Z DOS/Windows na Linux

Guido Gonzato, ggonza@itin.it v1.3.5, 31. avgust 2000

Ta HOWTO je posvečen vsem bodočim bivšim uporabnikom sistemov DOS in Windows, ki so se odločili za prehod na Linux, prost Unixu podoben operacijski sistem. Ta spis naj bi bralcu pomagal prestaviti svoje znanje sistemov DOS in Windows v novo okolje, kot tudi ponudil nasvete glede izmenjave datotek in virov med enim in drugim operacijskim sistemom.

Kazalo

1 Uvod

1.1 Je Linux prava stvar za vas?

Začnimo s politično pravovernim uvodom. Čeprav v spisu vseskozi uporabljamo izraz Linux, z njim mislimo operacijski sistem GNU/Linux. Spis na naslovu `<http://www.lugos.si/linux/linux-in-gnu.html>` razloži razliko med enim in drugim.

Bi tudi vi radi prešli z okolja DOS/Windows na Linux? Dobra zamisel: Linux je tehnično superioren glede na DOS, Windows 95 in celo glede na Windows NT. Ampak pazite: če niste iz prave zvrsti uporabnikov, morda za vas ne bo uporaben. V naslednjih točkah bomo našteli razlike med sistemi DOS/Windows in Linux:

- Na DOS in Windows teče Microsoft Office in veliko igric; sistem je enostaven za namestitev in nastavitev; je razvpit po svoji nestabilnosti; ni najhitrejši; pogosto zmrzne.
- Na Linuxu teče StarOffice, precej tehničnega programja, malo igric; namestitev in nastavitev je lahko problematična; je zelo stabilen; je hiter; zmrzne zelo poredko.

Na vas je, da se odločite, kaj potrebujete. Na kratko: Linux ponuja več moči, vendar traja nekaj časa, da se ga naučite uporabljati. Če večinoma uporabljate komercialno programje, ali pa ne želite izgubljati časa z učenjem novih ukazov in konceptov, verjetno ni za vas. Zavedajte se, da mnogi novopečeni uporabniki odnehajo zaradi začetnih težav.

Dela se sicer na tem, da bi bil Linux enostavnejši za uporabo, vseeno pa *ne pričakujte, da ga boste obvladali, ne da bi prebrali veliko dokumentacije in ga vsaj nekaj mesecev uporabljali.* Linux vam ne bo dal takojšnjih rezultatov. Navzlic tem opozorilom pa vam stoodstotno zagotavljam, da ste, če ste le pravi tip uporabnice ali uporabnika, v Linuxu našli svojo računalniško Nirvano, in da ne boste nikoli več hoteli iti nazaj na DOS ali Windows. In mimogrede: Linux ter DOS in Windows lahko mirno sobivajo na istem stroju.

Predpogoji za uporabo tega spisa: predpostavili bomo, da

- poznate osnovne ukaze in pojme okolja DOS
- je Linux, po možnosti skupaj z X Window System, že pravilno nameščen na vašem osebnem računalniku
- je vaša ukazna lupina v Linuxu (ekvivalent `COMMAND.COM`) `bash`.

Če ni poudarjeno drugače, se vse informacije v tem spisu nanašajo na slabi stari DOS. Tu in tam se najde kakšna informacija o Windows, védite pa, da sta Windows in Linux popolnoma različna pojma, za razliko od okolja DOS, ki je, resda uborna, paralela Unixu. Prosim, razumite tudi, da ta spis ni ne celovit uvod v Linux, niti priročnik za njegovo nastavitev.

Najnovejšo izdajo tega spisa najdete na naslovu `<http://www.linuxdoc.org>`.

1.2 Je. Povejte mi več o njem!

Na svoj računalnik ste namestili Linux in vse potrebne programe. Ustvarili ste svoj uporabniški račun (če ga še niste, napravite to *takoj* z ukazom `adduser`), in Linux lepo teče. Ravnokar ste vnesli svoje uporabniško ime in geslo, in zdaj ste se zazrli v zaslon z mislijo: „No, kaj pa zdaj?“

Ne obupujte. Skoraj ste že pripravljeni, da boste počeli vse stvari, ki ste jih počeli v okolju DOS/Windows, in še mnogo več. Pod prejšnjim okoljem ste verjetno počeli naslednje stvari:

- poganjali ste programe; ustvarjali, prepisovali, pregledovali, brisali, tiskali in preimenovali datoteke
- z ukazi CD, MD, RD in DIR ste se premikali po vaših imenikih, ustvarjali nove, jih brisali, in izpisovali njihovo vsebino
- formatirali ste diskete in prepisovali datoteke z njih na disk in nazaj
- okolje ste si prikrojili po svoje
- surfali ste po internetu
- pisali ste datoteke .BAT in programirali v svojem najljubšem programskem jeziku
- preostali 1%

Razveselili se boste novice, da se vsa omenjena opravila izvajajo pod Linuxom na podoben način. V okolju DOS povprečni uporabnik uporablja le kak ducat od nekaj čez 100 obstoječih ukazov, podobno do neke mere velja za Linux.

1.2.1 Uvodni pojmi

Najboljši način, da se naučite česa novega je, da tudi sami skočite v vodo. Eksperimentiranje in igranje z Linuxom je nadvse priporočljivo, saj z njim ne morete poškodovati sistema. Nekaj točk:

- Najprej se naučite, kako varno končate delo z Linuxom. Če ste v tekstovnem načinu, pritisnite <CTRL-ALT-DEL>, počakajte, da se sistem znova zažene, in računalnik ugasnite, še preden začne z diska nalagati operacijski sistem. Če delate v okolju X Window System, pritisnite najprej <CTRL-ALT-BACKSPACE>, da se vrnete v tekstovni način, potem pa še <CTRL-ALT-DEL>. *Nikoli* ne ugašajte računalnika ali pritiskajte tipke Reset sredi dela, to lahko poškoduje vaše podatke na disku.
- Za razliko od okolja DOS/Windows ima Linux vgrajene varnostne mehanizme. Datoteke in imeniki imajo dovoljenje za pisanje in branje, in kot rezultat tega, do nekaterih kot navaden uporabnik ne morete (oglejte si še razdelek 4.3 (Dovolilnice in lastništvo datotekam). DOS ali Windows vam po drugi strani dovoli, da pobrišete celotno vsebino vašega diska.
- Obstaja posebni „superuporabnik“ z imenom „root“. To je upravnik sistema, ki ima dovolj moči, da lahko odloča o življenju in smrti stroja. Če delate na svojem osebnem računalniku, boste sami morali tudi upravljati sistem. Delati kot root je *nevarno*: vsaka napaka lahko resno poškoduje ali celo uniči celoten sistem, enako kot pod DOS/Windows. Zato se prijavite kot root le, če res ne gre drugače.
- V precejšnji meri je za kompleksnost Linuxa kriva njegova izjemna nastavljivost; takorekoč vsako njegovo lastnost in vsak program si lahko prek ene ali večih nastavitvenih datotek prikrojite po lastnih željah in potrebah. Zapletenost je pač cena, ki smo jo plačali za to možnost.
- Preusmeritev in nizanje ukazov sta obstranski strani v DOS, v Linuxu pa zelo pomembni in predvsem neprimerno močnejši. Preproste ukaze lahko nanizate za dosego zapletenih opravil. Močno priporočam, da se naučite ravnanja z njimi.

1.2.2 Kako do pomoči

Mnogo načinov je, kako priti do pomoči, če se z Linuxom kaj zatakne. Najpomembnejši so:

- *Branje navodil*. Čisto resno. Čeprav lahko spis, ki ga berete, služi kot uvod v Linux, ne more nadomestiti branja katere od naslednjih knjig: Matt Welsh „Linux Installation and Getting Started“ (slovenski prevod je dosegljiv na naslovu `<http://www.lugos.si/delo/slo/LIGS-sl/>`), Larry Greenfield „Linux User Guide“ (`<ftp://metalab.unc.edu/pub/Linux/docs/linux-doc-project/users-guide>`), in spisek pogosto zastavljanih vprašanj o Linuxu (`<http://www.lugos.si/delo/slo/HOWTO-sl/Linux-FAQ-sl.html>`). Imejte slabo vest, dokler ne preberete vsaj ene od njih.
- Dokumentacijo za nameščene programske pakete dostikrat najdete v podimenikih imenika `/usr/doc/`.
- Navodila za katerega od internih ukazov ukazne lupine dobite z ukazom `help`. Izčrpnejše informacije dobite z ukazoma `man bash` ali `info bash`.
- Navodila za posamezen ukaz dobite z ukazom `man ukaz`. Ta prikaže stran iz priročnika (angl. manual), ki pripada iskanemu ukazu. Alternativa je ukaz `info ukaz`, ki, če je na voljo, prikliče poglavje `ukaz` iz informacijskega sistema info. Hipertekstni informacijski sistem info prvi hip morda ne deluje najbolj intuitivno. In končno lahko poskusite `apropos ukaz` ali `whatis ukaz`. Iz vseh naštetih dokumentacijskih sistemov se izmotate s pritiskom na tipko ‚q‘.

1.3 Dogovori

V tem spisu bodo zgledi pogosto sledili naslednjemu vzorcu: z <...> označimo obvezni argument ukaza, z `[...]` pa neobveznega. Zgled:

```
$ tar -tf <datoteka.tar> [> preusmerjena_datoteka]
```

`datoteka.tar` mora biti določena, preusmeritev v `preusmerjeno_datoteko` pa je neobvezna.

Koderkoli je kot pozivnik pri zgledu ukaza zapisan znak ‚#‘, to pomeni, da lahko ukaz izvede samo root.

2 Za nestrpne

Vas že srbijo prsti? Oglejte si naslednjo tabelo:

```
DOS                     Linux                   Opombe
------------------------------------------------------------------------------


ATTRIB (+-)attr file    chmod <mode> file       čisto nekaj drugega
BACKUP                  tar -Mcvf device dir/   prav tako
CD dirname\             cd dirname/             skoraj enaka skladnja
COPY file1 file2        cp file1 file2          prav tako
DEL file                rm file                 pazite -- ukaza undelete ni
DELTREE dirname         rm -R dirname/          prav tako
DIR                     ls                      različna skladnja
DIR file /S             find . -name file       povsem različna
```

```
EDIT file            vi file              morda vam ne bo všeč
                     jstar file           ta je tak kot edit v DOS
                     ed file              pozabite nanj
FORMAT               fdformat,
                     mount, umount        precej drugačna skladnja
HELP command         man command,         ista filozofija
                     info command
MD dirname           mkdir dirname/       skoraj enaka skladnja
MOVE file1 file2     mv file1 file2       prav tako
NUL                  /dev/null            prav tako
PRINT file           lpr file             prav tako
PRN                  /dev/lp0,
                     /dev/lp1             prav tako
RD dirname           rmdir dirname/       prav tako
REN file1 file2      mv file1 file2       ne gre za več datotek
RESTORE              tar -Mxpvf device    različna skladnja
TYPE file            less file            bistveno boljši
WIN                  startx               povsem drug svet!
```

Če vam ta tabela ne zadošča, si oglejte naslednje razdelke.

3 Predstavljamo vam Bash

Dobra novica: v Linuxu je tipkanja v ukazni vrstici manj kot v okolju DOS, ker ukazna lupina `bash` marsikaj postori namesto vas, in vsebuje vrsto sijajnih možnosti za urejanje ukazne vrstice. Za začetek: puščica navzgor prikliče prejšnje ukaze. Ampak to ni še nič. Pritisk na tipko <TAB> dopolni imena datotek in imenikov. Če natipkamo

```
$ ls /uTABloTABbTAB
```

je to isto kot

```
$ ls /usr/local/bin
```

Kadar okrajšave niso enolične in so mogoči nesporazumi, kot denimo v zgledu

```
$ ls /uTABloTABiTAB
```

se `bash` ustavi, saj ne ve, ali imamo v mislih imenik `/usr/local/info` ali `/usr/local/include`. V tem primeru dopišite še kako črko, da se izognete nejasnosti, in ponovno pritisnite TAB.

Drugi uporabni triki s tipkami so <ESC-BACKSPACE>, ki pobriše besedo levo od kazalca, in <ESC-D>, ki pobriše besedo desno od njega; s kombinacijo <ESC-F> premaknemo kazalec za besedo v desno (F kot Forward, naprej), z <ESC-B> (B kot Backward, nazaj) pa v levo. <CTRL-A> nas premakne na začetek vrstice, <CTRL-E> pa na konec. Tipka <ALT> je enakovredna tipki <ESC>.

Bodi dovolj za zdaj. Ko se boste enkrat privadili na te bližnjice, se vam bo zdela ukazna lupina v DOS nadvse okorna in nadležna...

4 Datoteke in programi

4.1 Datoteke: uvodni pojmi

Struktura imenikov in datotek na Linuxu je zelo podobna tisti v DOS/Windows. Datoteke imajo imena, za katera veljajo določena pravila, in so shranjena v imenikih. Nekatere od njih so programi, in večina med temi pozna ukazne izbire. Nadalje, uporabljate lahko džokerje, preusmeritve in nizanje ukazov. Obstaja le nekaj manjših razlik:

- V DOS velja za imena datotek omejitev 8+3 (do osem znakov imena, do tri znake pripone), npr. `NOTENOUG.TXT`. V Linuxu so stvari lepše. Če ste namestili Linux z datotečnim sistemom, kot sta na primer ext2 ali umsdos, lahko uporabljate daljša imena datotek (do 255 znakov) in več kot eno piko v imenu, na primer `To_je.zelo_dolgo.IME.datoteke`. Verjetno ste opazili, da sem uporabil velike in male črke, to je zato, ker...
- Linux razlikuje med velikimi in malimi črkami v imenih datotek ali ukazov. Zato sta `DATOTEKA.tar.gz` in `datoteka.tar.gz` dve različni datoteki. `ls` pomeni ukaz; `LS` pomeni napako.
- Posebno opozorilo za uporabnike Windows 95. Če ime datoteke vsebuje presledke (kar ni pametno, je pa načeloma možno), morate pri vsakem sklicu nanj navesti takšno ime znotraj dvojnih narekovajev. Zgled:

  ```
  $ # naslednji ukaz ustvari imenik "Stare datoteke"
  $ mkdir "Stare datoteke"
  $ ls
  Stare datoteke  bin            tmp
  ```

  Nadalje: nekateri znaki niso dovoljeni, med njimi so `!*$&#`.
- Linux ne pozna nobenih obveznih pripon, kot na primer .COM in .EXE za programe ali .BAT za paketne datoteke. Izvedljive datoteke so označene z zvezdico ‚`*`' na koncu, kadar izpišete vsebino imenika z ukazom `ls -F`. Na primer:

  ```
  $ ls -F
  I_am_a_dir/   cindy.jpg    cjpg*   letter_to_Joe    my_1st_script*  old~
  ```

  Datoteki `cjpg*` in `my_1st_script*` sta izvedljivi, torej programa. V DOS imajo varnostne kopije pripono .BAK; v Linuxu se končajo s tildo ‚~'. Še nekaj: če se ime datoteke začne s piko, na primer `.skrita_datoteka`, ga ukaz `ls` ne prikaže.
- Izbire v programih za DOS navadno uvede poševnica (npr. `/izbira`), v programih na Linuxu in Unixu nasploh pa ena ali dve črtici, torej `-izbira` ali `--dolga-izbira`. Ukaz `dir /s`, denimo, tako postane `ls -R`. Verjetno ste že opazili, da tudi mnogi programi za DOS uporabljajo ta stil določanja izbir, med njimi `PKZIP` in `ARJ`.

Zdaj lahko tudi preskočite na razdelek 4.4 (Datoteke: primerjava ukazov), na vašem mestu pa bi bral dalje.

4.2 Simbolne povezave

Unix pozna zvrst datoteke, ki v DOS ne obstaja: simbolne povezave. Lahko si jih zamišljamo kot kazalce na datoteke ali imenike, in jih lahko uporabljamo namesto datotek ali imenikov, na katere kažejo; nekaj podobnega so bližnjice v Windows 95. Primera simbolnih povezav sta imenik `/usr/X11`, ki kaže na `/usr/X11R6/` in enota `/dev/modem`, ki kaže bodisi na `/dev/ttyS0`, bodisi na `/dev/ttyS1`.

Simbolno povezavo napravimo takole:

```
$ ln -s <datoteka_ali_imenik> <ime_povezave>
```

Zgled:

```
$ ln -s /usr/doc/g77/DOC g77manual.txt
```

Zdaj se lahko sklicujemo na `g77manual.txt` namesto na `/usr/doc/g77/DOC`. Simbolnim povezavam se pri izpisu vsebine imenika doda znak ‚`@`' na koncu.

```
$ ls -F
g77manual.txt@
$ ls -l
(druge reči...)           g77manual.txt -> /usr/doc/g77/DOC
```

4.3 Dovolilnice in lastništvo nad datotekami

Datoteke in imeniki v DOS imajo lahko naslednje prilastke: A (arhivski), H (skrit, angl. hidden), R (dovoljen samo za branje, angl. read-only), ali S (sistemski). Samo H in R sta smiselna tudi pod Linuxom: imena skritih datotek in imenikov se začnejo s piko, o prilastku R pa več v nadaljevanju.

V Unixu ima vsaka datoteka svoje dovolilnice (angl. permissions) in lastnika (angl. owner), ki nadalje pripada eni ali večim skupinam uporabnikov (group). Poglejmo si naslednji primer:

```
$ ls -l /bin/ls
-rwxr-xr-x  1  root  bin  27281 Aug 15 1995 /bin/ls*
```

Ime datoteke, `/bin/ls`, je izpisano v zadnjem polju. Prvo polje so dovolilnice za datoteko. Vidimo tudi, da je lastnik datoteke root, pripada pa skupini bin. Ostale informacije za zdaj pustimo ob strani in si oglejmo dovolilnice. `-rwxr-xr-x` pomeni, od leve proti desni:

Prvi `-` je zvrst datoteke (`-` je navadna datoteka, `d` imenik, `l` simbolna povezava in tako dalje). Preostale znake beremo v trojicah. `rwx` so dovolilnice za lastnika. Ta lahko datoteko bere (read), piše oz. briše (write) in izvaja (execute). `r-x` so dovolilnice za uporabnike iz skupine bin. Ti jo lahko berejo (read) in izvajajo (execute), ne morejo pa je spreminjati (pisati ali brisati). Pojma skupine za zdaj ne bomo obravnavali; dokler ste začetnik ali začetnica, lahko preživite tudi brez njega. Zadnji `r-x` so dovolilnice za vse ostale uporabnike. Tudi ti lahko datoteko berejo in izvajajo, ne morejo pa je spreminjati.

Tudi imenik `/bin` ima svoje dovolilnice; za podrobnosti si oglejte 5.2 (Imeniki in dovolilnice). Dovolilnice so razlog, zakaj ne morete pobrisati datoteke `/bin/sl`: razen če ste root, nimate dovoljenja za spreminjanje te datoteke. Dovolilnice za posamezno datoteko lahko spreminjate z ukazom:

```
$ chmod <kdoXdovoljenje> <datoteka>
```

„Kdo" je lahko `u` (uporabnik, torej lastnik), `g` (skupina), ali `o` (ostali). Namesto X stoji bodisi `+` (izdamo dovolilnico) bodisi `-` (prekličemo dovolilnico). In končno, „dovoljenje" je lahko `r` (dovoljenje za branje, angl. read), `w` (dovoljenje za pisanje oz. spreminjanje; angl. write), ali `x` (dovoljenje za izvajanje, angl. execute). Sledi nekaj običajnih primerov uporabe ukaza `chmod`:

```
$ chmod +x datoteka
```

Datoteko smo napravili izvedljivo.

```
$ chmod go-rw datoteka
```

Umaknili smo dovoljenje za branje in pisanje za skupino in vse ostale, datoteko lahko bere in spreminja le še lastnik.

```
$ chmod ugo+rwx datoteko.
```

S takšno dovolilnico lahko vsakdo bere, spreminja ali izvaja datoteko.

```
# chmod +s datoteka
```

Izdali smo dovolilnico, s katero lahko vsakdo izvaja to datoteko z enakimi pooblastili, kot jih ima lastnik te datoteke (običajno datoteke izvajamo z lastnimi pooblastili). To je takoimenovana datoteka vrste „setuid“ ali „suid“. Navadno gre za sistemske datoteke kot npr. strežnik X, katerih lastnik je root, saj lahko le z njegovimi pooblastili dostopamo do nekaterih sistemskih virov.

Na dovolilnice se lahko sklicujemo tudi z osmiško kodo: `rwxr-xr-x` tako na primer izrazimo kot 755 (vsaki črki iz trojice ustreza en bit: `--` je 0, `-x` je 1, `-w-` je 2, `-wx` je 3 itd.). Na prvi pogled morda izgleda težavno, z nekaj prakse pa boste hitro razumeli koncept.

Edino superuporabnik, root, lahko spreminja dovolilnice za katerokoli datoteko na sistemu, ostali pa le svoje. Ve č informacij v priročniku

4.4 Datoteke: primerjava ukazov

Na levi je ukaz DOS, na desni enakovredni ukaz v Linuxu

```
ATTRIB:        chmod
COPY:          cp
DEL:           rm
MOVE:          mv
REN:           mv
TYPE:          more, less, cat
```

Preusmeritev in nizanje ukazov: $< \ > \ >> \ |$

Džokerji: `* ?`

`nul: /dev/null`

`prn, lpt1: /dev/lp0 or /dev/lp1; lpr`

4.4.1 Zgledi

```
DOS                                    Linux
---------------------------------------------------------------------
```

```
C:\GUIDO>ATTRIB +R FILE.TXT           $ chmod 400 file.txt
C:\GUIDO>COPY JOE.TXT JOE.DOC         $ cp joe.txt joe.doc
C:\GUIDO>COPY *.* TOTAL               $ cat * > total
C:\GUIDO>COPY FRACTALS.DOC PRN        $ cp fractals.doc /dev/lp1
C:\GUIDO>DEL TEMP                     $ rm temp
C:\GUIDO>DEL *.BAK                    $ rm *~
C:\GUIDO>MOVE PAPER.TXT TMP\          $ mv paper.txt tmp/
C:\GUIDO>REN PAPER.TXT PAPER.ASC      $ mv paper.txt paper.asc
C:\GUIDO>PRINT LETTER.TXT             $ lpr letter.txt
C:\GUIDO>TYPE LETTER.TXT              $ more letter.txt
C:\GUIDO>TYPE LETTER.TXT              $ less letter.txt
C:\GUIDO>TYPE LETTER.TXT > NUL        $ cat letter.txt > /dev/null
        n/a                           $ more *.txt *.asc
        n/a                           $ cat section*.txt | less
```

Opombe:

- Džoker `*` je na Linuxu pametnejši: `*` ujame vse datoteke razen skritih; `.*` ujame vse skrite datoteke (vendar tudi trenutni imenik ‚`.`' in imenik nad njim, ‚`..`', zato previdno!); `*.*` ujame le tiste datoteke, ki imajo sredi imena piko, ali pa se končajo s piko. `p*r` ujame „parameter" kot „papir"; `*c*` ujame tako „Kitajec" kot „akcija".
- Pri uporabi `more` pritiskajte <SPACE> za naslednjo stran, ‚q' za konec. Ukaz `less` je bolj intuitiven in dovoljuje uporabo kurzorskih tipk.
- Nič takega kot `UNDELETE` ne obstaja, zato *dvakrat premislite*, preden karkoli pobrišete.
- Dodatno k preusmeritvam < > >>, ki jih poznamo iz DOS, dovoljuje Linux tudi prusmeritve oblike `2>`. S slednjo preusmerimo standardni izhod za napake (stderr); konstrukt `2>&1` tako preusmeri standardni izhod za napake (stderr) na standardni izhod (stdout), `1>&2` pa obratno.
- Linux pozna še eno zvrst džokerjev `[]`. Uporaba: `[abc]*` ujame vse datoteke, katerih imena se začno z a, b, c; `*[I-N1-3]` pa vsa imena, ki se končajo na I, J, K, L, M, N, 1, 2, 3;
- Ukaz `lpr` <file> natisne datoteko v ozadju. Če vas zanima, kaj se z njo dogaja in na katerem mestu v vrsti za tiskanje se nahaja, uporabite `lpq`; iz vrste za tiskanje jo umaknemo z ukazom `lprm`;
- Nobenega ukaza ni, ki bi se obnašal kot `RENAME` v DOS; konkretno, `mv *.xxx *.yyy` ne napravi tega, kar si morda želite, da bi. Namesto tega lahko uporabite skript, dostopen na naslovu <ftp://metalab.unc.edu/pub/Linux/utils/file>
- Uporabljajte ukaza `cp -i` and `mv -i`, ki vas opozorita, preden bi kakšna datoteka bila prepisana.

4.5 Poganjanje programov: večopravilnost in seanse

Program poženete tako, da vtipkate njegovo ime, torej enako kot v DOS. Če se imenik (glejte razdelek 5 (Uporaba imenikov)), kjer se program nahaja, nahaja v poti (PATH; razdelek 8.1 (Sistemske inicializacijske datoteke)), boste s tem program pognali. Za razliko od DOS pa Linux ne po pognal programa iz trenutnega imenika, če ta imenik ni naveden v poti. Izogib: iz trenutnega imenika program poženite z ukazom `./program`.

Običajna ukazna vrstica izgleda nekako takole:

```
$ ukaz [-s1 [-s2] ... [-sn]] [par1 [par2] ... [parn]] [< vhod] [> izhod]
```

Pri tem so `-s1` ... `-sn` izbire programa, `par1` ... `parn` pa parametri. V isti vrstici lahko zaporedoma poženete več ukazov, če jih med seboj ločite s podpičji:

```
$ ukaz1 ; ukaz2 ; ... ; ukazn
```

To je pravzaprav vse, kar morate vedeti o poganjanju programov. Čisto enostaven pa je tudi naslednji korak. Eden od glavnih razlogov za uporabo Linuxa je ta, da je večopravilni sistem – več programov (od zdaj bomo programom med tekom rekli procesi) lahko teče hkrati. V ozadju lahko poženete nov proces in nemotemo nadaljujete s svojim delom. Še več: pod Linuxom ste lahko hkrati večkrat prijavljeni – kot bi imeli več računalnikov!

- Med prijavnimi seansami 1-6 na virtualni konzoli lahko preklapljate s pritiskom na tipke <ALT-F1> do <ALT-F6>
- Znotraj prijavne seanso v virtualni konzoli lahko z ukazom `su -` <prijavno_ime> poženete novo prijavno seanso, ne da bi s tem končali tekočo prijavno seanso. Primer `su - root`. Stvar je uporabna na primer pri kratkih opravilih, ki pa jih lahko izvaja samo root.
- Prijavno seanso zaključite z ukazom `exit`. Če imate še kakšne ustavljene procese (več o njih v nadaljevanju), boste na to opozorjeni.
- Proces poženete v ozadju tako, da na koncu ukazne vrstice dodate znak ‚`&`‘:

  ```
  $ program [-izbire] [parametri] [< vhod] [> izhod] &
  [1] 123
  ```

  Ukazna lupina prepoznava procese po številki opravila (npr. `[1]`; več o opravilih v naslednji točki) in po identifikacijski številki procesa (angl. PID, Process Identification Number; v našem zgledu je to 123).
- Seznam trenutno tekočih procesov dobite z ukazom `ps -ax`.
- Proces prekinemo z ukazom `kill` <PID>. K temu se moramo včasih zateči, če ne vemo, kako bi zaključili po običajni poti. Prekinjate lahko samo svoje procese, izjema je root, ki lahko prekine katerikoli proces. Občasno je proces možno prekiniti samo z ukazom `kill -SIGKILL` <PID>.

  Ukazna lupina poleg prekinjanja dovoljuje tudi, da proces ustavimo ali začasno zamrznemo, določimo, da se izvaja v ozadju, ali pa prikličeno iz ozadja nazaj v ospredje. V tem kontekstu pravimo procesom „opravila“.
- Seznam tekočih opravil dobite z ukazom `jobs`. Tu so opravila navedena le s številko opravila, ne s PID.
- Proces ki teče v ospredju ustavimo s pritiskom na <CTRL-C> (ne dela nujno vedno).
- Proces v ospredju zamrznemo s pritiskom na <CTRL-Z> (tudi to de deluje vedno).
- Zamrznjen proces pošljemo v izvajanje v ozadje z ukazom `bg` <job> (s tem postane opravilo).
- Opravilo prikličemo iz ozadja z ukazom `fg` <job>. Brez dodatnega argumenta ukaz `fg` prikliče v ospredje zadnje opravilo, ki smo ga poslali v ozadje.
- Opravilo prekinemo z ukazom `kill` <%opravilo>, pri čemer je <opravilo> številka posla: 1, 2, 3...

S temi ukazi lahko obenem formatirate disk, pospravljate kopico datotek v arhiv, prevajate program in dekomprimirate drugo arhivsko datoteko, pa imate še vedno dostop do pozivnika. Pa poskusite kaj takega v DOS! Poskusite še v Windows (če bodo preživeli), da vidite razliko v hitrosti.

4.6 Poganjanje programov na drugih računalnikih

Programe na drugem računalniku, katerega popolno mrežno ime je `nekje.drugje.si`, lahko poganjamo tako, da se nanj prijavimo s programom telnet:

```
$ telnet nekje.drugje.si
```

Ko ste prijavljeni, preprosto poženite program, ki ga želite. Verjetno je odveč povedati, da morate tudi na tem drugem računalniku imeti odprt uporabniški račun.

V okenskem okolju X11 lahko na drugem računalniku celo poganjate programe, pisane za to okolje, na svojem pa prikazujete rezultate. Naj bo `nekje.drugje.si` oddaljeni računalnik, `moj.linux.si` pa naš mlinček z Linuxom. Postopek spodaj opisuje, kako z računalnika `moj.linux.si` poženemo program, ki se nahaja na drugem računalniku, `nekje.drugje.si`:

- Poženemo X11 in v njih `xterm` ali ekvivalentni terminalski emulator. Potem tipkamo:

  ```
  $ xhost +nekje.drugje.si
  $ telnet nekje.drugje.si
  ```

- Po prijavi na drug računalnik vtipkamo:

  ```
  remote:$ DISPLAY=moj.linux.si:0.0
  remote:$ moj_program &
  ```

  (Odvisno od uporabljane ukazne lupine na drugem računalniku boste morda namesto vrstice `DISPLAY...` morali natipkati `setenv DISPLAY moj.linux.si:0.0`.)

Et voila! Zdaj se bo `moj_program` pognal na `nekje.drugje.si` in uporabljal naš zaslon za prikaz. Ne poskušajte pa tega prek modema – prepočasi gre, da bi bilo zares uporabno. Kar smo povedali, je tudi zelo groba metoda, ki ni povsem v skladu s pogledi na računalniško varnostjo. Več o tem lahko preberete v spisu „Remote X Apps mini-HOWTO“, `<http://www.linuxdoc.org/HOWTO/mini/Remote-X-Apps.html>`.

5 Imeniki

5.1 Imeniki: pripravljalni pojmi

Ogledali smo si že razlike med datotekami pod DOS in pod Linuxom, zdaj pa so na vrsti imeniki. V DOS se vrhnji imenik imenuje `\`, medtem ko je v Linuxu `/`. Podobno je tudi s podimeniki – v DOS so ravni ločene z obrnjeno poševnico (`\`), v Linuxu pa s poševnico (`/`). Primera poti do datotek v enem in drugem primeru:

```
DOS:    C:\PAPERS\GEOLOGY\MID_EOC.TEX
Linux:  /home/guido/papers/geology/middle_eocene.tex
```

Kot običajno `..` označuje imenik nad trenutnim, `.` pa trenutni imenik. Vsak uporabnik ima svoj domači imenik, ki mu ga ustvari upravnik sistema; na mojem domačem računalniku je to na primer `/home/guido`. Spomnite se, da vam sistem ne bo dovolil izvajati ukazov `cd`, `rd`, ali `md` koderkoli bi si želeli.

5.2 Imeniki in dovolilnice

Ne le datoteke, tudi imeniki imajo dovolilnice. Kar smo se v razdelku 4.3 (Lastništvo nad datotekami in dovolilnice) naučili o lastniku, skupini in ostalih uporabnikih, velja tudi za imenike. Pri imeniku `rx` pomeni da lahko napravite `cd` v ta imenik, `w` pa pomeni, da lahko v njem ustvarite novo datoteko, zbrišete obstoje čo datoteko, če njena dovolilnica to dovoljuje, ali zbrišete sam imenik.

Za zgled si oglejmo, kako preprečimo, da bi uporabniki iz drugih skupin vohljali po naših besedilih: `/home/guido/text`:

```
$ chmod o-rwx /home/guido/text
```

5.3 Imeniki: vzporedba ukazov

```
DIR:          ls, find, du
CD:           cd, pwd
MD:           mkdir
RD:           rmdir
DELTREE:      rm -R
MOVE:         mv
```

5.3.1 Zgledi

```
DOS                                     Linux
---------------------------------------------------------------------

C:\GUIDO>DIR                            $ ls
C:\GUIDO>DIR FILE.TXT                   $ ls file.txt
C:\GUIDO>DIR *.H *.C                    $ ls *.h *.c
C:\GUIDO>DIR/P                          $ ls | more
C:\GUIDO>DIR/A                          $ ls -l
C:\GUIDO>DIR *.TMP /S                   $ find / -name "*.tmp"
C:\GUIDO>CD                             $ pwd
        ga ni - glejte opombe           $ cd
        enako                           $ cd ~
        enako                           $ cd ~/temp
C:\GUIDO>CD \OTHER                      $ cd /other
C:\GUIDO>CD ..\TEMP\TRASH               $ cd ../temp/trash
C:\GUIDO>MD NEWPROGS                    $ mkdir newprogs
C:\GUIDO>MOVE PROG ..                   $ mv prog ..
C:\GUIDO>MD \PROGS\TURBO                $ mkdir /progs/turbo
C:\GUIDO>DELTREE TEMP\TRASH             $ rm -R temp/trash
C:\GUIDO>RD NEWPROGS                    $ rmdir newprogs
C:\GUIDO>RD \PROGS\TURBO                $ rmdir /progs/turbo
```

Opombe:

1. Pri uporabi `rmdir` mora biti imenik, ki ga želimo zbrisati, prazen. Če bi radi pobrisali imenik z vsemi podimeniki in datotekami v njih, uporabite ukaz `rm -rf`. Na lastno odgovornost, seveda.

2. Tilda („~‘) je bližnjica za ime vašega domačega imenika. Ukaz `cd` ali `cd ~` vas prestavi naravnost v vaš domač imenik, kjerkoli že ste; ukaz `cd ~/tmp` vas prestavi v imenik `tmp` vašega domačega imenika.
3. Ukaz `cd -` prakliče nazadnje izvedeni `cd`.

6 Disketniki, diski in podobno

Z enotami lahko v Linuxu delamo na dva načina: tako kot dela DOS, ali pa tako, kot dela Unix. Izbira je vaša.

6.1 Delo z enotami po dosovsko

Večina distribucij Linuxa vsebuje paket Mtools, zbirko ukazov, ki so popolnoma enakovredni svojim dvojnikom iz DOS. Od njih se bolj ali manj ločijo le po začetnem ‚m‘ v imenu; torej `mformat`, `mdir`, `mdel`, `mmd` in tako dalje. Ukazi iz paketa podpirajo tudi dolga imena, dovolilnic za datoteke in imenike pa ne. Paket ukrojite po svojih željah z datoteko `/etc/mtools.conf` (vzorec je priložen zbirki), in z njim lahko dostopate do particij z DOS in Windows, CD-ROM, ali enote ZIP.

Neformatirane diskete pa ne morete formatirati z ukazom `mformat`. Pred njim morate kot root prej pognati `fdformat`:

```
# fdformat /dev/fd0H1440
```

Pazite: pri tem načinu do datotek na disketah ne morete dostopati enako kot do datotek na disku; ne morete na primer napraviti `less a:besedilo.txt`. To je slabost opisanega načina dostopanja do enot.

6.2 Delo z enotami v duhu Unixa

Unix upravlja z enotami drugače kot DOS ali Windows. Logičnih enot ne ločuje z oznakami kot A: ali C:, ampak vse enote (diske, diskete, CD-ROM, prek NFS tudi diske na drugih računalnikih) vidimo kot del enotnega datotečnega sistema. Operaciji, s katero dodamo enoto v datotečni sistem, pravimo „priklop“ (angl. mount). Ko enote ne potrebujemo več (npr. ko želimo disketo izvreči iz disketnika), jo moramo „odklopiti“ (angl. unmount).

Fizično formatiranje diska in ustvarjanje datotečnega sistema na njem sta dve različni stvari. Ukaz `FORMAT A:` v DOS napravi oboje, v Linuxu pa sta ukaza ločena. Ukaz za formatiranje smo si že ogledali malo prej, datotečni sistem na disketi pa napravimo takole:

```
# mkfs -t ext2 -c /dev/fd0H1440
```

Namesto `ext2` lahko navedete tudi drugo vrsto datotečnega sistema, npr. `vfat` (priporočeno), `dos`, `minix` in podobno. Ko ste s tem končali, disketo priklopite z ukazom:

```
# mount -t ext2 /dev/fd0 /mnt
```

Če ste v prejšnjem ukazu namesto `ext2` uporabili kako drugo vrsto datotečnega sistema, morate tudi zdaj navesti isto. Zdaj lahko dostopate do datotek na disketi. Vse, kar ste doslej počeli z A: ali B: se zdaj počne z `/mnt`. Primeri:

```
DOS                                     Linux
---------------------------------------------------------------------

C:\GUIDO>DIR A:                         $ ls /mnt
C:\GUIDO>COPY A:*.*                     $ cp /mnt/* .
C:\GUIDO>COPY *.ZIP A:                  $ cp *.zip /mnt
C:\GUIDO>EDIT A:FILE.TXT                $ jstar /mnt/file.txt
C:\GUIDO>A:                             $ cd /mnt
A:\>_                                   /mnt/$ _
```

Ne pozabite, da morate po opravljenem delu *obvezno* odklopiti enoto:

```
# umount /mnt
```

Verjetno ni treba poudarjati, da ukaza `fdformat` in `mkfs` uporabljate samo na neformatiranih disketah, ne pa na tistih, ki ste jih že uporabljali. Če bi radi uporabljali disketno enoto B:, zamenjajte vse `fd0H1440` in `fd0` v prejšnjih zgledih s `fd1H1440` oziroma `fd1`.

Odveč je verjetno tudi poudarjati, da vse, kar velja za disketnike, velja tudi za ostale enote, kot so diski in CD-ROM. Takole priklopimo slednjega:

```
# mount -t iso9660 /dev/cdrom /mnt
```

To je bil „uradni“ način za priklop disketnikov in drugih enot, zdaj pa še zvijača, s katero si lahko delo malce olajšate. Nerodno je namreč, da lahko priklaplja in odklaplja enote samo root; včasih bi bilo prav uporabno, če bi vsaj diskete in CD-ROM lahko priklapljali tudi ostali uporabniki.

- Kot root napravite naslednje (zgled predpostavlja, da je CD-ROM priključen na `/dev/hdb`, torej kot druga enota na vodilu EIDE; prilagodite vaši situaciji):

  ```
  # mkdir /mnt/floppy ; mkdir /mnt/cdrom
  # chmod 777 /mnt/floppy /mnt/cd*
  # # Preverite, da je enota za CD-ROM pravilna!
  # chmod 666 /dev/hdb ; chmod 666 /dev/fd*
  ```

- V datoteko `/etc/fstab` dodamo naslednje tri vrstice:

  ```
  /dev/hdb        /mnt/cdrom  iso9660 ro,user,noauto          0       0
  /dev/fd0        /mnt/floppy vfat    user,noauto             0       0
  ```

Disketo z datotečnim sistemom DOS, disketo z datotečnim sistemom ext2 ali CD-ROM lahko zdaj priklopimo enostavno z ukazi:

```
$ mount /mnt/floppy
$ mount /mnt/cdrom
```

Enoti `/mnt/floppy` in `/mnt/cdrom` lahko zdaj priklaplja in odklaplja vsak uporabnik. Opozoriti pa vas moramo, da je prav to s stališča varnosti vašega sistema tudi slabost.

Dva uporabna ukaza sta še `df`, ki poda informacije o trenutno priklopljenih enotah, ter `du imenik`, ki izpiše porabo prostora za podani imenik.

6.3 Varnostne kopije

Več paketov obstaja, s katerimi si lahko pomagate pri izdelavi varnostnih kopij. Najosnovneši ukaz, s katerim lahko kot root napravite varnostno kopijo imenika na disketo (ali več disket, odvisno od velikosti), pa je:

```
# tar -M -cvf /dev/fd0H1440 arhivirani_imenik/
```

Pazite, da imate ob tem formatirano disketo v disketniku, še več nadaljnih pa pripravljenih. Podatke lahko v celoti restavrirate tako, da vtaknete prvo disketo v disketnik in natipkate:

```
# tar -M -xpvf /dev/fd0H1440
```

7 Kaj pa okna?

Ekvivalent okolju Windows je na Linuxu okensko okolje X Window System, ali na kratko X11. Za razliko od Macintosha ali Windows prvo izhodišče pri zasnovi X11 ni bila enostavnost uporabe ali lep izgled, ampak le nudenje grafičnih možnosti za delovne postaje s sistemom Unix. Glavne razlike med Windows in X11 so naslednje:

- Windows izgledajo bolj ali manj podobno širom sveta, X11 pa ne: je neprimerno bolj ukrojljiv. Izgled okolja X11 določa ključni element, imenovan „upravljalnik oken" (angl. window manager). In izbirate lahko med široko paleto slednjih: `fvwm` – osnoven, a prijeten in učinkovit pri delu s pomnilnikom; `fvwm2-95`, s katerim vaš Linux izgleda kot Windows 95; `Afterstep`, ki mu da izgled okolja NEXTstep, in tako naprej. Upravljalnika oken navadno poženemo v datoteki `\~{}/.xinitrc`.
- Upravljalnika oken lahko nastavimo tako, da se okna obnašajo tako kot v okolju Windows: z miško kliknemo na okno in ga tako prikličemo na površje. Druga možnost pa je, da priplava na površje avtomatično, ko z miško zapeljemo kazalček na okno. Tudi postavljanje novih oken na zaslon gre lahko avtomatično ali pa interaktivno. Slednjega imate, če se ob zagonu programa namesto programa na zaslonu pojavi nekakšen okvir. Z miško pripeljite okvir na željeno mesto in pritisnite levi gumb.
- Večino obnašanja upravljalnika oken lahko ukrojite po svoje s spremembo te ali one nastavitvene datoteke. Preberite navodila za upravljalnika oken, ki ga uporabljate; datoteka se lahko imenuje `.fvwmrc`, `.fvwm2rc95`, `.steprc` in tako dalje. Vzorčno nastavitveno datoteko navadno najdete v `/etc/X11/upravljalnik\_oken/system.upravljalnik\_oken`.
- Programi za X11 so pisani tako, da uporabljajo posebne knjižnice grafičnih elementov (angl. „widget set"). Ker je slednjih več, tudi programi med seboj izgledajo različno. Najosnovnejši nabor grafičnih elementov je Athena, uporabljajo ga `xdvi`, `xman` ali `xcalc`. Drugi uporabljajo Motif (npr. `netscape`), Tcl/Tk, Qt, Gtk, XForms ali kaj četrtega. Nekatere, ne pa vse od teh knjižnic ponujajo približno enak izgled in občutek kot Windows.
- Za občutek maloprej povedano žal ne velja vedno. Na primer, če z miško pobarvate vrstico besedila in pritisnete <BACKSPACE>, verjetno pričakujete, da bo vrstica izginila, kajne? Res bo, če program uporablja knjižnice Motif, Qt, Gtk ali Tcl/Tk, ne pa tudi, če uporablja Atheno.
- Delovanje drsnikov in spreminjanje velikosti oken je tudi odvisno od uporabljenega upravljalnika oken ter knjižnice grafičnih elementov. Namig: če se drsnik ne obnaša tako, kot bi pričakovali, uporabite namesto levega srednji gumb na miški (ali levega in desnega hkrati, če imate miško s samo dvema gumboma).

- Programi nimajo nujno privzete ikone; lahko pa jih imajo mnogo. V večini upravljalnikov oken se pokaže menu, če kliknete na ozadje („korensko okno“, angl. root window). Odveč je govoriti, da se da tudi menu ukrojiti po svoje. Izgled korenskega okna lahko spremenite z ukazoma `xsetroot` ali `xloadimage`.
- V odložišče (angl. clipboard) lahko odlagamo le besedilo, in še to se obnaša neobičajno. Tisti hip, ko ste pobarvali besedilo, je to tudi že v odložišču; premaknite se kamorkoli želite in pritisnite srednji gumb na miški, pa ga boste prenesli na nov kraj. Obstaja tudi program `xclipboard`, ki vas oskrbi z več predalčki za odlaganje.
- „Povleci in spusti“ deluje samo v tistih programih za X11, ki to podpirajo.

Pa vseeno nekaj dobrih novic. Projekta K Desktop Environment (KDE) <http://www.kde.org> in Gnome <http://www.gnome.org> sta si zastavila, da napravita X11 v obnašanju enako koherentne kot Windows. Oba sta videti sijajno, in namizja Windows ne boste več pogrešali!

8 Ukrojitev sistema

8.1 Sistemske inicializacijske datoteke

Dve pomembni datoteki v DOS sta `AUTOEXEC.BAT` in `CONFIG.SYS`, ki se prebereta ob zagonu in inicializirata sistem, nastavita nekaj spremenljivk okolja (npr. PATH in FILES), in po možnosti zaženeta kakšen program ali skripta. Windows 95 in poznejše različice imajo tudi nesrečni register, ki je vir mnogih težav in zapletov.

Pod Linuxom je inicializacijskih datotek mnogo. Nekatere od njih so takšne, da se jih raje ne dotikajte, če ne veste zelo dobro, kaj počenjate. Te navadno najdemo v imeniku `/etc`. Vse nastavitvene datoteke so tekstovne datoteke, ki jih enostavno spreminjamo s katerimkoli urejevalnikom besedil. Če želite samo nastaviti PATH ali katero drugo spremenljivko okolja, spremeniti prijavno sporočilo ali ob zagonu avtomatično zagnati katerega od programov, vam bo prišel prav naslednji spisek datotek:

```
DATOTEKE                               OPOMBE

/etc/issue                             nastavi sporočilo pred prijavo
/etc/motd                              nastavi sporočilo takoj po prijavi
/etc/profile                           nastavi $PATH in druge spremenljivke
/etc/bashrc                            nastavi vzdevke, funkcije ipd.
/home/uporabnik/.bashrc                nastavi uporabniške vzdevke in funkcije
/home/uporabnik/.bash_profile   ali
/home/uporabnik/.profile               nastavi uporabniško okolje, zažene prog.
```

Če slednja datoteka obstaja (opazite, da je skrita datoteka), se bo prebrala in tolmačila po prijavi.

Za zgled si oglejmo ta `.bash_profile`:

```
# To je komentar
echo Spremenljivke okolja:
printenv | less   # podobno kot ukaz SET pod DOS
alias d='ls -l'   # enostaven in razumljiv zgled, kaj so vzdevki
alias up='cd ..'
echo "Ponovimo, pot je "$PATH
```

```
echo "Današnji datum je `date`"  # uporabimo izpis ukaza ,,date``
echo "Lep dan želim, "$LOGNAME
# Sledi funkcija
ctgz() # Izpis vsebine arhivov .tar.gz
{
  for file in $*
  do
    gzip -dc ${file} | tar tf -
  done
}
# Konec .profile
```

Pravilno ste uganili, `$PATH` in `$LOGNAME` sta spremenljivki okolja. Še več zanimivih je, preberite si priročnik za `less` in `bash`.

Naslednja nastavitev v datoteki `/etc/profile` predstavlja približek za nastavitev pozivnika `PROMPT $P$G` v DOS:

```
export PS1="\w\\$ "
```

8.2 Programske inicializacijske datoteke

V Linuxu si lahko takorekoč karkoli ukrojite po svojih potrebah. Večina programov ima eno ali tudi več inicializacijskih datotek, s katerimi lahko eksperimentirate. Navadno se inicializacijski datoteki za program `moj_program` pravi kar `.moj\_programrc` in počiva v domačem imeniku. Prve, ki bodo verjetno prišle na vrsto za spreminjanje, so:

- `.inputrc`: uporablja jo ukazna lupina `bash` za določanje pomena tipk
- `.xinitrc`: prebere jo `startx` pri zagonu X Window System
- `.fvwmrc`: uporablja jo upravljalnik oken `fvwm`
- `.joerc`: uporablja jo urejevalnik `joe`
- `.jedrc`: uporablja jo urejevalnik `jed`
- `.pinerc`: uporablja jo program `pine` za delo z elektronsko pošto
- `.Xdefault`: uporabljajo jo mnogi programi za X11

Za vse naštete in vse ostale, na katere boste naleteli: preberite priročnik. Za konec mi dovolite, da vam priporočim branje spisa Configuration HOWTO. Najdete ga na <http://www.linuxdoc.org/HOWTO/Config-HOWTO.html>.

9 Osnove povezave v omrežja

Ne le, da je klicna povezava (angl. dialup networking) v internet dostopna tudi v Linuxu, ampak je tudi hitrejša in bolj stabilna. Vse o tem najdemo pod geslom „PPP“, ki je ime protokola za povezavo v internet prek modemov. Vse, kar potrebujete, je orodje, ki pokliče vašega ponudnika internetnih storitev (PIS) in vzpostavi povezavo.

Elektronsko pošto prenašate s strežnika svojega ponudnika internetnih storitev z programom, ki mu pravimo prinašalnik epošte (angl. email fetcher). Ta za povezavo s strežnikom PIS uporabi protokol PPP. Ko je pošta prenešena, je enako, kot če bi bila dostavljena neposredno na vaš računalnik. S katerimkoli programom za branje pošte (angl. mail user agent, MUA), npr. `pine`, `mutt`, `elm` itn. lahko zdaj pošto prebiramo, odgovarjamo nanjo ipd.

V Windows se klicna zveza vzpostavi avtomatično, ko poženemo kakšen program za delo z internetom. V Linuxu je obratno: najprej vzpostavimo zvezo, potem pa poženemo program. Program ček `diald` pa omogoča tudi način dela, kot ste ga navajeni. Namestitev in ukrojitev omrežja na klic je bilo svoj čas eno najtežjih opravil, kar pa ne drži več. Več podrobnosti je v „Configuration HOWTO"

Za konec še beseda o „Omrežni soseščini". Vaš računalnik z Linuxom se lahko obnaša enako kot računalniki z Windows NT/9x v krajevnem omrežju Microsoft Windows! Čudežna beseda tu je Samba – ne temperamentni brazilski ples, ampak izvedba protokola SMB za Linux. Več o Sambi je na voljo na naslovu `<http://samba.anu.edu.au/samba>`.

10 Kanček programiranja

10.1 Skripte: datoteke .BAT na steroidih

Če ste, tako kot pisec, uporabljali datoteke .BAT za to, da ste dolge ukazne vrstice nadomestili z bližnjicami, ste medtem verjetno že ugotovili, da gre na Linuxu to z vzdevki v datoteki `.profile`. Če pa so bile vaše datoteke .BAT bolj zapletene, boste gotovo navdušeni nad skriptnim jezikom, ki ga ponuja ukazna lupina. Vsaj tako močan je kot stari dobri QBasic, če ne bolj. Pozna spremenljivke, strukture kot `while`, `for`, `case`, `if..then..else`, in veliko drugih dobrot; lahko ga imamo za dobro alternativo „pravim" programskim jezikom.

Skript – ekvivalent datoteki .BAT v DOS – je enostavna tekstovna datoteka z ukazi, ki jo lahko napišemo s katerimkoli urejevalnikom, shranimo, in napravimo izvedljivo z ukazom `chmod +x <moja_skript>`. Poženemo ga enostavno tako, da vtipkamo njegovo ime.

Majhno opozorilo. S sistemskim urejevalnikom `vi` ima večina novih uporabnikov precejšnje težave. Tukaj ne bomo razlagali, kako se ga uporablja, bralec naj poišče razlago v knjigi „Namestitev in začetek dela z Linuxom" ali pa v priročnikih, ki so dostopni v internetu. Zadovoljiti se boste morali z najnujnejšim:

- Novo besedilo vnesemo tako, da pritisnemo tipko `i` in začnemo z vnašanjem
- Znak pobrišemo tako, da pritisnemo <ESC>, zatem pa `x`
- Urejevalnik zapustimo, ne da bi shranili spremembe, s pritiskom na tipko <ESC>, zatem pa vtipkamo `:q!`
- Shranimo in končamo s pritiskom na <ESC>, zatem vtipkamo `:wq`.

Dober urejevalnik za začetnike je `joe`. Če ga pokličemo z imenom `jstar`, se obnaša enako kot urejevalnik v DOS. `jed` v načinu WordStar ali IDE je še boljši. O tem, kje najdete enega in drugega, se poučite v razdelku 11.5 (Kje najdemo programe).

Pisanje skript v `bash` je tako obsežna tema, da bi sama zahtevala celo knjigo. Tu se ne bomo več spuščali v globine te teme, ampak bomo navrgli samo še en primer skripte, iz katerega se lahko naučite nekaj osnovnih pravil:

```
#!/bin/sh
# zgled.sh
```

```
# To je komentar
# Ne spreminjajte prve vrstice!
echo "Sistem: `uname -a`"       # uporabimo izpis zunanjega ukaza uname
echo "Tej skripti je ime $0"    # vgrajene spremenljivke
echo "Podali ste $# parametrov; ti so: "$*
echo "Prvi parameter je: "$1
echo -n "Vaše ime? " ; read ime
echo Opazite razliko: "hi $ime" # dvojni narekovaji
echo Opazite razliko: 'hi $ime' # enojni narekovaji
DIRS=0 ; FILES=0
for file in `ls .` ; do
  if [ -d ${file} ] ; then      # imenik?
    DIRS=`expr $DIRS + 1`       # DIRS = DIRS + 1
  elif [ -f ${file} ] ; then
    FILES=`expr $FILES + 1`
  fi
  case ${file} in
    *.gif|*jpg) echo "${file}: slika" ;;
    *.txt|*.tex) echo "${file}: besedilo" ;;
    *.c|*.f|*.for) echo "${file}: izvorna koda programa" ;;
    *) echo "${file}: druga datoteka" ;;
  esac
done
echo "Imenikov je ${DIRS}, datotek pa ${FILES}"
ls | grep "ZxY--!!!WKW"
if [ $? != 0 ] ; then           # izhodna koda zadnjega ukaza
  echo "ZxY--!!!WKW not found"
fi
echo "dovolj... 'man bash' vam bo postregel z dodatnimi informacijami."
```

10.2 C za vas

V Unixu je sistemski jezik C, najsi vam je všeč ali ne. Na voljo pa je tudi kopica drugih jezikov: Java, Fortran, Pascal, Lisp, Perl, Awk...

Ker verjamemo, da C že znate, bomo tu le navedli nekaj napotkov za tiste, ki vas je razvadil Turbo C++ ali kaj sorodnega. Prevajalnik za C na Linuxu se imenuje `gcc` in ne vključuje lepot, kot jih njegovi dvojniki v DOS: ne vsebuje integriranega okolja za delo, interaktivne pomoči, integriranega razhroščevalnika in podobnega. Je res samo surovi prevajalnik, ki ga poženemo iz ukazne vrstice (kot prevajalnik pa je resda zelo močan in učinkovit). Običajni programček `hello.c` prevedemo z ukazom:

```
$ gcc hello.c
```

To bo proizvedlo izvedljiv program z imenom `a.out`. Če želimo izvedljiv program poimenovati kako drugače, lahko to navedemo v ukazni vrstici, npr.:

```
$ gcc -o hola hello.c
```

Če bi radi v izvedljiv program povezali še knjižnico, dodamo izbiro -l<ime_knjižnice>. Takole na primer povežemo program s standardno matematično knjižnico:

```
$ gcc -o matemprog matemprog.c -lm
```

(Izbira -l<nekaj> pove prevajalniku gcc, naj v izvedljiv program poveže še knjižnico /usr/lib/lib$<$nekaj$>$.a; tako -lm iz prejšnjega zgleda poveže knjižnico /usr/lib/libm.a).

To je takorekoč vse, kar boste potrebovali za prevajanje kratkih programčkov. Kadar pa pišete dolg program, katerega izvorna koda je razbita v več manjših datotek, si boste precej olajšali delo s programom make. Recimo, da pišete razčlenjevalnik izrazov. Izvorna koda se imenuje parser.c in kliče deklaracije iz glav parser.h in xy.h. Funkcije, definirane v parser.c pa želite uporabiti v drugem programu, recimo calc.c, ki prav tako prebere glavo parser.h. Kakšna zmešnjava! Kaj morate torej napraviti, da prevedete calc.c?

Kar morate napraviti, je napisati takoimenovani Makefile, ki vodi evidenco o odvisnostih med izvornimi ter prevedenimi datotekami. V našem primeru bi bila takšna:

```
# makefile za prevajanje calc.c
# Vnesite <TAB> kjer je označeno

calc: calc.o parser.o
<TAB>gcc -o calc calc.o parser.o -lm
# Program calc je odvisen od dveh prevedenih datotek: calc.o in parser.o

calc.o: calc.c parser.h
<TAB>gcc -c calc.c
# calc.o je odvisen od izvorne datoteke calc.c in glave parser.h

parser.o:  parser.c parser.h xy.h
<TAB>gcc -c parser.c
# parser.o je odvisen od izvorne datoteke parser.c in dveh glav

# To je vse.
```

Shranite to datoteko pod imenom Makefile. Zdaj preprosto napišete make, ki bo prevedel cel paket. Alternativno jo lahko shranite tudi pod drugačnim imenom, denimo calc.mak, le da boste to ime zdaj morali navesti kot parameter: make -f calc.mak. Več o tem se boste naučili ob branju priročnika. Tudi funkcije iz standardne knjižnice imajo svoje strani v priročniku, in sicer v tretjem poglavju. Zgled kaže, kako do navodil za funkcijo printf:

```
$ man 3 printf
```

Napake v programih lahko odpravljate z razhroščevalnikom gdb. Ukaz info gdb vas bo poučil o podrobnostih.

Programerju je na voljo zelo veliko število različnih knjižnic. Med prvimi, ki se jih boste morda ogledali, sta ncurses, ki podpira delo z znakovnim zaslonom, ter svgalib za delo v grafičnem načinu. Mnogi urejevalniki se tudi lahko obnašajo kot integrirano okolje.emacs in jed, na primer, podpirata barvanje sintakse, zamikanje programske kode in podobno. Alternativa pa je paket rhide, ki ga lahko snamete z <ftp://metalab.unc.edu:/pub/Linux/devel/debuggers/>. Je kopija Borlandovega integriranega okolja, in precej verjetno je, da vam bo všeč.

10.3 Programiranje v oknih X

Če ste dovolj pogumni, da se boste lotili programiranja za X11 (ni tako zapleteno, koz se sprva zdi!), obstaja več knjižnic, ki pisanje programov za X11 znatno olajšajo. Glavni mesti, ki ju boste želeli obiskati, sta domača stran knjižnice GTK+, <http://www.gtk.org>, in domača stranj knjižnice Qt, <http://www.troll.no>. Knjižnica GTK+, zasnovana za jezik C, obsega nabor grafičnih elementov, ki je bil prvotno uporabljen za grafični paket GIMP <http://www.gimp.org>, zdaj pa ga uporablja tudi namizje GNOME. Kdeveloper, na drugi strani, temelji na knjižnici Qt, prirejeni za delo s C++, in uporabljeni v namizju KDE. Najverjetneje boste uporabili eno od teh dveh knjižnic.

Nekatera najboljša orodja za slikovno programiranje so Kdevelop za Qt, <http://www.kdevelop.org>, in Glade za GTK+, <http://glade.pn.org>. Več informacij najdete tudi na strani <http://www.free-soft.org/guitool/>.

10.3.1 Programiranje za več strojnih okolij

Kaj ne bi bilo sijajno, če bi lahko pisali programe, ki bi se s prevajalnikom `gcc` brez sprememb prevedli tako v Linuxu kot v Windows? V času tega pisanja obstaja nekaj kolikor toliko stabilnih naborov grafilčnih elementov, ki delujejo v več okoljih. Če kot prioriteti postavimo stabilnost in popolnost, se izbor bolj ali manj zoži na eni samo ime: FLTK, Fast Light Tool Kit <http://www.fltk.org>. Presenetljivo majhen, hiter in stabilen je. Ima tudi delno vizualni sestavljalnik programov, imenovan Fluid.

11 Preostali 1%

Pravzaprav več kot le 1%...

11.1 Poganjanje programov za DOS/Windows

Da, do neke mere lahko v Linuxu poganjamo tudi programe, pisane in prevedene za DOS ali Windows! Dva razmeroma dobra emulatorja obstajata, Dosemu <http://www.dosemu.org/> in WINE <http://www.winehq.com/>. Slednji je iz izdaje v izdajo boljši, in seznam programov, ki tečejo v njem, je vse daljši. Med njimi sta tudi Microsoft Word in Excel.

11.2 Arhiviranje datotek: tar in gzip

Programa `tar` in `gzip` sta standardno uporabljana programa za arhiviranje in stiskanje datotek v Unixu. Prvi služi za izdelavo arhivov – podobno kot `PKZIP` ali `Winzip`, le da arhiva ne stisne. Nov arhiv ustvarimo takole:

```
$ tar -cvf <arhuvska_datoteka.tar> <datoteka> [datoteka...]
```

Iz arhiva izluščimo posamezno datoteko ali več datotek:

```
$ tar -xpvf <arhivska_datoteka.tar> [datoteka...]
```

Vsebino arhiva izpišemo z ukazom:

```
$ tar -tf <arhivska_datoteka.tar> | less
```

Arhiv (ali kakršnokoli drugo datoteko) lahko stisnemo s programoma `compress` ali `gzip`. Drugi je novejši in precej bolj učinkovit pri stiskanju, zato uporabo prvega odsvetujemo.

```
$ compress <datoteka>
$ gzip <datoteka>
```

Rezultat je stisnjena datoteka, ki ima bodisi pripono .Z (`compress`) ali .gz (`gzip`). Nobeden od njiju ni arhivski program in stisneta le po eno datoteko naenkrat. Stisnjene datoteke razpnemo z obratnima ukazoma:

```
$ compress -d <datoteka.Z>
$ gzip -d <datoteka.gz>
```

Več o enem in drugem preberite v priročniku.

Obstajajo tudi programi `unarj`, `zip` in `unzip`. Slednja sta povsem združljiva s parom PKZIP/PKUNZIP. Arhivske datoteke s pripono `.tar.gz` ali `.tgz` so v svetu Unixa enako pogoste kot datoteke .ZIP v svetu DOS. Gre za datoteke, ki so bile najprej arhivirane s programom `tar`, zatem pa še stisnjene s programom `gzip`. Vsebino stisnjene arhivske datoteke si ogledamo tako, da progranu `tar` dodatno podamo še izbiro `-z`:

```
$ tar ztf <datoteka.tar.gz> | less
```

11.3 Nameščanje dodatnih programov

Najprej in predvsem: nameščanje dodatnih programov je delo sistemskega skrbnika. Ve čina programskih paketov za Linux je dostopnih kot arhiv `.tar.gz`. Arhiv navadno vsebuje podimenik z imenom programskega paketa, ta pa vsebuje vse datoteke in eventuelne podimenike. Dodatne programe navadno nameš čamo v imenik `/usr/local`. Arhiv razpakiramo z ukazom:

```
# tar zxf <arhiv.tar.gz>
```

Navodila za namestitev so navadno v datoteki `README` ali `INSTALL`. Dostikrat je programski paket na voljo le kot izvorna koda, kar pomeni, da ga morate najprej prevesti. Navadno gre to enostavno z zaporedjem ukazom `./configure`, `make` in `make install`. Za prevajanje seveda potrebujete prevajalnik `gcc` ali `g++`.

Drugi arhivi morajo biti razpakirani iz korenskega imenika `/`; takšni arhivi so recimo Slackware (`.tgz`). Spet tretji ne vsebujejo podimenika, ampak se datoteke razpakirajo kar v trenutnem imeniku. Vedno je pametno prej z izbiro `-t` izpisati vsebino arhiva, preden ga z izbiro `-x` v resnici razpakiramo.

Distribucije Debian, Red Hat in druge uporabljajo lasten arhivski format; prva `.deb`, ostale dve `.rpm`. Posebno drug format je precej razširjen; takšne pakete namestimo enostavno z ukazom:

```
# rpm -i paket.rpm
```

11.4 Nasveti brez katerih ne gre

Pomikanje nazaj po izpisu: pri izpisovanju na zaslon se dostikrat zgodi, da je tisto zanimivo ravnokar izginilo na vrhu zaslona. Nekaj zadnjih strani izpisa (koliko, je odvisno od grafične kartice) je kljub vsemu še shranjeno, do njih pridete s pritiskom na tipki <SHIFT>+<PAGE UP>.

Vrnitev zaslona v normalno stanje: če ste na zaslon s programoma `more` ali `cat` izpisali kakšno čudno, recimo binarno, datoteko, je zaslon lahko ostal v neuporabnem stanju. Poskusite na slepo natipkati `reset` ali pa naslednji niz znakov: `echo CTRL-V ESC c RETURN`.

Odreži in prilepi: za znakovni zaslon glej spodaj; v X11 kliknite in povlecite miško, da označite (pobarvate) besedilo, potem kazalček prestavite na željeno mesto in pritisnite srednji gumb na miški (ali levega in desnega hkrati, če imate miš z dvema gumboma). Obstaja tudi odlagališče `xclipboard` (kamor lahko odlagamo samo besedilo); naj vas pri njem ne zmede zelo dolg odzivni čas.

Miška in znakovni zaslon: če ste namestili paket `gpm`, gonilnik za miško za znakovni zaslon, lahko z klikom in potegom označite besedilo, s klikom na desni gumb pa ga prenesete na nov kraj. Deluje tudi prek več virtualnih konzol.

Sporočila jedra operacijskega sistema: v datoteko `/var/adm/messages` (ali `/var/log/messages`) se zapisujejo sporočila jedra, vključno z izpisom ob zagonu. Tudi ukaz `dmesg` je uporaben.

11.5 Kje najdemo programe

Če vas skrbi, kje boste našli nadomestila za svoje stare programe z DOS/Windows, vam svetujemo, da se najprej obrnete na glavni arhiv programja za Linux: `<ftp://metalab.unc.edu/pub/Linux>`. Druga imenitna začetna mesta so „Linux Applications and Utilities Page“ na `<http://www.xnet.com/~blatura/linapps.shtml>`, „uradna“ stran Linuxa `<http://www.linux.org>` ter `<http://freshmeat.net>`.

11.6 Sladkorčki

Na Linuxu lahko počenjate cel kup stvari, ki jih je bilo nerodno, težko ali pa sploh nemogoče izvesti pod DOS/Windows. Tole je kratek seznam; le toliko, da dobite okus:

- Ukaz `at` omogoča izvajanje ukazov ob določenem času
- `awk` je preprost, a močan jezik za delo s podatkovnimi datotekami (in ne samo z njimi). Če je `podatki.dat` tabela s podatki, lahko z naslednjim enovrstičnim programčkom izpišete prvo in četrto polje vseh tistih zapisov, ki v drugem polju vsebujejo niz „abc“:

  ```
  $ awk '$2 ~ "abc" {print $1, "\t", $4}' podatki.dat
  ```

- Ukaz `cron` je uporaben za izvajanje periodičnih opravil (ob danem dnevu in/ali uri). Glejte `man 5 crontab` za podrobnosti.
- Ukaz `file <datoteka>` poskusi uganiti zvrst `datoteke` (besedilo, program, arhiv ipd.).
- Ukaz `find` (glejte tudi razdelek 5.3 (Imeniki: vzporedba ukazov)) je eden najmočnejših in najbolj uporabnih. Uporablja se za iskanje datotek, ki zadoščajo navedenim pogojem, obenem pa lahko na njih tudi delujemo. Splošna skladnja ukaza je:

```
$ find <imenik> <izraz>
```

kjer <izraz> vključuje pogoje iskanja ter delovanje na najdene datoteke. Primeri:

```
$ find . -type l -exec ls -l {} \;
```

poišče vse datoteke, ki so simbolične povezave, in izpiše, kam kažejo.

```
$ find / -name "*.old" -ok rm {} \;
```

poišče vse datoteke s pripono `.old` in jih pobriše, pri vsaki pa vpraša za potrditev.

```
$ find . -perm +111
```

poišče vse izvedljive datoteke.

```
$ find . -user root
```

poišče vse datoteke, katerih lastnik je root. Možnosti je še veliko, oglejte si priročnik.

- Ukaz `grep` poišče regularne izraze (seveda tudi enostavne nize) v datotekah. Spodnji primer preišče vse datoteke s pripono `.tex` v trenutnem imeniku in izpiše tiste, v katerih nastopa niz „geološk*“: geološka, geološkega, geološkim...

  ```
  $ grep -l "geološk*" *.tex
  ```

  Izvedba `zgrep` deluje na datotekah, stisnjenih z `gzip`.

- **Regularni izraz** so navidez zapleten, vendar hudirjevo močan način za iskanje po besedilih. Naslednji regularni izraz, `^a[^a-m]X{4,}txt$` poišče vse vrstice, ki zadoščajo naslednjim pogojem: začnejo se z ‚a‘, temu sledi katerikoli znak razen črk z intervala a-m, temu sledijo štiri ali več črk X, in se končajo s „txt“. Regularne izraze poznajo vsi spodobni urejevalniki, `less`, in mnogi drugi programi. `man grep` vam lahko služi kot uvod v regularne izraze.

- Ukaz `script <dnevnik>` piše dnevniško datoteko komunikacije uporabnik-računalnik v navedeno datoteko, dokler ne vtipkamo ukaza `exit`. Uporabno pri odkrivanju napak.

- Ukaz `sudo` selektivno dovoljuje uporabnikom nekatera opravila, ki so sicer dovoljena samo sistemskemu skrbniku, npr. formatiranje in priklapljanje diskov. Preberite priročnik za dodatne informacije.

- Ukaz `uname -a` izpiše osnovne informacije o sistemu.

- Ukaza `zcat` in `zless` sta uporabna za pregledovanje stisnjenih datotek, ne da bi jih zares dekomprimirali. Uporabimo jih lahko tudi pri nizanju ukazov. Zgled:

  ```
  $ zless textfile.gz
  $ zcat textfile.gz | lpr
  ```

- Naslednji ukazi tudi lahko kdaj pridejo prav: `bc, cal, chsh, cmp, cut, fmt, head, hexdump, nl, passwd, printf, sort, split, strings, tac, tail, tee, touch, uniq, w, wall, wc, whereis, write, xargs, znew`. Preberite ustrezne strani priročnika.

11.7 Trening Unixa v DOS

Če verjamete ali ne, tudi za DOS in Windows obstajajo imenitna orodja, ki ponujajo okolje, podobno Unixu. Eno je zbirka DJGPP (`<http://www.delorie.com/djgpp/>`) za DOS, medtem ko je Cygwin (`<http://cygwin.com/>`) nekoliko popolnejši nabor, ki pa deluje samo v okolju Win32. Eden in drugi ponujata enake pripomo čke in razvojna orodja GNU kot Linux, seveda pa ne pri čakujte enake stabilnosti sistema ali hitrosti.

Če bi radi poskusili, približno kakšno je življenje v Linuxu, poskusite DJGPP. Snemite in prenesite naslednje datoteke (v času pisanja tega prevoda je zadnja izdaja 2.03): `djdev203.zip`, `bnu2951.zip`, `bsh204b.zip`, `fil40b.zip`, `find41b.zip`, `grep24b.zip`, `gwk306b.zip`, `lss358b.zip`, `shl112b.zip`. Pakete spremljajo navodila za namestitev, dodatno pomoč pa najdete v novičarski skupini `<news:comp.os.msdos.djgpp>`.

Še posebej uporaba ukazne lupine `bash` je prijetna osvežitev v DOS/Windows. Nastavite jo tako, da popravite priloženo datoteko `BOOT.BAT` tako, da bo primerna za vašo namestitev. Potem prepišite obstoje če datoteke v domači imenik v razdelku DOS z naslednjimi:

```
# to je _bashrc

LS_OPTIONS="-F -s --color=yes"
alias cp='cp -i'
alias d='ls -l'
alias l=less
alias ls="ls $LS_OPTIONS"
alias mv='mv -i'
alias rm='rm -i'
alias u='cd ..'

# to je _bprof
if [ -f ~/_bashrc ]; then
  . ~/_bashrc
fi
PS1='\w\$ '
PS2='> '
CDPATH="$CDPATH:~"
# spremenljivke, ki jih potrebuje less(1)
LESS="-M-Q"                     # dolgi pozivnik, tiho delovanje
LESSEDIT="%E ?lt+%lt. %f"       # urejaj zgornjo vrstico
VISUAL="jed"                    # urejevalnik
LESSCHARSET=latin1              # prikaži ne-ASCII znake
export PS1 PS2 CDPATH LS_OPTIONS LESS LESSEDIT LESSOPEN VISUAL LESSCHARSET
```

11.8 Pripone imen datotek in programi, ki jih uporabljajo

Pri imenih datotek ste morda naleteli na celo kopico razli čnih pripon. Če izpustimo nekatere bolj eksoti čne (datoteke s pisavami in podobno), naslednja tabela podaja, kaj je kaj.

- `1 ... 8`: sestavki referenčnega priročnika (angl. man pages). Uporablja jih program `man`.
- `arj`: arhivi, napravljeni s programom `arj`.

- `dvi`: izhodna datoteka, ki jo izdela TeX (glejte spodaj). Prikažete jo s programom `xdvi`, s programom `dvips` pa jo lahko odtisnete ali pa predelate v format PostScript (pripona `.ps`).
- `gz`: stisnjena datoteka; razpnemo jo z ukazom `gzip -d`.
- `info`: hipertekstna datoteka informacijskega sistema GNU texinfo. Prikažete jo s programom `info`.
- `lsm`: datoteka Linux Software Map. Navadna besedilna datoteka z opisom programskega paketa.
- `ps`: datoteka v formatu PostScript. Prikažemo ali odtisnemo jo s paketom GhostScript (ukaz `gs`) in, neobvezno, `ghostview` ali `gv`.
- `rpm`: paket Red Hat. Namestite ga z namestitvenim programom `rpm`.
- `taz, tar.Z`: arhivska datoteka izdelana s programom `tar` in stisnjena s programom `compress`.
- `tgz, tar.gz`: arhivska datoteka izdelana s programom `tar` in stisnjena s programom `gzip`.
- `tex`: izvorna datoteka za stavni sistem TeX. Namestite paket `tex`, ki je na voljo v mnogih distribucijah. Paket je precej obsežen.
- `texi`: izvorna datoteka za dokumentacijski sistem GNU texinfo, iz katere lahko pridelamo datoteko `.tex` in interaktivno hipertekstno datoteko `.info`. Potrebujete paket `texinfo`.
- `xbm, xpm, xwd`: bitne slike. Prikazujete in popravljate jih lahko s programom `xpaint`.
- `Z`: datoteka, stisnjena s programom `compress`.

11.9 Pretvorba datotek

Pri izmenjavi besedilnih datotek med okolji DOS/Windows in Linux pazite na dejstvo, da sistema zaklju čujeta vrstice na različen način. V DOS se vsaka vrstica zaključi z dvema znakoma, CR/LF (Carriage Return/Line Feed; pomik na začetek vrstice, skok v novo vrstico; kodi ASCII 13 in 10), pod Linuxom pa samo z LF. Datoteke z DOS zato na Linuxu izgledajo okrancljane z znaki ˆM na koncu vsake vrstice; obratno pa datoteke z Linuxa DOS razume kot eno samo kilometrsko vrstico. Orodja kot so `dos2unix` in `unix2dos` prevajajo iz enega formata v drugega.

Za zapis naših znakov se uporablja kar nekaj razli čnik kodnih razporedov. Pod DOS se najve č uporablja CP852 ali stari sedembitni standard JUS I.B1.002, pod Windows CP1250, na Linuxu pa ISO 8859-2 (Latin 2). Med razli čnimi kodnimi nabori lahko pretvarjate z orodjem `recode`.

11.10 Prosti pisarniški paketi

Da tudi ti obstojajo!

StarOffice (<http://www.sun.com/staroffice>) je trenutno najbolj zrel in popoln pisarniški paket. Hitro pa poteka tudi razvoj paketa Koffice (<http://www.koffice.org>). StarOffice je velikanski in počasen, vseeno pa čisto dober. Ponuja nekatere lastnosti, ki jih v Microsoft Office ni. Poleg tega lahko prebira tudi datoteke, ki jih zapišeta Word in Excel, čeravno pretvorba ni vedno popolna.

Drug dostojen urejevalnik je Corel WordPerfect; prosto izdajo je mo č sneti z interneta. Je sploh treba povedati še kaj več? Tu je: <http://www.corel.com>.

12 Konec, za zdaj

Čestitamo! Zdaj imate že nekaj pojma o Unixu in lahko začnete z delom. Zavedajte pa se, da je vaše poznavanje sistema za zdaj še vedno omejeno, in da potrebujete več prakse, preden se boste v Linuxu počutili kot doma. Če pa je vaš namen le, da se naučite katerega od uporabniških programov, s katerimi boste delali, pa je tukaj zapisano za najosnovnejše poznavanje sistema že čisto dovolj.

Prepričan sem, da boste uživali ob Linuxu in se naučili vse več in več o njem – tako kot vsi. Stavim tudi, da se ne boste nikoli več vrnili nazaj v DOS in Windows. Nazadnje upam, da je zapisano jasno povedano, in da je bilo v pomoč vsem trem ali štirim bralcem, ki so se prebili do tod.

12.1 Copyright

Z morebitnimi vprašanji se obrnite na domačo stran projekta LDP (Linux Documentation Project), `<http://www.linuxdoc.org>`.

12.2 Pravni poduk

Ta spis je ponujen takšen, kot je. Veliko napora je bilo vloženega v to, da bi bila zapisana informacija čimbolj ažurna in točna, vendar pa jo uporabljate izključno na lastno odgovornost. Ne pisec in ne prevajalec ne odgovarjata za kakršnokoli škodo, ki bi morebiti sledila iz uporabe tega dela.

Pisec se zahvaljuje Mattu Welshu, avtorju "Linux Installation and Getting Started", Ianu Jacksonu, avtorju "Linux frequently asked questions with answers", Giuseppu Zanettiju, avtorju italijanske knjige "Linux", vsem, ki so mi po elektronski pošti poslali predloge, posebej pa Linusu Torvaldsu in skupini GNU, ki so nam dali Linux.

Odziv – prošnje, predlogi, opljuvanja in podobno – je vedno dobrodošel.

Uživajte Linux, uživajte življenje,

Guido `=8-)`