simbolus sauc par *aizstājējzīmēm* (*wildcards*). Visbiežāk sastopamās aizstājējzīmes ir `*` un `?`. Parasti `?` nozīmē jebkuru vienu simbolu. Pieņemsim, ka jūs atrodaties direktorijā, kurā ir trīs faili: `ex1.txt`, `ex2.txt` un `ex3.txt`. Jūs vēlaties nokopēt visus šos failus (`cp` komandas lietojums aplūkots sadaļā 10.5) uz citu direktoriju, piemēram, `/tmp`. Ziniet, komandas `cp ex1.txt ex2.txt ex3.txt /tmp` ievade ir pārāk gara. Ir daudz vienkāršāk ievadīt `cp ex?.txt /tmp`. `?` aizstās katru no simboliem "1", "2" un "3".

Ko jūs teicāt? Tas joprojām ir pārāk gari? Jums taisnība. Galu galā, mums ir pat darba likumi, kas aizsargā mūs no rutīnas. Par laimi, mums ir arī aizstājējzīme `*`. Tā aizstāj jebkuru skaitu jebkuru simbolu - arī nevienu. Tātad, ja šie trīs faili ir vienīgie mūsu direktorijā, mēs varam vienkārši ievadīt `cp * /tmp` un miers. Pieņemsim, ka direktorijā ir arī fails `ex.txt` un `hejaz.txt`. Mēs vēlamies kopēt `ex.txt`, bet nevēlamies kopēt `hejaz.txt`. To veiks komanda `cp ex* /tmp`.

Komanda `cp ex?.txt /tmp` nokopēs tikai mūsu sākotnējos trīs failus - faila `ex.txt` nosaukumā nav neviena simbola, kuru aizstāj `?`, tādēļ tas tiks izlaists.

Vēl kāda pieņemta aizstājējzīme ir iekavu pāris `[ ]`. Jebkuri simboli iekavās tiks ievietoti `[ ]` vietā, lai meklētu rezultātus. Dīvaini? Nav nemaz tik slikti. Piemēram, mums ir direktorija ar 8 failiem: `a1`, `a2`, `a3`, `a4`, `aA`, `aB`, `aC` un `aD`. Mēs vēlamies atrast tikai tos failus, kuru nosaukumi beidzas ar cipariem - šeit mums palīdzēs `[ ]`.

```
% ls a[1-4]
a1 a2 a3 a4
```

Ko darīt, ja mēs vēlamies atrast tikai **a1**, **a2** un **a4**? Iepriekšējā piemērā mēs izmantojām `-`, lai apzīmētu visas vērtības no 1 līdz 4. Atsevišķas vērtības mēs varam atdalīt ar komatu.

```
% ls a[1,2,4]
a1 a2 a4
```

Zinu, ko jūs šobrīd domājat. "Kā ir ar burtiem?" *Linux* vidē ir atšķirība starp lielajiem un mazajiem burtiem, kas nozīmē, ka **a** un **A** ir dažādi simboli, kuriem kāds kopsakars ir tikai jūsu prātā. Lielie burti vienmēr ir pirms mazajiem, tādēļ **A** un **B** ir pirms **a** un **b**. Turpinot iepriekšējo piemēru, pieņemsim, ka vēlamies atrast failus **a1** un **A1**. Arī to varam ātri veikt ar `[ ]`.

```
% ls [A,a]1
A1 a1
```

Ievērojiet, ka, ja mēs būtu izmantojuši `-` komata vietā, mēs būtu ieguvuši nepareizus rezultātus.

```
% ls [A-a]1
A1 B1 C1 D1 a1
```

Varam arī kombinēt `-` un `,`.

```
% ls [A,a-d]
A1 a1 b1 c1 d1
```

Ievades/izvades pāradresācija un kanāli

(Šeit ir kas jauks)

```
% ps > blargh
```

Zināt, ko tas nozīmē? Tas esmu es, kas izpilda komandu `ps`, lai aplūkotu startētos procesus. `ps` ir aplūkots sadaļā 11.3. Tā nav jaukāka daļa. Jaukāka daļa ir `> blargh`, kas, vienkārši sakot, nozīmē, ka `ps` izvadītie dati tiek "sagrābti" un ierakstīti failā `blargh`. Bet pagaidiet, tūlīt būs vēl jaukāk.

```
% ps | less
```

Šī komanda "sagrābj" `ps` izvadītos datus un novirza tos `less` kanālā (*pipe*), lai es varētu tos ērti aplūkot.

```
% ps >> blargh
```

Šis ir trešais visbiežāk lietotais pāradresācijas simbols - tas dara ko līdzīgu, kā `>`, tikai `>>` `ps` izvadītos datus pievienos faila `blargh` beigās, ja šāds fails jau eksistē. Ja nē, tad, līdzīgi kā `>`, tas radīs jaunu failu. `>` jebkurā gadījumā dzēsīs iepriekšējo faila saturu.

Eksistē arī operators `<`, kura nozīme ir programmas ievaddatus nolasīt no faila, tomēr tas netiek lietots tik bieži.

```
% fromdos < dosfile.txt > unixfile.txt
```

Pāradresācija kļūst patiešām interesanta, kad to sāk pielietot virknē:

```
% ps | tac >> blargh
```

Šī komanda izpildīs `ps`, sakārtos rindiņas apgrieztā secībā un pievienos tās faila `blargh` beigās. Šādi var savienot virknē tik daudz komandas, cik vēlaties - vienīgi jāatceras, ka tās tiek izpildītas no kreisās puses uz labo.

Aplūkojiet `bash(1)` `man` dokumentu, lai atrastu vairāk informācijas par pāradresāciju.

## 8.3. Bourne Again Shell (bash)

### Vides mainīgie

*Linux* sistēma ir sarežģīts zvērs, un ir daudz lietas, kam jāseko līdz, daudz sīku detaļu, kas parādās, sadarbojoties ar dažādām programmām (dažas no šīm detaļām jums pat nav nepieciešamas).

Neviens nevēlas nodot katrai programmai kaudzi dažādu parametru, norādot, kādu terminālu lietot, datora nosaukumu, kā jāizskatās komandrindai...

Piemērs. Vides mainīgo izvade ar `set`

```
% set
PATH=/usr/local/lib/qt/bin:/usr/local/bin:/usr/bin:/bin:/usr/X11R6/bin:/usr
/openwin/bin:/usr/games:./usr/local/ssh2/bin:/usr/local/ssh1/bin:/usr/shar
e/texmf/bin:/usr/local/sbin:/usr/sbin:/home/logan/bin
PIPESTATUS=( [0]="0" )
PPID=4979
PS1='\h:\w\$ '
PS2='> '
PS4='+ '
PWD=/home/logan
QTDIR=/usr/local/lib/qt
REMOTEHOST=ninja.tdn
SHELL=/bin/bash
```

Visu šo sauc par *lietotāja vidi*. Vide nosaka, kādos apstākļos darboties programmām, pie tam daži no šiem apstākļiem ir maināmi; lietotājs vai mainīt un spēlēt ar šiem apstākļiem, jo ir vienīgais pareizais savā *Linux* sistēmā. Gandrīz jebkurai čaulai ir vides mainīgie (ja to nav, tā acīmredzot nav pārāk lietojama čaula). Aplūkosim `bash` čaulas piedāvātās vides mainīgo manipulācijas komandas.

`set` pats par sevi parādīs jums visus vides mainīgos, kas ir uzstādīti, kā arī to vērtības. Kā lielākā daļa `bash` iebūvēto komandu, `set`, izmantojot parametrus, var veikt arī citas darbības - tās atstāsim `bash(1)` man dokumentam. Piemērā jūs redzējat fragmentu no `set` komandas, kas izpildīta vienā no autora datoriem. Ievērojiet `PATH` mainīgo, par ko runājām iepriekš. Programmas, kuras atrodas norādītajās direktoriņās, varam izpildīt, vienkārši ievadot to faila nosaukumus.

```
% unset VARIABLE
```

`unset` nodzēsīs jebkurus mainīgos, ko jūs tai norādīsiet, izdzēšot no sistēmas gan mainīgo, gan tā vērtību - `bash` "aizmirsīs", ka šāds mainīgais jebkad būtu eksistējis. (Neuztraucieties. Kamēr vien tas nav kaut kas, ko jūs norādījāt tekošajā sesijā, tas, ticamākais, atkal tiks noteikts jebkurā citā sesijā.)

```
% export VARIABLE=some_value
```

`export` ir ļoti noderīga komanda. Izmantojot to, jūs piešķirat vides mainīgajam `VARIABLE` vērtību "some\_value". Ja `VARIABLE` iepriekš neeksistēja, tad tagad tas eksistē. Ja `VARIABLE` jau bija kāda vērtība, tad... ziniet, tagad tās vairs nav. Tas nav labi, ja jūs gribējāt tikai pievienot direktoriju mainīgajam `PATH`. Tādā gadījumā jūs, ticamākais, gribējāt izpildīt komandu

```
% export PATH=$PATH:/some/new/directory
```

Ievērojiet šeit izmantoto `$PATH` lietojumu - kad jūs vēlaties, lai `bash` interpretē mainīgo (aizstāj to ar tā vērtību), mainīgā nosaukuma priekšā pievienojiet `$`. Piemēram, `echo $PATH` izvadīs `PATH` vērtību, manā gadījumā:

```
% echo $PATH
/usr/local/lib/qt/bin:/usr/local/bin:/usr/bin:/bin:/usr/X11R6/bin:/usr/open
win/bin:/usr/games:./usr/local/ssh2/bin:/usr/local/ssh1/bin:/usr/share/tex
mf/bin:/usr/local/sbin:/usr/sbin:/home/logan/bin
```

## Rindiņas aizpilde ar Tab

(Šeit atkal kas jauks)

1. Komandrindas saskarne nozīmē daudz rakstīšanas.
2. Rakstīšana nozīmē darbu.

3. Nevienam nepatīk strādāt.

No (3) un (2) varam noteikt, ka (4) būs "nevienam nepatīk rakstīt". Par laimi, `bash` paglābj mūs no (5) - "nevienam nepatīk komandrindas saskarne".

Kā `bash` to panāk? Papildus aizstājējzīmju izmantošanai `bash` piedāvā arī rindiņu aizpildi ar `Tab` taustiņu.

Rindiņu aizpilde ar `Tab` darbojas apmēram šādi - jūs rakstāt faila nosaukumu. Varbūt tas ir jūsu `PATH`, bet varbūt jūs rakstāt pilnu ceļu. Viss, ko jums jāizdara, ir jāuzraksta pietiekoši informācijas, lai varētu nekļūdīgi identificēt failu. Tad nospiediet taustiņu `Tab`. `bash` noteiks, ko jūs vēlaties, un pabeigs rakstīšanu jūsu vietā!

Laiks piemēram. `/usr/src` ir divas apakšdirektorijas: `/usr/src/linux` un `/usr/src/sendmail`. Es gribu aplūkot, kas atrodas direktorijā `/usr/src/linux`. Tādēļ es vienkārši uzrakstu `ls /usr/src/1`, nospiežu taustiņu `Tab` un `bash` komandrindā ieraugu `ls /usr/src/linux`.

Tagad iedomāsimies, ka ir divas direktorijas - `/usr/src/linux` un `/usr/src/linux-old`. Ja es ierakstīšu `/usr/src/1` un nospiedīšu taustiņu `Tab`, `bash` aizpildīs rindiņu tik tālu, cik spēs, un es saņemšu `/usr/src/linux`. Šeit es varu apstāties, bet varu arī nospiegt `Tab` vēlreiz, un `bash` parādīs direktoriju sarakstu, kas atbilst tam, ko es esmu ievadījis līdz šim.

Tas viss nozīmē mazāk rakstīšanas (un tādēļ cilvēkiem patīk komandrindas saskarne). Es jau teicu, ka tas ir jauki.

## 8.4. Virtuālie termināļi

Tātad, jūs veicat kādu darbu un esat nolēmis, ka jums jāizdara vēl ko. Jūs varētu vienkārši pārtraukt veikto darbu un darīt ko citu, tomēr šī taču ir daudzlietotāju sistēma, vai ne? Un jūs varat autorizēties sistēmā vienlaikus tik reizes, cik vēlaties, tā? Tad kādēļ vienlaikus darīt tikai vienu darbu?

Protams, nav tā jādara. Tiesa, mums visiem nav daudzas tastatūras, peles un monitori vienam datoram. Skaidrs, ka aparatūra nav īstais risinājums. Paliek programmatūra, un *Linux* to izmanto, piedāvājot *virtuālos termināļus* (VT).

Nospiežot taustiņu `Alt` un kādu funkcionālo taustiņu, jūs varat pārslēgties starp virtuālajiem termināļiem - katrs funkcionālais taustiņš atbilst vienam. Slackware pēc noklusējuma nodrošina autorizēšanās iespējas sešos VT. `Alt+F2` ieslēgs otro, `Alt+F3` - trešo u.t.t.

Pārējie funkcionālie taustiņi ir rezervēti `x` sesijām. Katra `x` sesija izmanto savu VT, sākot ar septīto (`Alt+F7`). Kad atrodaties `x` logu vidē, `Alt+Funkcionālais taustiņš` kombinācija tiek aizstāta ar `Ctrl+Alt+Funkcionālais taustiņš`, tādēļ, ja atrodaties `x` logu vidē un vēlaties, neaizverot `x` sesiju, atgriezties teksta komandrindā, `Ctrl+Alt+F3` ieslēgs trešo VT. (`Alt+F7` jūs nogādās atpakaļ grafiskajā vidē, pieņemot, ka jūs izmantojat pirmo `x` sesiju.)

### screen

Ko darīt situācijās, kad nav pieejami virtuālie termināļi? Ko tad? Par laimi, *Slackware* ir iekļauts lielisks ekrānu pārvaldnieks, saukts par `screen`. `screen` ir termināla emulators ar iespējām, kas atgādina virtuālo termināli. `screen` izpilde parāda īsu ievadu un tūlīt pēc tam - termināli. Atšķirībā no standarta virtuālajiem termināļiem `screen` ir savas komandas. Visas komandas sākas ar `Ctrl+A`. Piemēram, `Ctrl+A+C` uzsāks jaunu termināļa sesiju. `Ctrl+A+N` pārslēgsies un nākamo termināli (*Next*), bet `Ctrl+A+P` - uz iepriekšējo (*Previous*).

`screen` atbalsta arī atvienošanos un pievienošanos `screen` sesijām, kas ir īpaši lietderīgi attālinātām `ssh` un `telnet` sesijām (par tām vēlāk). `Ctrl+Alt+D` atvienosies no tekošās `screen` sesijas. `screen -r` izpilde parādīs visas strādājošās `screen` sesijas, pie kurām jūs varat pievienoties.

```
% screen -r
There are several suitable screens on:
    1212.pts-1.redtail      (Detached)
    1195.pts-1.redtail      (Detached)
    1225.pts-1.redtail      (Detached)
    17146.pts-1.sanctuary   (Dead ???)
Remove dead screens with 'screen -wipe'.
Type "screen [-d] -r [pid.]tty.host" to resume one of them.
```

`screen -r 1212` izpilde ļaus jums pievienoties pie pirmās `screen` sesijas, kas redzama sarakstā. Jau iepriekš minēju, cik ļoti noderīgi tas ir attālinātām sesijām. Ja es esmu pieslēdzies attālinātais *Slackware* sistēmai ar `ssh` un mans pieslēgums tiek pārtraukts elektrības problēmu dēļ, viss, ko es daru šajā sesijā, tūlīt pat pazūd. Tas var būt diezgan briesmīgi. `screen` izmantošana novērš šādu situāciju, pieslēguma pārtraukuma gadījumā vienkārši atvienojot manu sesiju. Kad pieslēgums ir atjaunots, es atkal varu pieslēgties savai `screen` sesijai un turpināt no vietas, kur pārtraucu.



## 9.nodaļa. Failsistēmas struktūra

Mēs jau iepriekš apspriedām *Slackware Linux* direktoriju struktūru. Šobrīd jums būtu jāprot atrast jums nepieciešamos failus un direktorijas, tomēr failsistēma ir kas vairāk, kā direktoriju struktūra.

*Linux* ir daudzlietotāju operētājsistēma. Katra sistēmas komponente ir veidota daudzlietotāju režīmam - pat failsistēma. Sistēma glabā informāciju par lietotāju tiesībām - piemēram, kam pieder fails un kas to drīkst lasīt. Failsistēmām ir arī īpašas komponentes, kā saites un NFS montēšanas punkti. Šī sadaļa izskaidros visas šīs niansas.

### 9.1. Īpašumtiesības

Failsistēma glabā katra faila un direktorijas īpašumtiesību informāciju - arī to, kuram lietotājam un grupai pieder fails. Vienkāršākais veids, kā aplūkot šo informāciju, ir `ls` komandas izmantošana:

```
% ls -l /usr/bin/wc
-rwxr-xr-x  1 root    bin      7368 Jul 30  1999 /usr/bin/wc
```

Mums interesē izvadītās informācijas trešā un ceturtā kolona. Tajās norādītas faila īpašnieka lietotājavārds un grupa. Redzam, ka šis fails pieder lietotājam `root` un grupai `bin`.

Mēs varam viegli mainīt faila īpašniekus, izmantojot komandas `chown(1)` (*change owner* - mainīt lietotāju) un `chgrp(1)` (*change group* - mainīt grupu). Lai nomainītu faila īpašnieku uz `daemon`, mēs izmantosim `chown`:

```
# chown daemon /usr/bin/wc
```

Lai nomainītu grupu uz `root`, izmantosim `chgrp`:

```
# chgrp root /usr/bin/wc
```

Ir iespējams nomainīt gan lietotāju, gan grupu ar vienu komandu:

```
# chown daemon:root /usr/bin/wc
```

Iepriekšējā piemērā kola `:` vietā varēja izmantot arī punktu `.`, tomēr tiek uzskatīts, ka labāk izmantot kolu. Punkta lietojums tiek uzskatīts par novecojušu un, iespējams, nākamajās `chown` versijās nebūs pieejams, lai atļautu lietotājavārdus ar punktu tajos. Šādi lietotājavārdi ir ļoti populāri *Windows Exchange* serveros un tos bieži redzam e-pasta adresēs, piemēram, `mr.jones@example.com`. *Slackware* sistēmā administratoriem tiek ieteikts izvairīties no šādiem lietotājavārdiem, jo daži skripti punktu joprojām izmanto, lai norādītu lietotāju un grupu. Mūsu piemērā `chmod` interpretētu `mr.jones` kā lietotāju `mr` un grupu `jones`.

Failu īpašumtiesības ir ļoti svarīgas *Linux* sistēmas izmantošanā - pat, ja jūs esat vienīgais lietotājs. Reizēm jums var nākties salabot īpašumtiesības failiem un iekārtu mezgliem (*device nodes*).

### 9.2. Tiesības

Tiesības ir otrs svarīgais daudzlietotāju failsistēmas jēdziens. Izmantojot tiesības, jūs varat noteikt, kurš drīkst lasīt, rakstīt un izpildīt failus.

Tiesību informācija tiek glabāta kā četri oktālie cipari, kur katrs norāda citu tiesību komplektu. Tās ir lietotāja tiesības, grupas tiesības un pasaules (jeb visas pārējās) tiesības. Ceturtais cipars nosaka īpašu papildinformāciju, kā lietotāja ID uzstādīšana, grupas ID uzstādīšana un t.s. *sticky bit*. Oktālās vērtības, kas piešķirtas tiesību režīmiem, ir šādas: (patiesībā ir arī burti, kas saistīti ar šīm vērtībām - tos izmanto `ls` tiesību attēlošanai un tos var izmantot ar `chmod`)

| Tiesību tips | Oktālā vērtība | Burta vērtība |
|--------------|----------------|---------------|
| sticky bit   | 1              | t             |

| Tiesību tips             | Oktālā vērtība | Burta vērtība |
|--------------------------|----------------|---------------|
| lietotāja ID uzstādīšana | 4              | s             |
| grupas ID uzstādīšana    | 2              | s             |
| lasīt                    | 4              | r             |
| rakstīt                  | 2              | w             |
| izpildīt                 | 1              | x             |

Katrai tiesību grupai oktālās vērtības vienkārši saskaita. Piemēram, ja jūs vēlaties, lai grupas tiesības būtu "lasīt" un "rakstīt", izmantojiet 6 (=4+2) tiesību informācijas grupas iedaļā.

**bash** noklusētās tiesības ir

```
% ls -l /bin/bash
-rwxr-xr-x 1 root bin 477692 Mar 21 19:57 /bin/bash
```

Pirmais – simbols tiktu aizstāts ar **d**, ja tā būtu direktorija. Kā nākamās tiek parādītas trīs tiesību grupas (īpašnieks, grupa un pasaule). Redzam, ka īpašniekam ir lasīšanas, rakstīšanas un izpildes tiesības. Grupai un pasaulei ir tiesības šo failu tikai lasīt un izpildīt.

Kā var uzstādīt tiesības citam failam, lai tās atgādinātu **bash** tiesības? Vispirms izveidosim piemēra failu:

```
% touch /tmp/example
% ls -l /tmp/example
-rw-rw-r-- 1 david users 0 Apr 19 11:21 /tmp/example
```

Lai uzstādītu šī faila tiesības, izmantosim **chmod(1)** (kas nozīmē "change mode" - "mainīt režīmu"). Saskaitiet vēlamo tiesību oktālos apzīmējumus. Lai lietotājam būtu lasīšanas, rakstīšanas un izpildes tiesības, jāsanāk 7. Lasīšana un izpilde būs 5. Salieciet šos ciparus kopā un nododiet tos **chmod**:

```
% chmod 755 /tmp/example
% ls -l /tmp/example
-rwxr-xr-x 1 david users 0 Apr 19 11:21 /tmp/example
```

Tagad jūs varētu iedomāties: "Bet kādēļ sistēma jau uzreiz neizveidoja failu ar šādām tiesībām?" Atbilde ir vienkārša. **bash** ir iekļauta iebūvētā komanda **umask**. Šāda komanda ir iekļauta lielākajā daļā *Unix* čaulu un tā nosaka, kādas failu tiesības tiks piešķirtas jauniem failiem. Mēs jau nedaudz aplūkojām **bash** iebūvētās komandas sadaļā 8.3.1. Pie **umask** ir nedaudz jāpierod. Tā darbojas līdzīgi, kā **chmod**, tikai apgriezti. Jūs norādāt oktālās vērtības, kuras NEVĒLATIES redzēt jauno failu tiesībās. Noklusētā **umask** vērtība ir 0022.

```
% umask
0022
% umask 0077
% touch tempfile
% ls -l tempfile
-rw----- 1 david users 0 Apr 19 11:21 tempfile
```

Vairāk informācijas meklējiet **man** dokumentā par **bash**.

Lai uzstādītu īpašās tiesības ar **chmod**, saskaitiet atbilstošos skaitļus un novietojiet rezultātu pirmajā kolonā. Piemēram, lai liktu uzstādīt lietotāja ID uzstādīšanu un grupas ID uzstādīšanu, pirmajā kolonā jāatrodas 6:

```
% chmod 6755 /tmp/example
```

```
% ls -l /tmp/example
-rwsr-sr-x 1 david      users    0 Apr 19 11:21 /tmp/example
```

Ja oktālās vērtības jūs mulsina, varat arī izmantot burtus. Tiesību iedaļas tiek attēlotas kā:

```
Īpašnieks      u
Grupa           g
Pasaule         o
Visi            a
```

Lai izpildītu iepriekš veikto darbību, mums jāizmanto vairākas komandas:

```
% chmod a+rx /tmp/example
% chmod u+w /tmp/example
% chmod ug+s /tmp/example
```

Dažiem lietotājiem labāk patīk burti. Jebkurā gadījumā rezultāts būs vienāds.

Oktālais formāts bieži ir ātrāks, un tas ir visbiežāk izmantotais skriptos. Reizēm tomēr burtu izmantošana piedāvā lielākas iespējas. Piemēram, nav viegla veida, kā nomainīt vienu tiesību iedaļu, nemainot citas. Izmantojot burtus, tas ir elementāri:

```
% ls -l /tmp/
-rwxr-xr-x 1 alan      users    0 Apr 19 11:21 /tmp/example0
-rwxr-x--- 1 alan      users    0 Apr 19 11:21 /tmp/example1
----r-xr-x 1 alan      users    0 Apr 19 11:21 /tmp/example2
% chmod g-rwx /tmp/example?
% ls -l /tmp/
-rwx---r-x 1 alan      users    0 Apr 19 11:21 /tmp/example0
-rwx----- 1 alan      users    0 Apr 19 11:21 /tmp/example1
-----r-x 1 alan      users    0 Apr 19 11:21 /tmp/example2
```

Mēs jau pieminējām lietotāja ID uzstādīšanas un grupas ID uzstādīšanas tiesības. Jūs varētu brīnīties - kas tās tādas? Normāli, izpildos programmu, tā darbojas ar jūsu lietotāja tiesībām. Tas nozīmē, ka tai ir visas tās tiesības, kas jums kā sistēmas lietotājam. Tas pats attiecas arī uz grupu. Kas jūs izpildat programmu, tā izpildas jūsu tekošajā grupā. Ar lietotāja ID uzstādīšanas tiesībām, jūs varat likt programmai darboties ar programmas īpašnieka tiesībām (piemēram, `root`). Grupas ID uzstādīšanas tiesības ir tas pats, tikai attiecas uz grupu.

Uzmanieties ar šo - lietotāja ID uzstādīšanas un grupas ID uzstādīšanas tiesības var atvērt būtiskus drošības caurumus jūsu sistēmā. Uzstādot šādas tiesības programmām, kuras pieder `root` lietotājam, jūs ļaujat jebkurai sistēmas lietotājam tās izpildīt ar `root` tiesībām. Tā kā `root` nav nekādu ierobežojumu sistēmas izmantošanā, redzams, ka tas var radīt nopietnas drošības problēmas. Īsāk sakot - nav slikti uzstādīt šīs tiesības, tikai to jādara prātīgi.

### 9.3. Saites

Saites ir norādes starp failiem. Ar saitēm iespējams panākt, ka viens un tas pats fails atrodas dažādās vietās ar dažādiem nosaukumiem. Ir divi saišu tipi - cietās saites un simbolsaites.

Cietās saites ir faila dažādi nosaukumi. Tās var eksistēt tikai vienā failsistēmā un tās tiek dzēstas tikai, kad no sistēmas tiek dzēsts reālais fails. Šīs saites reizēm ir noderīgas, tomēr daudziem lietotājiem noderīgākas šķiet simbolsaites.

Simbolsaite var norādīt uz failu citā failsistēmā. Patiesībā simbolsaite ir neliels fails, kas satur saitei nepieciešamo informāciju. Simbolsaites ir iespējams pievienot un dzēst, neskarot oriģinālo failu. Tā kā simbolsaite patiesībā ir fails, kas satur savu informāciju, tas var norādīt pat uz direktoriju. Piemēram, ir pieņemts, ka `/var/tmp` patiesībā ir simbolsaite uz `/tmp`.

Saitēm nav sava īpašumtiesību un lietotāja tiesību komplekta - tās izmanto tādu tiesību komplektu, kā failam, uz kuru tās norāda. *Slackware* lielākoties izmanto simbolsaites. Tipisks piemērs:

```
% ls -l /bin/sh
lrwxrwxrwx 1 root root 4 Apr 6 12:34 /bin/sh -> bash
```

`sh` čaula *Slackware* sistēmā patiesībā ir `bash`. Saišu dzēšana tiek veikta, izmantojot `rm`. Saišu radīšana tiek veikta, izmantojot komandu `ln`. Šīs komandas plašāk tiks aplūkotas 10.nodaļā.

Ir ļoti svarīgi uzmainīties ar simbolsaišu izmantošanu. Reiz es strādāju ar datoru, kuram pastāvīgi radās problēmas ar rezerves kopiju izveidi lentē. Izrādījās, ka divas direktorijas viena zem otras bija simbolsaites. Rezerves kopēšanas programma turpināja pievienot šīs direktorijas lentes arhīvam, kamēr beidzās vieta. Parasti šādā situācijā attiecīgs pārbaužu komplekss ļauj izvairīties no simbolsaišu veidošanas, bet šis bija īpašs gadījums.

## 9.4. Iekārtu montēšana

Kā aplūkojam sadaļā 4.1, visas iekārtas jūsu datorā veido vienu lielu failsistēmu. Vienā kokā tiek savietotas visas cieto disku sadaļas, CD-ROM un diskešu iekārtas. Lai šīs iekārtas pievienotu failsistēmai tā, ka tām iespējams piekļūt, jāizmanto `mount (1)` un `umount (1)` komandas.

Dažas iekārtas tiek automātiski montētas, startējot datoru. Tās ierakstītas failā `/etc/fstab`. Šajā failā būs ierakstīta informācija par visu, ko vēlaties montēt automātiski. Citām iekārtām jums būs jāievada montēšanas komandu katru reizi, kad vēlēsit izmantot šo iekārtu.

### fstab

Aplūkosim `/etc/fstab` faila piemēru:

```
% cat /etc/fstab
/dev/sda1      /          ext2          defaults      1 1
/dev/sda2      /usr/local ext2          defaults      1 1
/dev/sda4      /home     ext2          defaults      1 1
/dev/sdb1      swap      swap         defaults      0 0
/dev/sdb3      /export   ext2          defaults      1 1
none          /dev/pts  devpts       gid=5,mode=620 0 0
none          /proc     proc         defaults      0 0
/dev/fd0       /mnt/floppy msdos        defaults      0 0
/dev/cdrom     /mnt/cdrom iso9660       ro            0 0
```

Pirmajā kolonā norādīts iekārtas mezgla nosaukums. Šajā gadījumā iekārtas ir piecas disku sadaļas, kas sadalītas pa diviem SCSI cietajiem diskiem, divas speciālās failsistēmas, kam nav nepieciešama iekārta, diskešu iekārta un CD-ROM iekārta. Otrajā kolonā tiek norādīts, kur iekārta tiks montēta. Tam jābūt direktorijas nosaukumam, izņemot maiņatmiņas (*swap*) sadaļu. Trešajā kolonā norādīts failsistēmas tips. Parasti *Linux* failsistēmām tas būs `ext2` (otrā paplašinātā failsistēma) (mūsdienās gan prātīgāk būtu izmantot `ext3` - tulk.piez.). CD-ROM iekārtām šī failsistēma ir `iso9660`, bet *Windows* tipa iekārtām (piemēram, disketēm) tā ir `msdos` vai `vfat`.

Ceturta kolona ir dažādu iestatījumu saraksts, kas tiek pielietots piemontētajai failsistēmai. Parasti pietiek ar noklusētajiem iestatījumiem - `defaults` - tomēr tikai lasāmajām iekārtām jānorāda iestatījumu `ro`. Šo iestatījumu ir daudz - aplūkojiet `fstab (5)` `man` dokumentu. Pēdējās divas kolonas izmanto `fsck` un citas komandas, kam jāveic dažādas darbības ar iekārtām. `man` dokumentā arī par to atrodama papildinformācija.

Kad jūs instalējat *Slackware Linux*, instalēšanas programma jau izveidoja lielāko daļu `fstab` faila.

## mount un umount

Iekārtas pieslēgšana failsistēmai ir vienkārša. Viss, ko jums jāveic - jāstartē `mount` komandu, tai norādot dažus parametrus. `mount` lietošana var tikt vienkāršota, ja iekārta norādīta failā `/etc/fstab`. Piemēram, pieņemsim, ka es vēlos montēt savu CD-ROM iekārtu un ka mans `fstab` fails izskatās līdzīgi, kā iepriekšējā piemērā. Es varu izsaukt `mount` šādi:

```
% mount /dev/cdrom
```

Tā kā `fstab` ir atbilstošs ieraksts, `mount` zinās, kādus parametrus izmantot. Ja šāda ieraksta nebūtu, man būtu tos jānorāda pašam:

```
% mount -t iso9660 -o ro /dev/cdrom /mnt/cdrom
```

Šī komandrinda iekļauj to pašu informāciju, ko `fstab`, tomēr aplūkosim to tuvāk. `-t iso9660` norāda iekārtas failsistēmas tipu. Šajā gadījumā tā ir *ISO9660* failsistēma, ko parasti izmanto CD-ROM iekārtas. `-o ro` norāda `mount` programmai, ka iekārtu jāmontē tikai lasāmā režīmā. `/dev/cdrom` ir montējamās iekārtas mezgla nosaukums, un `/mnt/cdrom` ir montējamā vieta failsistēmā.

Pirms jūs varat izņemt disketi no diskešu iekārtas, CD-ROM disku no CD-ROM iekārtas, vai jebkuru noņemamo datu nesēju, kas ir piemontēts, to vispirms jāatmontē. To veic ar komandu `umount`. Nejautājiet, kur šīs komandas nosaukumā pazudis burts `n` (*atmontēt* - *umount* - tulk. piez.) - arī mēs to nezinām. Kā `umount` argumentu varat izmantot vai nu piemontētās iekārtas mezgla nosaukumu, vai piemontētās direktorijas nosaukumu. Piemēram, ja jūs vēlaties atmontēt CD-ROM iekārtu no iepriekšējā piemēra, darbosies jebkura no šīm komandām:

```
# umount /dev/cdrom
```

```
# umount /mnt/cdrom
```

## 9.5. NFS montēšana

NFS nozīmē "Tīkla failsistēma" ("Network Filesystem"). Patiesībā tā nav daļa no reālas failsistēmas, tomēr to var izmantot piemontētas failsistēmas papildināšanai.

Lielās *Unix* sistēmās bieži vien ir kopīgas programmas, mājas direktorijas un e-pasta rindas. NFS atrisina problēmas, kas saistītas ar šo direktoriju un failu vienādu kopiju nodrošināšanu visās sistēmās. Izmantojot NFS, iespējams izdalīt vienu mājas direktoriju komplektu visām tīkla darbstacijām. Darbstacijas pēc tam montē šo NFS resursu tā, it kā tas atrastos pašās darbstacijās.

Vairāk informācijas iespējams atrast sadaļā 5.6 un `man` dokumentos `exports(5)`, `nfsd(8)` un `mountd(8)`.

## 10.nodaļa. Darbs ar failiem un direktoriņām

*Linux* tiek veidots tik līdzīgs *Unix* operētājsistēmām, cik vien iespējams. Tradicionāli *Unix* operētājsistēmās darbs ir noticis komandrindā. Mēs *Slackware* piedāvājam grafisko lietotāja saskarni, tomēr komandrinda joprojām ir galvenais līdzeklis darbam ar sistēmu. Tādēļ ir svarīgi izprast dažas galvenās failu vadības komandas.

Turpmākajās sadaļās izskaidrotas vispārējās failu vadības komandas un sniegti to pielietojuma piemēri. Ir arī daudzas citas komandas, tomēr sākumam pietiks ar šīm. Bez tam, komandas šeit aplūkotas tikai virspusēji. Vairāk informācijas par katru komandu iespējams atrast tās `man` dokumentā.

### 10.1. Navigācija: *ls*, *cd* un *pwd*

#### *ls*

Šī komanda izvada sarakstu ar direktoriņā izvietotajiem failiem. *Windows* un *DOS* lietotāji pamanīs tās līdzību ar komandu `dir`. Bez parametriem `ls (1)` izvadīs sarakstu ar failiem, kas izvietoti tekošajā direktoriņā. Lai aplūkotu, kas atrodas saknes direktoriņā, iespējams ievadīt šādas komandas:

```
% cd /
% ls
bin   cdr   dev   home  lost+found  proc  sbin   tmp   var
boot  cdrom etc   lib   mnt         root  suncd  usr   vmlinuz
```

Daudziem lietotājiem šāds izvades veids sagādās problēmas noteikt, kas no izvadītā ir direktoriņa un kas ir fails, tādēļ tie izvēlas norādīt `ls`, lai katram saraksta elementam tiktu pievienots tipa identifikators:

```
% ls -FC
bin/   cdr/   dev/   home/  lost+found/  proc/  sbin/  tmp/  var/
boot/  cdrom/ etc/   lib/   mnt/         root/  suncd/ usr/  vmlinuz
```

Direktoriņu nosaukumiem galā tiek pievienota slīpsvītra, izpildāmajiem failiem - zvaigznīte, u.c.

`ls` var izmantot arī, lai noskaidrotu failu statistiku. Piemēram, lai aplūkotu failu radīšanas datumus, īpašniekus un tiesības, jums jāaplūko garo (*long*) izvades formu:

```
% ls -l
drwxr-xr-x  2 root   bin      4096 May  7 09:11 bin/
drwxr-xr-x  2 root   root     4096 Feb 24 03:55 boot/
drwxr-xr-x  2 root   bin      4096 Feb 18 01:10 cdr/
drwxr-xr-x 14 root   bin      6144 Oct 23 19:37 cdrom/
drwxr-xr-x  4 root   bin     28672 Mar  5 19:01 dev/
drwxr-xr-x 10 root   bin      4096 Mar  8 03:32 etc/
drwxr-xr-x  8 root   bin      4096 Mar  8 03:31 home/
drwxr-xr-x  3 root   bin      4096 Jan 23 21:29 lib/
drwxr-xr-x  2 root   bin     16384 Nov  1 09:53 lost+found/
drwxr-xr-x  2 root   bin      4096 Oct  6 12:47 mnt/
dr-xr-xr-x 62 root   bin         0 Mar  4 15:32 proc/
drwxr-xr-x 12 root   bin      4096 Feb 26 02:06 root/
drwxr-xr-x  2 root   bin      4096 Feb 17 02:02 sbin/
drwxr-xr-x  5 root   bin      2048 Oct 25 10:51 suncd/
drwxrwxrwt  4 root   bin    487424 Mar  7 20:42 tmp/
drwxr-xr-x 21 root   bin      4096 Aug 24 03:04 usr/
```

```
drwxr-xr-x 18 root bin 4096 Mar 8 03:32 var/
```

Iedomāsimies, ka vēlamies aplūkot tekošās direktorijas slēptos failus. Nākamā komanda veiks tieši to.

```
% ls -a
```

```
. bin cdrom home mnt sbin usr
.. boot dev lib proc suncd var
.pwrchute_tmp cdr etc lost+found root tmp vmlinuz
```

Faili, kuru nosaukumi sākas ar punktu (.), ir slēpti jeb neredzami, izpildot `ls`. Jūs tos ieraudzīsiet tikai, izpildot `ls` ar parametru `-a`.

`ls` komandai ir arī vēl daudzi citi parametri. Tos iespējams aplūkot `man` dokumentā. Neaizmirstiet, ka `ls` parametrus iespējams arī kombinēt.

## cd

`cd` komandu izmanto, lai nomainītu darba direktoriju. To lieto, vienkārši ievadot `cd` ar parametru, kas norāda ceļu, uz kuru darba direktoriju jānomaina. Piemēri:

```
darkstar:~$ cd /bin
darkstar:/bin$ cd usr
bash: cd: usr: No such file or directory
darkstar:/bin$ cd /usr
darkstar:/usr$ ls
bin
darkstar:/usr$ cd bin
darkstar:/usr/bin$
```

Ievērojiet, ka, ceļa sākumā nenorādot slīpsvītras simbolu, `cd` mēģinās nomainīt direktoriju uz atbilstošo tekošās direktorijas apakšdirektoriju. `cd` izpilde bez parametriem nomainīs darba direktoriju uz jūsu mājas direktoriju.

`cd` komanda nav, kā visas citas komandas. Tā ir čaulā iebūvēta komanda. Čaulā iebūvētās komandas aplūkojām sadaļā 8.3. Patiesībā tas šobrīd varētu jums neko nenožīmēt, tomēr būtībā tas nozīmē, ka šai komandai nav `man` dokumenta. Tā vietā jums jāizmanto čaulas iebūvētā palīdzība:

```
% help cd
```

Šī komanda izvadīs `cd` pieejamos parametrus un informāciju par to pielietojumu.

## pwd

`pwd` komandu izmanto, lai uzzinātu tekošo direktoriju. Lai izmantotu `pwd` komandu, vienkārši ievadiet `pwd`. Piemēram:

```
% cd /bin
% pwd
/bin
% cd /usr
% cd bin
% pwd
/usr/bin
```

## 10.2. Peidžeri: *more*, *less* un *most*

### **more**

**more** (1) sauc arī par peidžera (lappušu pārlūka) rīku. Bieži vien kādas noteiktas komandas izvadītā informācija ir pārāk apjomīga, lai ietilptu vienā ekrānā. Komandas nezina, kā izvadīt savu informāciju vairākos atsevišķos ekrānos, atstājot šo uzdevumu peidžera rīkam.

**more** komanda sadala izvadāmo informāciju atsevišķos ekrānos un, pirms turpināt nākamā ekrāna izvadi, gaida, līdz jūs nospiedīsiet atstarpes taustiņu. **Enter** taustiņa nospiešana turpinās izvadi līdz nākamajai rindiņai. Labs piemērs:

```
% cd /usr/bin
% ls -l
```

Šī komanda ritinās ekrānu labu brīdi. Lai sadalītu izvadīto informāciju ekrānos, novirziet to **more** kanālā:

```
% ls -l | more
```

Kā jau aplūkojām sadaļā 8.2, kanālu novirzīšana nozīmē ņemt visu **ls** izvadīto informāciju un nodot to **more** kā ievaddatus. Šādi **more** kanālā iespējams ievadīt jebko, ne tikai **ls** izvadīto informāciju.

### **less**

**more** komanda ir visnotaļ noderīga, tomēr bieži vien jūs konstatēsiet, ka esat "pārskrējis" vēlamajam ekrānam, tādēļ noderētu iespēja pārvietoties arī par ekrānu atpakaļ. Šo funkcionalitāti nodrošina komanda **less** (1). To lieto tieši tāpat, kā **more** komandu, tādēļ visi iepriekšējie piemēri attiecas arī uz to. Tātad, ņemot vērā, ka **more** angļu valodā nozīmē vairāk un **less** - mazāk, **less** ir vairāk, kā **more** (*less is more than more*). Džūsts Kremers, izmantojot angļu valodas vārdu spēli, to aprakstījis šādi (netiek tulkots - tulk. piez.):

**less is more, but more than more is, so more is less less, so use more less if you want less more.**

### **most**

**most** (1) noder tur, kur neder ne **more**, ne **less**. Ja **less** ir kas vairāk par **more**, **most** ir vairāk par **less**. Ja citi peidžeri spēj attēlot vienu failu, **most** spēj attēlot jebkuru skaitu failu vienlaikus, ja vien katra faila izvades logs ir vismaz 2 rindiņas plats. **most** ir pieejami daudzi parametri - aplūkojiet **man** dokumentu pilnai informācijai.

## 10.3. Vienkārša izvade: *cat* un *echo*

### **cat**

**cat** (1) ir saīsinājums no *concatenate* (savienot). Sākotnēji izveidota, lai savienotu vairākus teksta failus vienā, tomēr to iespējams izmantot daudziem citiem mērķiem.

Lai savienotu divus vai vairāk teksta failus vienā, vienkārši norādiet failu nosaukumus kā **cat** parametrus, un izvadītos datus pāradresējiet uz failu. **cat** darbojas ar standarta ievadi un standarta izvadi, tādēļ jāizmanto čaulas pāradresācijas simbolus. Piemēram:

```
% cat file1 file2 file3 > bigfile
```

Šī komanda nolasa **file1**, **file2** un **file3** saturu un apvieno tos vienā. Rezultāts tiek nosūtīts uz standarta izvadi.



`cat` var izmantot arī failu aplūkošanai. Daudzi izmanto `cat` un tā izvadīto informāciju novirza uz `more` vai `less` komandu kanālu:

```
% cat file1 | more
```

Šī komanda attēlos failu `file1` un nosūtīs to uz `more` kanālu, lai jūs varētu to lasīt pa ekrānam.

Vēl kāds biežs `cat` pielietojums ir failu kopēšana. Jebkuru failu iespējams kopēt šādi:

```
% cat /bin/bash > ~/mybash
```

`/bin/bash` programma tiks nokopēta uz jūsu mājas direktoriju un nosaukta `mybash`.

`cat` ir daudzi pielietojumi un šeit aplūkoti ir tikai daži no tiem. Tā kā `cat` intensīvi izmanto standarta ievadi un standarta izvadi, tā ir ideāla izmantošanai čaulas skriptos vai kā citu, sarežģītāku, komandu daļu.

## echo

`echo` (1) komanda izvada uz ekrāna norādīto tekstu. Izvadāmo tekstu norāda kā teksta rindu pēc `echo` komandas. Pēc noklusējuma `echo` izvadīs šo tekstu un jaunas rindas simbolu pēc tā. Norādot `-n` parametru, iespējams, novērst jaunas rindas simbola izvadi. Parametrs `-e` liks `echo` meklēt teksta rindā vadības simbolus un tos izpildīt.

## 10.4. Radīšana: touch un mkdir

### touch

`touch` (1) tiek izmantota, lai mainītu faila laiku. Iespējams mainīt gan piekļuves, gan izmaiņu laiku. Ja norādītais fails neeksistē, `touch` izveidos failu ar nulles garumu un norādīto nosaukumu. Lai atzīmētu failu ar tekošo sistēmas laiku, ievadiet komandu:

```
% ls -al file1
-rw-r--r--  1 root    root      9779 Feb  7 21:41 file1
% touch file1
% ls -al file1
-rw-r--r--  1 root    root      9779 Feb  8 09:17 file1
```

`touch` komandai ir vairāki izvēles parametri - iespēja norādīt, kuru no laikiem (piekļuves vai modifikācijas) mainīt, laiku, uz kuru to mainīt u.c. Vairāk informācijas par šiem parametriem atradīsiet `touch man` dokumentā.

### mkdir

`mkdir` (1) izveidos jaunu direktoriju. Vienkārši norādiet veidojamo direktoriju kā `mkdir` parametru. Piemēram, izveidosim direktoriju `hejaz` tekošajā direktorijā:

```
% mkdir hejaz
```

Var arī norādīt pilnu ceļu:

```
% mkdir /usr/local/hejaz
```

`mkdir` var norādīt parametru `-p`, kas liks tam izveidot arī visas veidojamās direktorijas saknes direktorijas. Piemēram, iepriekšējais piemērs nedarbosies, ja neeksistēs direktorija `/usr/local`. Parametrs `-p` izveidos gan `/usr/local` (ja tās neeksistē), gan `/usr/local/hejaz`:

```
% mkdir -p /usr/local/hejaz
```

## 10.5. Kopēšana un pārvietošana

### cp

**cp**(1) kopē failus. DOS lietotāji pamanīs šīs komandas līdzību ar komandu **copy**. **cp** atbalsta daudzus komandrindas parametrus, tādēļ pirms **cp** izmantošanas ieteicams aplūkot **man** dokumentu.

Tipisks pielietojums ir **cp** izmantošana, lai kopētu failu no vienas direktorijas uz citu. Piemēram:

```
% cp hejaz /tmp
```

Šī komanda kopēs failu **hejaz** no tekošās direktorijas uz direktoriju **/tmp**.

Daudzi lietotāji vēlas, lai tiktu saglabāti failu modifikācijas un piekļuves laiki:

```
% cp -a hejaz /tmp
```

Parametrs **-a** nodrošina, ka laiki netiek modificēti.

Lai rekursīvi nokopētu veselas direktorijas saturu uz citu direktoriju, izmantojiet komandu:

```
% cp -R mydir /tmp
```

Šī komanda nokopēs direktoriju **mydir** uz direktoriju **/tmp**.

Ja jūs vēlaties nokopēt direktoriju vai failu un saglabāt faila tiesības un laikus, izmantojiet **cp -p**.

```
% ls -l file
```

```
-rw-r--r--  1 root    vlad      4 Jan 1 15:27 file
```

```
% cp -p file /tmp
```

```
% ls -l /tmp/file
```

```
-rw-r--r--  1 root    vlad      4 Jan 1 15:27 file
```

**cp** ir daudz citu parametru, kas sīkāk aplūkoti **cp man** dokumentā.

### mv

**mv**(1) pārvieto failus no vienas direktorijas uz citu. Izklusās vienkārši, vai ne?

```
% mv oldfile /tmp/newfile
```

**mv** atbalsta dažus noderīgus komandrindas parametrus, kas aplūkoti **man** dokumentā. Praksē **mv** gandrīz vienmēr izmanto bez papildparametriem.

## 10.6. Dzēšana: rm un rmdir

### rm

**rm**(1) dzēš failus un direktoriju kokus. DOS lietotāji pamanīs šīs komandas līdzību ar **del** un **deltree**. Neuzmanīgi pielietojot, **rm** var būt ļoti bīstama komanda. Lai arī reizēm ir iespējams atgūt nejauši dzēstu failu, tas var būt sarežģīti (un arī dārgi) un šāda darbība šajā grāmatā netiek aplūkota.

Lai dzēstu vienu failu, norādiet tā nosaukumu:

```
% rm file1
```

Ja failam ir noņemtas rakstīšanas tiesības, jūs varat saņemt paziņojumu par piekļuves aizliegumu. Lai piespiestu sistēmu dzēst failu neatkarīgi ne no kā, izmantojiet parametru **-f** (*force* - piespiest):

```
% rm -f file1
```

Lai dzēstu veselu direktoriju, vienlaikus jāizmanto parametri **-f** un **-r**. Šis ir labs piemērs, kā nejauši nodzēst visu cietā diska saturu. **Jūs patiešām nevēlaties to veikt**. Jebkurā gadījumā, šī nejaušā komanda izskatās šādi:

```
# rm -rf /
```

Esiet ļoti uzmanīgi ar `rm`, jūs varat viegli *savainoties*. `rm` ir arī daži komandrindas parametri, kas sīkāk aplūkoti `man` dokumentā.

## rmdir

`rmdir(1)` dzēš direktorijas no failsistēmas. Direktorijai pirms dzēšanas jābūt tukšai. Komandas sintakse ir vienkārša:

```
% rmdir <direktorija>
```

Piemēram, šī komanda dzēsīs tekošās direktorijas apakšdirektoriju `hejaz`:

```
% rmdir hejaz
```

Ja šāda direktorija neeksistē, `rmdir` to paziņos. Jūs arī varat norādīt pilnu ceļu uz dzēšamo direktoriju, piemēram:

```
% rmdir /tmp/hejaz
```

Šāda komanda mēģinās dzēst direktoriju `hejaz`, kas atrodas direktorijā `/tmp`.

Jūs arī varat dzēst direktoriju un visas tās saknes direktorijas, norādot parametru `-p`.

```
% rmdir -p /tmp/hejaz
```

Vispirms šī komanda mēģinās dzēst direktoriju `hejaz`. Ja tas izdosies, tā mēģinās dzēst `/tmp`. Tā tas turpināsies, līdz radīsies kļūda vai arī viss norādītais direktoriju koks būs dzēsts.

## 10.7. Failu aizstātvārdu izveide ar ln

`ln(1)` tiek izmantots, lai starp failiem veidotu saites. Šīs saites var būt gan cietās saites, gan simbolsaites. Atšķirības starp šiem saišu veidiem tika aplūkotas sadaļā 9.3. Ja jūs vēlētos izveidot simbolsaiti direktorijai `/var/media/mp3` un izvietot to jūsu mājas direktorijā, jūs izmantotu komandu

```
% ln -s /var/media/mp3 ~/mp3
```

Parametrs `-s` norāda `ln`, ka jāveido simbolsaiti. Nākamais parametrs ir saites mērķis, un pēdējais - informācija, kā nosaukt saiti. Šajā gadījumā jūsu mājas direktorijā tiks izveidots fails ar nosaukumu `mp3`, kurš norādīs uz `/var/media/mp3`. Saiti jūs varat nosaukt pēc saviem ieskatiem, vienkārši mainot pēdējo parametru.

Cietās saites izveide ir tikpat vienkārša. Viss, ko jums jāizdara - jāizlaiž parametru `-s`. Cietās saites parasti nevar atsaukties uz direktorijām vai norādīt uz failiem citās failsistēmās. Lai izveidotu cieto saiti `/usr/bin/email`, kas norādītu uz `/usr/bin/mutt`, ievadiet komandu

```
# ln /usr/bin/mutt /usr/bin/email
```

## 11.nodaļa. Procesu vadība

Jebkuru strādājošu programmu sauc par *procesu*. Sistēmā darbojas dažādi procesi - sākot ar X logu sistēmu un beidzot ar sistēmas programmām (*dēmoniem*), kuri tiek startēti sistēmas ielādes laikā. Jebkurš process darbojas ar kāda lietotāja tiesībām. Procesi, kuri tiek startēti sistēmas ielādes laikā, parasti darbojas ar `root` vai `nobody` tiesībām. Procesi, kurus startējat jūs, darbosies ar jūsu lietotāja tiesībām. Procesi, kurus startē citi lietotāji, darbosies ar attiecīgo lietotāju tiesībām.

Jūs varat pārvaldīt visus paša startētos procesus. Papildus, `root` lietotājs var pārvaldīt visus sistēmas procesus - arī tos, kurus startējuši citi lietotāji. Procesus var kontrolēt un uzraudzīt, izmantojot vairākas programmas, kā arī dažas čaulas komandas.

### 11.1. Procesu darbināšana fonā

Programmas, kuras tiek startētas komandrindā, darbojas kā priekšplāna programmas. Tas ļauj jums redzēt visu programmas izvadīto informāciju, kā arī sazināties ar to. Tomēr ir gadījumi, kad jūs varētu vēlēties, lai programma darbojas, neaizņemot jūsu termināli. To sauc par programmas darbināšanu fonā, un ir vairāki veidi, kā to veikt.

Pirmais veids ir procesu startēt fona režīmā, pievienojot komandrindai simbolu `&`. Piemēram, pieņemsim, ka mēs vēlamies izmantot komandrindas MP3 atskaņotāju `amp`, lai atskaņotu visus direktoriņā izvietotos MP3 failus, tomēr vienlaikus vēlamies savā terminālī veikt ko citu. `amp` fona startēšanai izmantosim komandu:

```
% amp *.mp3 &
```

Programma tiek startēta, un jūs varat turpināt strādāt ar sistēmu.

Otrs veids, kā procesu darbināt fona režīmā, ir tā pārslēgšana darba laikā. Vispirms startējiet programmu. Kamēr tā darbojas, nospiediet tastatūras taustiņu kombināciju **Control+Z** - tā procesu iemidzinās. Iemidzināts process nedarbojas - tā darbība ir apturēta, bet to var turpināt jebkurā laikā. Kad esat iemidzinājis procesu, jūs atgriezāties komandrindā. Tad ir iespējams procesu iedarbināt fona režīmā ar komandu

```
% bg
```

Tagad iemidzinātais process darbojas fona režīmā.

### 11.2. Procesu darbināšana priekšplānā

Ja jums jānodod informācija procesam, kurš darbojas fona režīmā, jūs to varat iedarbināt priekšplāna režīmā. Ja jums ir tikai viens fona process, jūs to varat atgriezt priekšplāna režīmā, ievadot komandu

```
% fg
```

Ja programma nav beigusi darbu, tā pārņems termināļa vadību. Reizēm programma, strādājot fona režīmā, būs beigusi darbu. Tad jūs ieraudzīsiet paziņojumu

```
[1]+ Done /bin/ls $LS_OPTIONS
```

Šis paziņojums nozīmē, ka fona process (šajā gadījumā `ls` - ne pārāk interesanti) ir beidzis darbu.

Ir iespējams fona režīmā vienlaikus darbināt vairākus procesus. Ja tā, tad jums jāzina, kuru procesu jūs vēlaties atgriezt priekšplāna režīmā. Vienkārša `fg` ievade priekšplānā atgriezīs procesu, kurš tika pārslēgts fona režīmā kā pēdējais. Ko darīt, ja jūs fona režīmā darbinājat vairākus procesus? Par laimi `bash` ir iekļauta komanda, kas parāda visus fona procesus. Šo komandu sauc `jobs` un tā izvada informāciju, līdzīgu šai:

```
% jobs
[1] Stopped vim
[2]- Stopped amp
```

```
[3]+ Stopped          man ps
```

Šajā sarakstā norādīti visi fona režīmā pārslēgtie procesi. Kā redzams, tie visi ir apturēti jeb iemidzināti. Skaitlis ir tāds kā identifikators visiem fona procesiem. Identifikators ar + simbolu (**man ps**) nozīmē, ka šis ir process, kurš tiks pārslēgts darbam priekšplāna režīmā, vienkārši ievadot **fg**.

Ja jūs vēlaties priekšplāna režīmā pārslēgt **vim**, ievadiet

```
% fg 1
```

**vim** pārņems konsoles vadību. Procesu pārslēgšana fona režīmā var būt ļoti noderīga, ja jums ir tikai viens terminālis (piemēram, izmantojot iezvanpieejas pieslēgumu). Jūs varat izmantot vairākas programmas, periodiski pārslēdzoties starp tām.

## 11.3. ps

Tagad jūs zināt, kā pārslēgties uz priekšu un atpakaļ starp dažādiem procesiem, kas startēti komandrindā. Jūs zināt arī to, ka vienlaikus sistēmā darbojas daudzi procesi. Kā tos aplūkot? Nu, jūs varat izmantot komandu **ps** (1). Šai komandai ir daudz dažādu parametru, tādēļ šeit aplūkosim tikai svarīgākos. Pilnam parametru sarakstam aplūkojiet **ps man** dokumentu. **man** dokumentu izmantošana ir aplūkota sadaļā 2.1.

Vienkārša **ps** ievade attēlos to programmu sarakstu, kuras darbojas jūsu terminālī, t.sk. priekšplāna procesus (čaulu, kuru jūs izmantojat un, protams, pašu **ps**). Tiks parādīti arī fona procesi. Bieži vien tas būs ļoti īss saraksts.

```
% ps
  PID TTY          TIME CMD
 7923 tty0      00:00:00 bash
 8059 tty0      00:00:00 ps
```

Pat, ja šāds saraksts nav garš, informācija ir ļoti tipiska. Izmantojot **ps**, jūs iegūsit šādas pat kolonas neatkarīgi no tā, cik procesi darbojas. Tātad, ko tas viss nozīmē?

**PID** ir procesa identifikators. Katram strādājošam procesam tiek piešķirts unikāls identifikators no 1 līdz 32767. Katram jaunam procesam tiek piešķirts nākamais brīvais PID. Kad process beidz darbu (vai tiek nogalināts, kā aplūkosim nākamajā sadaļā), tas atbrīvo savu PID. Kad tiek sasniegts maksimālais PID, nākamais brīvais tiek izvēlēts, sākot skaitīt no sākuma.

Kolona **TTY** norāda, kurā terminālī process darbojas. Ievadot vienkārši **ps**, sarakstā tiks izvadītas tikai tās programmas, kas darbojas tekošajā terminālī, tādēļ visiem procesiem **TTY** kolonā būs tā vienāda informācija. Kā redzams, abi attēlotie procesi darbojas terminālī **ttyp0**. Tas norāda, ka tie vai nu tiek darbināti attālināti, vai no kāda X termināļa.

**TIME** kolona norāda, cik CPU laika process ir patērējis. Tas atšķiras no laika, cik ilgi process darbojas. Atcerieties, ka *Linux* ir daudzuzdevumu operētājsistēma. Vienlaikus darbojas daudzi procesi, un katrs var izmantot tikai nelielu daļu procesora laika. Tātad, **TIME** kolonai būtu jārāda daudz mazāk laika katram procesam, nekā tas patiesībā darbojas. Ja jūs šajā kolonā redzat laiku, kas lielāks par dažām minūtēm, kaut kas nav labi.

Beidzot, **CMD** kolonā norādīts, kas konkrētā programma īsti ir. Tajā parādīts tikai programmas pamatnosaukums, nevis komandrindas parametri vai līdzīga informācija. Lai iegūtu arī papildinformāciju, jāizmanto vienu no daudzajiem **ps** parametriem. To aplūkosim pēc brīža.

Jūs varat iegūt pilnu sistēmā strādājošo procesu sarakstu, izmantojot pareizo parametru kombināciju. Rezultāts, ticamākais, būs garš procesu saraksts (piecdesmit pieci manā klēpj datorā brīdī, kad rakstu šo teikumu), tādēļ es saīsināšu šo sarakstu:

```
% ps -ax
  PID TTY          STAT TIME COMMAND
   1  ?            S     0:03  init [3]
```

```

 2 ?      SW      0:13 [kflushd]
 3 ?      SW      0:14 [kupdate]
 4 ?      SW      0:00 [kpiod]
 5 ?      SW      0:17 [kswapd]
11 ?      S       0:00 /sbin/kerneld
30 ?      SW      0:01 [cardmgr]
50 ?      S       0:00 /sbin/rpc.portmap
54 ?      S       0:00 /usr/sbin/syslogd
57 ?      S       0:00 /usr/sbin/klogd -c 3
59 ?      S       0:00 /usr/sbin/inetd
61 ?      S       0:04 /usr/local/sbin/sshd
63 ?      S       0:00 /usr/sbin/rpc.mountd
65 ?      S       0:00 /usr/sbin/rpc.nfsd
67 ?      S       0:00 /usr/sbin/crond -l10
69 ?      S       0:00 /usr/sbin/atd -b 15 -l 1
77 ?      S       0:00 /usr/sbin/apmd
79 ?      S       0:01 gpm -m /dev/mouse -t ps2
94 ?      S       0:00 /usr/sbin/automount /auto file /etc/auto.misc
106 tty1   S       0:08 -bash
108 tty3   SW      0:00 [agetty]
109 tty4   SW      0:00 [agetty]
110 tty5   SW      0:00 [agetty]
111 tty6   SW      0:00 [agetty]
[saraksts saīsināts]

```

Lielākajā daļā sistēmu lielākā daļa šo procesu tiek startēti sistēmas ielādes laikā. Es esmu veicis dažas izmaiņas savā sistēmā, tādēļ jūsu sistēmas procesu saraksts varētu būt nedaudz cits, tomēr lielāko daļu no šiem procesiem jūs redzēsiet arī savā sistēmā. Nesen atklātās kodola *ptrace* ievainojamības rezultātā tika radīts labojums, kas vairs nerāda komandrindas parametrus daudziem procesiem. Tos tagad rāda kvadrātiekvās kā procesiem ar PID no 108 līdz 111. Tas arī parāda dažas citas kolonas un citu interesantu informāciju.

Vispirms jums jāievēro, ka lielākajai daļai šo procesu norādīts, ka tie darbojas terminālī ?. Šie procesi nav piesaistīti nevienam terminālim. Visbiežāk šāda situācija sastopama dēmoniem, kas ir procesi, kuri darbojas, nepiesaistoties terminālim. Tipiski dēmoni ir **sendmail**, **BIND**, **apache** un **NFS**. Tie parasti gaida pieprasījumu no klienta un atgriež tam informāciju.

Otrkārt, redzam jaunu kolonu: **STAT**. Tā parāda procesa statusu. **S** nozīmē, ka process guļ (*sleeping*) - process, gaida, kad kaut kas notiks. **Z** nozīmē, ka process ir *zombēts* - tas ir, šī procesa mātesprocess ir miris, neaizvācos savus apakšprocesus. Tas nav labi. **D** nozīmē, ka process atrodas nepārtraucamā guļas stāvoklī - bieži vien šie procesi atsakās *mirt* pat tad, ja tiem nosūta **SIGKILL**. Par **SIGKILL** jūs vairāk varat izlasīt nākamajā sadaļā par **kill**. **W** nozīmē, ka process ierakstīts maiņatmiņā. Miris process tiek apzīmēts ar **x**. Process, kas apzīmēts ar **T**, tiek trasēts vai ir apturēts. **R** nozīmē, ka process ir izpildāms.

Ja jūs vēlaties ieraudzīt pat vēl vairāk informācijas par procesiem, izmēģiniet šo:

```

% ps -aux
USER          PID  %CPU  %MEM    VSZ   RSS TTY   STAT  START   TIME  COMMAND
root           1   0.0   0.0   344    80 ?     S     Mar02   0:03  init [3]
root           2   0.0   0.0     0     0 ?     SW    Mar02   0:13  [kflushd]
root           3   0.0   0.0     0     0 ?     SW    Mar02   0:14  [kupdate]
root           4   0.0   0.0     0     0 ?     SW    Mar02   0:00  [kpiod]

```

```

root      5  0.0  0.0    0    0 ?    SW   Mar02  0:17 [kswapd]
root     11  0.0  0.0  1044   44 ?    S    Mar02  0:00 /sbin/kerneld
root     30  0.0  0.0  1160    0 ?    SW   Mar02  0:01 [cardmgr]
bin      50  0.0  0.0  1076  120 ?    S    Mar02  0:00 /sbin/rpc.port
root     54  0.0  0.1  1360  192 ?    S    Mar02  0:00 /usr/sbin/sysl
root     57  0.0  0.1  1276  152 ?    S    Mar02  0:00 /usr/sbin/klog
root     59  0.0  0.0  1332   60 ?    S    Mar02  0:00 /usr/sbin/inet
root     61  0.0  0.2  1540  312 ?    S    Mar02  0:04 /usr/local/sbi
root     63  0.0  0.0  1796   72 ?    S    Mar02  0:00 /usr/sbin/rpc.
root     65  0.0  0.0  1812   68 ?    S    Mar02  0:00 /usr/sbin/rpc.
root     67  0.0  0.2  1172  260 ?    S    Mar02  0:00 /usr/sbin/cron
root     77  0.0  0.2  1049  316 ?    S    Mar02  0:00 /usr/sbin/apmd
root     79  0.0  0.1  1100  152 ?    S    Mar02  0:01 gpm -m /dev/mo
root     94  0.0  0.2  1396  280 ?    S    Mar02  0:00 /usr/sbin/auto
chris   106  0.0  0.5  1820  680 tty1 S    Mar02  0:08 -bash
root   108  0.0  0.0  1049    0 tty3 SW   Mar02  0:00 [agetty]
root   109  0.0  0.0  1049    0 tty4 SW   Mar02  0:00 [agetty]
root   110  0.0  0.0  1049    0 tty5 SW   Mar02  0:00 [agetty]
root   111  0.0  0.0  1049    0 tty6 SW   Mar02  0:00 [agetty]
[saraksts saisināts]

```

Šeit redzam daudz informācijas. Būtībā tiek pievienota informācija par to, kurš lietotājs startējis procesu, cik sistēmas resursu process patērē (%CPU, %MEM, VSZ un RSS kolonas) un kurā datumā process startēts. Saprotams, ka tas ir liels daudzums informācijas, kas var palīdzēt sistēmas administratoram. Tas arī parāda ko citu - informāciju nav iespējams ierakstīt ekrānā, tādēļ katras rindiņas labā mala tiek apgriezta, līdz ar to nav redzama. Parametrs `-w` liks `ps` aplauzt garas rindiņas.

Protams, nav izcili skaisti, tomēr nepieciešamā informācija ir iegūta. Tagad jums ir pilns katra procesa saraksts. Patiesībā par katru procesu iespējams attēlot vēl vairāk informācijas. Aplūkojiet ļoti detalizēto `ps man` dokumentu. Jebkurā gadījumā, šajā sadaļā aplūkoti parametri ir vispopulārākie un tos jums būs jāizmanto visbiežāk.

## 11.4. kill

Reizēm programmas sāk slikti uzvesties un tās ir jāsaved kārtībā. Šāda veida administrēšanas rīks ir `kill(1)`, un to var izmantot dažādām manipulācijām ar procesiem. Visuzskatāmākais `kill` pielietojums ir procesa *nogalināšana* jeb likvidēšana. Šādi jārikojas, ja programma uzvedas slikti un izmanto pārāk daudz sistēmas resursu, vai ja jums tā vienkārši apnikusi.

Lai nogalinātu procesu, jums jāzina tā PID vai nosaukumu. Lai noskaidrotu PID, izmantojiet komandu `ps`, kā aplūkojām iepriekšējā sadaļā. Piemēram, lai nogalinātu procesu `4747`, jāievada komandu:

```
% kill 4747
```

Ievērojiet, ka, lai nogalinātu procesu, jums jābūt tā īpašniekam. Tā ir sistēmas drošības īpašība. Ja jums būtu tiesības nogalināt arī citu lietotāju startētos procesus, būtu iespējams veikt visdažādāko kaitniecību. Protams, `root` var nogalināt jebkuru sistēmas procesu.

Ir arī otrs `kill` komandas veids - to sauc par `killall(1)`. Šī programma veic tieši to, ko izsaka tās nosaukums - nogalina visus procesus ar noteiktu nosaukumu. Ja jūs vēlaties nogalināt visus `vim` procesus, ievadiet komandu:

```
% killall vim
```

Tiks nogalināti visi jūsu startētie `vim` procesi. Izpildot šo komandu ar `root` tiesībām, tiks nogalināti visi `vim` procesi. Tā piemēram, aplūkosim interesantu veidu, kā izmest no sistēmas visus lietotājus (arī

sevi):

```
# killall bash
```

Nogalinot procesus ar vienkāršu `kill`, tie ne vienmēr grib mirt. Atsevišķi procesi nemirst ar `kill`. Jums būs jāizmanto *spēcīgāku* `kill` paveidu. Ja tas riebīgais PID 4747 neatbildēja uz jūsu nogalināšanas pieprasījumu, jūs varat ievadīt:

```
% kill -9 4747
```

Tas gandrīz noteikti nogalinās procesu 4747. To pašu iespējams veikt ar `killall`. Atšķirība starp šo `kill` un iepriekšējo ir tāda, ka procesam tiek nosūtīts cits signāls. Parasts `kill` nosūta signālu `SIGTERM` (*terminate* - beigt darbu), kas nosaka, ka procesam jāpabeidz darbu, ko tas veic, jādzēš visus pagaidu datus, kas radīti darba procesā un jābeidzas. `kill -9` nosūta signālu `SIGKILL` (*kill* - nogalināt), kas vienkārši beidz procesu, neļaujot tam likvidēt pagaidu datus. Tā reizēm var notikt sliktas lietas, kā piemēram, datu zudumi. Jūsu rīcībā ir vesels saraksts ar signāliem. To var uzzināt ievadot:

```
% kill -l
```

```
1) SIGHUP      2) SIGINT      3) SIGQUIT     4) SIGILL
5) SIGTRAP     6) SIGABRT    7) SIGBUS      8) SIGFPE
9) SIGKILL     10) SIGUSR1   11) SIGSEGV    12) SIGUSR2
13) SIGPIPE    14) SIGALRM   15) SIGTERM    17) SIGCHLD
18) SIGCONT    19) SIGSTOP   20) SIGTSTP    21) SIGTTIN
22) SIGTTOU    23) SIGURG    24) SIGXCPU    25) SIGXFSZ
26) SIGVTALRM 27) SIGPROF   28) SIGWINCH   29) SIGIO
30) SIGPWR
```

Skaitli jāizmanto kā `kill` parametru, kamēr `killall` komandai jāizmanto nosaukumu bez `SIG`. Piemērs:

```
% killall -KILL vim
```

Vēl kāds `kill` pielietojums ir procesu pārstartēšana. `SIGHUP` nosūtīšana lielākajai daļai programmu liks atkārtoti nolasīt savus konfigurācijas failus. Tas ir īpaši noderīgi, lai norādītu sistēmas procesiem, ka jānolasa konfigurācijas failus pēc to modificēšanas.

## 11.5. top

Visbeidzot, *Linux* sistēmā ir arī komanda, ko var izmantot, lai attēlotu dinamiski atjauninātu informāciju par sistēmā strādājošajiem procesiem. Šo komandu sauc `top` (1) un to startē šādi:

```
% top
```

Tā visā termināļa ekrānā attēlos informāciju par strādājošajiem procesiem, kā arī dažādu vispārēju sistēmas informāciju - kā vidējo noslodzi, procesu skaitu, CPU statusu, informāciju par pieejamo atmiņu, kā arī detalizētu informāciju par procesiem, iekļaujot PID, lietotāju, prioritāti, procesora un atmiņas izmantojuma informāciju, darbības laiku un programmas nosaukumu.



```
6:47pm up 1 day, 18:01, 1 user, load average: 0.02, 0.07, 0.02
61 processes: 59 sleeping, 2 running, 0 zombie, 0 stopped
CPU states: 2.8% user, 3.1% system, 0.0% nice, 93.9% idle
Mem: 257992K av, 249672K used, 8320K free, 51628K shrd, 78248K buff
Swap: 32764K av, 136K used, 32628K free, 82600K cached
```

| PID  | USER  | PRI | NI | SIZE  | RSS  | SHARE | STAT | LIB | %CPU | %MEM | TIME  | COMMAND     |
|------|-------|-----|----|-------|------|-------|------|-----|------|------|-------|-------------|
| 112  | root  | 12  | 0  | 19376 | 18M  | 2468  | R    | 0   | 3.7  | 7.5  | 55:53 | X           |
| 4947 | david | 15  | 0  | 2136  | 2136 | 1748  | S    | 0   | 2.3  | 0.8  | 0:00  | screenshot  |
| 3398 | david | 7   | 0  | 20544 | 20M  | 3000  | S    | 0   | 1.5  | 7.9  | 0:14  | gimp        |
| 4946 | root  | 12  | 0  | 1040  | 1040 | 836   | R    | 0   | 1.5  | 0.4  | 0:00  | top         |
| 121  | david | 4   | 0  | 796   | 796  | 644   | S    | 0   | 1.1  | 0.3  | 25:37 | wmSMPmon    |
| 115  | david | 3   | 0  | 2180  | 2180 | 1452  | S    | 0   | 0.3  | 0.8  | 1:35  | wmaker      |
| 4948 | david | 16  | 0  | 776   | 776  | 648   | S    | 0   | 0.3  | 0.3  | 0:00  | xwd         |
| 1    | root  | 1   | 0  | 176   | 176  | 148   | S    | 0   | 0.1  | 0.0  | 0:13  | init        |
| 189  | david | 1   | 0  | 6256  | 6156 | 4352  | S    | 0   | 0.1  | 2.4  | 3:16  | licq        |
| 4734 | david | 0   | 0  | 1164  | 1164 | 916   | S    | 0   | 0.1  | 0.4  | 0:00  | rxvt        |
| 2    | root  | 0   | 0  | 0     | 0    | 0     | SW   | 0   | 0.0  | 0.0  | 0:08  | kflushd     |
| 3    | root  | 0   | 0  | 0     | 0    | 0     | SW   | 0   | 0.0  | 0.0  | 0:06  | kupdate     |
| 4    | root  | 0   | 0  | 0     | 0    | 0     | SW   | 0   | 0.0  | 0.0  | 0:00  | kpiod       |
| 5    | root  | 0   | 0  | 0     | 0    | 0     | SW   | 0   | 0.0  | 0.0  | 0:04  | kswapd      |
| 31   | root  | 0   | 0  | 340   | 340  | 248   | S    | 0   | 0.0  | 0.1  | 0:00  | kerneld     |
| 51   | root  | 0   | 0  | 48    | 48   | 32    | S    | 0   | 0.0  | 0.0  | 0:00  | dhcpcd      |
| 53   | bin   | 0   | 0  | 316   | 316  | 236   | S    | 0   | 0.0  | 0.1  | 0:00  | rpc.portmap |
| 57   | root  | 0   | 0  | 588   | 588  | 488   | S    | 0   | 0.0  | 0.2  | 0:01  | syslogd     |

Šo programmu sauc par `top` (augšgalis) tādēļ, ka procesoru visintensīvāk noslogojošās programmas tiks attēlotas saraksta augšgalā. Interesanti, ka, pateicoties CPU noslodzei, `top` tiks attēlots pirmais lielākajā daļā maznoslogoto sistēmu (un dažās noslogotās). Neraugoties uz to, `top` ir samērā noderīga, lai noteiktu, kura programma uzvedas slikti un ir nogalināma.

Tomēr pieņemsim, ka jūs vēlaties aplūkot tikai savus procesus, vai tikai kāda cita lietotāja procesus. Tie var arī nebūt starp visintensīvāk strādājošajiem. Parametrs `-u` ļauj jums norādīt lietotājevārdu vai UID un aplūkot tikai tos, procesus, kas pieder noteiktajam UID.

```
% top -u alan
PID USER PR NI VIRT RES SHR S %CPU %MEM TIME+ COMMAND
3622 alan 13 0 11012 10m 6956 S 1.0 2.1 0:03.66 gnome-terminal
3739 alan 13 0 1012 1012 804 R 0.3 0.2 0:00.06 top
3518 alan 9 0 1312 1312 1032 S 0.0 0.3 0:00.09 bash
3529 alan 9 0 984 984 848 S 0.0 0.2 0:00.00 startx
3544 alan 9 0 640 640 568 S 0.0 0.1 0:00.00 xinit
3548 alan 9 0 8324 8320 6044 S 0.0 1.6 0:00.30 gnome-session
3551 alan 9 0 7084 7084 1968 S 0.0 1.4 0:00.50 gconfd-2
3553 alan 9 0 2232 2232 380 S 0.0 0.4 0:00.05 esd
3555 alan 9 0 2552 2552 1948 S 0.0 0.5 0:00.10 bonobo-activati
3557 alan 9 0 2740 2740 2224 S 0.0 0.5 0:00.05 gnome-smproxy
3559 alan 9 0 6496 6492 5004 S 0.0 1.3 0:00.31 gnome-settings-
3565 alan 9 0 1740 1740 1440 S 0.0 0.3 0:00.28 xscreensaver
3568 alan 9 0 7052 7052 4960 S 0.0 1.4 0:02.28 metacity
```

```
3572 alan      9    0 11412  11m 7992 S  0.0  2.2   0:01.58 gnome-panel
3574 alan      9    0 12148  11m 8780 S  0.0  2.4   0:00.64 nautilus
3575 alan      9    0 12148  11m 8780 S  0.0  2.4   0:00.00 nautilus
3576 alan      9    0 12148  11m 8780 S  0.0  2.4   0:00.00 nautilus
```

Kā redzams, es šobrīd darbinu **x**, **top**, **gnome-terminal** (kurā es rakstu šo tekstu), kā arī daudzus citus ar X logu sistēmu saistītus procesus, kas manā sistēmā aizņem visvairāk procesora laika. Šī ir laba metode, lai novērotu, cik intensīvi lietotāji strādā ar jūsu sistēmu.

**top** arī atbalsta procesu aplūkošanu pēc to PID, gulošo un zombijprocesu ignorēšanu, kā arī daudzus citus konfigurācijas parametrus. Vislabākā vieta, kur aplūkot šos iestatījumus, ir **top man** dokuments.

## 12.nodaļa. Sistēmas administrēšanas pamati

Oppā... Es zinu, ko jūs domājat. "Es neesmu nekāds sistēmadministrators! Es nemaz negribu būt sistēmadministrators!"

Patiesībā ir tā - jūs esat administrators jebkuram datoram, kuram jūs zināt `root` paroli. Tas var būt gan jūsu galddators ar vienu vai diviem lietotājiem, gan liels serveris ar daudziem simtiem lietotāju. Neatkarīgi no tā, jums jāzina, kā pārvaldīt lietotājus un pareizi sagatavot sistēmu izslēgšanai. Šie uzdevumi liekas vienkārši, tomēr ir dažas nianšes, ko jāatceras.

### 12.1. Lietotāji un grupas

Kā minēts 8.nodaļā, jums parasti nevajadzētu strādāt sistēmā ar `root` privilēģijām. Tā vietā jums būtu jāizveido vienkāršu lietotāju, ko izmantot ikdienā, un jāizmanto `root` tikai sistēmas administrēšanas darbiem. Lai izveidotu lietotāju, jūs varat izmantot gan *Slackware* komplektā iekļautos rīkus, gan manuāli modificēt paroli failus.

#### Iekļautie rīki

Vienkāršākais veids, kā veikt lietotāju un grupu pārvaldību, ir sistēmā iekļauto skriptu un programmu izmantošana. Lietotāju pārvaldībai *Slackware* ir iekļautas programmas `adduser`, `userdel(8)`, `chfn(1)`, `chsh(1)` un `passwd(1)`. Komandas `groupadd(8)`, `groupdel(8)` un `groupmod(8)` ir paredzētas grupu pārvaldībai. Visi šie rīki, izņemot `chfn`, `chsh` un `passwd`, ir paredzēti izpildei ar `root` tiesībām, un tādēļ izvietoti direktoriņā `/usr/sbin`. `chfn`, `chsh` un `passwd` var izpildīt jebkurš lietotājs, un tie ir izvietoti direktoriņā `/usr/bin`.

Lietotāju pievienošanai jāizmanto programmu `adduser`. Sāksim ar visu programmas izpildes soļu aplūkošanu, apskatot visus tās uzdotos jautājumus un noskaidrojot, kas ko nozīmē. Noklusētā atbilde sniegta kvadrātiekvās, un to iespējams izvēlēties kā atbildi gandrīz visiem jautājumiem, ja vien jūs nevēlaties kaut ko mainīt.

```
# adduser
```

```
Login name for new user []: jellyd
```

Šis ir vārds, ko lietotājs izmantos, lai autorizētos sistēmā. Parasti lietotājevārdi sastāv no astoņiem vai mazāk simboliem un tiek pierakstīti ar mazajiem burtiem. Protams, jūs varat izmantot vairāk, kā astoņus simbolus, arī izmantot burtus, tomēr iesakām no tiem izvairīties, ja vien jums nav kāds svarīgs iemesls.

Lietotājevārdu iespējams norādīt arī kā komandrindas argumentu:

```
# adduser jellyd
```

Katrā gadījumā pēc lietotājevārda norādes `adduser` pieprasīs ievadīt lietotāja ID:

```
User ID ('UID') [ noklusētā vērtībā ir kārtējais brīvais ID ]:
```

Lietotāja ID (UID - User ID) ir veids, kā patiesībā *Linux* sistēmā tiek noteiktas lietotāju tiesības. Katram sistēmas lietotājam ir piekārtots unikāls skaitlis - *Slackware* sistēmā šie skaitļi sākas ar 1000. Jūs varat brīvi ievadīt UID, vai arī ļaut `adduser` piešķirt veidojamajam lietotājam kārtējo brīvo vērtību.

```
Initial group [users]:
```

Pēc noklusējuma visi lietotāji tiek pievienoti grupai `users`. Jūs varētu vēlēties jauno lietotāju pievienot citai grupai, tomēr tas nav ieteicams, ja nu vienīgi jūs patiešām zināt, ko darāt.

```
Additional groups (comma separated) []:
```

Šis jautājums ļauj jums pievienot jauno lietotāju papildgrupām. Katrs lietotājs var vienlaikus atrasties vairākās grupās - tas noder, ja jūs esat izveidojis grupas dažādiem sistēmas darbiem - kā tīmekļa lapas failu modificēšana, spēļu spēlēšana u.c. Piemēram, dažos serveros ir definēta grupa `wheel` kā

vienīgā, kurai piederošie lietotāji var izmantot komandu `su`. Noklusētā *Slackware* sistēma izmanto grupu `sys` tiem lietotājiem, kuriem ir tiesības atskaņot skaņu, izmantojot datora skaņas karti.

```
Home directory [/home/jellyd]
```

Mājas direktorijas pēc noklusējuma tiek izvietotas `/home` direktorijā. Ja jūs uzturat ļoti lielu sistēmu, var gadīties, ka mājas direktorijas ir pārvietotas citur (vai tiek izvietotas dažādās direktorijās). Šis solis ļauj noteikt, kur atradīsies lietotāja mājas direktorija.

```
Shell [ /bin/bash ]
```

`bash` ir *Slackware Linux* noklusētā čaula, un tā derēs lielākajai daļai lietotāju. Ja jūsu jaunajam lietotājam ir *Unix* pieredze, viņš var būt pazīstams ar citu čaulu. Šeit jūs varat norādīt, kādu čaulu šis lietotājs izmantos. Lietotājs to var veikt arī vēlāk, izmantojot komandu `chsh`.

```
Expiry date (YYYY-MM-DD) []:
```

Lietotājus iespējams izveidot tā, ka noteiktā datumā tiem beidzas lietošanas termiņš. Pēc noklusējuma šāds lietošanas termiņš netiek uzstādīts, tomēr to iespējams mainīt pēc jūsu ieskatiem. Šī iespēja var būt noderīga *Internet* pakalpojumu sniedzējiem, kuri vēlas, lai lietotāja autorizācijas derīguma termiņš beigtos noteiktā datumā, ja vien netiek saņemts maksājums par nākamo gadu.

```
New account will be created as follows:
```

```
-----  
Login name:          jellyd  
UID:                 [ Next available ]  
Initial group:      users  
Additional groups:  [ None ]  
Home directory:    /home/jellyd  
Shell:              /bin/bash  
Expiry date:        [ Never ]
```

Tas arī viss... Ja jūs vēlaties atcelt lietotāja konta izveidi, nospiediet **Control+C**. Ja esat nolēmis, ka visi ievadītie dati ir korekti un vēlaties izveidot lietotāju, nospiediet **ENTER**.

```
Creating new account...
```

```
Changing the user information for jellyd  
Enter the new value, or press return for the default  
  Full Name []: Jeremy  
  Room Number []: Smith 130  
  Work Phone []:  
  Home Phone []:  
  Other []:
```

Visa šī informācija ir ievadāma pēc jūsu ieskatiem. Ja nevēlaties, varat neievadīt neko no šīs informācijas. No otras puses, lietotājs to var jebkurā brīdī ievadīt pats, izmantojot komandu `chfn`. Jebkurā gadījumā var izrādīties noderīgi ievadīt vismaz pilnu lietotāja vārdu un tālruna numuru, ja nu vēlāk nepieciešams ar šo lietotāju sazināties.

```
Changing password for jellyd
```

```
Enter the new password (minimum of 5, maximum of 127 characters)  
Please use a combination of upper and lower case letters and numbers.  
New password:  
Re-enter new password:  
Password changed.  
  
Account setup complete.
```

Šeit jums jāievada jaunā lietotāja paroli. Parasti, ja jaunais lietotājs nav pieejams un līdz ar to nevar pats uzstādīt savu paroli, tiek uzstādīta kāda administratora izvēlēta noklusētā parole, un lietotājam tiek pieteikts, lai nomaina paroli uz ko drošāku.

**Piezīme: paroles izvēle.** Droša parole ir svarīgākais pasākums, kā cīnīties pret datora uzlaušanu. Jūs nevēlaties viegli uzminamu paroli, jo tāda padara ielaušanos sistēmā daudz vienkāršāku. Ideālā gadījumā droša parole ir gadījuma kārtībā izvēlēta simbolu virkne, kas sastāv no burtiem (gan augšējā, gan apakšējā reģistra jeb lielajiem un mazajiem), skaitļiem un dažādiem speciālajiem simboliem (Tab simbols nebūtu gudra izvēle, jo šis simbols ir atkarīgs no tā, kādu datoru jūs izmantojat, lai pieslēgtos sistēmai). Internet tīklā ir pieejamas daudzas programmatūras pakotnes, kas ļaus jums ģenerēt gadījuma paroles.

Vispārējs ieteikums: paroļu izvēlē izmantojiet veselo saprātu. Neizvēlieties paroli, kas ir jums pazīstamas personas dzimšanas diena, vispārzināmu teicienu, kaut ko, kas atrodams uz jūsu rakstāmgalda, vai jebko citu, ko viegli saistīt ar jums. Parole `secure1` vai jebkura cita parole, kas atrodama presē vai *Internet* tīklā, arī ir slikta.

Lietotāju dzēšana ir ļoti vienkārša. Vienkārši izpildiet `userdel`, norādot lietotājvārdu. Jums jāpārliecinās, ka lietotājs nav autorizējies sistēmā, un ka ar šī lietotāja tiesībām nedarbojas neviens process. Atcerieties arī, ka, dzēšot lietotāju, visa informācija par lietotāja paroli arī tiks neatgriezeniski dzēsta.

```
# userdel jellyd
```

Šī komanda no sistēmas dzēš to riebīgo lietotāju `jellyd`. Jauka atbrīvošanās! :) Lietotājs tiek dzēsts no `/etc/passwd`, `/etc/shadow` un `/etc/group` failiem, tomēr netiek dzēsta lietotāja mājas direktorijs.

Lai dzēstu arī lietotāja mājas direktoriju, dzēšanai jāizmanto komandu

```
# userdel -r jellyd
```

Lietotāja pagaidu bloķēšanu aplūkosim nākamajā sadaļā par parolēm, jo pagaidu bloķēšana patiesībā ir lietotāja paroles maiņa. Cita lietotāju informācijas maiņa aplūkota nedaudz tālāk.

Programmas grupu pievienošanai un dzēšanai ir ļoti vienkāršas. `groupadd` vienkārši pievienos jaunu ierakstu failam `/etc/group` ar unikālu grupas identifikatoru, kamēr `groupdel` dzēsīs norādīto grupu. Lai pievienotu lietotājus kādai noteiktai grupai, jūs varat manuāli modificēt failu `/etc/groups`. Piemēram, pievienosim grupu `cvs`:

```
# groupadd cvs
```

Tagad dzēsīsim to:

```
# groupdel cvs
```

## Paroļu maiņa

Paroles maina ar programmu `passwd`, kura modificē failu `/etc/shadow`. Šis fails satur visas sistēmas lietotāju paroles šifrētā veidā. Lai mainītu savu paroli, ievadiet

```
% passwd
```

```
Changing password for chris
```

```
Old password:
```

```
Enter the new password (minimum of 5, maximum of 127 characters)
```

```
Please use a combination of upper and lower case letters and numbers.
```

```
New password:
```

Kā redzams, jums vispirms jāievada savu veco paroli. To ievadot, tā neparādīsies uz ekrāna - gluži, kā autorizējoties darbam sistēmā. Tad jums tiks piedāvāts ievadīt jauno paroli. `passwd` veic dažādas jūsu izvēlētas paroles pārbaudes, un *protestēs*, ja jūsu parole neatbilst `passwd` kritērijiem. Protams, ja vēlaties, varat ignorēt `passwd` brīdinājumus. Lai apstiprinātu paroles maiņu, jums to būs jāievada

atkārtoti.

Ja jūs esat autorizējies ar `root` tiesībām, jūs varat mainīt arī citu lietotāju paroles:

```
# passwd ted
```

Jums būs jāizpilda iepriekš aprakstīto procedūru, vienīgi nebūs jāievada lietotāja veco paroli (viens no daudzajiem `root` labumiem).

Ja nepieciešams, jūs varat arī uz laiku bloķēt lietotāju, un atļaut tam autorizēties vēlāk. Gan lietotāja bloķēšanu, gan atbloķēšanu veic ar `passwd`. Lai bloķētu lietotāju, kā `root` izpildiet

```
# passwd -l david
```

Šī komanda nomainīs `david` paroli uz tādu, kas nevar atbilst nevienai šifrētai vērtībai. Vēlāk lietotāju iespējams atjaunot, izmantojot komandu

```
# passwd -u david
```

Tagad `david` lietotājs atkal var autorizēties. Lietotāja bloķēšana var būt noderīga, ja lietotājs negrib spēlēties pēc jūsu noteikumiem, vai ir eksportējis ļoti lielu `xeyes` (1) uz jūsu `x` darbvirsma.

## Lietotāja informācijas maiņa

Ir divi parametri, ko lietotāji var vienmēr mainīt - čaulu un *finger* informāciju (*finger information*). *Slackware Linux* šo vērtību maiņai izmanto komandas `chsh` (*change shell* - mainīt čaulu) un `chfn` (*change finger* - mainīt *finger* informāciju).

Lietotājs var izvēlēties jebkuru čaulu, kas ierakstīta failā `/etc/shells`. Lielākajai daļai lietotāju ļoti labi derēs `/bin/bash`. Citiem varētu patikt čaula, kas atrodama to sistēmā darbā vai skolā, un tie varētu vēlēties lietot to, ko jau pārziņa. Lai mainītu savu čaulu, izmantojiet `chsh`:

```
% chsh
Password:
Changing the login shell for chris
Enter the new value, or press return for the default
Login Shell [/bin/bash]:
```

Pēc paroles ievades ievadiet pilnu ceļu uz jauno čaulu. Vispirms pārlicinieties, ka tā atrodama failā `/etc/shells(5)`. `root` lietotājs var mainīt jebkura lietotāja čaulu, izpildot `chsh` un tam kā parametru norādot atbilstošo lietotājvārdu.

*Finger* informācija ir lietotāja pilnais vārds, tālruņa numurs un informācija par kabinetu. To iespējams mainīt, izmantojot `chfn` - maiņas process ir tāds pats, kā lietotāja veidošanas laikā. Kā vienmēr, `root` var mainīt jebkura lietotāja *finger* informāciju.

## 12.2. Lietotāji un grupas - grūtais ceļš

Protams, ir iespējams arī pievienot, modificēt un dzēst lietotājus un grupas, neizmantojot *Slackware* iekļautos skriptus un programmas. Tas nav sarežģīti, tomēr pēc iepazīšanās ar procesu jūs, ticamākais, sapratīsiet, ka skriptus izmantot ir daudz vienkāršāk. Tomēr ir svarīgi saprast, kā patiesībā tiek glabāta informācija par parolēm, ja gadījumā ir nepieciešams šo informāciju atjaunot un jums nav piekļuves *Slackware* rīkiem.

Vispirms mēs pievienosim jaunu lietotāju failiem `/etc/passwd(5)`, `/etc/shadow(5)` un `/etc/group(5)`. `passwd` fails satur informāciju par sistēmas lietotājiem, tomēr (dīvainā kārtā) nesatur informāciju par parolēm. Savulaik arī parolu informācija tika glabāta šajā failā, tomēr drošības dēļ jau sen vairs netiek tur glabāta. `passwd` failam jābūt lasāmam visiem lietotājiem, tomēr jūs nevēlaties, lai visi varētu lasīt šifrētas paroles, jo potenciālie iebrucēji šo informāciju varētu izmantot, lai tās atšifrētu. Tādēļ šifrētās paroles tiek glabātas failā `shadow`, kas ir lasāms tikai `root` lietotājam, un visas paroles `passwd` failā tiek norādītas kā `x`. Failā `group` norādītas visas grupas un informācija

par to, kādi lietotāji katrā iekļauti.

Lai droši labotu `/etc/passwd` failu, jūs varat izmantot komandu `vipw`, un komandu `vigr` - lai labotu `/etc/group`. Izmantojiet `vipw -s`, lai droši labotu failu `/etc/shadow` ("droši" šeit nozīmē to, ka neviens cits nevarēs modificēt šos failus, kamēr jūs tos labojat. Ja jūs esat vienīgais sistēmas administrators, droši vien, ka jūs jau esat drošībā, tomēr vislabāk ir apgūt labas iemaņas jau sākumā).

Aplūkosim failu `/etc/passwd` un mēģināsim pievienot jaunu lietotāju. Tipisks ieraksts `passwd` failā izskatās šādi:

```
chris:x:1000:100:Chris Lumens,Room 2,,:/home/chris:/bin/bash
```

Katra faila rindiņa ir ieraksts vienam lietotājam, un katras rindiņas lauki ir atdalīti ar kolu. Lauki atbilst lietotājevārdam, šifrētai parolei ("**x**", jo Slackware sistēma izmanto `shadow` failu paroli glabāšanai), lietotāja ID, grupas ID, izvēles *finger* informācijai, mājas direktorijai un čaulai. Lai manuāli pievienotu jaunu lietotāju, pievienojiet jaunu rindiņu faila beigās, aizpildot nepieciešamo informāciju.

Ievadītajai informācijai jāatbilst dažām prasībām - pretējā gadījumā jaunajam lietotājam var rasties problēmas ar autorizēšanos sistēmā. Vispirms pārliecinieties, ka paroles lauks satur **x**, un ka gan lietotājevārds, gan lietotāja ID ir unikāli. Piešķiriet lietotājam grupu - vai nu 100 (*Slackware* grupa `users`), vai jūsu izvēlēto noklusēto grupu (tās numuru, nevis nosaukumu). Piešķiriet lietotājam derīgu mājas direktoriju (to izveidosim vēlāk) un čaulu (atcerieties - derīgās čaulas norādītas failā `/etc/shells`).

Pēc tam mums jāpievieno ierakstu failam `/etc/shadow`. Tas saturēs lietotāja paroli šifrētā veidā. Tipisks ieraksts izskatās šādi:

```
chris:$1$w9bsw/N9$uwLr2bRER6YyBS.CAEp7R.:11055:0:99999:7:::
```

Arī šeit katra rindiņa atbilst vienam lietotājam un lauki ir atdalīti ar kolu. Norādāmie lauki ir: lietotājevārds, parole šifrētā veidā, dienas kopš Ēras sākuma (1970.gada 1.janvāris), kad parole pēdējoreiz mainīta, dienas pirms paroles termiņa beigām, kad lietotājs tiek brīdināts, dienas pēc paroles termiņa beigām, kad lietotājs tiek bloķēts, dienas kopš Ēras sākuma, kad lietotājs ir bloķēts, kā arī rezervētais lauks.

Kā redzams, lielākā daļa lauku satur informāciju par paroles termiņu. Ja jūs to neizmantojat, jums jāizpilda tikai daži lauki - pretējā gadījumā vispirms jāveic dažus aprēķinus un jāpieņem atbilstošus lēmumus. Jaunam lietotājam paroles laukā ievadiet jebkādu gadījuma informāciju - neuztraucieties par paroli, jo jūs to tūlīt mainīsiet. Vienīgais simbols, ko nevar iekļaut paroles laukā, ir kols. Lauku "dienas kopš paroles maiņas" arī atstājiet tukšu. Aizpildiet 0, 99999 un 7 tieši tāpat, kā piemērā, un pārējos laukus atstājiet tukšus.

(Tiem, kas domā, ka nupat no manas šifrētās paroles varēs uzzināt to nešifrētā veidā - uz priekšu! Ja jūs to uzlauzīsiet, jūs zināsiet manu paroli no testa sistēmas, kas aizsargāta ar ugunssienu. Tas būs ļoti noderīgi! :))

Visi parastie lietotāji tipiskā `Slackware` sistēmā pieder grupai `users`. Ja jūs tomēr vēlaties izveidot jaunu grupu, vai pievienot jaunu lietotāju papildgrupām, jums būs jālabo failu `/etc/group`. Šeit ir tipisks ieraksts:

```
cvs: :102:chris,logan,david,root
```

Šīs rindiņas lauki satur informāciju par grupas nosaukumu, grupas paroli, grupas ID un grupas dalībniekiem, kuri atdalīti ar komatu. Jaunas grupas izveidei vienkārši pievienojiet failam jaunu rindiņu ar unikālu grupas ID, un norādiet visus lietotājus, kuri pieder šai grupai. Visiem lietotājiem, kuri pieder šai jaunajai grupai un ir autorizējušies sistēmā, nāksies atteikties no autorizācijas un autorizēties atkārototi, lai izmaiņas stātos spēkā.

Šobrīd laba doma ir izmantot `pwck` un `grpck` komandas, lai pārliecinātos, ka jūsu veiktās izmaiņas ir savietojamas. Vispirms izpildiet `pwck -r` un `grpck -r`. Parametrs `-r` nosaka, ka netiek veiktas nekādas izmaiņas, tomēr tiek attēlotas visas izmaiņas, kuras būtu jāveic. Jūs varat izmantot šo programmu sniegto informāciju, lai izlemtu, vai nepieciešams veikt papildmodifikācijas šajos failos, izpildīt `pwck` vai `grpck` bez parametra `-r`, vai vienkārši atstāt visu, kā ir.

Tagad jūs varat izmantot komandu `passwd`, lai jaunajam lietotājam piešķirtu korektu paroli. Tad, izmantojot komandu `mkdir`, izveidojiet lietotāja mājas direktoriju, kā norādījat failā `/etc/passwd`, un, izmantojot `chown`, nomainiet jaunās direktorijas īpašnieku uz izveidoto lietotāju.

Lietotāja dzēšana ir vienkārša - jāizdzēš visi šim lietotājam atbilstošie ieraksti. Izdzēsiet lietotāja ierakstu no `/etc/passwd` un `/etc/shadow`, un nodzēsiet lietotājevārdu no visām grupām failā `/etc/group`. Ja vēlaties, izdzēsiet lietotāja mājas direktoriju, e-pasta `spool` (spolēšanas) failu un `crontab` ierakstu (ja tāds eksistē).

Grupu dzēšana ir līdzīga - izdzēsiet grupas ierakstu no `/etc/group`.

### 12.3. Sistēmas sagatavošana izslēgšanai

Ir ļoti svarīgi sistēmu izslēgt korekti. Vienkārša datora barošanas pārtraukšana ar slēdža palīdzību var radīt nopietnus failsistēmas bojājumus. Kamēr sistēma darbojas, pat, ja jūs neko nedarāt, faili vienalga tiek lietoti. Atcerieties, ka sistēmā nepārtraukti darbojas daudzi fona procesi. Šie procesi vada sistēmu un darbojas ar daudziem failiem. Kad sistēmas elektroapgāde tiek pēkšņi pārtraukta, šie faili netiek korekti aizvērti un var tikt bojāti. Atkarībā no tā, kādi tieši faili tiek bojāti, sistēmā var kļūt nelietojama. Katrā ziņā nākamajā sistēmas ielādes reizē būs jāveic ilgstošu failsistēmas pārbaudes procedūru.

**Piezīme:** ja jūs konfigurējat savu sistēmu darbam ar žurnālētu failsistēmu, kā `ext3` vai `reiserfs`, jūs būsiet daļēji pasargāts no failsistēmas bojājumiem, un jūsu failsistēmas pārbaudes procedūra arī būs īsāka, kā lietojot failsistēmu bez žurnālēšanas iespējas (piemēram, `ext2`). Tomēr šis drošības tīkls nav attaisnojums nekorekti izslēgtai sistēmai. Žurnālētas failsistēmas būtība ir pasargāt jūsu failus no bojājumiem, kas rodas gadījumos, kurus jūs nevarat kontrolēt, nevis no jūsu slinkuma.

Jebkurā gadījumā, pārstartējot vai izslēdzot datoru, ir svarīgi to veikt pareizi. Ir vairāki veidi, kā to veikt - jūs varat izvēlēties to, kas jums vislabāk patīk (vai prasa vismazāk darbu). Tā kā izslēgšana un pārstartēšana ir līdzīgas darbības, lielākā daļa metožu, ko izmanto sistēmas izslēgšanai, attiecas arī uz pārstartēšanu.

Pirmā metode ir programma `shutdown(8)`. Tā, šķiet, ir arī vispopulārākā. `shutdown` var izmantot, lai pārstartētu vai izslēgtu sistēmu iepriekš norādītā laikā, un tā var arī paziņot visiem sistēmā autorizētajiem lietotājiem, ka sistēma tiek sagatavota izslēgšanai.

Visvienkāršākais `shutdown` pielietojums datora izslēgšanai ir

```
# shutdown -h now
```

Šajā gadījumā mēs nesūtīsim pielāgotu paziņojumu lietotājiem - lietotāji redzēs `shutdown` noklusēto paziņojumu. `now` (*tagad*) ir laiks, kad mēs vēlamies veikt izslēgšanu, un parametrs `-h` nozīmē, ka sistēma tiek sagatavota izslēgšanai (nevis pārstartēta - tulk. piez.). Šis nav lietotājiem visdraudzīgākais veids daudzlietotāju sistēmā, tomēr labi darbojas mājas datorā. Labāka metode daudzlietotāju sistēmā būtu par sistēmas izslēgšanu lietotājiem paziņot jau nedaudz iepriekš:

```
# shutdown -h +60
```

Šī komanda sagatavos sistēmu izslēgšanai pēc 60 minūtēm, kas ir tieši laikā daudzlietotāju sistēmai. Kritiskām sistēmām izslēgšanu jāplāno jau krietnu laiku iepriekš, un jums jābrīdina lietotājus, izmantojot tam piemērotus apziņošanas veidus (e-pasts, ziņojumu dēlis, `/etc/motd` vai citus).

Sistēmas pārstartēšana ir līdzīga, vienīgi parametrs `-h` tiek aizstāts ar `-r`:

```
# shutdown -r now
```

Izmantojot `shutdown -r`, laiku norāda tieši tāpat, kā komandai `shutdown -h`. Ar `shutdown` komandu iespējams kontrolēt arī citas pārstartēšanas vai izslēgšanas laika nianses - vairāk informācijas meklējiet `man` dokumentā.

Otrs veids, kā sagatavot izslēgšanai vai pārstartēt jūsu *Slackware* sistēmu, ir `halt(8)` vai `reboot(8)` komandu izmantošana. Kā jau to norāda komandu nosaukumi, `halt` nekavējoties veiks



sistēmas sagatavošanu izslēgšanai, bet **reboot** veiks sistēmas pārstartēšanu (**reboot** patiesībā ir simbolsaite uz komandu **halt**). Tās iedarbina šādi:

```
# halt
# reboot
```

Sistēmas pārstartēšanai vai izslēgšanai iespējams izmantot arī zemāka līmeņa metodi - nodot atbilstošās komandas tieši **init**. Visas pārējās metodes patiesībā ir ērtāki veidi, kā "sarunāties" ar **init**, tomēr ir iespējams arī izmantot zemā līmeņa komandu - **telinit(8)** (*tell init* - paziņot **init**; ievērojiet, ka **telinit** komandā ir tikai viens "1"). **telinit** komanda paziņos **init**, uz kuru darblīmeni jāpārslēdzas, kas, savukārt, tiks izpildīts noteiktu skriptu. Šis skripts nogalinās vai startēs attiecīgajam darblīmenim nepieciešamos procesus. Šāda pārslēgšanās ir izmantojama arī sistēmas pārstartēšanai un sagatavošanai izslēgšanai, jo šīs abas sistēmas darba stadijas arī ir īpaši darblīmeņi.

```
# telinit 0
```

Darblīmenis 0 ir sistēmas darba pārtraukuma (**halt**) režīms. Pārslēgšanās uz darblīmeni 0 nogalinās visus sistēmas procesus, atmontēs visas failsistēmas un dators tiks sagatavots izslēgšanai. Šī ir pilnībā pieņemama metode datora sagatavošanai izslēgšanai. Daudzos klēpj datoros un mūsdienu galddatoros pāreja uz darblīmeni 0 arī fiziski izslēgs datoru.

```
# telinit 6
```

Darblīmenis 6 ir sistēmas pārstartēšanas režīms. Visi procesi tiks nogalināti, visas failsistēmas atmontētas, un sistēma tiks pārstartēta. Šī ir pilnībā pieņemama metode sistēmas pārstartēšanai.

Ja kādam interesē - kad notiek pārslēgšanās uz darblīmeņiem 0 vai 6 - vienalga, izmantojot **shutdown**, **halt**, vai **reboot** - tiek izpildīts skripts **/etc/rc.d/rc.6** (**/etc/rc.d/rc.0** ir simbolsaite uz **/etc/rc.d/rc.6**). Jūs šo skriptu varat pielāgot savai gaumei, tomēr esiet uzmanīgi un vispirms notestējiet izmaiņas!

Ir vēl kāda metode sistēmas pārstartēšanai. Visu pārējo metožu izmantošanai jums nepieciešamas **root** tiesības, tomēr izrādās, ka ir iespējams pārstartēt sistēmu pat, ja jums šo tiesību nav - ja vien jums ir fiziska piekļuve datora tastatūrai. **Control+Alt+Delete** ("trīspirkstu kombinācija") izmantošana tiks sistēmai pārstartēties gluži tāpat, kā, izmantojot iepriekšējās metodes. Šī kombinācija ne vienmēr darbojas **x** logu vidē - jums var nākties izmantot **Control+Alt+F1** (vai citu funkcionālo taustiņu), lai pārslēgtos uz termināli, kuru neaizņem **X** logu sistēma.

Un beidzot - fails, kas patiesībā vada visas sistēmas startēšanās un izslēgšanās nianses, ir **/etc/inittab(5)**. Parasti jums nevajadzētu modificēt šo failu, tomēr tas var sniegt ieskatu sistēmas procesos un ļaut noskaidrot, kādēļ ir lietas, kas darbojas tieši tā un ne citādi. Kā vienmēr - sīkākai informācijai aplūkojiet **man** dokumentāciju.

## 13.nodaļa. Svarīgākās tīkla komandas

Datortīkls sastāv no vairākiem savā starpā savienotiem datoriem - tie var būt gan daži mājās vai nelielā birojā saslēgti datori, gan liels augstskolas tīkls vai pat viss *Internet*. Ja jūsu dators arī ir tīkla daļa, arī jūs varat piekļūt šīm sistēmām - tieši vai izmantojot dažādus servisu, kā tīmeklis vai e-pasts.

Jūs varat izmantot dažādas tīkla programmas. Dažas ir ērtas tīkla diagnostikai - lai pārliecinātos, ka viss darbojas. Citas (kā e-pasta lasītāji vai tīmekļa pārlūki) noder, lai veiktu nepieciešamo darbu un sazinātos ar citiem cilvēkiem.

### 13.1. ping

`ping`(8) norādītajam resursdatoram nosūta `ICMP ECHO_REQUEST` paketi. Ja resursdators atbild, jūs saņemat šo paketi atpakaļ. Izklusās dīvaini? Nu jā - jūs varat "nopingot" jeb *ehotestēt* IP adresi, lai pārliecinātos, ka konkrētais resursdators darbojas. Ja ehotestēšanas atbildes nav, skaidrs - kaut kas nav labi. Aplūkosim piemēru sarunai starp diviem *Linux* lietotājiem:

Lietotājs A: *Loki* atkal nedarbojas

Lietotājs B: Tu esi pārliecināts?

Lietotājs A: Jā, es mēģināju to "nopingot", bet nav atbildes.

Šādas situācijas padara `ping` par ļoti noderīgu ikdienas komandu, piedāvājot ātru metodi, lai pārliecinātos, ka dators darbojas un ir pieslēgts datortīklam. Pamata sintakse ir šāda:

```
% ping www.slackware.com
```

Protams, ir iespējams norādīt arī vairākus parametrus. Aplūkojiet `ping(1)` man dokumentu.

### 13.2. traceroute

Slackware iekļautā `traceroute`(8) komanda ir ļoti noderīgs tīkla diagnostikas rīks. `traceroute` attēlo katru resursdatoru, caur kuru, cenšoties sasniegt mērķi, ceļo pakete. Lai aplūkotu, cik "lecienu" attālumā no *Slackware* serveriem jūs atrodaties, izmantojiet komandu:

```
% traceroute www.slackware.com
```

Tīks attēlots katrs resursdators un informācija par tā atbildes laiku. Piemērs:

```
% traceroute www.slackware.com
```

```
traceroute to www.slackware.com (204.216.27.13), 30 hops max, 40 byte packets
```

```
1  zuul.tdn (192.168.1.1) 0.409 ms  1.032 ms  0.303 ms
2  207.171.227.254 (207.171.227.254) 18.218 ms  32.873 ms  32.433 ms
3  border-sf-2-0-4.sirius.com (205.134.230.254) 15.66 ms  15.73 ms  16.142 ms
4  pb-nap.crl.net (198.32.128.20) 20.741 ms  23.672 ms  21.378 ms
5  E0-CRL-SFO-03-E0X0.US.CRL.NET (165.113.55.3) 22.29 ms  21.532 ms  21.29 ms
6  T1-CDROM-00-EX.US.CRL.NET (165.113.118.2) 24.54 ms  42.955 ms  58.443 ms
7  www.slackware.com (204.216.27.13) 38.115 ms  53.033 ms  48.328 ms
```

`traceroute`, līdzīgi `ping`, izmanto `ICMP` paketes. `traceroute` var norādīt vairākus parametrus - tie izskaidroti man dokumentā.

### 13.3. DNS rīki

Domēnu vārdu serviss (Domain Name Service - DNS) ir maģiskais protokols, kas ļauj jūsu datoram pārvērst bezjēdzīgos domēnu vārdus (kā `www.slackware.com`) saprotamās IP adresēs (kā

64.57.102.34). Datori nevar maršrutēt paketes uz `www.slackware.com`, bet var maršrutēt paketes uz šim domēna vārdam piekārtoto IP adresi. Tas ļauj mums ērti atcerēties datoru nosaukumus. Bez DNS mums nāktos atcerēties, kura IP adrese piesaistīta kuram datoram - pieņemot, ka IP adreses nemainās. Protams, ka datoru vārdu izmantošana ir ērtāka, tomēr kā sasaistīt vārdus ar adresēm?

## host

Domēnu vārdu sasaisti ar IP adresēm var veikt `host(1)` - ātrs un vienkāršs rīks bez bagātīgas funkcionalitātes.

```
% host www.slackware.com
www.slackware.com is an alias for slackware.com.
slackware.com has address 64.57.102.34
```

Pieņemsim, ka mums ir kāds iemesls, kādēļ vēlamies sasaistīt IP adresi ar domēna vārdu - ko tad?

## nslookup

`nslookup` ir programma, kas izturējusi laika pārbaudi, tomēr ir novecojusi un var tikt neiekļauta nākotnes versijās. Šai programmai pat nav `man` dokumenta.

```
$ nslookup 64.57.102.34
Note: nslookup is deprecated and may be removed from future releases.
Consider using the `dig' or `host' programs instead. Run nslookup with
the `-sil[ent]` option to prevent this message from appearing.
Server:          192.168.1.254
Address:         192.168.1.254#53
Non-authoritative answer:
www.slackware.com      canonical name = slackware.com.
Name:   slackware.com
Address: 64.57.102.34
```

## dig

Visspēcīgākais no šiem rīkiem ir domēnu informācijas savācējs (*Domain Information Groper*) `dig(1)`. `dig` no DNS servera var saņemt praktiski jebkādu informāciju - arī reversos ierakstus, **A**, **CNAME**, **MX**, **SP** un **TXT** ierakstus. `dig` ir pieejami daudzi komandrindas parametri un, ja jūs ar šo rīku neesat pazīstams, jums noteikti jāizlasa tā lielisko `man` dokumentu.

```
% dig @192.168.1.254 www.slackware.com mx

; <<>> DiG 9.2.2 <<>> @192.168.1.254 www.slackware.com mx
;; global options:  printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 26362
;; flags: qr rd ra; QUERY: 1, ANSWER: 2, AUTHORITY: 2, ADDITIONAL: 2

;; QUESTION SECTION:
;www.slackware.com.          IN      MX

;; ANSWER SECTION:
www.slackware.com.          76634  IN      CNAME   slackware.com.
```

```

slackware.com.      86400   IN      MX      1 mail.slackware.com.

;; AUTHORITY SECTION:
slackware.com.      86400   IN      NS      ns1.cwo.com.
slackware.com.      86400   IN      NS      ns2.cwo.com.

;; ADDITIONAL SECTION:
ns1.cwo.com.        163033  IN      A       64.57.100.2
ns2.cwo.com.        163033  IN      A       64.57.100.3

;; Query time: 149 msec
;; SERVER: 192.168.1.254#53(192.168.1.254)
;; WHEN: Sat Nov 6 16:59:31 2004
;; MSG SIZE rcvd: 159

```

Šīs komandas darbības rezultāts parāda `dig` darbības pamatprincipus. `@192.168.1.254` norāda izmantojamo DNS serveri, `www.slackware.com` ir domēna vārds, par kuru jāmeklē informāciju, savukārt `mx` ir meklējamā ieraksta tips. Pieprasījuma atbilde informē, ka e-pastu uz `www.slackware.com` patiesībā jāsūta serverim `mail.slackware.com`.

### 13.4. *finger*

`finger` (1) saņem un parāda informāciju par norādīto lietotāju. `finger` rīkam jānorāda lietotājvārdu vai e-pasta adresi, pēc kā tas mēģinās sazināties ar atbilstošo serveri un saņemt informāciju par lietotājvārdu, biroju, tālruņa numuru u.c. Piemērs:

```
% finger johnc@idsoftware.com
```

`finger` var atgriezt lietotājvārdu, e-pasta statusu, tālruņa numurus, kā arī failus `.plan` un `.project`. Protams, atgrieztā informācija dažādiem `finger` serveriem atšķiras. *Slackware* iekļautais serveris pēc noklusējuma atgriež šādu informāciju:

- lietotājvārds
- telpas numurs
- mājas tālruņa numurs
- darba tālruņa numurs
- autorizācijas statuss
- e-pasta statuss
- lietotāja mājas direktorijas `.plan` faila saturs
- lietotāja mājas direktorijas `.project` faila saturs

Pirmos četrus elementus iespējams uzstādīt ar komandu `chfn`. Tie tiek glabāti failā `/etc/passwd`. Lai mainītu informāciju `.plan` un `.project` failos, tos vienkārši jāmodificē ar jebkuru tekstu redaktoru. Šiem failiem jāatrodas lietotāja mājas direktoriņā un to nosaukumiem jābūt `.plan` un `.project`.

Daudzi lietotāji izmanto `finger`, lai uzzinātu, vai attālinātā datorā nav jauns e-pasts. Iespējams uzzināt arī lietotāja plānu vai projektu, pie kura tiek strādāts.

Līdzīgi, kā daudzām komandām, arī `finger` ir pieejami komandrindas parametri. Ar tiem iespējams iepazīties `man` dokumentā.

## 13.5. telnet

Kāds reiz teicis, ka `telnet` (1) ir vislabākais rīks, kas jebkad redzēts datoros. Spēja autorizēties un veikt darbības attālinātos datoros ir galvenā īpašība, kas atšķir *Unix* un tā paveidus no citām operētājsistēmām.

`telnet` ļauj jums autorizēties datorā tieši tāpat, kā, izmantojot datora lokālo konsoli. Kad lietotājvārds un parole ir pārbaudīti, jums tiek piedāvāta komandrinda, kuru izmantojot, iespējams veikt visas darbības, kuras iespējams veikt teksta konsolē - e-pasta rakstīšanu, vēstkopu lasīšanu, failu pārvietošanu u.t.t. Ja jūs izmantojat X logu sistēmu un ar `telnet` pieslēdzaties citam datoram, jūs varat darbināt X programmas attālinātajā datorā un attēlot tās savā.

Lai autorizētos attālinātā datorā, izmantojiet šādu sintaksi:

```
% telnet <resursdators>
```

Ja resursdators atbild, jūs saņemsit autorizācijas uzvedni. Ievadiet savu lietotājvārdu un paroli, un atradīsities attālinātā datora čaulā. Lai aizvērtu `telnet` sesiju, ievadiet komandu `exit` vai `logout`.

**Brīdinājums:** `telnet` nešifrē sūtīto informāciju. Tā visa tiek sūtīta vienkārša teksta veidā - pat paroles. Nav ieteicams izmantot `telnet` *Internet* tīklā - tā vietā apsveriet iespēju izmantot drošo čaulu (*Secure Shell* - SSH). Tā šifrē visu datu plūsmu un ir bezmaksas.

## Cits telnet lietojums

Tagad, kad esat pārliecināts, ka `telnet` protokolu labāk neizmanto autorizācijai attālinātos datoros, aplūkosim dažus lietderīgus `telnet` pielietojumus.

Jūs varat izmantot `telnet`, lai pieslēgtos attālināta datora noteiktam portam.

```
% telnet <resursdators> [ports]
```

Šī iespēja var būt visai noderīga, ja jums ātri jāpārbauda noteiktu servisu, jums nepieciešama pilna kontrole pār komandām, un jums jāredz, kas tieši notiek. Jūs varat interaktīvi pārbaudīt vai pat izmantot *SMTP* serveri, *POP3* serveri, *HTTP* serveri u.c.

Aplūkosim piemēru `telnet` lietojumam, lai pieslēgtos *HTTP* serverim pie 80.porta, un saņemtu pamatinformāciju no tā.

```
% telnet store.slackware.com 80
Trying 69.50.233.153...
Connected to store.slackware.com.
Escape character is '^]'.
HEAD / HTTP/1.0

HTTP/1.1 200 OK
Date: Mon, 25 Apr 2005 20:47:01 GMT
Server: Apache/1.3.33 (Unix) mod_ssl/2.8.22 OpenSSL/0.9.7d
Last-Modified: Fri, 18 Apr 2003 10:58:54 GMT
ETag: "193424-c0-3e9fda6e"
Accept-Ranges: bytes
Content-Length: 192
Connection: close
Content-Type: text/html

Connection closed by foreign host.
%
```

To pašu iespējams veikt arī citiem tekstuālajiem protokoliem, ja vien jūs zināt, pie kāda porta pieslēgties un kādas ir protokola izmantotās komandas.

### 13.6. Drošā čaula - SSH

Mūsdienās drošā čaula jeb SSH (*Secure SHell*) bauda to pašu apbrīnu, ko kādreiz baudīja `telnet`. `ssh(1)` ļauj pieslēgties pie attālināta datora un izpildīt komandas tā, it kā atrastos pie tā konsoles, tomēr `ssh` šifrē visu informāciju, kas ceļo starp abiem datoriem - pat, ja kāds pārķer tīkla datu plūsmu, šifrētā informācija nebūs saprotama. Aplūkosim tipisku drošās čaulas pieslēgumu.

```
% ssh carrier.lizella.net -l alan
The authenticity of host 'carrier.lizella.net (192.168.1.253)' can't be
established.
RSA key fingerprint is 0b:e2:5d:43:4c:39:4f:8c:b9:85:db:b2:fa:25:e9:9d.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'carrier.lizella.net' (RSA) to the list of
known hosts.
Password: password
Last login: Sat Nov  6 16:32:19 2004 from 192.168.1.102
Linux 2.4.26-smp.
alan@carrier:~$ ls -l MANIFEST
-rw-r--r--  1 alan users 23545276 2004-10-28 20:04 MANIFEST
alan@carrier:~$ exit
logout
Connection to carrier.lizella.net closed.
```

Šeit mēs pieslēdzamies serverim `carrier.lizella.net` un pārbaudam faila `MANIFEST` lietotāja tiesības.

### 13.7. E-pasts

E-pasts ir viena no populārākajām *Internet* tīklā veicamajām darbībām. 1998.gadā tika ziņots, ka nosūtīts vairāk e-pasta, kā parastā. E-pasts patiešām ir populārs un noderīgs.

*Slackware* sistēmā ir iekļauts standarta e-pasta serveris un vairāki e-pasta klienti. Visi šajā sadaļā aplūkoti e-pasta klienti darbojas teksta režīmā. Daudzi *Windows* lietotāji varētu protestēt, tomēr teksta režīma e-pasta klienti ir ļoti ērti - īpaši, ja e-pastu jāpārbauda attālināti, bez tam nav jāuztraucas - eksistē arī daudzi grafiskā režīma e-pasta klienti, kā *KDE* iekļautais *KMail*. Ja jūs vēlaties izmantot kādu no tiem, aplūkojiet to tiešsaistes palīdzību.

#### pine

`pine(1)` is not `elm` (`pine` nav `elm`). Apmēram tāds ir šīs programmas nosaukuma atšifrējums. Vašingtonas universitāte radīja šo programmu *Internet* vēstkopu un e-pasta aplūkošanai, jo šīs augstskolas studentiem bija nepieciešams vienkārši izmantojams e-pasta klients. `pine` ir viens no populārākajām mūsdienu e-pasta klientiem un ir pieejams gandrīz visām *Unix* sistēmām un pat *Windows*.

```

PINE 4.58      MAIN MENU                               Folder: INBOX  2 Messages

?      HELP      - Get help using Pine
C      COMPOSE MESSAGE - Compose and send a message
I      MESSAGE INDEX - View messages in current folder
L      FOLDER LIST - Select a folder to view
A      ADDRESS BOOK - Update address book
S      SETUP     - Configure Pine Options
Q      QUIT     - Leave the Pine program

Copyright 1989-2003. PINE is a trademark of the University of Washington.
[Folder "INBOX" opened with 2 messages]
? Help          P PrevCmd      R ReINotes
0 OTHER CMDS  > [Compose]  N NextCmd     K KBlock

```

Jūs redzēsiet izvēlni ar komandām un rindu ar komandu tastatūras taustiņu aprakstiem. `pine` patiesībā ir jaudīga programma, tādēļ neaplūkosit katru tās iespēju.

Lai aplūkotu savu ienākošo e-pastu, ievadiet `i`. jūsu e-pasta ziņojumi tiek attēloti, izmantojot saņemšanas datumu, autoru un tematu. Iezīmējiet vēlamo ziņojumu un nospiediet **Enter**, lai to aplūkotu. `r` nospiešana ļaus jums rakstīt atbildi uz šo vēstuli. Kad vēstule uzrakstīta, nospiediet **Ctrl+X**, lai to nosūtītu. Lai atgrieztos ziņojumu sarakstā, nospiediet `i`.

Ja jūs vēlaties dzēst ziņojumu, nospiediet `d`. Šī komanda atzīmēs izvēlēto ziņojumu dzēšanai. `pine` dzēš e-pastu, beidzot darbu ar programmu. `pine` ļauj arī sakārtot e-pastu mapēs. Mapju sarakstu iespējams aplūkot, nospiežot `l`. Lai saglabātu ziņojumu citā mapē, nospiediet `s`. jums būs jāizvēlas mapi, kurā ierakstīt ziņojumu.

`pine` piedāvā ļoti daudz iespējas - jums noteikti jāaplūko `man` dokumentu, jo tas satur jaunāko informāciju par programmu.

## elm

`elm(1)` ir vēl viens populārs teksta režīma e-pasta klients. Lai arī tas nav tik draudzīgs lietotājam, kā `pine`, tas noteikti ir vecāks.

```
Mailbox is '/var/mail/root' with 2 messages [ELM 2.5 PL6]

  1  Sep 17  Patrick J. Volkerd (174)  Welcome to Linux (Slackware 9.1)!
0  2  Sep 17                               (45)  Register with the Linux counter pr

You can use any of the following commands by pressing the first character:
d)delete or u)ndelete mail, m)ail a message, r)eply or f)orward mail, q)uit
  To read a message, press <return>.  j = move down, k = move up, ? = help

Command:
```

Atverot `elm`, pēc noklusējuma jūs atradīsities ienākošo e-pasta ziņojumu mapē. Ziņojumi ir attēloti ar to kārtas numuru, datumu, sūtītāju un tematu. Izmantojiet tastatūras kursora taustiņus, lai iezīmētu vēlamo ziņojumu. Nospiediet **Enter**, lai to izlasītu.

Lai sagatavotu jaunu ziņojumu, galvenajā ekrānā nospiediet `m`. Taustiņš `d` atzīmēs ziņojumu dzēšanai. `r` ļaus sagatavot atbildi atvērtajam ziņojumam. Visu šo taustiņu apzīmējumi attēloti ekrāna apakšdaļā.

`man` dokumentā `elm` aplūkots sīkāk, tādēļ jūs, ticamākais, vēlēsities pirms `elm` lietošanas ar to iepazīties.

## mutt

"Visi e-pasta klienti ir mēsli. Šis vienkārši ir mazāks mēsli." `mutt` sākotnējais interfeiss bija balstīts uz `elm`, pievienojot citos populāros e-pasta klientos sastopamas īpašības. Tā radās hibrīds - `mutt`.

Dažas no `mutt` īpašībām:

- krāsu atbalsts
- ziņojumu kārtošana pēc tematiem
- MIME un PGP/MIME atbalsts
- POP3 un IMAP atbalsts
- vairāku e-pasta kastīšu formātu atbalsts (*mbox, MMDF, MH, maildir*)
- ļoti plašas pielāgošanas iespējas



```
q:Quit d:Del u:Undel s:Save m:Mail r:Reply g:Group ?:Help
1 0 F Sep 17 To root ( 23) Register with the Linux counter project
2 + Sep 17 Patrick J. Volk ( 153) Welcome to Linux (Slackware 9.1)!

-----Mutt: /var/mail/root [Msgs:2 Old:1 9.5K]---(date/date)----- (all)-----
```

Ja jūs vēlaties e-pasta klientu, kas ļaus jums pilnībā kontrolēt visu, tad jums patiks `mutt`. Visus noklusētos iestatījumus iespējams pielāgot; iespējams arī mainīt tastatūras taustiņu piesaisti. Jūs vēlaties izveidot makrokomandu? Arī tas ir iespējams.

Jūs, iespējams, vēlēsit iepazīties arī ar `mutt.rc man` dokumentu - tajā aprakstīts, kā konfigurēt `mutt`. Iesakām aplūkot arī *Slackware* iekļauto `mutt.rc` faila piemēru.

## nail

`nail(1)` ir e-pasta klients ar komandrindas vadību. Tas ir ļoti primitīvs un lietotāja saskarnes ziņā nepiedāvā gandrīz neko, tomēr ir noderīgs, kad nepieciešams ātri nosūtīt e-pastu, veidot e-pasta sūtīšanas skriptus, pārbaudīt sava e-pasta servera instalāciju vai veikt līdzīgas darbības. Ievērojiet, ka Slackware izveido `nail` simbolsaites - `/usr/bin/mail` un `/usr/bin/mailx`. Visas šīs trīs komandas izpilda vienu un to pašu programmu. Patiesībā jūs bieži ieraudzīsiet, ka `nail` tiek saukts par `mail`.

Pamata komandrinda ir

```
% mailx <temats> <adresāts>
```

`nail` nolasa ziņojuma tekstu no standarta ievades kanāla, līdz ar to ir iespējams, izmantojot `cat`, nodot `nail` faila saturu, kas tiks nosūtīts kā e-pasts. Ir iespējams arī ievadīt ziņojuma tekstu komandrindā, beigās ievadot **Control+D**.

Piemērs programmas teksta nosūtīšanai:

```
% cat randomfunc.c | mail -s "Sheit ir taa funkcija" asdf@example.net
```

`man` dokuments izskaidro arī citas `nail` iespējas, tādēļ jūs, ticamākais, vēlēsities pirms `nail` izmantošanas ar to iepazīties.

## 13.8. Pārlūki

Pirmais, par ko parasti iedomājas, izdzirdot vārdu *Internet*, ir "*Internet* pārlūkošana" jeb tīmekļa lapu aplūkošana ar tīmekļa pārlūkprogrammu. Vidējam lietotājam šis, šķiet, ir vispopulārākais *Internet* lietošanas veids.

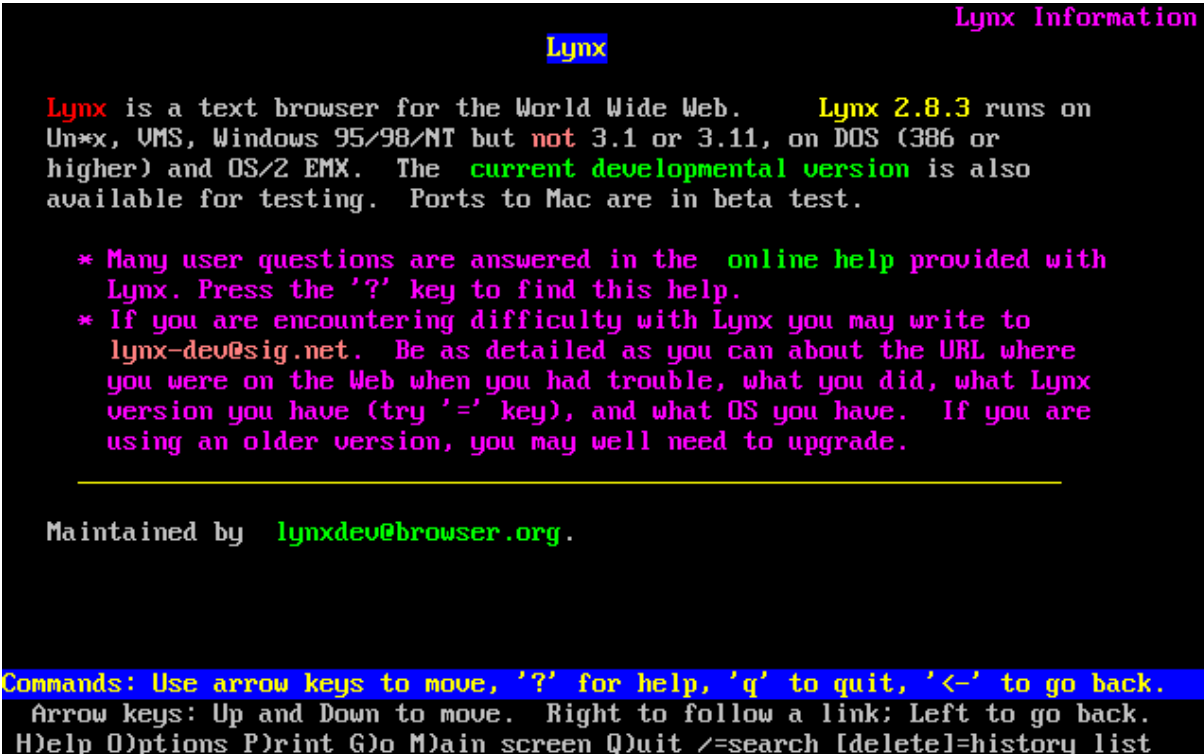
*Slackware* piedāvā populārus grafiskā režīmā tīmekļa pārlūkus programmatūras pakotņu sērijā **XAP**, kā arī teksta režīma pārlūkus **N** sērijā. Aplūkosim vispopulārākās pakotnes.

### lynx

**lynx** (1) ir teksta režīma tīmekļa pārlūks, kurš piedāvā ļoti ātru veidu, kā aplūkot informāciju *Internet* tīklā. Reizēm grafiskie elementi tikai traucē, ja jūs zināt, ko tieši vēlaties atrast.

Lai startētu **lynx**, vienkārši ievadiet **lynx** komandrindā:

```
% lynx
```



```

                                     Lynx Information
                                     _____
                                     Lynx
Lynx is a text browser for the World Wide Web.   Lynx 2.8.3 runs on
Un*x, VMS, Windows 95/98/NT but not 3.1 or 3.11, on DOS (386 or
higher) and OS/2 EMX. The current developmental version is also
available for testing.  Ports to Mac are in beta test.

* Many user questions are answered in the online help provided with
  Lynx. Press the '?' key to find this help.
* If you are encountering difficulty with Lynx you may write to
  lynx-dev@sig.net. Be as detailed as you can about the URL where
  you were on the Web when you had trouble, what you did, what Lynx
  version you have (try '=' key), and what OS you have. If you are
  using an older version, you may well need to upgrade.

-----
Maintained by lynxdev@browser.org.

Commands: Use arrow keys to move, '?' for help, 'q' to quit, '<' to go back.
Arrow keys: Up and Down to move. Right to follow a link; Left to go back.
H)elp O)ptions P)rint G)o M)ain screen Q)uit /=search [delete]=history list
```

Ir iespējams arī norādīt atveramo lapu:

```
% lynx http://www.slackware.com
```

**lynx** ekrāna apakšdaļā atrodama informācija par tastatūras komandu taustiņiem. Tastatūras taustiņi kursora pārvietošanai uz augšu un leju ļauj pārvietoties pa dokumentu. **Enter** ļauj izvēlēties hipersaiti un taustiņš kursora pārvietošanai pa kreisi ļauj atgriezties iepriekšējā lapā. **d** ievade lejuplādēs izvēlēto failu. **g** komanda parāda *Go* (*let uz* - tulk. piez.) komandrindu, kurā iespējams ievadīt atveramo URL.

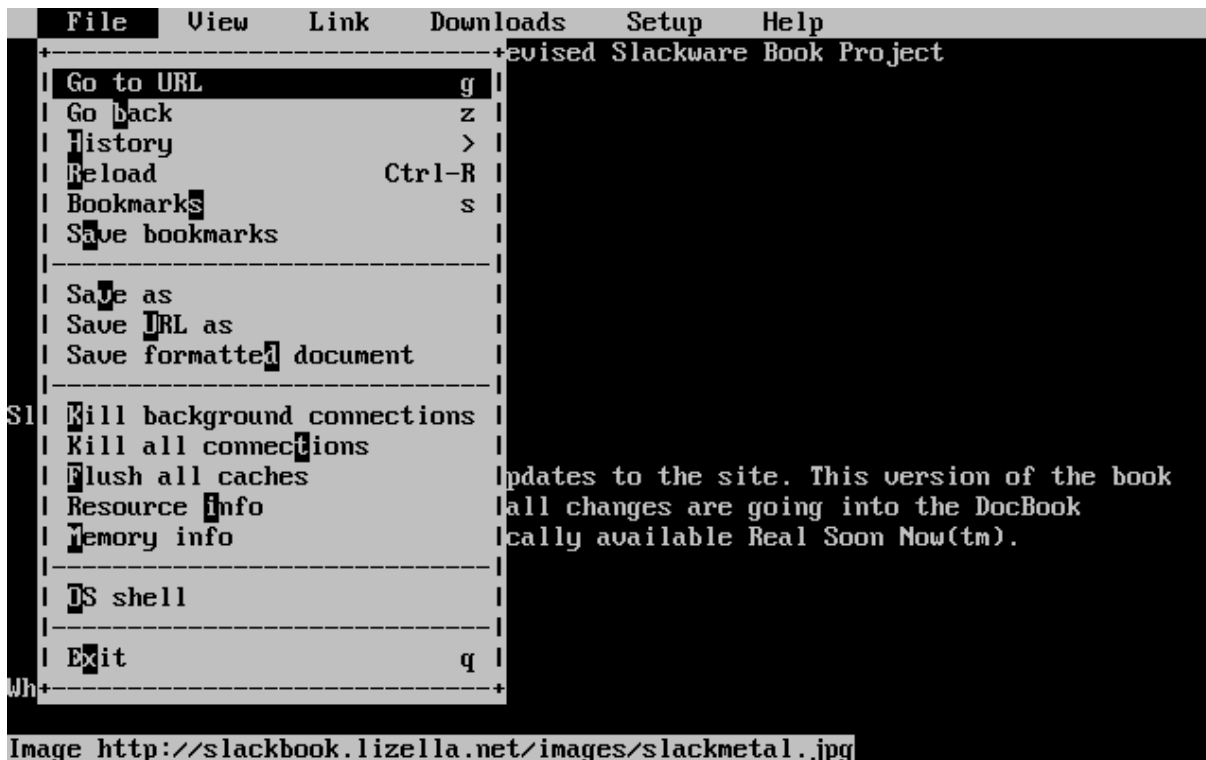
**lynx** ir pieejamas arī daudzas citas komandas. Ieteicams vai nu aplūkot **man** dokumentu, vai ievadīt **h**, lai aplūkotu palīdzības ekrānu.

### links

Līdzīgi, kā **lynx**, **links** ir teksta režīma tīmekļa pārlūkprogramma, kas ļauj visu navigāciju veikt ar

tastatūras palīdzību. Atšķirībā no `lynx`, `Esc` taustiņa nospiešana aktivizēs ļoti ērtu izvēlni ekrāna augšdaļā. Tā padara `links` ļoti ērti izmantojamu, atbrīvojot no nepieciešamības apgūt tastatūras vadības komandas. Lietotāji, kas teksta režīma pārlūku neizmanto katru dienu, noteikti novērtēs šo iespēju.

`links`, salīdzinot ar `lynx`, ir labāks kadru (frames) un tabulu atbalsts.



## wget

`wget` (1) ir komandrindas rīks, kas lejuplādē failus no norādītā URL. Lai gan tas nav īsts tīmekļa pārlūks, `wget` tiek lietots, lai lejuplādētu visu vai daļu tīmekļa vietnes lokālai aplūkošanai, vai ātrai failu lejuplādei no HTTP vai FTP serveriem. Pamatsintakse izskatās šādi:

```
% wget <URL>
```

Ir iespējams arī norādīt parametrus. Piemēram, šis parametrs lejuplādēs *Slackware* tīmekļa vietni:

```
% wget --recursive http://www.slackware.com
```

`wget` radīs direktoriju `www.slackware.com` un tajā saglabās failus tieši tā, kā tie tiek glabāti vietnē.

`wget` var arī lejuplādēt failus no FTP vietnēm - vienkārši norādiet FTP URL.

```
% wget ftp://ftp.gnu.org/gnu/wget/wget-1.8.2.tar.gz
--12:18:16--  ftp://ftp.gnu.org/gnu/wget/wget-1.8.2.tar.gz
           => `wget-1.8.2.tar.gz'
Resolving ftp.gnu.org... done.
Connecting to ftp.gnu.org[199.232.41.7]:21... connected.
Logging in as anonymous ... Logged in!
==> SYST ... done.    ==> PWD ... done.
==> TYPE I ... done.  ==> CWD /gnu/wget ... done.
==> PORT ... done.   ==> RETR wget-1.8.2.tar.gz ... done.
```

```
Length: 1,154,648 (unauthoritative)

100%[=====>] 1,154,648      209.55K/s   ETA
00:00

12:18:23 (209.55KB/s) - `wget-1.8.2.tar.gz' saved [1154648]
```

`wget` ir pieejami daudzi parametri, kas to padara ērtu specifisku skriptu (tīmekļa vietņu spoguļošanas u.c.) veidošanai. Vairāk informācijas `man` dokumentā.

### 13.9. FTP klienti

FTP nozīmē *File Transfer Protocol* (failu pārraides protokols). Tas ļauj sūtīt un saņemt failus starp diviem datoriem, no kuriem viens ir serveris, bet otrs - klients. Šajā sadaļā aplūkosim FTP klientu.

Ziņkārīgajiem - klients esat jūs. Serveris ir dators, kurš atbild jūsu FTP pieprasījumam un ļauj jums autorizēties. jūs lejuplādēsiet un augšuplādēsiet failus serverī. Klients nevar apkalpot FTP pieslēgumus - tas var tikai pieslēgties FTP serveriem.

#### ftp

Lai pieslēgtos FTP serverim, izpildiet komandu `ftp` (1) un norādiet resursdatoru:

```
% ftp <resursdators> [ports]
```

Ja resursdatorā darbojas FTP serveris, tas pieprasīs jums ievadīt lietotājvārdu un paroli. Jūs varat autorizēties kā lietotājs `anonymous` (anonīms). Anonīmās FTP vietnes ir ļoti populāras programmatūras arhīviem. Piemēram, lai lejuplādētu *Slackware Linux*, izmantojot FTP, jums jāizmanto anonīmo FTP pieslēgumu.

Pēc veiksmīgas autorizācijas jums tiks piedāvāta komandrinda `ftp`. FTP tiek izmantotas īpašas komandas, tomēr tās ir līdzīgas standartkomandām. Tabulā norādītas dažas no pamatkomandām darbam ar FTP.

| Komanda                             | Nozīme                                                    |
|-------------------------------------|-----------------------------------------------------------|
| <code>ls</code>                     | Attēlot failu sarakstu                                    |
| <code>cd &lt;direktorija&gt;</code> | Nomainīt darba direktoriju                                |
| <code>bin</code>                    | Uzstādīt bināro datu pārraides režīmu                     |
| <code>ascii</code>                  | Uzstādīt ASCII datu pārraides režīmu                      |
| <code>get &lt;fails&gt;</code>      | Lejuplādēt failu                                          |
| <code>put &lt;fails&gt;</code>      | Augšuplādēt failu                                         |
| <code>hash</code>                   | Pārslēgt heša atzīmes statistikas indikatoru              |
| <code>tick</code>                   | Pārslēgt baitu skaitītāja indikatoru                      |
| <code>prom</code>                   | Pārslēgt lejupielāžu interaktīvo režīmu                   |
| <code>mget &lt;maska&gt;</code>     | Lejuplādēt failu grupu. Šeit atļautas arī aizstājējzīmes  |
| <code>mput &lt;maska&gt;</code>     | Augšuplādēt failu grupu. Šeit atļautas arī aizstājējzīmes |

| Komanda | Nozīme                    |
|---------|---------------------------|
| quit    | Atslēgties no FTP servera |

Jūs varat arī izmantot dažas pašsaprotamas komandas: `chmod`, `delete`, `rename`, `rmdir`. Lai aplūkotu pilnu komandu sarakstu, ievadiet `help` vai `?`.

FTP ir samērā vienkārša programma, tomēr tai trūkst mūsdienās tik pierastās lietotāja saskarnes. `man` dokumentā aplūkoti `ftp` (1) komandrindas parametri.

```
ftp> ls *.TXT
200 PORT command successful.
150 Opening ASCII mode data connection for /bin/ls.
-rw-r--r--  1 root    100      18606 Apr  6  2002 BOOTING.TXT
-rw-r--r--  1 root    100      10518 Jun 13  2002 COPYRIGHT.TXT
-rw-r--r--  1 root    100         602 Apr  6  2002 CRYPTO_NOTICE.TXT
-rw-r--r--  1 root    100     32431 Sep 29  02:56 FAQ.TXT
-rw-r--r--  1 root    100    499784 Mar  3 19:29 FILELIST.TXT
-rw-r--r--  1 root    100    241099 Mar  3 19:12 PACKAGES.TXT
-rw-r--r--  1 root    100     12339 Jun 19  2002 README81.TXT
-rw-r--r--  1 root    100     14826 Jun 17  2002 SPEAKUP_DOCS.TXT
-rw-r--r--  1 root    100     15434 Jun 17  2002 SPEAK_INSTALL.TXT
-rw-r--r--  1 root    100      2876 Jun 17  2002 UPGRADE.TXT
226 Transfer complete.
ftp> tick
Tick counter printing on (10240 bytes/tick increment).
ftp> get README81.TXT
local: README81.TXT remote: README81.TXT
200 PORT command successful.
150 Opening BINARY mode data connection for README81.TXT (12339 bytes).
Bytes transferred: 12339
226 Transfer complete.
12339 bytes received in 0.208 secs (58 Kbytes/sec)
```

## ncftp

`ncftp` (1) (izrunā "*Nik FTP*") ir alternatīva tradicionālajam *Slackware* iekļautajam FTP klientam. Tā arī ir teksta režīma programma, tomēr tai ir vairākas priekšrocības, piemēram:

- Rindiņas aizpilde ar Tab
- Grāmatzīmju fails
- Brīvāka aizstājējzīmju izmantošana
- Komandu vēsture

Pēc noklusējuma `ncftp` mēģinās anonīmi autorizēties jūsu norādītajā serverī. Jūs varat arī likt `ncftp` parādīt autorizācijas komandrindu, izmantojot parametru `-u`. Kad būsit autorizējies, jūs varat izmantot tās pašas komandas, kā ar `ftp`. Atšķirība būs redzama ērtākā saskarnē - līdzīgā tai, ko izmanto `bash`.

```
ncftp /pub/linux/slackware > cd slackware-current/
Please read the file README81.TXT
it was last modified on Wed Jun 19 16:24:21 2002 - 258 days ago
```

```
CWD command successful.
ncftp ...ware/slackware-current > ls
BOOTING.TXT          FAQ.TXT              bootdisks/
CHECKSUMS            FILELIST.TXT        extra/
CHECKSUMS.asc       GPG-KEY             isolinux/
CHECKSUMS.md5       PACKAGES.TXT        kernels/
CHECKSUMS.md5.asc   PRERELEASE_NOTES   pasture/
COPYING             README81.TXT        rootdisks/
COPYRIGHT.TXT       SPEEKUP_DOCS.TXT    slackware/
CRYPTO_NOTICE.TXT    SPEEK_INSTALL.TXT   source/
CURRENT.WARNING     Slackware-HOWTO
ChangeLog.txt       UPGRADE.TXT
ncftp ...ware/slackware-current > get README81.TXT
README81.TXT:                12.29 kB  307.07
kB/s
```

## 13.10. Tērzēšana ar citiem

### wall

`wall(1)` piedāvā ātru metodi ziņojumu nosūtīšanai citiem sistēmas lietotājiem. Komandas pamatsintakse ir

```
% wall [fails]
```

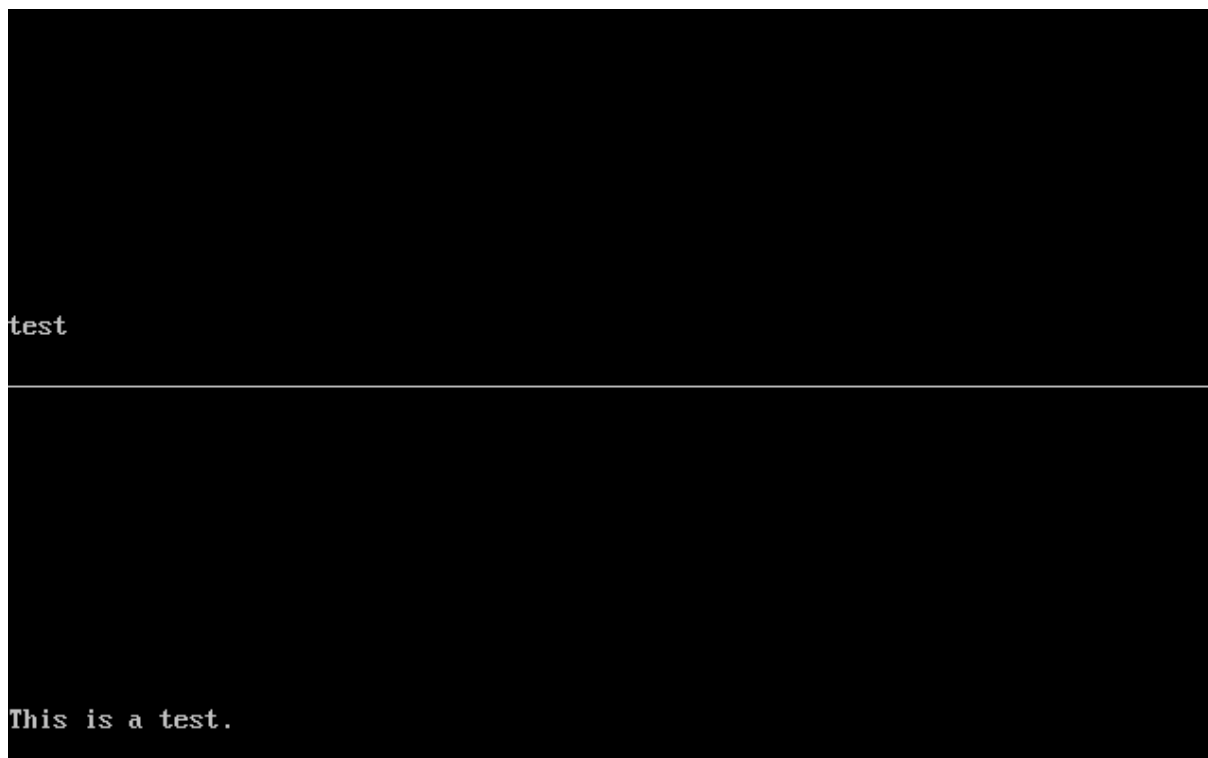
Šīs komandas izpildes rezultātā faila `[fails]` saturs tiks attēlots visu sistēmā autorizēto lietotāju termināļos. Ja jūs nenorādīsiet failu, `wall` attēlojamo tekstu nolasīs no standarta ievades kanāla - tekstu var ievadīt konsolē, beigās ievadot `Control+D`.

`wall` nav daudz papildiespēju, līdz ar to tā ir izmantojama paziņojumam, ka tiks veikti nopietni sistēmas tehniskās apkopes darbi vai tā pat tiks pārstartēta, lai dotu lietotājiem laiku saglabāt savu darbu un atslēgties no sistēmas. :)

### talk

`talk(1)` ļauj diviem lietotājiem tērzēt savā starpā. Šī programma sadala ekrānu divās daļās pa horizontāli. Lai pieprasītu tērzēšanu ar citu lietotāju, izmantojiet komandu

```
% talk <lietotājs> [TTY nosaukums]
```



Ja jūs norādīsiet tikai lietotājevārdu, tiek uzskatīts, ka tērzēšanas pieprasījums ir lokāls, tādēļ tiks nosūtīts tikai lokālam lietotājam. TTY nosaukumu ir jānorāda, ja vēlaties tērzēt ar lietotāju, kas pieslēdzies noteiktam terminālim (ja lietotājs ir autorizējies vairāk, kā vienreiz). `talk` nepieciešamo informāciju iespējams uzzināt, izmantojot komandu `w(1)`.

`talk` var arī sazināties ar lietotājiem attālinātos datoros, ja lietotājevārds norādīts kā e-pasta adrese.

`talk` ir samērā ierobežotas iespējas. Tā atbalsta tikai divus lietotājus un darbojas pusduplekta režīmā.

## ytalk

`ytalk(1)` ir atpakaļsavietojams `talk` aizstājējs. *Slackware* tas ir iekļauts kā komanda `ytalk`. Komandas sintakse ir līdzīga, tomēr tai ir dažas atšķirības:

```
% ytalk <lietotājs>[#TTY nosaukums]
```

```
-----= YTalk version 3.1.1 -----
test

-----= root@192.168.1.132#tty1 -----
this is a test
```

Lietotājvārdu un terminālu norāda līdzīgi, kā `talk`, izņemot to, ka šos abus parametrus jāsavieno kopā, izmantojot simbolu `#`.

`ytalk` piedāvā vairākus uzlabojumus:

- Tas atbalsta vairāk, kā divus lietotājus
- Jebkurā brīdī, izmantojot tastatūras taustiņu **Esc**, iespējams iedarbināt izvēlni ar iestatījumiem
- Ir iespējams iedarbināt čaulu, neaizverot `talk` sesiju
- Un vēl...

Ja jūs esat servera administrators, pārliedzinieties, ka `/etc/inetd.conf` failā ir atļauta `ntalk` porta izmantošana - tā nepieciešama `ytalk` korektam darbam.



## 14.nodaļa. Drošība

Jebkuras sistēmas drošība ir svarīga - tā var novērst attālinātu uzbrukumu veikšanu no jūsu sistēmas, kā arī aizsargāt svarīgus datus. Šajā nodaļā aplūkosim, kā aizsargāt jūsu *Slackware* datoru no pubertātes vecumu sasniegušiem jauniešiem, hakeriem un arī no negantiem kāmīšiem. Atcerieties, ka šeit aplūkotais ir tikai sistēmas drošības ieviešanas sākums - sistēmas drošība ir process, nevis stāvoklis.

### 14.1. Servisu aizliegšana

Pirmais solis, ko jāveic pēc *Slackware* instalēšanas, ir visu nevajadzīgo servisu aizliegšana. Potenciāli jebkurš sistēmas serviss var būt drauds drošībai, tādēļ ir svarīgi, lai sistēmā darbotos pēc iespējas mazāk servisu (t.i., tikai nepieciešamie). Servisi tiek startēti divos veidos - izmantojot *inetd* un inicializācijas skriptus.

#### Ar *inetd* startētie servisi

Daudzi *Slackware* iekļautie dēmoni tiek startēti ar *inetd(8)*. *inetd* ir dēmons, kas klausās visus tīkla portus, kurus tas ir konfigurēts klausīties un, tiklīdz pie kāda no tiem kāds mēģina pieslēgties, startē atbilstošo dēmonu. *inetd* var aizlielt startēt kādu dēmonu, komentējot atbilstošās rindiņas failā */etc/inetd.conf*. Lai to veiktu, atveriet šo failu jūsu iecienītajā tekstu redaktorā (piemēram, *vi*). Jūs redzēsiet rindiņas, kas līdzīgas šai:

```
telnet stream tcp nowait root /usr/sbin/tcpd in.telnetd
```

jūs varat aizlielt šo servisu, kā arī jebkuru citu, kurš nav nepieciešams, komentējot atbilstošo rindiņu, t.i., rindiņas sākumā pievienojot simbolu *#*. Iepriekšējā rindiņa pēc tās komentēšanas izskatīsies šādi:

```
#telnet stream tcp nowait root /usr/sbin/tcpd in.telnetd
```

Pēc *inetd* pārstartēšanas serviss būs aizliegts. *inetd* pārstartē ar komandu

```
# kill -HUP $(cat /var/run/inetd.pid)
```

#### Ar inicializācijas skriptiem startētie servisi

Pārējie servisi tiek startēti sistēmas ielādes laikā, izmantojot inicializācijas skriptus, kas atrodas direktoriņā */etc/rc.d/*. Tos iespējams aizlielt divos veidos - noņemot izpildes tiesības atbilstošajam inicializācijas skriptam vai komentējot atbilstošās inicializācijas skriptu rindiņas.

Piemēram, SSH tiek startēts, izmantojot skriptu */etc/rc.d/rc.sshd*. To iespējams aizlielt, izmantojot komandu:

```
# chmod -x /etc/rc.d/rc.sshd
```

Servisiem, kam nav sava inicializācijas skripta, jāveic rindiņu komentēšana. Piemēram, *portmap* dēmons tiek startēts ar šajās faila */etc/rc.d/rc.inet2* rindiņās:

```
# This must be running in order to mount NFS volumes.
# Start the RPC portmapper:
if [ -x /sbin/rpc.portmap ]; then
    echo "Starting RPC portmapper: /sbin/rpc.portmap"
    /sbin/rpc.portmap
fi
# Done starting the RPC portmapper.
```

Šo servisu iespējams aizlielt, pievienojot *#* to rindiņu sākumā, kas nesākas ar šo simbolu:

```
# This must be running in order to mount NFS volumes.
```

```
# Start the RPC portmapper:
#if [ -x /sbin/rpc.portmap ]; then
# echo "Starting RPC portmapper: /sbin/rpc.portmap"
# /sbin/rpc.portmap
#fi
# Done starting the RPC portmapper.
```

Šīs izmaiņas stāsies spēkā tikai pēc sistēmas pārstartēšanas vai darblīmeņu pārslēgšanas no 3 vai 4. Jūs to varat veikt, komandrindā ievadot (pēc pārslēgšanās uz darblīmeni 1 būs atkārtoti jāautorizējas):

```
# telinit 1
# telinit 3
```

## 14.2. Resursdatora piekļuves kontrole

### iptables

`iptables` ir Linux 2.4 un jaunāku kodolu pakešu filtra konfigurācijas programma. Kodola versija 2.4 (2.4.5, ja būsīm precīzi) pirmo reizi *Slackware* komplektā tika iekļauta (kā izvēles iespēja) versijā 8.0 un kļuva par noklusēto versijā 8.1. Šajā sadaļā aplūkoti tikai `iptables` lietojuma pamati - plašākai informācijai noteikti aplūkojiet <http://www.netfilter.org/>. Šīs komandas iespējams ievadīt failā `/etc/rc.d/rc.firewall`, kuru jāpadara izpildāmu, lai `iptables` iestatījumi stātos spēkā sistēmas ielādes laikā. Ievērojiet, ka nekorektas `iptables` komandas patiesībā var aizliegt jums pieslēgties jūsu datoram, tādēļ, ja vien neesat 100% pārliecināts par savām iemaņām, vienmēr pārliecinieties, ka jums ir lokāla pieeja datoram.

Pirmais, ko lielākajai daļai būtu jāveic, ir katras ienākošo datu ķēdes noklusētās politikas uzstādīšana uz `DROP` (nomest):

```
# iptables -P INPUT DROP
# iptables -P FORWARD DROP
```

Kad viss ir aizliegts, ir iespējams arī kaut ko atļaut. Pirmais, ko jāatļauj, ir jebkāda tīkla datu plūsma, kas notiek jau nodibinātās sesijās:

```
# iptables -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
```

Lai netraucētu atgriezenisko adresi izmantojošo programmu darbu, parasti laba doma ir pievienot šādu nosacījumu:

```
# iptables -A INPUT -s 127.0.0.0/8 -d 127.0.0.0/8 -i lo -j ACCEPT
```

Šis nosacījums atļauj jebkuru tīkla datu plūsmu no un uz `127.0.0.0/8` (`127.0.0.0-127.255.255.255`) atgriezeniskajā tīkla interfeisā (`lo`). Veidojot `iptables` nosacījumus, vēlams censties būt tik noteiktam, cik iespējams, lai būtu pārliecināts, ka izveidotie nosacījumu nejauši nepieļauj ko sliktu. Vienlaikus jāatceras, ka nosacījumi, kas pieļauj par maz, nozīmē vairāk papildnosacījumus un vairāk rakstīšanas.

Nākamais, ko jāveic, ir datu plūsmas atļaušana specifiskiem servisiem, kas darbojas jūsu datorā. Piemēram, ja jūs vēlaties savā datorā darbināt tīmekļa serveri, jūs izmantotu ko šādu:

```
# iptables -A INPUT -p tcp --dport 80 -i ppp0 -j ACCEPT
```

Šī komanda atļauj piekļuvi no jebkura datora pie jūsu datora porta 80, izmantojot tīkla interfeisu `ppp0`. Jūs varētu vēlēt ierobežot piekļuvi šim servisam tā, ka tam varētu piekļūt tikai noteikti datori. Šis nosacījums atļaus piekļuvi jūsu tīmekļa servisam tikai no adreses `64.57.102.34`:

```
# iptables -A INPUT -p tcp -s 64.57.102.34 --dport 80 -i ppp0 -j ACCEPT
```

Diagnostikas nolūkiem var noderēt ICMP datu plūsmas atļaušana. Lai no veiktu, izveidojiet šādu nosacījumu:

```
# iptables -A INPUT -p icmp -j ACCEPT
```

Daudzi vēlēsies savā vārtejas datorā izmantot tīkla adresu translāciju (Network Address Translation - NAT), lai citi lokālā tīkla datori varētu piekļūt *Internet* tīklam. Tam jāizveido šādu nosacījumu:

```
# iptables -t nat -A POSTROUTING -o ppp0 -j MASQUERADE
```

Lai šis nosacījums darbotos, jāatļauj IP pārsūtīšanu. To iespējams veikt, izmantojot šādu komandu:

```
# echo 1 > /proc/sys/net/ipv4/ip_forward
```

Lai atļautu IP pārsūtīšanu pastāvīgi (tā, ka izmaiņas saglabājas arī pēc sistēmas pārstartēšanas), jums jāmodificē failu `/etc/rc.d/rc.inet2`, nomainot rindiņu

```
IPV4_FORWARD=0
```

uz

```
IPV4_FORWARD=1
```

Vairāk informācijas par NAT atradīsiet dokumentā "NAT HOWTO"

(<http://www.netfilter.org/documentation/HOWTO/NAT-HOWTO.txt>).

## tcpwrappers

`tcpwrappers` kontrolē piekļuvi dēmoniņiem aplikāciju līmenī, nevis IP līmenī. Šāda iespēja var nodrošināt papildus drošības slāni gadījumos, kad IP līmeņa piekļuves kontrole (piemēram, *Netfilter*) nedarbojas korekti. Piemēram, ja jūs pārkompilējat kodolu un aizmirstat tajā iekļaut `iptables` atbalstu, jūsu IP līmeņa aizsardzība nedarbosies, tomēr `tcpwrappers` joprojām aizsargās jūsu sistēmu.

`tcpwrappers` aizsargāto servisu piekļuvi iespējams konfigurēt, izmantojot failus `/etc/hosts.allow` un `/etc/hosts.deny`.

Lielākai daļai lietotāju `/etc/hosts.deny` failā atradīsies tikai viena rindiņa, kas pēc noklusējuma aizliedz piekļuvi visiem dēmoniņiem:

```
ALL : ALL
```

Kad šī rindiņa ievadīta, varat koncentrēties uz atsevišķu servisu piekļuves atļaušanu noteiktiem resursdatoriem, domēniem vai IP adresu apgabaliem. To iespējams veikt failā `/etc/hosts.allow`, kura formāts ir līdzīgs.

Daudzi lietotāji sāks ar visu `localhost` pieslēgumu atļaušanu. To var veikt ar rindiņu

```
ALL: 127.0.0.1
```

Lai atļautu SSH piekļuvi no `192.168.0.0/24`, var izmantot kādu no šiem nosacījumiem:

```
sshd: 192.168.0.0/24
```

```
sshd: 192.168.0.
```

Ir iespējams arī ierobežot piekļuvi resursdatoriem noteiktos domēnos. To var veikt, izmantojot šādu nosacījumu (ievērojiet, ka šis nosacījums ir atkarīgs no reversā DNS ieraksta eksistences, tādēļ es neiesaku to izmantot *Internet* tīklam pieslēgtās sistēmās):

```
sshd : .slackware.com
```

## 14.3. Jauninājumu uzturēšana

### Vēstkopa *slackware-security*

Ja *Slackware* tiek atklāta drošības problēma, visiem vēstkopas `slackware-security@slackware.com` abonentiem tiek nosūtīts e-pasta ziņojums. Ziņojumi tiek izsūtīti par ievainojamībām jebkurā *Slackware* distribuīva daļā, izņemot programmatūru `/extra` un `/pasture` sadaļās. Šie drošības ziņojumi satur informāciju par jauninājumu saņemšanu, vai problēmas apiešanu,

ja tāda iespējama.

Informācija par parakstīšanos uz *Slackware* vēstkopām atrodama sadaļā 2.2.

## Direktorija */patches*

Kad *Slackware* tiek izlaistas atjauninātas pakotnes (jau izlaistu *Slackware* versiju gadījumā parasti tie ir tikai drošības jauninājumi), tās tiek novietotas direktorijā */patches*. Pilns ceļš uz jauninājumiem ir atkarīgs no izmantotā spoguļservera, tomēr tas būs pierakstīts formātā */ceļš/uz/slackware-x.x/patches/*.

Pirms pakotņu instalēšanas vēlams pārbaudīt pakotnes MD5 summu. `md5sum(1)` ir komandrindas rīks, kas jebkuram failam var izveidot unikālu matemātisku kontrolsummu. Ja kaut viens faila bits tiek mainīts, `md5sum` izveidotā kontrolsumma būs cita.

```
% md5sum package-<ver>-<arch>-<rev>.tgz
6341417aa1c025448b53073a1f1d287d package-<ver>-<arch>-<rev>.tgz
```

Šo summu jāsalīdzina ar jaunās pakotnes kontrolsummu, kas norādīta `slackware-$VERSION` saknes direktorijas failā `CHECKSUMS.md5` (arī */patches* direktorijā labojumiem) vai `slackware-security` vēstkopas ziņojumā.

Ja jums ir fails ar `md5sum` vērtībām tajā, jūs varat izmantot to, norādot parametru `-c`:

```
# md5sum -c CHECKSUMS.md5
./ANNOUNCE.10_0: OK
./BOOTING.TXT: OK
./COPYING: OK
./COPYRIGHT.TXT: OK
./CRYPTO_NOTICE.TXT: OK
./ChangeLog.txt: OK
./FAQ.TXT: FAILED
```

Kā redzams, jebkuri `md5sum` pārbaudītie faili, kuru kontrolsummas saskan, tiek atzīmēti ar "OK", kamēr faili, kuru kontrolsummas atšķiras, tiek atzīmēti ar "FAILED". (Jā, tas bija trieciens jūsu intelīģencei. Kādēļ jūs sacenšaties ar mani?)

## 15.nodaļa. Failu arhivēšana

### 15.1. gzip

`gzip` (1) ir GNU failu saspiešanas programma. Tā saspiež vienu failu. Programmas pamatlietoņums ir šāds:

```
% gzip fails
```

Iegūtā faila nosaukums būs `fails.gz`, un tā apjoms parasti būs mazāks, kā sākotnējam failam. Ievērojiet, ka fails `fails.gz` aizstās failu `fails` - tas nozīmē, ka `fails` vairs neeksistēs, lai arī ar `gzip` saspiegtā faila versija eksistēs. Ļoti labi saspiešanas rezultāti iegūstami teksta failiem, kamēr JPEG attēli, MP3 faili un tamlīdzīgi nebūs saspiežami pārāk labi, jo tie jau ir saspiesti. Šī `gzip` pamatlietoņuma rezultāts ir kompromiss starp saspiegtā faila apjomu un saspiešanas laiku. Maksimālo saspiešanu iespējams iegūt šādi:

```
% gzip -9 fails
```

Faila saspiešana notiks ilgāk, tomēr rezultātā iegūtais saspiestais fails būs tik mazs, cik `gzip` to spēj saspiest. Mazāku vērtību izmantošana parametrā ļaus saspiešanu veikt ātrāk, tomēr fails netiks tik labi saspiests.

Ar `gzip` saspiestu failu atspiešanu var veikt, izmantojot kādu no divām komandām, kas patiesībā nav viena un tā pati programma. `gzip` atspiedīs jebkuru failu ar atpazītu faila nosaukuma paplašinājumu. `gzip` atpazīst šādus paplašinājumus: `.gz`, `-gz`, `.z`, `-z`, `.Z` un `-Z`. Pirmā metode ir `gunzip` (1) izmantošana:

```
% gunzip fails.gz
```

Pēc šīs programmas izpildes tekošajā direktoriņā atradīsies faila fails atspiesta versija, pie tam paplašinājums `.gz` tiks noņemts. `gunzip` ir `gzip` pakotnes sastāvdaļa un tās izpilde ir identiska `gzip -d` izpildei. `gzip` bieži vien izrunā kā `gunzip`, jo šis nosaukums izklausās labāk :^)

### 15.2. bzip2

`bzip2` (1) ir *Slackware Linux* iekļautā alternatīvā failu saspiešanas komanda. Tā izmanto citu failu saspiešanas algoritmu, iegūstot dažas priekšrocības un dažus trūkumus. Svarīgākā `bzip2` priekšrocība ir saspiestā faila apjoms - `bzip2` gandrīz vienmēr saspiež failus labāk, kā `gzip`. Reizēm saspiesto failu apjomu atšķirība var būt dramatiska. Tas var ļoti noderēt tiem, kas izmanto lēnus modema pieslēgumus. Atcerieties, ka, lejupielādējot programmatūru no publiska FTP servera, parasti labais tonis prasa lejupielādēt `.bz2` failus, nevis `.gz` failus, jo tā tiks samazinātas serveri uzturošo labo cilvēku datu plūsmas izmaksas.

`bzip2` trūkums ir lielāka CPU noslodze. Tas nozīmē, ka faila saspiešana ar `bzip2` būs ilgāka un prasīs vairāk datora resursus, kā saspiešana ar `gzip`. Izvēloties saspiešanas programmu, jāsalīdzina faila saspiešanas ātrumu ar saspiestā faila apjomu un jāizvēlas - kas ir svarīgāk.

`bzip2` lietoņums ir gandrīz tāds pats, kā `gzip`, tādēļ to aplūkosim ļoti īsi. Līdzīgi kā `gunzip`, `bunzip2` ir identisks `bzip2 -d`. Galvenā atšķirība ir, ka `bzip2` izmanto faila nosaukuma paplašinājumu `.bz2`.

```
% bzip2 fails
```

```
% bunzip2 fails.bz2
```

```
% bzip2 -9 fails
```

### 15.3. tar

`tar` (1) ir GNU lenšu arhivators. Tas ņem vairākus failus vai direktorijas un izveido vienu lielu failu.

`tar` izmantošana ļauj jums saspiest veselu direktoriju koku, ko nav iespējams veikt, vienkārši izmantojot `gzip` vai `bzip2`. `tar` ir pieejami daudzi komandrindas parametri, kas izskaidroti `tar man` dokumentā. Šajā sadaļā aplūkosim tikai visbiežāk izmantotos `tar` lietojumus.

Visbiežāk sastopamais `tar` lietojums ir no tīmekļa vai FTP resursa lejupielādētas pakotnes atspiešana un atarhivēšana. Lielākā daļa pakotņu tiek izplatītas failos, kuru paplašinājums ir `.tar.gz`. Parasti to sauc par `tarball`. Tar nozīmē, ka vairāki faili ir arhivēti ar `tar` un iegūtais arhīvs ir saspiests ar `gzip`. To pašu nozīmē arī faili ar paplašinājumu `.tar.z`, tomēr šāds paplašinājums parasti sastopams vecākās *Unix* sistēmās.

Reizēm sastopami arī faili ar paplašinājumu `.tar.bz2`. Piemēram, šādi tiek izplatīts kodola pirmkods, jo tā tiek iegūts mazāks faila apjoms. Kā jau jūs varat iedomāties, šādā gadījumā vairāki faili ir arhivēti ar `tar` un saspiesti ar `bzip2`.

Pie šādi arhivētiem failiem iespējams piekļūt, izmantojot `tar` ar īpašiem komandrindas parametriem. Saspiesta `tar` faila atarhivēšanai jāizmanto parametru `-z`, kas nozīmē, ka fails vispirms tiek atspiests ar `gunzip`, un tikai tad atarhivēts. Visbiežāk sastopamā komandrinda saspiesta tar arhīva atspiešanai un atarhivēšanai izskatās šādi:

```
% tar -xvzf fails.tar.gz
```

Diezgan daudz parametru - ko tie visi nozīmē? `-x` nozīmē "atarhivēt" (*extract*). Šis parametrs ir svarīgs, jo norāda `tar`, ko tieši darīt ar arhīva failu. Šajā gadījumā mēs arhīvu sadalīsim failos, no kuriem tas tika izveidots. `-v` nozīmē izvadīt vairāk sistēmas paziņojumu (*verbose*). Šis parametrs liks `tar` programmai attēlot visus failus, kuri tiek atarhivēti. Ja jūs to nevēlaties, varat vienkārši izlaist šo parametru. No otras puses, iespējams izmantot parametru `-vv` (*very verbose* - ļoti daudz sistēmas paziņojumu), kas izvadīs vēl vairāk informācijas par katru atarhivējamo failu. Parametrs `-z` norāda, ka failu vispirms jāatspiež ar `gunzip`. Beidzot, parametrs `-f` norāda, ka komandrindā norādīts apstrādājamais fails.

Ir vairāki citi veidi, kā pierakstīt šo pašu komandu. Vecākās sistēmās, kurās nav sakarīgas GNU `tar` kopijas, to pieraksta šādi:

```
% gunzip fails.tar.gz | tar -xvf -
```

Šī komandrinda atspiedīs failu no iegūtos datus nosūtīs programmai `tar`, jo `gzip` var sava darba rezultātus rakstīt arī standarta izvades kanālā. `tar` parametrs `-` nozīmē, ka jāapstrādā standarta ievades kanālu - tiks atarhivēta no `gzip` saņemtā datu plūsma un iegūtie faili tiks ierakstīti diskā.

Vēl kāds veids, kā pierakstīt šo komandu, ir izlaist simbolu `-` pirms parametriem:

```
% tar xvzf fails.tar.gz
```

Ir sastopami arī ar `bzip2` saspiesti arhīvi. `tar` versija, kas iekļauta *Slackware Linux*, tos spēj apstrādāt tieši tāpat, kā ar `gzip` saspiestus arhīvus, vienīgi parametra `-z` vietā jāizmanto `-j`:

```
% tar -xvjf fails.tar.bz2
```

Ir svarīgi atcerēties, ka `tar` atarhivētos failus novietos tekošajā direktorijā. Līdz ar to, ja arhīvs atrodas direktorijā `/tmp`, bet jūs vēlaties to atspiest savā mājas direktorijā, ir pieejami vairāki veidi, kā to veikt. Arhīvu iespējams pārvietot jūsu mājas direktorijā un izpildīt `tar`. Ir iespējams arī komandrindā norādīt pilnu ceļu uz arhīvu. Visbeidzot, ir iespējams izmantot parametru `-c`, lai atarhivētu arhīvu norādītajā direktorijā.

```
% cd $HOME
% cp /tmp/filename.tar.gz .
% tar -xvzf filename.tar.gz

% cd $HOME
% tar -xvzf /tmp/filename.tar.gz
```

```
% cd /
% tar -xvzf /tmp/filename.tar.gz -C $HOME
```

Visas šīs komandas ir ekvivalentas. Visos gadījumos arhīvs tiek atarhivēts jūsu mājas direktorijā un sākotnējais arhīva fails tiek saglabāts.

Kāds gan labums no iespējas atspiest šos arhīvus, ja tos nevar pats izveidot? `tar` piedāvā arī šādu iespēju. Lielākajā daļā gadījumu arhīvu veidošana ir tik vienkārša, kā parametra `-x` aizstāšana ar `-c`.

```
% tar -cvzf fails.tar.gz .
```

Šajā komandrindā parametrs `-c` norāda `tar`, ka jāveido jaunu arhīvu, kamēr `-z` nosaka, ka iegūto arhīvu jāsaspiež ar `gzip`. `fails.tar.gz` ir arhīvs, kuru vēlaties izveidot.

Parametra `-f` norādīšana ne vienmēr ir nepieciešama, tomēr parasti ir laba prakse. Bez šī parametra norādīšanas `tar` informāciju ieraksta standarta izvades kanālā, kas parasti tiek izmantots, lai `tar` izvaddatus nodotu citai programmai:

```
% tar -cv fails.tar . | gpg --encrypt
```

Šī komanda izveido nespīestu tekošās direktorijas `tar` arhīvu un nodod to programmai `gpg`, kas to šifrē un saspiež, padarot neiespējamu to lasīt trešajai personai, kura nezina slepeno atslēgu.

## 15.4. zip

Ir arī divi rīki, ko var izmantot ZIP failu apstrādei. Šie faili ir ļoti populāri *Windows* vidē, tādēļ arī *Linux* sistēmā ir programmas, kas tos apstrādā. Saspiešanas programma ir `zip(1)`, bet atspiešanas - `unzip(1)`.

```
% zip foo *
```

Šī komanda izveidos failu `foo.zip`, kas saturēs visus tekošās direktorijas failus. `zip` pievienos nepieciešamo faila paplašinājumu automātiski, tādēļ to nav jānorāda faila nosaukumā. Ir iespējams arī rekursīvi apceļot tekošo direktoriju, ZIP failam pievienojot visas apakšdirektorijas:

```
% zip -r foo *
```

Failu atspiešana arī ir vienkārša.

```
% unzip foo.zip
```

Šī komanda atspiedīs visus failus, kas iekļauti `foo.zip` - arī visas direktorijas.

`zip` rīkiem ir vairākas papildiespējas, kas ļauj veidot pašatarhivējošus arhīvus, izlaist noteiktus failus, kontrolēt iegūtā arhīva apjomu, izdrukāt informāciju par plānoto procesu un daudzas citas. Lai uzzinātu, kā tās izmantot, aplūkojiet `zip` un `unzip man` dokumentus.

## 16.nodaļa. vi

**vi** (1) ir *Unix* sistēmu standarta tekstu redaktors un, lai arī mūsdienās tā pārzināšana nav tik svarīga kā kādreiz, tomēr joprojām ir ļoti noderīga. Ir pieejamas vairākas **vi** versijas (jeb kloni) - **vi**, **elvis**, **vile** un **vim**. Kāda no tām ir pieejama praktiski jebkurā *Unix* sistēmā - arī *Linux*. Visām versijām ir līdzīgs pamatkomandu komplekts, tādēļ vienas apgūšana ļaus viegli apgūt arī citas. Tā kā mūsdienu *Linux* un *Unix* sistēmās iekļauti daudz dažādi tekstu redaktori, daudzi izvēlas nelietot **vi**, tomēr tas joprojām ir visplašāk pieejamais tekstu redaktors *Unix* tipa sistēmās. **vi** pārzināšana nozīmē, ka jums nekad nenāksies strādāt ar *Unix* sistēmu un justies neērti, jo nebūs pieejams neviens jums zināms tekstu redaktors.

**vi** iekļautas plašas iespējas - sintakses izgaismošana, koda formatēšana, jaudīgs meklēšanas un aizstāšanas mehānisms, makrokomandas un daudzas citas. Šīs iespējas padara **vi** ļoti pievilcīgu programmētājiem, tīmekļa izstrādātājiem un citiem. Sistēmu administratori novērtēs automatizāciju un integrāciju ar čaulu.

*Slackware Linux* sistēmā noklusētā **vi** versija ir **elvis**. Ir pieejamas arī citas versijas - **vim** un **gvim**, ja vien ir instalētas attiecīgās pakotnes. **gvim** ir **vim** X logu sistēmas versija, kurā iekļautas rīkjoslas, izvēlnes un dialoglogi.

### 16.1. vi startēšana

**vi** var startēt komandrindā vairākos veidos. Visvienkāršākais veids ir ievadīt šādu komandu:

```
% vi
*/
void
TabReset(Tabs tabs)
{
    int i;

    for (i = 0; i < TAB_ARRAY_SIZE; ++i)
        tabs[i] = 0;

    for (i = 0; i < MAX_TABS; i += 8)
        TabSet(tabs, i);
}

tabs.c
if ((tmp = TypeCallocN(Char, length)) == 0)
    SysError(ERROR_SREALLOC);
*sbufaddr = tmp;

for (i = k = 0; i < nrow; i++) {
    k += BUF_HEAD;
    for (j = BUF_HEAD; j < old_max_ptrs; j++) {
        memcpy(tmp, base[k++], ncol);
        tmp += ncol;
    }
}

screen.c
```

Šī komanda startēs **vi** ar tukšu buferi. Šajā brīdī jūs redzēsiet gandrīz tukšu ekrānu. **vi** darbojas komandu režīmā, gaidot, kad tiks veikta kāda darbība. **vi** darba režīmi ir aplūkoti sadaļā 16.2. Lai aizvērtu **vi**, ievadiet komandu

```
:q
```



Pieņemot, ka failā nav veiktas nekādas izmaiņas, šī komanda aizvērs **vi**. Ja ir veiktas izmaiņas, **vi** brīdinās par tām un ieteiks, kā tās ignorēt. Izmaiņu ignorēšanu parasti veic, **q** pievienojot izsaukuma zīmi:

```
:q!
```

Izsaukuma zīme parasti ļauj veikt norādīto darbību *piespiedu kārtā*. Vēlāk aplūkosim šo un citas komandu kombinācijas.

**vi** iespējams startēt arī ar jau eksistējošu failu. Piemēram, failu `/etc/resolv.conf` atver šādi:

```
% vi /etc/resolv.conf
```

Beidzot, **vi** var startēt, norādot atveramā faila tekošo rindiņu. Tas ir īpaši ērti programmētājiem, kad kompilators norāda programmas teksta kļūdaino rindiņu. Piemēram, iespējams startēt **vi**, atverot failu `/usr/src/linux/init/main.c` ar tekošo rindiņu **47**:

```
% vi +47 /usr/src/linux/init/main.c
```

**vi** attēlos norādīto failu un novietos kursoru norādītajā rindiņā. Gadījumā, ja norādīta rindiņa, kuras numurs ir lielāks kā rindiņu skaits failā, kurors tiks novietots pēdējā rindiņā. Šī iespēja noderēs programmētājiem, ļaujot atvērt failu rindiņā, kurā notikusi kļūda bez nepieciešamības to meklēt.

## 16.2. Režīmi

**vi** darbojas dažādos režīmos, kurus izmanto, lai veiktu dažādus uzdevumus. Kad **vi** tiek startēts pirmo reizi, jūs atrodaties komandu režīmā, kurā iespējams ievadīt dažādas komandas darbam ar tekstu, pārvietoties pa failu, saglabāt to, aizvērt **vi** un mainīt režīmus. Tekstu labošana tiek veikta ievietošanas režīmā (*insert mode*). Starp režīmiem ir iespējams ātri pārslēgties, izmantojot dažādas komandas.

### Komandu režīms

Atverot **vi**, jūs atrodaties komandu režīmā. Šajā režīmā nav iespējams tieši ievadīt vai labot tekstu, tomēr ir iespējams veikt dažādas darbības ar to, meklēt, aizvērt, saglabāt, ielādēt u.c. Šis apraksts ir tikai ievads komandu režīmā. Dažādās **vi** komandas aplūkotas sadaļā 16.7.

Šķiet, visbiežāk izmantotā komandu režīma komanda ir pārslēgšanās uz ievietošanas režīmu. To veic, nospiežot tastatūras taustiņu **i**. Kursors maina izskatu un ekrāna apakšā tiek attēlots teksts `-- INSERT --` (tas gan nenotiek visos **vi** klonos). Šajā režīmā visa ievadītā informācija tiek ievietota tekošajā buferī un tiek attēlota uz ekrāna. Lai pārslēgtos uz komandu režīmu, nospiediet taustiņu **Esc**.

Komandu režīms ļauj arī pārvietoties pa failu. Dažās sistēmās iespējams izmantot kursora vadības taustiņus, bet citās jāizmanto tradicionālie taustiņi - **hjk1**. Aplūkosim, kā šos taustiņus izmanto, lai pārvietotos pa tekstu:

- h** pārvietoties vienu simbolu pa kreisi
- j** pārvietoties vienu simbolu uz leju
- k** pārvietoties vienu simbolu uz augšu
- l** pārvietoties vienu simbolu pa labi

Lai pārvietotos, vienkārši nospiediet atbilstošo taustiņu. Kā redzēsiet vēlāk, šos taustiņus iespējams kombinēt ar skaitļiem, lai pārvietotos pa tekstu daudz efektīvāk.

Daudzas komandu režīma komandas sākas ar kolu. Piemēram, **:q**, kuru aplūkojām jau iepriekš. Kols norāda, ka sekos komanda, kamēr **q** norāda **vi**, ka jābeidz darbu. Citas komandas sastāv no skaitļa un burta. Tās nesākas ar kolu un parasti tiek izmantotas manipulācijām ar tekstu.

Piemēram, vienas rindiņas dzēšana no teksta tiek veikta, ievadot **dd**. Pēc šīs komandas ievades tiks

dzēsta rindiņa, kurā atrodas kursora. Komandas **4dd** ievade norādīs **vi**, ka jādzēš rindiņu, kurā atrodas kursora, un vēl trīs rindiņas pēc tam. Vispār skaitlis nosaka, cik reizes **vi** jāizpilda komandu.

Jūs varat arī kombinēt skaitli ar kursora pārvietošanas taustiņiem, lai pārvietotos uzreiz vairākus simbolus noteiktajā virzienā. Piemēram, **10k** pārvietos kursoru desmit rindiņas uz augšu.

Komandu režīmu iespējams arī izmantot, lai izgrieztu un ievietotu tekstu, kā arī tekošajā buferī ielādētu citus failus. Teksta kopēšanu veic ar **y** komandu (*yank* - raut). Tekošās rindiņas kopēšanu veic, ievadot **yy**. Norādot skaitli, iespējams kopēt vairākas rindiņas uzreiz. Pēc tam pārvietojieties uz rindiņu, kurā kopēt tekstu un nospiediet taustiņu **p**. Teksts tiek ievietots aiz tekošās rindiņas.

Teksta izgriešanu veic, izmantojot komandu **dd**, pēc kā tā ievietošanai izmanto komandu **p**. Teksta nolasišana no cita faila ir vienkārša - ievadiet **:r**, atstarpes simbolu un ievietojamā faila nosaukumu. Faila saturs tiks ievietots tekošajā buferī aiz rindiņas, kas atrodas pēc kursora. Jaudīgāki **vi** kloni pat atbalsta faila nosaukuma aizpildi - līdzīgi, kā čaula.

Pēdējā iespēja, ko aplūkosim, ir meklēšana. Komandu režīms ļauj veikt gan vienkāršu teksta meklēšanu, gan izpildīt sarežģītas meklēšanas un aizstāšanas komandas, kas izmanto regulārās izteiksmes. Regulāro izteiksmju aplūkošana ir ārpus šīs nodaļas ietvariem, tādēļ aplūkosim tikai vienkāršākās meklēšanas iespējas.

Vienkārša meklēšana tiek veikta, nospiežot taustiņu **/**, pēc tā norādot meklējamo tekstu. **vi** meklēs uz priekšu no kursora līdz faila beigām, apstājoties pie pirmā atrastā teksta fragmenta, kas atbilst meklētajam. Atcerieties, ka **vi** apstāsies arī pie neprecīziem fragmentiem - piemēram, meklējot tekstu **the**, **vi** apstāsies pie vārdiem, **then**, **therefore** u.c. Tas tādēļ, ka visi šie teksta fragmenti atbilst meklētajam tekstam - **the**.

Kad **vi** atradis pirmo atbilstošo fragmentu, jūs varat turpināt meklēšanu, vienkārši ievadot komandu **/** un nospiežot taustiņu **Enter**. Jūs varat arī meklēt tekstu pretējā virzienā, **/** vietā ievadot **?**. Piemēram, **the** meklēšana pretējā virzienā tiek veikta ar komandu **?the**.

## Ievietošanas režīms

Teksta ievadi un aizstāšanu veic ievietošanas režīmā. Kā jau iepriekš aplūkojām, ievietošanas režīmu iespējams ieslēgt, komandu režīmā ievadot komandu **i**. Pēc tā viss teksts, ko jūs ievadīsiet, tiks ievadīts tekošajā buferī. **Esc** taustiņa nospiešana ļaus jums atgriezties komandu režīmā.

Teksta aizvietošanu iespējams veikt dažādi. Komandu režīma komanda **r** ļaus aizstāt vienu simbolu, kas atrodas kursora vietā. Ievadiet jaunu simbolu un tas aizstās iepriekšējo, pēc kā atkal tiks ieslēgts komandu režīms. Komandas **R** ievade ļaus jums aizstāt tik simbolus, cik vēlaties. Lai pārslēgtos atpakaļ uz komandu režīmu, nospiediet taustiņu **Esc**.

Ir vēl kāds veids, kā pārslēgties starp ievietošanu un aizstāšanu. **Insert** taustiņa nospiešana komandu režīmā ieslēgs ievietošanas režīmu. Strādājot ievietošanas režīmā, **Insert** taustiņš pārslēdz ievietošanas un aizstāšanas režīmus.

## 16.3. Failu atvēršana

**vi** ļauj atvērt failus gan komandu režīmā, gan norādot atveramo failu kā komandrindas parametru. Lai atvērtu failu **/etc/lilo.conf**, ievadiet

```
:e /etc/lilo.conf
```

Ja veiktas izmaiņas tekošajā buferī bez saglabāšanas, **vi** protestēs. Jūs tomēr varat atvērt failu bez tekošā bufera saglabāšanas, ievadot **:e!**, atstarpi un faila nosaukumu. Kopumā **vi** brīdinājumus iespējams noslēpt, ja komandu papildina ar izsaukuma zīmi.

Ja jūs vēlaties atkārtoti atvērt tekošo failu, to iespējams veikt, vienkārši ievadot **e!**. Tas noder, ja jūs nejauši esat sabojājis failu un vēlaties to atvērt atkārtoti.

Daži `vi` kloni (piemēram, `vim`) ļauj vienlaikus atvērt vairākus buferus. Piemēram, lai atvērtu failu `09-vi.sgml`, kas atrodas manā mājas direktoriņā, vienlaikus neaizverot jau atvērtu failu, es varu ievadīt komandu:

```
:split ~/09-vi.sgml
```

Jaunais fails tiek atvērts ekrāna augšējā pusē, kamēr vecais tiek attēlots ekrāna apakšdaļā. Eksistē daudzas komandas darbam ar vairākiem buferiem, un daudzas no tām sāk atgādināt `Emacs` redaktoru. Labākais papildinformācijas avots ir jūsu izvēlētajā `vi` klona `man` dokuments. Ievērojiet, ka daudzi kloni neatbalsta vairāku buferu režīmu, tādēļ tajos šīs komandas nav iespējams izmantot.

## 16.4. Failu saglabāšana

Ir vairāki veidi, kā `vi` redaktorā saglabāt failus. Ja jūs vēlaties saglabāt tekošo buferi failā `randomness`, ievadiet

```
:w randomness
```

Kad fails saglabāts, tā atkārtota saglabāšana tiek veikta ar komandu `:w`, kas iepriekš norādītajā failā saglabās visas izmaiņas. Kad fails saglabāts, tiek ieslēgts komandu režīms. Ja jūs vēlaties saglabāt failu un aizvērt `vi` (ļoti bieži veikta darbība), ievadiet komandu `:wq`. Šī komanda norāda `vi`, ka tekošo failu jāsavāc un jāatgriežas čaulā.

Reizēm jūs varētu vēlēties saglabāt failu, kas atzīmēts kā tikai lasāms. To iespējams veikt, komandu `w` papildinot ar izsaukuma zīmi:

```
:w!
```

Reizēm tomēr var gadīties situācijas, kurās failu saglabāt nav iespējams (piemēram, mēģinot labot failu, kurš pieder citam lietotājam). Šādās situācijās `vi` paziņos, ka failu nav iespējams saglabāt. Ja jūs tiešām vēlaties labot šo failu, jums nāksies to darīt ar `root` vai faila īpašnieka (vēlams) tiesībām.

## 16.5. vi aizvēršana

Viens no `vi` aizvēršanas veidiem ir, izmantojot komandu `:wq`, kas pirms aizvēršanas saglabās tekošo buferi. `vi` iespējams arī aizvērt, nesaglabājot buferi, ar komandu `:q` vai (biežāk) `:q!`. Pēdējo izmanto, ja esat modificējis failu, bet nevēlaties saglabāt izmaiņas.

Reizēm jūsu dators vai `vi` var avarēt, tomēr gan `elvis`, gan `vim` veic visu iespējamo, lai samazinātu bojājumus visiem atvērtajiem buferiem - abi redaktori atvērtos buferus regulāri saglabā pagaidu failā. Šī faila nosaukums parasti ir tāds pats, kā atvērtajam failam, tomēr sākas ar punktu, kas padara šo failu neredzamu.

Šis pagaidu fails tiek dzēsts, ja redaktors tiek korekti aizvērts - tas nozīmē, ka avārijas gadījumā šī pagaidu kopija tomēr būs pieejama. Ja jūs atkārtoti atverat failu, jums tiks piedāvātas iespējamās datu atjaunošanas darbības. Lielākajā daļā gadījumu lielāko daļu jūsu datu ir iespējams atjaunot. `elvis` jums arī nosūtīs e-pasta ziņojumu informējot, ka eksistē faila rezerves kopija.

## 16.6. vi konfigurēšana

Jūsu izvēlēto `vi` klonu ir iespējams dažādi konfigurēt.

Komandu režīmā ir iespējams ievadīt dažādas komandas, lai pielāgotu `vi` izskatu jūsu gaumei. Atkarībā no izvēlētajā klona ir iespējams ieslēgt režīmus, kas atvieglo programmēšanu (sintakses izgaismošana, automātiskās atkāpes u.c.), iestatīt makrokomandas automake uzdevumiem, ieslēgt teksta aizstāšanu u.c.

Gandrīz visas šīs komandas ir iespējams ierakstīt konfigurācijas failā jūsu mājas direktoriņā. `elvis` gadījumā šī faila nosaukums ir `.exrc`, kamēr `vim` izmanto failu `.vimrc`. Konfigurācijas failā

iespējams iekļaut lielāko daļu komandu režīmā izpildāmo iestatījumu komandu - iestatījumu informāciju, teksta aizstāšanu, makrokomandas u.c.

Visu iestatījumu un redaktoru savstarpējo atšķirību aplūkošana ir samērā apjomīgs uzdevums, tādēļ iesakām aplūkot jūsu izvēlēto `vi` redaktora `man` dokumentu vai tīmekļa vietni. Dažiem redaktoriem (kā `vim`) pašā redaktorā ir pieejama laba palīdzības sistēma, kurai var piekļūt ar komandu `:help` vai līdzīgu komandu. Iesakām arī iepazīties ar O'Reilly izdoto Lamb un Robbins grāmatu *Learning the vi Editor*.

Daudzas *Linux* programmas, piemēram, `crontab` rediģēšanas režīms, teksta failus atvērš tieši `vi` redaktorā. Ja jums nepatīk `vi` un vēlaties, lai tā vietā tiktu atvērts cits tekstu redaktors, viss, ko jums jāveic, ir vides mainīgā `VISUAL` iestatīšana, lai tas saturētu informāciju par noklusēto tekstu redaktoru. Informāciju par vides mainīgo iestatīšanu iespējams atrast 8.nodaļas sadaļā "Vides mainīgie". Ja jūs vēlaties, lai jūsu izvēlētais redaktors būtu noklusētais vienmēr, pievienojiet `VISUAL` vides mainīgo savam `.bash_profile` vai `.bashrc` failam.

## 16.7. vi komandas

Šī sadaļa ir visbiežāk lietoto `vi` komandu apskats. Dažas no šīm komandām jau aplūkojām, bet citas būs jaunas.

### Kursora pārvietošana

| Darbība                               | Komanda              |
|---------------------------------------|----------------------|
| Pa kreisi, pa labi, uz augšu, uz leju | <code>h,j,k,l</code> |
| Uz rindiņas beigām                    | <code>\$</code>      |
| Uz rindiņas sākumu                    | <code>^</code>       |
| Uz faila beigām                       | <code>G</code>       |
| Uz faila sākumu                       | <code>:1</code>      |
| Uz 47.rindiņu                         | <code>:47</code>     |

### Labošana

| Darbība                                             | Komanda                          |
|-----------------------------------------------------|----------------------------------|
| Rindiņas dzēšana                                    | <code>dd</code>                  |
| Piecu rindiņu dzēšana                               | <code>5dd</code>                 |
| Simbola aizstāšana                                  | <code>r</code>                   |
| Simbola dzēšana                                     | <code>x</code>                   |
| Desmit simbolu dzēšana                              | <code>10x</code>                 |
| Iepriekšējās darbības atcelšana                     | <code>u</code>                   |
| Apvienot tekošo ar nākamo rindiņu                   | <code>J</code>                   |
| Aizstāt <code>vecais</code> ar <code>jaunais</code> | <code>%s'vecais'jaunais'g</code> |

| Darbība | Komanda |
|---------|---------|
|         |         |

### Meklēšana

| Darbība                                   | Komanda            |
|-------------------------------------------|--------------------|
| Meklēt <code>asdf</code>                  | <code>/asdf</code> |
| Meklēt <code>asdf</code> pretējā virzienā | <code>?asdf</code> |
| Atkārtot iepriekšējo meklējumu            | <code>/</code>     |
| Atkārtot meklējumu pretējā virzienā       | <code>?</code>     |
| Atkārtot meklējumu tajā pašā virzienā     | <code>n</code>     |
| Atkārtot meklējumu otrā virzienā          | <code>N</code>     |

### Saglabāšana un aizvēršana

| Darbība                                   | Komanda               |
|-------------------------------------------|-----------------------|
| Aizvērt                                   | <code>:q</code>       |
| Aizvērt bez saglabāšanas                  | <code>:q!</code>      |
| Saglabāt un aizvērt                       | <code>:wq</code>      |
| Saglabāt                                  | <code>:w</code>       |
| Atkārtoti ielādēt tekošo failu            | <code>:e!</code>      |
| Saglabāt buferi failā <code>asdf</code>   | <code>:w asdf</code>  |
| Atvērt failu <code>hejaz</code>           | <code>:e hejaz</code> |
| Ielādēt failu <code>asdf</code> buferī    | <code>:r asdf</code>  |
| Ielasīt <code>ls</code> izvaddatus buferī | <code>:r !ls</code>   |

## 17.nodaļa. Emacs

Ja `vi` (ar kloniem), bez šaubām, ir visplašāk izmantotais tekstu redaktors *Unix* tipa sistēmās, tad **Emacs** noteikti ir otrajā vietā. Atšķirībā no `vi`, kas izmanto dažādus darba "režīmus", **Emacs** komandu ievadei izmanto **Control** un **Alt** taustiņu kombinācijas, līdzīgi, kā tekstu procesoros un, patiesībā, ļoti daudzās programmās. (Jāatzīst, ka šīs taustiņu kombinācijas reti atbilst citu programmu kombinācijām - piemēram, ja daudzās mūsdienu programmās tiek izmantotas **Ctrl-C/X/V** kombinācijas teksta kopēšanai, izgriešanai un ielīmēšanai, tad **Emacs** izmanto citas kombinācijas un patiesībā arī citu mehānismu.)

Arī, atšķirībā no `vi`, kas ir (lielisks) tekstu redaktors un nekas vairāk, **Emacs** ir programma ar gandrīz bezgalīgām iespējām. **Emacs** lielākā daļa ir rakstīta valodā *Lisp*, kas ir ļoti spēcīga programmēšanas valoda ar īpašību, ka jebkura tajā sarakstīta programma pati par sevi ir *Lisp* kompilators. Tas nozīmē, ka lietotājs var paplašināt **Emacs**, un patiesībā arī izveidot pilnīgi jaunas programmas "**Emacs** vidē".

Līdz ar to **Emacs** vairs nav vienkārši redaktors. **Emacs** ir pieejamas daudzas paplašinājumu pakotnes (daudzas ar brīvi pieejamu pirmkodu), kas nodrošina ļoti dažādu funkcionalitāti. Daudzas no tām ir saistītas ar tekstu rediģēšanu, kas, galu galā, ir **Emacs** galvenais uzdevums, tomēr ar to viss nebeidzas. Eksistē, piemēram, izklājlapu programmas, datu bāzes, spēles, e-pasta un vēstkopu klienti (vispopulārākais ir *Gnus*) u.c.

Ir divas galvenās **Emacs** versijas: *GNU Emacs* (versija, kas iekļauta arī *Slackware*) un *XEmacs*. Pēdējā nav **Emacs** versija X logu videi - gan **Emacs**, gan *XEmacs* darbojas gan konsolē, gan X logu sistēmā. *XEmacs* savulaik tika uzsākts kā projekts ar mērķi padarīt tīrāku **Emacs** kodu. Šobrīd abas versijas tiek aktīvi izstrādātas un īstenībā starp abām izstrādes komandām ir labi sakari. Šajā nodaļā nav svarīgi, vai jūs izmantojat **Emacs** vai *XEmacs* - atšķirības starp tiem nav gala lietotājam sevišķi pamanāmas.

### 17.1. Emacs startēšana

**Emacs** var startēt, čaulā ievadot komandu `emacs`. Strādājot X logu vidē, **Emacs** parasti atvērsies savā X logā ar izvēlni, kurā iespējams atrast svarīgākās funkcijas. Startējoties **Emacs** vispirms parādīs sākumpaziņojumu un pēc dažām sekundēm atvērs buferi `*scratch*` (sk. sadaļu 17.2).

```

File Edit Options Buffers Tools Help
Welcome to GNU Emacs, one component of the GNU/Linux operating system.

Get help          C-h (Hold down CTRL and press h)
Emacs manual      C-h r
Emacs tutorial    C-h t          Undo changes      C-x u
Buy manuals       C-h C-m        Exit Emacs        C-x C-c
Browse manuals    C-h i
Activate menubar  F10 or ESC ` or M-`
(`C-' means use the CTRL key. `M-' means use the Meta (or Alt) key.
If you have no Meta key, you may instead type ESC followed by the character.)

GNU Emacs 23.0.0.2 (i686-pc-linux-gnu, X toolkit, Xaw3d scroll bars)
of 2005-03-25 on slackware
Copyright (C) 2004 Free Software Foundation, Inc.

GNU Emacs comes with ABSOLUTELY NO WARRANTY; type C-h C-w for full details.
Emacs is Free Software--Free as in Freedom--so you can redistribute copies
of Emacs and modify it; type C-h C-c to see the conditions.
Type C-h C-d for information on getting the latest version.

If an Emacs session crashed recently, type M-x recover-session RET
to recover the files you were editing.

----- GNU Emacs -----
For information about the GNU Project and its goals, type C-h C-p.

```

**Emacs** var atvērt ar jau eksistējošu failu, ievadot komandu

```
% emacs /etc/resolv.conf
```

Šīs komandas izpilde atvērs norādīto failu, neparādot sākumpaziņojumu.

## Komandas

Kā jau minēju iepriekš, **Emacs** komandām izmanto **Control** un **Alt** kombinācijas. Parasti ir pieņemts tās pierakstīt kā **C-burts** un **M-burts**. Piemēram, **C-x** nozīmē **Control+x** un **M-x** nozīmē **Alt+x**. (Burtu **M** izmanto tādēļ, ka sākotnēji **Alt** taustiņu sauca par **Meta** taustiņu. **Meta** taustiņš mūsdienās vairs netiek izmantots datoros, tādēļ tā nozīmi **Emacs** vidē pilda taustiņš **Alt**.)

Daudzas **Emacs** komandas sastāv no taustiņu un to kombināciju virknēm. Piemēram, **C-x C-c** (**Control+X** un tad **Control+C**) aizver **Emacs**, bet **C-x C-s** saglabā tekošo failu. Atcerieties, ka **C-x C-b** nav tas pats, kas **C-x b**. Pirmajā virknē vispirms jānospiež taustiņu kombināciju **Control+x** un tad **Control+b**, bet otrajā - **Control+x** un tad vienkārši **b**.

## 17.2. Buferi

**Emacs** ļoti svarīgs elements ir buferis. Katrs atvērtais fails tiek ielādēts savā buferī. Vēl jo vairāk - **Emacs** ir arī vairāki īpaši buferi, kuros netiek ielādēti faili, bet kurus izmanto citām vajadzībām. Šādiem īpašajiem buferiem parasti ir nosaukums, kas sākas un beidzas ar simbolu **\*** (zvaigznīti). Piemēram, buferis, ko **Emacs** parāda pēc startēšanas, ir **\*scratch\*** buferis. Šajā buferī ir iespējams normāli ievadīt tekstu, tomēr ievadītais teksts, aizverot **Emacs**, netiek saglabāts.

Ir vēl viens īpašais buferis, par kuru jums neko nav jāzina - tas ir *minibuferis*. Šis buferis satur tikai vienu rindiņu, kura vienmēr redzama uz ekrāna - tā ir pēdējā **Emacs** loga rindiņa, kas atrodas zem tekošā bufera statusjoslas. Minibuferis ir vieta, kur **Emacs** rāda lietotājam paredzētus paziņojumus. Tajā tiek izpildītas arī komandas, kurām nepieciešama lietotāja ievadīta informācija. Piemēram, atverot failu, **Emacs** tā nosaukumu pieprasīs ievadīt tieši minibuferī.

Pārslēgšanās no viena bufera uz citu tiek veikta ar komandu **C-x b**. Šī komanda pieprasīs jums ievadīt bufera nosaukumu (parasti tas sakrīt ar rediģējamā faila nosaukumu), piedāvājot arī noklusēto izvēli - buferi, kurā jūs atradāties pirms pārslēgšanās. Taustiņa **Enter** nospiešana pārslēgs jūs uz šo buferi.

Ja jūs vēlaties pārslēgties uz citu buferi, ievadiet tā nosaukumu. Ievērojiet, ka ir iespējams izmantot rindiņu aizpildi ar **Tab** taustiņu - ievadiet pirmos dažus bufera nosaukuma simbolus un nospiediet taustiņu **Tab** - **Emacs** aizpildīs bufera nosaukumu. Aizpilde ar **Tab Emacs** vidē darbojas visur, kur vien no tās ir kāda jēga.

Atvērto buferu sarakstu iespējams aplūkot, ievadot komandu **C-x C-b**. Šī komanda parasti sadalīs ekrānu divās daļās, augšējā daļā attēlojot buferi, ar kuru jūs strādājat, bet apakšējā - jaunu buferi ar nosaukumu **\*Buffer List\***. Šajā buferī redzams visu buferu saraksts, to apjomi un režīmi, kā arī faili (ja tādi ir), kurus šie buferi *apciemo* (tā to sauc **Emacs** vidē). No šī ekrāna sadalījuma iespējams atbrīvoties, ievadot **C-x l**.

Piezīme: X logu vidē buferu saraksts ir pieejams arī izvēlnē **Buffer**.

### 17.3. Režīmi

Katram **Emacs** buferim ir piesaistīts *režīms*. Šie režīmi būtiski atšķiras no **vi** režīmiem - tie nosaka atvērtā bufera veidu. Piemēram, parasti teksta faili tiek atvērti teksta režīma buferī, tomēr eksistē arī tādi režīmi, kā **c** režīms **C** programmu labošanai, **sh** režīms čaulas skriptu labošanai, **latex** režīms **LATEX** failu labošanai, e-pasta režīms e-pasta un vēstkopu ziņojumu labošanai u.c. Režīms nodrošina īpašus pielāgojumus un funkcionalitāti, kas ir noderīga jūsu labotā faila tipam. Katram režīmam pat ir iespējams pārsaukt komandas un to taustiņu kombinācijas. Piemēram, teksta režīmā **Tab** taustiņš vienkārši tiek pārvietots uz nākamo tabulācijas pozīciju, tomēr daudzos programmēšanas valodu režīmos **Tab** taustiņš izveido atkāpi atkarībā no rindiņas bloka dziļuma.

Iepriekš minētie režīmi tiek saukti par galvenajiem režīmiem. Katram buferim ir tieši viens galvenais režīms. Papildus tam, buferim var būt viens vai vairāki papildrežīmi. Papildrežīms nodrošina papildiespējas, kas var būt noderīgas noteiktiem labošanas uzdevumiem. Piemēram, nospiežot **Insert** taustiņu, tiek ieslēgts aizstāšanas režīms. Eksistē arī automātiskās izlīdzināšanas režīms, kas ir noderīgs, to lietojot teksta vai **latex** režīmā - tas nodrošina, ka jūsu rakstītā teksta katra rindiņa tiek automātiski aplauzta, sasniedzot noteiktu simbolu skaitu. Bez šī režīma rindkopas izlīdzināšanai jāizmanto komandu **M-q**. (To var izmantot arī rindkopas pārformatēšanai, ja veikti labojumi tekstā un tas vairs nav skaisti izlīdzināts.)

### Failu atvēršana

Lai **Emacs** redaktorā atvērtu failu, ievadiet

**C-x C-f**

**Emacs** pieprasīs jums ievadīt faila nosaukumu, aizpildot noklusēto ceļu (parasti **~/**). Pēc faila nosaukuma ievades (tajā iespējams izmantot rindiņu aizpildi ar **Tab**) nospiediet taustiņu **Enter**. **Emacs** atvērs norādīto failu jaunā buferī un to attēlos.

**Piezīme: Emacs automātiski izveidos jaunu buferi - tas neielādēs failu tekošajā buferī.**

Lai izveidotu jaunu failu, jūs nevarat vienkārši sākt rakstīt. Vispirms jāizveido jaunu buferi un jānorāda failu, kurā tas tiks glabāts. To veic, ievadot **C-x C-f** un ierakstot faila nosaukumu - līdzīgi, kā atverot jau eksistējošu failu. **Emacs** konstatēs, ka šāds fails neeksistē, tādēļ izveidos jaunu buferi un minibuferī izvadīs tekstu (**New file**).

Ievadot **C-x C-f** un faila nosaukuma vietā ievadot direktorijas nosaukumu, **Emacs** izveidos jaunu buferi, kurš saturēs attiecīgās direktorijas visu failu sarakstu. Jūs varat pārvietot kursoru uz failu, kuru meklējat, un ievadīt **,**. **Emacs** atvērs izvēlēto failu. (Patiesībā šeit iespējams veikt daudz citas darbības



- kā failu dzēšanu, pārsaukšanu, pārvietošanu u.c. **Emacs** šajā brīdī darbojas **direc** (direktoriju redaktora - *Directory Editor*) režīmā, kas būtībā ir vienkāršs failu pārvaldnieks.)

Ievadot **C-x C-f** un pēkšņi pārdomājot, varat ievadīt komandu **C-g**, lai atceltu darbību. **C-g** darbojas gandrīz visur, kur vēlaties atcelt darbību vai komandu, kura ir iesākta, bet kuru nevēlaties pabeigt.

## 17.4. Rediģēšanas pamati

Kad esat atvēris failu, jūs, protams, varat pārvietoties pa to. Kursora vadības taustiņi un **PgUp/PgDn** dara tieši to, ko jūs varētu sagaidīt. **Home** un **End** pārvieto kursoru uz rindiņas sākumu un beigām. (Vecākās versijās šie taustiņi pārvieto kursoru uz bufera sākumu un beigām.) Tiesa, eksistē arī **Control** un **Meta (Alt)** taustiņu kombinācijas, ko arī var izmantot kursora pārvietošanai. Tā kā šo kombināciju izmantošanai nav nepieciešams pārvietot rokas uz citu tastatūras daļu, to izmantošana ļauj ietaupīt laiku.

Ievērojiet, ka **Meta** komandas darbojas paralēli **Control** komandām, izņemot to, ka tās darbojas ar lielākām teksta vienībām - ja **C-f** pārvieto kursoru vienu simbolu pa labi, tad **M-f** pārvieto kursoru vienu vārdu pa labi u.c.

Ievērojiet arī, ka **M-<** un **M->** ievadei patiesībā jāievada **Shift+Alt+**, un **Shift+Alt+**, jo **<** un **>** ir **Shift+**, un **Shift+**. (protams, ja vien jūsu tastatūras izklājums neatšķiras no standarta ASV izklājuma).

**C-k** dzēš (*nogalina*, kā pieņemts teikt) visu tekstu no kursora līdz rindiņas beigām, tomēr neizdzēš pašu rindiņu (t.i., neizdzēš rindiņas pārnese simbolu). Rindiņa tiek dzēsta tikai tad, ja aiz kursora nav teksta. Citiem vārdiem, lai dzēstu rindiņu, jums jānovieto kursoru rindiņas sākumā un jāievada **C-k** divreiz - vienreiz, lai dzēstu rindiņas tekstu, un otrreiz - lai dzēstu pašu rindiņu.

## Svarīgākās Emacs rediģēšanas komandas

| Komanda    | Rezultāts                             |
|------------|---------------------------------------|
| <b>C-b</b> | Pārvietoties vienu simbolu pa kreisi  |
| <b>C-f</b> | Pārvietoties vienu simbolu pa labi    |
| <b>C-n</b> | Pārvietoties vienu rindiņu uz leju    |
| <b>C-p</b> | Pārvietoties vienu rindiņu uz augšu   |
| <b>C-a</b> | Pārvietoties uz rindiņas sākumu       |
| <b>C-e</b> | Pārvietoties uz rindiņas beigām       |
| <b>M-b</b> | Pārvietoties vienu vārdu pa kreisi    |
| <b>M-f</b> | Pārvietoties vienu vārdu pa labi      |
| <b>M-}</b> | Pārvietoties vienu rindkopu uz leju   |
| <b>M-{</b> | Pārvietoties vienu rindkopu uz augšu  |
| <b>M-a</b> | Pārvietoties vienu teikumu atpakaļ    |
| <b>M-e</b> | Pārvietoties vienu teikumu uz priekšu |

| Komanda | Rezultāts                                                                                                  |
|---------|------------------------------------------------------------------------------------------------------------|
| C-d     | Dzēst simbolu zem kursora                                                                                  |
| M-d     | Dzēst tekstu līdz vārda beigām                                                                             |
| C-v     | Pārvietoties vienu ekrānu uz leju (PgDn)                                                                   |
| M-v     | Pārvietoties vienu ekrānu uz augšu (PgUp)                                                                  |
| M-<     | Pārvietoties uz bufera sākumu                                                                              |
| M->     | Pārvietoties uz bufera beigām                                                                              |
| C-_     | Atcelt iepriekšējo izmaiņu (atkārtojama darbība). Ievērojiet, ka patiesībā jāievada <b>Shift+Control+-</b> |
| C-k     | Izdzēst tekstu līdz rindiņas beigām                                                                        |
| C-s     | Meklēt uz priekšu                                                                                          |
| C-r     | Meklēt atpakaļ                                                                                             |

## 17.5. Failu saglabāšana

Lai saglabātu failu, ievadiet

```
C-x C-s
```

**Emacs** nepieprasīs ievadīt faila nosaukumu - buferis tiks saglabāts tajā pat failā, no kurienes tas tika ielādēts. Ja jūs vēlaties saglabāt tekstu citā failā, ievadiet

```
C-x C-w
```

Saglabājot failu pirmoreiz, **Emacs** parasti faila iepriekšējo versiju saglabās rezerves kopijas failā, kam nosaukums būs līdzīgs, kā oriģinālajam, ar tildes simbolu beigās - labojot failu `cars.txt`, **Emacs** tam izveidos rezerves kopiju ar nosaukumu `cars.txt~`.

Šī kopija ir faila saturs atvēršanas brīdī. Kamēr jūs strādāsiet ar failu, **Emacs** regulāri veiks automātisku datu saglabāšanu failā, kura nosaukuma sākums un beigas papildināti ar simboliem # - `#cars.txt#`. Šī kopija tiek dzēsta, saglabājot failu ar komandu `C-x C-s`.

Kad faila labošana pabeigta, to saturošo buferi iespējams *nogalināt*, ievadot komandu

```
C-x k
```

**Emacs** pieprasīs ievadīt nogalināmā bufera nosaukumu, kā noklusēto piedāvājot tekošo buferi, kuru iespējams izvēlēties, nospiežot taustiņu **Enter**. Ja vēl neesat saglabājis failu, **Emacs** pārjautās, vai tiešām vēlaties nogalināt šo buferi.

## 17.6. Emacs aizvēršana

Kad darbs ar **Emacs** pabeigts, ievadiet

```
C-x C-c
```

Šī komanda aizver **Emacs**. Ja ir atvērti kādi nesaglabāti faili, **Emacs** to paziņos un piedāvās katru no tiem saglabāt. Ja uz kādu no piedāvājumiem atbildēsiet ar negatīvu atbildi, **Emacs** pieprasīs galīgo apstiprinājumu un beigs darbu.

## 18.nodaļa. Slackware pakotņu vadība

Programmatūras pakotne ir instalēšanai sagatavots saistītu programmu kopums. Lejuplādējot pirmkoda arhīvu, jums jāveic tā konfigurēšana, kompilēšana un to manuāli jāinstalē. Izmantojot programmatūras pakotni, tas viss jau ir veikts. Viss, ko jāveic - jāinstalē šo pakotni. Vēl kāda ērta pakotņu īpašība ir iespēja tās ļoti vienkārši dzēst un atjaunināt. *Slackware* ir iekļautas visi pakotņu vadībai nepieciešamie rīki, ar kuru palīdzību pakotnes iespējams ļoti vienkārši instalēt, dzēst, atjaunināt, izveidot un pārbaudīt.

Kopš *RedHat* ieviesa *RedHat* pakotņu vadības rīku *RPM (RedHat Package Manager)*, radies mīts, ka *Slackware* neesot pakotņu vadības rīka. Nekas nevar būt tālāk no patiesības. *Slackware* vienmēr ir bijis savs pakotņu vadības rīks - vēl tad, kad *RedHat* neeksistēja. Lai arī ne tik pilnīgs un universāls kā *RPM* (vai, piemēram, *DEB*), *pkgtool* un ar to saistītās programmas spēj instalēt pakotnes tikpat veiksmīgi, kā *RPM*. Patiesība par *pkgtool* ir tāda, ka tas nevis neeksistē, bet gan nepārbauda savstarpējo pakotņu atkarību.

Nez kādēļ daudzi *Linux* lietotāji un izstrādātāji iedomājas, ka pakotņu vadības rīkam noteikti jāveic pakotņu savstarpējās atkarības pārbaudi. Šis nav tas gadījums, jo *Slackware* to neveic. Tas nenozīmē, ka *Slackware* pakotnēm nav savstarpējo atkarību, bet gan to, ka pakotņu vadības rīks tās nepārbauda. Šī pārbaude ir atstāta sistēmas administratora ziņā, un mums tas patīk.

### 18.1. Pakotņu formāta apskats

Pirms rīku aplūkošanas jāiepazīstas ar *Slackware* pakotņu formātu. *Slackware* sistēmā pakotne ir vienkāršs *tar* arhīvs, kas saspriests ar *gzip*. Pakotnes tiek veidotas tā, lai tiktu atspiestas saknes direktorijā.

Piemēram aplūkosim kādu iedomātu programmu un tās pakotnes piemēru:

```
./
usr/
usr/bin/
usr/bin/makehejaz
usr/doc/
usr/doc/makehejaz-1.0/
usr/doc/makehejaz-1.0/COPYING
usr/doc/makehejaz-1.0/README
usr/man/
usr/man/man1
usr/man/man1/makehejaz.1.gz
install/
install/doinst.sh
```

Pakotņu vadības sistēma pakotnes instalēšanai to atarhivēs saknes direktorijā. Pakotņu datu bāzē tiks izveidots ieraksts ar pakotnes saturu, lai to vēlāk būtu iespējams atjaunināt vai dzēst.

Ievērojiet direktoriju `install/`. Šī direktorija satur pēcinstalācijas skriptu ar nosaukumu `doinst.sh`. Ja pakotņu vadības sistēmas atrod šādu failu, tas pēc pakotnes instalēšanas tiks izpildīts.

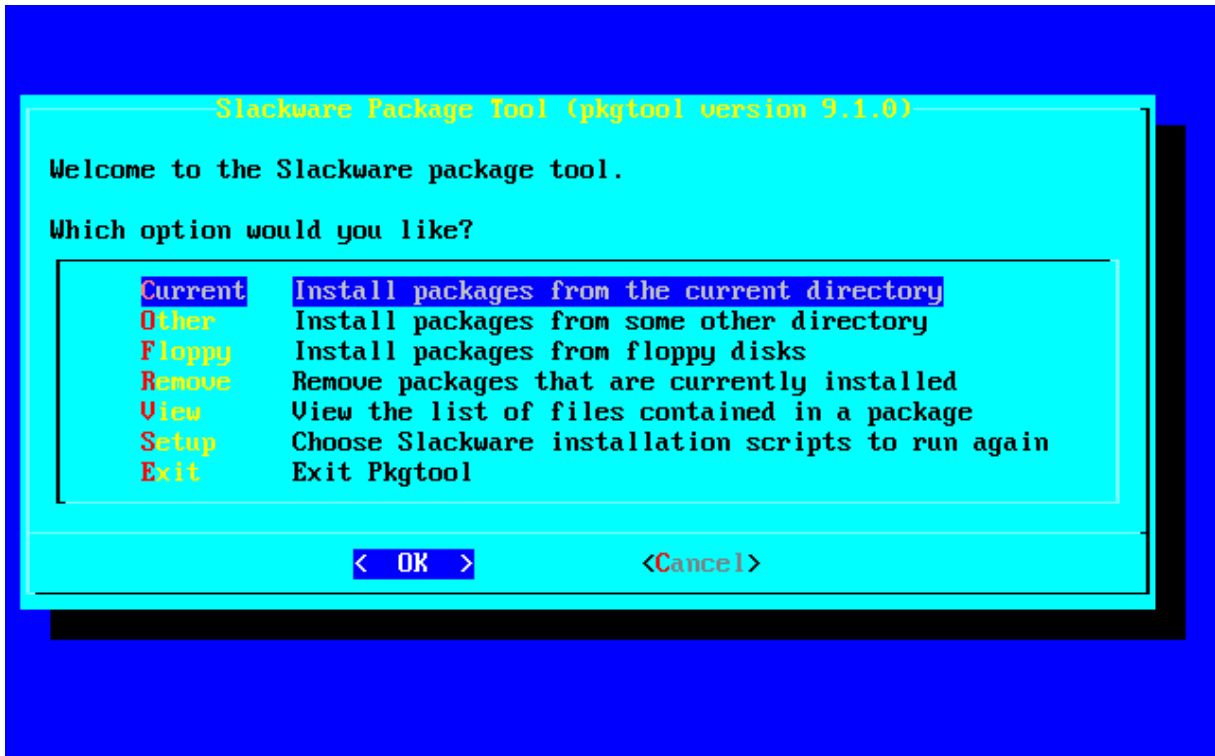
Pakotnē iespējams iekļaut arī citus skriptus - tos sīkāk aplūkosim sadaļā 18.3.

### 18.2. Pakotņu rīki

Pakotņu vadībai paredzēti četri galvenie rīki. Tie veic pakotņu instalēšanu, dzēšanu un atjaunināšanu.

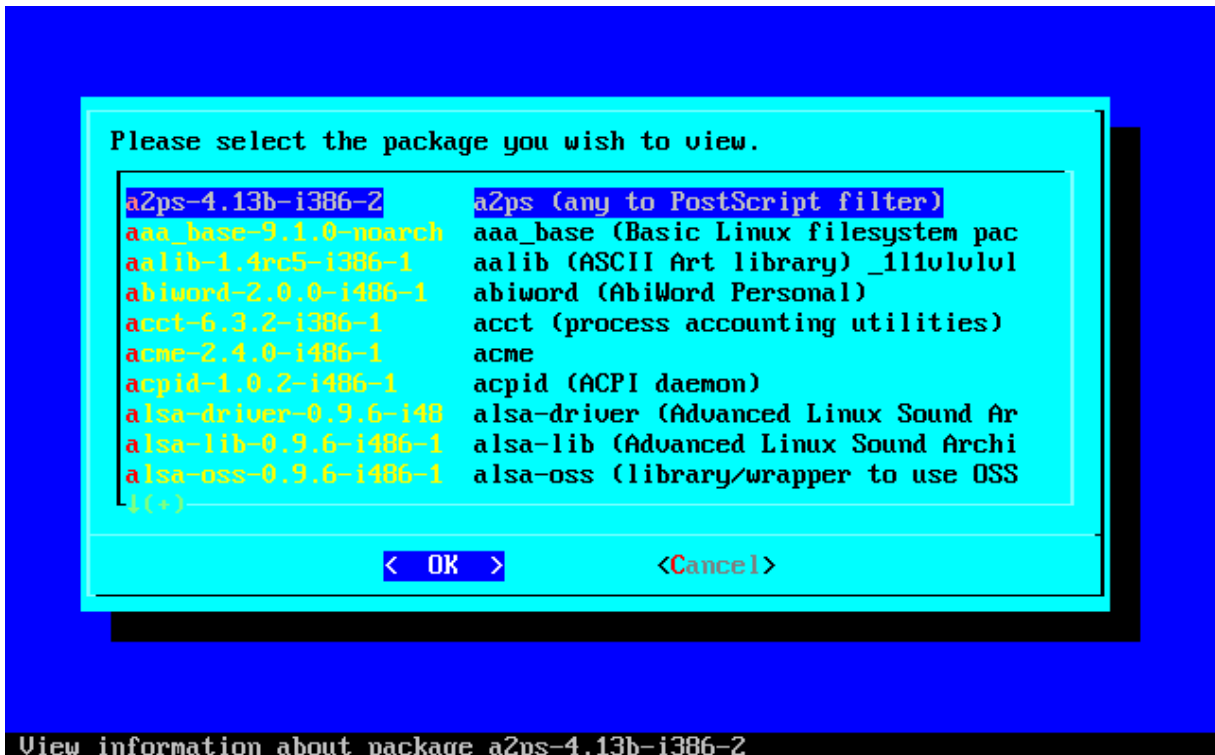
## pkgtool

pkgtool (8) ir programma ar izvēlnēm, kas ļauj instalēt un dzēst pakotnes.



Tiek piedāvāts instalēt pakotnes no tekošās direktorijas, citas direktorijas, vai no disketēm. Vienkārši izvēlieties instalēšanas metodi un pkgtool meklēs pieejamās pakotnes.

Ir iespējams arī aplūkot instalēto pakotņu sarakstu.



Ja jūs vēlaties dzēst pakotnes, izvēlieties dzēšanas režīmu, lai aplūkotu sarakstu ar visām instalētajām pakotnēm. Iezīmējiet tās, kuras vēlaties dzēst un izvēlieties OK. `pkgtool` tās nodzēsīs.

Dažiem lietotājiem šis rīks patīk labāk, kā komandrindas rīki, tomēr jāatceras, ka komandrindas rīki piedāvā vairāk konfigurācijas parametrus. Pakotņu atjaunināšana iespējama tikai, izmantojot komandrindas rīkus.

## installpkg

`installpkg` (8) veic pakotņu instalēšanu. To izpilda šādi:

```
# installpkg parametrs pakotnes_nosaukums
```

`installpkg` pieejami trīs parametri, no kuriem vienlaikus iespējams izmantot tikai vienu.

| Parametrs          | Nozīme                                                                                                                                                                                                |
|--------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>-m</code>    | Veic <code>makepkg</code> darbību tekošajā direktorijā                                                                                                                                                |
| <code>-warn</code> | Parāda, kas notiktu, ja tiktu instalēta norādītā pakotne. Šis parametrs noder ekspluatācijas sistēmās, jo pirms pakotnes instalēšanas iespējams aplūkot, kas tieši veikts.                            |
| <code>-r</code>    | Rekursīvi instalē visas pakotnes, kas atrodas tekošajā direktorijā un tās apakšdirektorijās. Pakotnes nosaukumā iespējams izmantot aizstājējzīmes, kuras tiks izmantotas kā pakotņu meklēšanas maska. |

Ja pirms `installpkg` izpildes tiek uzstādīts vides mainīgais `ROOT`, tā saturs tiks izmantots kā saknes direktorijas nosaukums. Šāda iespēja ir noderīga, sagatavojot jaunu disku saknes direktorijai, jo jauns disks parasti tiek montēts direktorijā `/mnt` vai citā, bet ne `/`.

Instalēto pakotņu datu bāze tiek glabāta direktorijā `/var/log/packages`. Datu bāzes ieraksts patiesībā ir vienkāršs teksta fails. Katrai pakotnei tiek izveidots atsevišķs fails. Ja pakotnē ir iekļauts pēcinstalācijas skripts, tas tiek ierakstīts `/var/log/scripts/`.

Ir iespējams norādīt vairākas instalējamās pakotnes, vai pakotņu nosaukumos izmantot aizstājējzīmes. Atcerieties, ka `installpkg` jūs nebrīdinās, veicot jau instalētas pakotnes pārrakstīšanu - jaunais pakotnes eksemplārs vienkārši tiks pārrakstīts pār iepriekšējo. Ja vēlaties pārlicināties, ka iepriekšējās pakotnes faili tiek droši saglabāti, izmantojiet `upgradepkg`.

## removepkg

`removepkg` (8) veic instalēto pakotņu dzēšanu no sistēmas. Izmantojama šāda sintakse:

```
# removepkg parametrs pakotnes_nosaukums
```

`removepkg` pieejami četri parametri, no kuriem vienlaikus iespējams izmantot tikai vienu.

| Parametrs              | Nozīme                                                                              |
|------------------------|-------------------------------------------------------------------------------------|
| <code>-copy</code>     | Pakotne tiek kopēta saglabāto pakotņu direktorijā                                   |
| <code>-keep</code>     | Saglabā dzēšanas laikā radītos pagaidu failus. Noder vienīgi atklūdošanas procesā.  |
| <code>-preserve</code> | Pakotne tiek dzēsta, tomēr vienlaikus tiek saglabāta saglabāto pakotņu direktorijā. |
| <code>-warn</code>     | Parāda, kas notiktu, ja tiktu dzēsta pakotne.                                       |

| Parametrs | Nozīme |
|-----------|--------|
|           |        |

Ja pirms `removepkg` izpildes tiek uzstādīts vides mainīgais `ROOT`, tā saturs tiks izmantots kā saknes direktorijas nosaukums. Šāda iespēja ir noderīga, sagatavojot jaunu disku saknes direktorijai, jo jauns disks parasti tiek montēts direktorijā `/mnt` vai citā, bet ne `/`.

`removepkg` pārbauda arī citas instalētās pakotnes un dzēš tikai tos failus, ko izmanto tikai norādītā pakotne. Tiks pārbaudīts arī norādītās pakotnes pēcināšanas skripts un dzēšanas visas tā radītās simbolsaites.

Dzēšanas laikā tiek attēlota statusa atskaite. Pēc dzēšanas pakotņu datu bāzes ieraksts tiek pārvietots uz `/var/log/removed_packages` un pēcināšanas skripts tiek pārvietots uz `/var/log/removed_scripts`.

Līdzīgi kā ar `installpkg`, pakotnes nosaukumā iespējams izmantot aizstājējzīmes, kā arī iespējams norādīt vairākas pakotnes.

## upgradepkg

`upgradepkg` (8) atjauninās jau instalētu *Slackware* pakotni. Jāizmanto šādu sintaksi:

```
# upgradepkg pakotnes_nosaukums
```

vai

```
# upgradepkg vecais_pakotnes_nosaukums%jaunais_pakotnes_nosaukums
```

`upgradepkg` vispirms instalē jauno pakotni, pēc tam dzēš veco, nodrošinot, ka vecie faili vairs netiek izmantoti. Ja atjauninātajai pakotnei mainījies nosaukums, izmantojiet procenta zīmi, lai norādītu veco pakotni (to, kas jau instalēta) un jauno (to, kuru vēlaties instalēt).

Ja pirms `upgradepkg` izpildes tiek uzstādīts vides mainīgais `ROOT`, tā saturs tiks izmantots kā saknes direktorijas nosaukums. Šāda iespēja ir noderīga, sagatavojot jaunu disku saknes direktorijai, jo jauns disks parasti tiek montēts direktorijā `/mnt` vai citā, bet ne `/`.

`upgradepkg` nav bez trūkumiem. Vienmēr jāizveido konfigurācijas failu rezerves kopijas. Ja tās tiek dzēstas vai pārrakstītas, jūs vēlēsities oriģinālu kopijas, lai veiktu bojājumu labošanu.

Līdzīgi kā ar `installpkg` un `removepkg`, ir iespējams norādīt vairākas pakotnes vai pakotnes nosaukumā izmantot aizstājējzīmes.

## rpm2tgz|rpm2targz

*RedHat* pakotņu vadības rīks mūsdienās ir ļoti populārs. Daudzi programmatūras izplatītāji piedāvā savus produktus tieši *RPM* formātā. Tā kā *RPM* nav *Slackware* pakotņu formāts, mēs neiesakām to izmantot. Diemžēl ir pakotnes, kas pieejamas tikai *RPM* formātā (pat to pirmkods).

Šim nolūkam mēs esam izveidojuši programmu, kas pārveidos *RPM* pakotnes mūsu `.tgz` formātā. Tas ļaus jums atarhivēt pakotni (šķiet, ar `explodepkg`) pagaidu direktorijā un aplūkot tās saturu.

Programma `rpm2tgz` izveidos *Slackware* pakotni ar paplašinājumu `.tgz`, kamēr `rpm2targz` izveidos arhīvu ar paplašinājumu `.tar.gz`.

## 18.3. Pakotņu izveide

*Slackware* pakotņu izveide var būt gan vienkārša, gan sarežģīta. Nav kādas noteiktas metodes pakotņu izveidei. Vienīgās prasības ir, lai pakotne būtu ar `gzip` saspriests `tar` arhīvs un, ja pakotnē

iekļauts pēcinstalācijas skripts, lai tas būtu `/install/doinst.sh`.

Ja jūs vēlaties izveidot pakotnes savai sistēmai vai jūsu pārvaldītajam datortīklam, jums jāaplūko dažādos *Slackware* pirmkoda koka kompilēšanas skriptus. Mēs pakotņu izveidei izmantojam vairākas metodes.

## explodepkg

`explodepkg` (8) veic to pašu, ko `installpkg` veic pakotnes atarhivēšanai, tomēr šis rīks neveic pakotnes instalēšanu un neregistrē savas darbības pakotņu datu bāzē. Tas vienkārši atarhivē pakotni tekošajā direktorijā.

Aplūkojot *Slackware* pirmkoda koku, jūs redzēsiet, kā mēs šo komandu izmantojam "struktūras" pakotnēm. Šīs pakotnes satur skeletu tam, kā izskatīsies pakotnes gala versija. Tās satur visus nepieciešamos failu nosaukumus (ar nulles garumu), tiesības un informāciju par failu īpašniekiem. Pakotnes būvēšanas skripts nokopē pakotnes saturu no pirmkoda direktorijas uz pakotnes direktoriju.

## makepkg

`makepkg` (8) no failiem tekošajā direktorijā sagatavo derīgu *Slackware* pakotni. Tas pakotnes kokā meklē simbolsaites un pēcinstalācijas skriptam pievieno bloku, kas tās izveidos instalēšanas laikā. Šis rīks arī brīdina par visiem pakotnes kokā atrastajiem failiem ar nulles garumu.

Šo komandu parasti izpilda pēc pakotnes koka sagatavošanas.

## SlackBuild skripti

Ja nepieciešams, *Slackware* pakotnes tiek būvētas dažādos veidos. Ne visas programmas tiek veidotas tā, lai tās būtu kompilējamas vienādi. Daudzām programmām ir pieejami kompilēšanas laika parametri, kas visi nav iekļauti *Slackware* pakotnēs. Iespējams, jums var nākties pievienot šo funkcionalitāti - jums nāksies pārkompilēt savu pakotni. Par laimi pakotnes pirmkodā iespējams iebūvēt *SlackBuild* skriptus.

Kas ir *SlackBuild* skripti? *SlackBuild* skripti ir izpildāmi čaulas skripti, kas tiek izpildīti ar `root` tiesībām, lai konfigurētu, kompilētu un izveidotu *Slackware* pakotnes. Šos skriptus var brīvi modificēt pirmkoda direktorijā un tad izpildīt, lai izveidotu savas *Slackware* pakotņu versijas.

## 18.4. Birkas un birku failu izveide

*Slackware* instalēšanas programma vada jūsu sistēmas pakotņu instalēšanu, izmantojot failus, kuri satur informāciju par to, kuras pakotnes ir jāinstalē obligāti, kuras var instalēt pēc izvēles un kuras tiek izvēlētas pēc noklusējuma.

Birku fails - `tagfile` - atrodas pirmajā programmatūras sēriju direktorijā. Tas satur attiecīgā disku komplekta pakotņu sarakstu un to statusu. Statuss var būt:

**ADD** - pakotne obligāti nepieciešama korektai sistēmas instalēšanai

**SKP** - pakotne tiks automātiski izlaista

**REC** - pakotne nav obligāti nepieciešama, tomēr ir ieteicama

**OPT** - pakotni var instalēt pēc izvēles

Ierakstu formāts ir vienkāršs:

```
pakotnes_nosaukums: statuss
```

Vienā rindiņā ierakstīts viens ieraksts. Sākotnējie katras pakotņu sērijas birku faili tiek glabāti kā

`tagfile.org`. Līdz ar to, nejauši sabojājot savu failu, ir iespējams atjaunot sākotnējo.

Daudzi administratori izvēlas izveidot savus birku failus, pēc tam startējot instalēšanas programmu un izvēloties "full". Instalēšanas programma nolasīs birku failus un veiks instalēšanu pēc tajos norādītajiem ierakstiem. Ja tiek izmantots statuss `REC` vai `OPT`, tiks piedāvāta izvēle, vai instalēt noteikto pakotni. Līdz ar to ir ieteicams pieturēties pie `ADD` un `SKP`, ja tiek veidoti birku faili sistēmas automātiskai instalēšanai.

Vēl jāpārlicinās, ka birku faili tiek ierakstīti turpat, kur oriģināli. Ir iespējams norādīt arī pielāgotu birku failu ceļu, ja tiek izmantoti pielāgoti birku faili.



## 19.nodaļa. ZipSlack

### 19.1. Kas ir ZipSlack?

*ZipSlack* ir *Slackware Linux* īpaša versija. Tā ir jau instalēta *Slackware* kopija, ko paredzēts izpildīt no *DOS* vai *Windows* diska sadaļas. Šī instalācija satur tikai svarīgākās programmas - tajā nav iekļauts viss *Slackware* programmu kopums.

*ZipSlack* nosaukums radies no tā izplatīšanas formāta - liela `.ZIP` faila. *DOS* un *Windows* lietotāji, domājams, jau ir pazīstami ar šiem failiem. Tie ir saspiesti arhīvi. *ZipSlack* arhīvs satur visu, kas nepieciešams *Slackware* startēšanai.

Ir svarīgi atcerēties, ka *ZipSlack* būtiski atšķiras no parastas instalācijas. Lai arī tās abas darbojas vienādi un satur vienas un tās pašas programmas, to mērķauditorijas un funkcijas ir dažādas. Zemāk aplūkosim vairākas *ZipSlack* priekšrocības un trūkumus.

Un beidzot - vienmēr iepazīstieties ar *ZipSlack* direktoriņā iekļauto dokumentāciju. Tā satur svaigāko informāciju par instalēšanu, ielādi un produkta lietošanu.

#### Priekšrocības

- Nav nepieciešama diska sadaļu veidošana
- Lieliska metode, kā apgūt *Slackware Linux* bez tās instalēšanas

#### Trūkumi

- Izmanto *DOS* failsistēmu, kas ir lēnāka par *Linux* failsistēmu
- Nedarbojas *Windows NT* sistēmās

### 19.2. Apgādājamies ar ZipSlack

*ZipSlack* iegūšana ir vienkārša. Ja jūs iegādājāties oficiālo *Slackware Linux* CD komplektu, tad jums jau ir *ZipSlack*. Atrodiet disku, kas satur `zipslack` direktoriņu un ievietojiet to CD-ROM iekārtā. Parasti tas ir trešais vai ceturtais disks, tomēr labāk uzticēties dokumentācijā rakstītajam, jo disks, kurā atrodas *ZipSlack*, var mainīties.

Ja jūs vēlaties lejuplādēt *ZipSlack*, vispirms apciemojiet mūsu "*Get Slack*" tīmekļa vietni, lai iepazītos ar jaunāko lejuplādes informāciju:

<http://www.slackware.com/getslack/>

*ZipSlack* ir katra *Slackware* laidiena sastāvdaļa. Atrodiet jūs interesējošo laidieni un atveriet atbilstošo direktoriņu FTP serverī. Jaunākā laidiena direktoriņa atrodama šeit:

<ftp://ftp.slackware.com/pub/slackware/slackware/>

*ZipSlack* atrodams apakšdirektoriņā `/zipslack`. *ZipSlack* tiek piedāvāts viena liela `.ZIP` faila formātā, kā arī sadalīts disketes izmēra fragmentos. Šie fragmenti atrodami direktoriņā `/zipslack/split`.

Neapstājieties pie `.ZIP` failiem. Jums arī jālejuplādē dokumentācijas failus un visus ielādes disku attēlus, kas atrodami šajā direktoriņā.

#### Instalēšana

Kad nepieciešamie komponenti lejuplādēti, jums jāatarhivē `.ZIP` failu. Izmantojiet 32 bitu

atrhivēšanas rīku. Arhīvā iekļauto failu apjomi un nosaukumi ir par daudz 16 bitu rīkam. Piemēri 32 bitu rīkiem *Windows* vidē ir *WinZip* un *PKZIP for Windows*.

*ZipSlack* ir veidots, lai to atspiestu tieši iekārtas saknes direktorijā (kā **C:** vai **D:**). Tajā tiks izveidota **\LINUX** direktorija, kas saturēs *Slackware* instalāciju. Šajā direktorijā jūs atradīsiet arī sistēmas ielādei nepieciešamos failus.

Pēc failu atspiešanas jūsu izvēlētajā iekārtā jāatrodas direktorijai **\LINUX** (turpmāk uzskatīsim, ka izvēlēta iekārta ir **C:**).

### **19.3. ZipSlack ielāde**

Ir vairāki veidi, kā ielādēt *ZipSlack*. Visbiežāk izmantotais ir tajā iekļautais fails **LINUX.BAT**, kas ielādē sistēmu no *DOS* vides (vai *Windows 9x* sistēmu *DOS* režīmā). Failā jāveic modifikācijas, lai tas atbilstu jūsu sistēmai.

Sāciet ar faila **C:\LINUX\LINUX.BAT** atvēršanu izvēlētajā tekstu redaktorā. Faila augšdaļā jūs pamanīsiet lielu komentāru. Tas izskaidro, ko jādara, lai korekti veiktu izmaiņas šajā failā (un arī ko darīt, ja ielādi jāveic no ārējās *ZIP* iekārtas). Neuztraucieties, ja neizprotat iestatījumu **root=**. Failā sniegti vairāki piemēri, tādēļ brīvi izvēlieties kādu no tiem un izmēģiniet. Ja tas nedarbojas, varat mēģināt atkārtoti rediģēt failu, komentēt atkomentēto rindiņu un izvēlēties citu.

Kad izvēlēta rindiņa atkomentēta, nodzēšot tās sākumā simbolus **rem**, saglabājiet failu un aizveriet redaktoru. Pārstartējiet datoru *DOS* režīmā.

*DOS* komandrinda *Windows 9x* vidē NEDARBOSIES.

Ievadiet **C:\LINUX\LINUX.BAT**, lai startētu sistēmu. Ja viss darbosies, jums tiks piedāvāts autorizēties sistēmā.

Autorizējieties kā **root** - bez paroles. Jūs, iespējams, vēlēsit uzstādīt **root** paroli, kā arī pievienot savu lietotāju. Šobrīd jūs varat iepazīties ar citām šīs grāmatas sadaļām, lai apgūtu sistēmas lietošanu.

Ja **LINUX.BAT** faila izmantošana nedarbojas, jums jāiepazīstas ar failu **C:\LINUX\README.1ST**, kurā aprakstīti citi ielādes veidi.

## Pielikums A. GNU vispārējā publiskā licence

Šis ir neoficiāls GNU Vispārējās publiskās licences tulkojums latviešu valodā. To nav publicējis Brīvas programmatūras fonds, un tas nenosaka likumīgus izplatīšanas noteikumus programmatūrai, kas izmanto GNU VPL--tos nosaka tikai oriģinālais teksts angļu valodā. Tomēr mēs ceram, ka šis tulkojums latviski lasošajiem palīdzēs labāk saprast GNU VPL.

This is an unofficial translation of the GNU General Public License into Latvian. It was not published by the Free Software Foundation, and does not legally state the distribution terms for software that use the GNU GPL--only the original English text of the GNU GPL does that. However, we hope that this translation will help Latvian speakers understand the GNU GPL better.

### GNU VISPĀRĒJĀ PUBLISKĀ LICENCE

2. versija 1991. gada jūnijs

Copyright (C) 1989, 1991 Free Software Foundation, Inc. 59 Temple Place, Suite 330, Boston, MA 02111-1307, USA

Autortiesības (C) 1998 Vitauts Stočka, tulkojums latviešu valodā

Ikvienam ir atļauts kopēt un izplatīt burtiskas šī licences dokumenta kopijas, taču to izmaiņšana nav atļauta.

### Ievads

Lielākajai programmatūras daļai licences tiek veidotas tā, lai neļautu jums to kopīgi izmantot un izmainīt. Pretēji tam GNU Vispārējās publiskās licences nolūks ir garantēt jums brīvību kopīgi izmantot un izmainīt brīvu programmatūru - lai nodrošinātu, ka šī programmatūra ir bezmaksas visiem tās lietotājiem. Šī Vispārējā publiskā licence attiecas uz vairumu no Brīvas programmatūras fonda programmatūru un uz jebkuru citu programmu, kuras autori ir piekrituši to izmantot. (Daļai Brīvas programmatūras fonda programmatūru savukārt tiek pielietota GNU Bibliotēku vispārējā publiskā licence.) Jūs to varat izmantot arī savām programmām.

Kad mēs runājam par brīvu programmatūru, mēs ar to saprotam brīvību nevis cenu. Mūsu Vispārējās publiskās licences ir radītas ar nolūku nodrošināt, lai jums būtu tiesības izplatīt brīvas programmatūras kopijas (un, ja vēlaties, par šo pakalpojumu pieprasīt samaksu), lai jūs saņemtu sākumkodu vai varētu to saņemt, ja vien vēlaties, lai jūs varētu izmainīt programmatūru vai izmantot tās fragmentus jaunās brīvās programmās; un, lai jūs zinātu, ka jūs varat darīt visas šīs lietas.

Lai aizsargātu jūsu tiesības, mums ir jāievieš ierobežojumi, kas aizliedz kādam atņemt jums šīs tiesības vai pieprasīt jūs atteikties no tiesībām. Šie ierobežojumi pārvēršas par noteiktiem jūsu pienākumiem, ja jūs izplatāt programmatūras kopijas vai modificējat to.

Piemēram, ja jūs izplatāt šādas programmas kopijas, vai bezmaksas, vai par samaksu, jums jādod saņēmējiem visas tās tiesības, kas ir jums. Jums jānodrošina, lai arī viņi tāpat saņemtu vai varētu saņemt sākumkodu. Un jums jāparāda viņiem šie noteikumi, lai viņi zinātu savas tiesības.

Mēs aizsargājam jūsu tiesības divos soļos: (1) saglabājam programmatūras autortiesības un (2) piedāvājam jums šo licenci, kas dod jums legālu atļauju kopēt, izplatīt un/vai modificēt programmatūru. Tāpat, lai aizsargātu katru autoru un sevi, mēs vēlamies būt droši, lai ikviens saprot, ka šai brīvajai programmatūrai nav nekādu garantiju. Ja kāds cits modificē programmu un nodod to tālāk, mēs vēlamies, lai tās saņēmēji zinātu, ka viņu rīcībā nav oriģināls, un tādējādi jebkuras citu radītas problēmas neietekmētu oriģināla autora slavu.

Visbeidzot, jebkuru brīvu programmu nepārtraukti apdraud programmatūras patenti. Mēs vēlamies izvairīties no briesmām, ka brīvas programmas tālākizplatītāji individuāli saņemtu patenta licences, rezultātā padarot programmu par savu īpašumu. Lai to novērstu, mēs esam skaidri pateikuši, ka jebkuru patentu ir jālicencē brīvai izmantošanai visiem vai arī nav jālicencē vispār.

Tālāk seko precīzi noteikumi un nosacījumi kopēšanai, izplatīšanai un modificēšanai.

## GNU VISPĀRĒJĀ PUBLISKĀ LICENCE

### NOTEIKUMI UN NOSACĪJUMI KOPĒŠANAI, IZPLATĪŠANAI UN MODIFICĒŠANAI

**0.** Šī Licence attiecas uz jebkuru programmu vai citu darbu, kas satur autortiesību īpašnieka ievietotu paziņojumu, kurā teikts, ka to drīkst izplatīt saskaņā ar šīs Vispārējās publiskās licences noteikumiem. Tālāk "Programma" apzīmē jebkuru šādu programmu vai darbu, un "darbs, kas izveidots uz Programmas bāzes" nozīmē vai nu Programmu, vai arī jebkuru saskaņā ar autortiesību likumu atvasinātu darbu: tāpat tas ir darbs, kas satur Programmu vai tās daļu, neizmainītu vai ar modifikācijām un/vai tulkotu citā valodā. (Turpmāk tulkošana bez ierobežojumiem tiek iekļauta terminā "modificēšana"). Katrs licences saņēmējs tiek uzrunāts kā "jūs".

Darbības, kas nav kopēšana, izplatīšana un modificēšana, šī Licence neaplūko; tās atrodas ārpus tās kompetences. Programmas darbināšana nav ierobežota, un uz Programmas darbības iznākumu šī Licence attiecas tikai tad, ja tās saturs veido darbu, kas izveidots uz Programmas bāzes (neatkarīgi no tā, vai tas ir radies, darbinot Programmu). Vai tā notiek, ir atkarīgs no tā, ko tieši Programma dara.

**1.** Jūs drīkstat kopēt un izplatīt neizmainītas Programmas sākumkoda kopijas, tiklīdz jūs tās saņemat, uz jebkura informācijas nesēja, ar nosacījumu, ka jūs labi redzamā vietā uz katras kopijas publicējat atbilstošu autortiesību paziņojumu un garantiju atteikumu; ka atstājat bez izmaiņām visus paziņojumus, kas atsaucas uz šo Licenci un norāda jebkādu garantiju trūkumu; un, ka kopā ar Programmu jebkuriem citiem Programmas saņēmējiem nododat šīs Licences kopiju. Jūs drīkstat ņemt maksu par fizisku kopijas nodošanas aktu, kā arī drīkstat pēc savas vēlēšanās piedāvāt par samaksu garantijas saistības.

**2.** Jūs drīkstat modificēt jūsu rīcībā esošo Programmas kopiju vai kopijas, vai arī jebkuru tās daļu, tādējādi veidojot darbu, kas izveidots uz Programmas bāzes, kā arī kopēt un izplatīt šādas modifikācijas vai darbu saskaņā ar iepriekšminētās 1. nodaļas noteikumiem, ja vien jūs arī ievērojat visus šos nosacījumus:

a) Jums jānodrošina, lai izmainītie faili saturētu labi ieraugāmus paziņojumus, ka jūs esat izmainījuši šos failus, kā arī katru izmaiņu datumu.

b) Jums jānodrošina, lai jebkurš jūsu izplatītais vai publicētais darbs, ja tas kopumā vai kādā daļā satur Programmu vai ir atvasināts no Programmas vai kādas tās daļas, tiktu licencēts kā viens veselums bez maksas visām trešajām personām saskaņā ar šīs Licences noteikumiem.

c) Ja modificētā programma darbības laikā parasti interaktīvi izpilda komandas, jums jānodrošina, lai tā, uzsākot šādu interaktīvu komandu izpildi, visierastākajā veidā izdrukātu vai parādītu paziņojumu ar atbilstošu autortiesību paziņojumu un paziņojumu, ka nav nekādu garantiju (vai, pretējā gadījumā, norādot, ka jūs piedāvājat garantijas) un, ka lietotāji var izplatīt tālāk šo programmu saskaņā ar šiem nosacījumiem, kā arī paskaidrotu lietotājam, kā izlasīt šīs Licences kopiju. (Izņēmums: ja Programma pati ir interaktīva, taču parasti neizvada šādu paziņojumu, jūsu darbs, kas izveidots uz Programmas bāzes, arī var neizvadīt šādu paziņojumu.)

Šīs prasības attiecas uz modificētu darbu kā vienu veselumu. Ja zināmas šī darba daļas nav iegūtas no Programmas un var būt pamatoti uzskatāmas par neatkarīgiem un atsevišķiem darbiem pašas par sevi, tad šī Licence un tās noteikumi neattiecas uz šīm daļām, kad jūs tās izplatāt kā atsevišķus darbus. Taču, ja jūs izplatāt tās pašas daļas kā viena veselā sastāvdaļu, kas ir darbs, izveidots uz Programmas bāzes, šai viena veselā izplatīšanai jānotiek saskaņā ar šīs Licences noteikumiem, kas citiem licences saņēmējiem nodrošina tiesības uz visu darbu, tādējādi arī uz katru un ikvienu tā daļu neatkarīgi no tā, kas to ir uzrakstījis.

Tādējādi šī punkta nolūks nav pieprasīt sev tiesības vai apstrīdēt jūsu tiesības uz darbu, ko esat uzrakstījuši pilnībā saviem spēkiem; mūsu nolūks drīzāk ir izmantot tiesības kontrolēt atvasinātu vai kolektīvu darbu izplatīšanu, kas izveidoti uz Programmas bāzes.

Bez tam, cita darba, kas nav izveidots uz Programmas bāzes, vienkārša atrašanās uz viena uzglabāšanas vai izplatīšanas datu nesēja kopā ar Programmu (vai ar darbu, kas izveidots uz Programmas bāzes) nenozīmē cita darba nonākšanu šīs Licences darbības sfērā.

**3.** Jūs drīkstat kopēt un izplatīt Programmu (vai darbu, kas izveidots uz tās bāzes saskaņā ar 2.

nodaļu) objektkodā vai izpildāmā formā saskaņā ar iepriekšminēto 1. un 2. nodaļas noteikumiem ar nosacījumu, ka jūs arī veicat vienu no sekojošajām darbībām:

a) Pavadīt to ar atbilstošu pilnīgu mašīnlasāmu sākumkodu, kam jābūt izplatītam saskaņā ar iepriekšminēto 1. un 2. nodaļas noteikumiem uz programmatūras apmaiņai parasti izmantota informācijas nesēja; vai,

b) Pavadīt to ar rakstisku piedāvājumu, spēkā esošu vismaz trīs gadus, ka jebkurai trešajai personai par maksu, kas nepārsniedz jūsu fiziskās sākumkoda izplatīšanas izmaksas, piedāvājat atbilstošu pilnīgu mašīnlasāmu sākumkodu, kam jābūt izplatītam saskaņā ar iepriekšminētajiem 1. un 2. nodaļas noteikumiem uz programmatūras apmaiņai parasti izmantota informācijas nesēja; vai,

c) Pavadīt to ar informāciju, ko jūs saņēmat kā piedāvājumu izplatīt atbilstošu sākumkodu. (Šī alternatīva ir atļauta tikai nekomerciālai izplatīšanai, un tikai tad, ja jūs saņēmat programmu objektkodā vai izpildāmā formā kopā ar šādu piedāvājumu, kas ir saskaņā ar iepriekšminēto apakšnodaļu b.)

Darba sākumkods nozīmē tādu darba formu, ko ir visērtāk modificēt. Pildāmam darbam pilnīgs sākumkods nozīmē sākumkodu visiem moduļiem, ko tas satur, kā arī jebkādas ar to saistītas interfeisa definīciju failus, tāpat scenārijus, ko izmanto kompilēšanas kontrolei un izpildāmā darba instalēšanai. Tomēr, kā īpašs izņēmums, izplatāmais sākumkods drīkst nesaturēt neko, kas parasti tiek izplatīts (vai nu sākumkodā, vai binārā formā) kopā ar tās operētājsistēmas galvenajiem komponentiem

(kompilatoru, kodolu utt.), kurai paredzēts izpildāmais darbs, ja vien šis komponents pats nepavada izpildāmo darbu.

Ja izpildāmais darbs vai objektkods tiek izplatīts, piedāvājot pieklūšanu darba kopijai noteiktā vietā, tad tāds pats piedāvājums no tās pašas vietas nokopēt sākumkodu ir uzskatāms par sākumkoda izplatīšanu pat tad, ja trešajām personām netiek piespiests nokopēt sākumkodu kopā ar objektkodu.

4. Jums nav tiesību kopēt, modificēt, licencēt tālāk vai izplatīt Programmu citos veidos, izņemot tos, kas tieši norādīti šajā Licencē. Jebkurš mēģinājums citādi kopēt, modificēt, licencēt tālāk vai izplatīt Programmu ir nelikumīgs un automātiski atņem jums šīs Licences piešķirtās tiesības. Tomēr pusēm, kas no jums saskaņā ar šo Licenci ir saņēmušas kopijas vai tiesības, licences paliek spēkā tik ilgi, kamēr šīs personas pilnībā tās ievēro.

5. No jums netiek pieprasīts akceptēt šo Licenci, kamēr jūs neesat to parakstījuši. Tomēr nekas cits nesniedz jums atļauju modificēt vai izplatīt Programmu vai no tās atvasinātos darbus. Likums aizliedz šīs darbības, ja jūs nepieņemat šo Licenci. Tāpēc, izmainot vai izplatot Programmu (vai jebkuru darbu, kas izveidots uz Programmas bāzes), jūs apstiprināt, ka esat pieņēmuši šīs Licences atļauju to darīt, kā arī tās noteikumus un nosacījumus Programmas vai darbu, kas izveidots uz tās bāzes, kopēšanai, izplatīšanai vai modificēšanai.

6. Katru reizi, kad jūs izplatāt tālāk Programmu (vai jebkuru darbu, kas izveidots uz Programmas bāzes), saņēmjēs automātiski no sākotnējā licences īpašnieka saņemt atļauju kopēt, izplatīt vai modificēt Programmu saskaņā ar šiem noteikumiem un nosacījumiem. Jūs nedrīkstat saņēmjējam uzspiest jebkādas papildus ierobežojumus izmantot šeit piešķirtās tiesības. Jūs neesat atbildīgs par to, kā trešās personas ievēro šīs Licences prasības.

7. Ja tiesas sprieduma rezultātā vai sakarā ar patenta pārkāpumu, vai arī jebkura cita iemesla dēļ (kas var būt saistīts ne tikai ar patentu jautājumiem) jūs piespiež (saskaņā ar tiesas lēmumu, vienošanos vai kā citādi) pārkāpt šīs Licences nosacījumus, tas neatbrīvo jūs no šīs Licences nosacījumiem. Ja jūs nevarat turpināt izplatīšanu tā, lai vienlaicīgi ievērotu savas saistības saskaņā ar šo Licenci un jebkuras citas piemērotās saistības, tad rezultātā jūs vispār nedrīkstat izplatīt Programmu. Piemēram, ja patenta licence neļautu visiem, kas no jums tieši vai netieši saņēmuši kopijas, bez maksas izplatīt tālāk Programmu, tad vienīgais veids, kā jūs varat apmierināt gan patenta, gan šīs Licences prasības, ir pilnībā atteikties no Programmas izplatīšanas.

Ja jebkura šī punkta daļa kādos konkrētos apstākļos nav spēkā vai nevar tikt piemērota, tad ir paredzēts, ka tiek piemērota atlikusī šī punkta daļa, un punkts kā viens vesels tiek piemērots citos apstākļos. Šī punkta nolūks nav pamudināt jūs pārkāpt jebkādas patentus vai citas tiesību prasības vai

apstrīdēt jebkādu šādu prasību pamatotību; šī punkta vienīgais nolūks ir aizsargāt brīvas programmatūras izplatīšanas sistēmas viengabalainību, kas tiek realizēta, izmantojot publiskas licences. Daudzi cilvēki pateicoties šai izplatīšanas sistēmai ir devuši dāsnu ieguldījumu plašā programmatūras spektrā, paļaujoties uz šīs sistēmas konsekventu ievērošanu; autora ziņā ir izlemt, vai viņš/viņa vēlas izplatīt programmatūru ar kādas citas sistēmas starpniecību, un licences saņēmējs nevar ietekmēt šo izvēli.

Šī punkta nolūks ir pavisam skaidri pateikt to, kas ir uzskatāms par šīs Licences pārējo daļu sekām.

8. Ja Programmas izplatīšana un/vai izmantošana noteiktās valstīs ir ierobežota sakarā ar patentiem vai autortiesību aizsargātiem interfeisiem, sākotnējais autortiesību īpašnieks, kas nodod Programmu šīs Licences pārziņā, var pievienot skaidri formulētu izplatīšanas ģeogrāfisko ierobežojumu, izslēdzot šīs valstis un tādējādi atļaujot izplatīšanu tikai tajās valstīs, kas nav šādi izslēgtas. Šajā gadījumā šī Licence iekļauj ierobežojumu gluži tāpat, it kā tas būtu ierakstīts Licences tekstā.9. Brīvas programmatūras fonds reizēm var publicēt pārskatītas un/vai jaunas Vispārējās publiskās licences versijas. Šādas jaunas versijas pēc būtības būs līdzīgas šai versijai, taču tās var atšķirties detaļās, lai aplūkotu jaunas problēmas vai aizdomas.

9. Katrai versijai tiek piešķirts atšķirīgs versijas numurs. Ja Programma norāda šīs Licences versijas numuru, kas uz to attiecas, un "jebkuru vēlāku versiju", jums ir izvēle ievērot norādītās vai jebkuras Brīvas programmatūras fonda vēlāk publicētas versijas noteikumus un nosacījumus. Ja Programma nenorāda šīs Licences versijas numuru, jūs varat izvēlēties jebkuru versiju, ko jebkad ir publicējis Brīvas programmatūras fonds.

10. Ja jūs vēlaties iekļaut Programmas daļas citās brīvās programmās, kuru izplatīšanas nosacījumi ir atšķirīgi, uzrakstiet autoram un palūdziet atļauju. Attiecībā uz programmatūru, kuras autortiesību īpašnieks ir Brīvas programmatūras fonds, rakstiet uz Brīvas programmatūras fondu; reizēm mēs izdarām šādus izņēmumus. Mūsu lēmums būs atkarīgs no diviem mērķiem saglabāt brīvas programmatūras statusu visiem mūsu brīvās programmatūras atvasinājumiem un kopumā veicināt programmatūras koplietošanu un atkārtotu izmantošanu.

## NEKĀDU GARANTIJU

11. SAKARĀ AR TO, KA ŠĪ PROGRAMMA TIEK LICENCĒTA BEZ MAKSAS, ŠAI PROGRAMMAI NAV NEKĀDU GARANTIJU, CIK LIELĀ MĒRĀ TO PIEĻAUJ ATTIECĪGI LIKUMI. JA VIEN PRETĒJAIS NAV APGALVOTS RAKSTISKI, AUTORTIESĪBU ĪPAŠNIEKI UN/VAI CITAS TREŠĀS PERSONAS PIEDĀVĀ ŠO PROGRAMMU "KĀ IR", BEZ JEBKĀDĀM GARANTIJĀM, TIEŠI IZTEIKTĀM VAI ŠĶIETAMĀM, TAJĀ SKAITĀ, BET NE TIKAI, BEZ ŠĶIETAMĀM KOMERCIĀLAS VĒRTĪBAS UN PIEMĒROTĪBAS KONKRĒTIEM NOLŪKIEM GARANTIJĀM. JŪS UZŅEMATIES PILNU RISKU, KAS SAISTĪTS AR PROGRAMMAS KVALITĀTI UN VEIKTSPĒJU. JA PROGRAMMA IZRĀDĀS DEFEKTĪVA, JŪS UZŅEMATIES VISUS AR APKALPOŠANU, ATJAUNOŠANU UN LABOJUMIEM SAISTĪTOS IZDEVUMUS.

12. NEVIENĀ GADĪJUMĀ, JA VIEN TO NEPIEPRASA ATTIECĪGS LIKUMS VAI RAKSTISKA VIENOŠANĀS, NEVIENS AUTORTIESĪBU ĪPAŠNIEKS VAI JEBKURA CITA PERSONA, KAS SASKAŅĀ AR IEPRIEKŠMINĒTO ATĻAUJU VAR MODIFICĒT UN/VAI IZPLATĪT TĀLĀK PROGRAMMU, NAV ATBILDĪGI JŪSU PRIEKŠĀ PAR BOJĀJUMIEM, IESKAITOT JEBKĀDUS VISPĀRĒJUS, ĪPAŠUS, NEJAUŠUS VAI IZRIETOŠUS BOJĀJUMUS, KAS RADUŠIES SAKARĀ AR PROGRAMMAS IZMANTOŠANU VAI NESPĒJU TO IZMANTOT (IESKAITOT, BET NE TIKAI, DATU PAZAUDĒŠANU, DATU SABOJĀŠANU VAI ZAUDĒJUMUS, KAS RADUŠIES JŪMS VAI TREŠAJĀM PERSONĀM, VAI PROGRAMMAS NESPĒJU DARBOTIES KOPĀ AR JEBKURĀM CITĀM PROGRAMMĀM), PAT JA ŠĀDAM AUTORTIESĪBU ĪPAŠNIEKAM VAI CITAI PERSONAI BIJA ZIŅOTS PAR ŠĀDU ZAUDĒJUMU IESPĒJAMĪBU.

NOTEIKUMU UN NOSACĪJUMU BEIGAS

## Kā pielietot šos noteikumus jūsu jaunajām programmām

Ja jūs izstrādājat jaunu programmu un vēlaties, lai tā būtu maksimāli noderīga sabiedrībai, labākais veids to sasniegt ir padarīt to par brīvu programmatūru, ko ikviens var izplatīt tālāk un izmainīt saskaņā ar šiem noteikumiem.

Lai to izdarītu, programmai pievienojiet sekojošus paziņojumus. Visdrošāk ir tos pievienot katra sākumkoda faila sākumā, lai visefektīvāk paziņotu par garantiju trūkumu; bez tam katrā failā jābūt vismaz "autortiesību" rindai un norādei uz vietu, kur var atrast pilnu paziņojumu.

<viena rinda ar programmas nosaukumu un īsu norādi, ko tā dara>

Copyright (C) <gads> <autora vārds>

Šī programma ir brīva programmatūra; jūs varat to izplatīt tālāk un/vai modificēt saskaņā ar GNU Vispārējās publiskās licences noteikumiem, ko publicējis Brīvas programmatūras fonds; izmantojiet Licences 2. versiju vai (pēc jūsu izvēles) jebkuru vēlāku versiju.

Šī programma tiek izplatīta cerībā, ka tā būs noderīga, taču BEZ JEBKĀDĀM GARANTIJĀM; pat bez šķietamām KOMERCIĀLAS VĒRTĪBAS un PIEMĒROTĪBAS KONKRĒTIEM NOLŪKIEM garantijām. Sīkāku informāciju meklējiet GNU Vispārējā publiskajā licencē.

Jums vajadzēja saņemt GNU Vispārējās publiskās licences kopiju kopā ar šo programmu; ja jūs to nesaņēmat, rakstiet uz Brīvas programmatūras fondu, Free Software Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA Tāpat pievienojiet informāciju, kā sarakstīties ar jums pa elektronisko un parasto pastu.

Ja programma ir interaktīva, lieciet tai izvadīt īsu, apmēram šādu paziņojumu, kad tā sāk darboties interaktīvajā režīmā:

Gnomovision versija 69, Autortiesības (C) autora vārds Gnomovision ir ABSOLŪTI BEZ JEBKĀDĀM GARANTIJĀM; sīkākai

informācijai ievadiet komandu `show w'. Šī ir brīva programmatūra, un jūs tiek ataicināti to izplatīt tālāk saskaņā ar zināmiem nosacījumiem; sīkākai informācijai ievadiet komandu `show c'.

Hipotētiskajām komandas `show w' un `show c' jāparāda atbilstošas Vispārējās publiskās licences daļas. Protams, jūsu izmantotās komandas var saukties citādi nekā `show w' un `show c'; tās pat var būt pieejamas ar peles klikšķi vai kā izvēlnes punkti--atkarībā no tā, kas labāk atbilst jūsu programmai.

Jums arī vajadzētu no sava darba devēja (ja jūs strādājat kā programmētājs) vai mācību iestādes, ja tāda ir, dabūt parakstu par "atteikšanos no autortiesībām" uz šo programmu, ja tas ir nepieciešams. Šeit ir piemērs, kurā varat izmainīt vārdus: Jojodini un Ko ar šo atsakās no visām autortiesību interesēm uz programmu `Gnomovision' (kas uzmācas kompilatoriem), ko uzrakstījis James Hacker.

<Mag Nāta paraksts>, 1989. gada 1. aprīlī

Mag Nāts, Viceprezidents

Šī Vispārējā publiskā licence neatļauj jums iekļaut jūsu programmu tādās programmās, kas ir privātīpašums. Ja jūsu programma ir apakšprogrammu bibliotēka, jums būtu jāpadomā, vai nav izdevīgāk atļaut šo bibliotēku iesaistīt arī programmās, kas ir privātīpašums. Ja jūs to vēlaties, šīs Licences vietā izmantojiet GNU Bibliotēku vispārējo publisko Licenci.