

Z DOS/Windows na Linux

Guido Gonzato, guido@ibogfs.cineca.it

v1.3.0, 15. april 1998

Ta HOWTO je posvečen vsem bodočim bivšim uporabnikom sistemov DOS in Windows, ki so se odločili za prehod na Linux, prost klon Unixa. 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	2
1.1	Je Linux prava stvar za vas?	2
1.2	Je. Povejte mi več o njem!	3
1.2.1	Uvodni pojmi	3
1.2.2	Kako do pomoči	4
1.3	Dogovori	4
2	Za nestrpne	4
3	Datoteke in programi	5
3.1	Datoteke: uvodni pojmi	5
3.2	Simbolne povezave	6
3.3	Dovolilnice in lastništvo nad datotekami	6
3.4	Datoteke: primerjava ukazov	7
3.4.1	Primeri	8
3.5	Poganjanje programov: večopravnost in seanse	9
3.6	Poganjanje programov na drugih računalnikih	10
4	Imeniki	11
4.1	Imeniki: pripravljalni pojmi	11
4.2	Imeniki in dovolilnice	11
4.3	Imeniki: vzporedba ukazov	11
4.3.1	Primeri	11
5	Disketniki, diski in podobno	12
5.1	Delo z enotami po dosovsko	12
5.2	Delo z enotami v duhu Unixa	12
5.3	Varnostne kopije	14
6	Kaj pa okna?	14

7	Ukrojitev sistema	15
7.1	Sistemske inicializacijske datoteke	15
7.2	Programske inicializacijske datoteke	16
8	Kanček programiranja	16
8.1	Skripte: datoteke .BAT na steroidih	16
8.2	C za vas	17
9	Preostali 1%	19
9.1	Arhiviranje datotek: tar in gzip	19
9.2	Nameščanje dodatnih programov	20
9.3	Nasveti brez katerih ne gre	20
9.4	Kje najdemo programe	20
9.5	Sladkorčki	21
9.6	Pripone imen datotek in programi, ki jih uporabljajo	22
9.7	Pretvorba datotek	22
10	Konec, zaenkrat	23
10.1	Copyright	23
10.2	Pravni poduk	23

1 Uvod

1.1 Je Linux prava stvar za vas?

Bi tudi vi radi prešli z okolja DOS/Windows na Linux? Dobra ideja: 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 resnici se DOS in Windows uporabljata večinoma za igranje igrice in pisarniška opravila, medtem ko Linux blesti pri omrežnih povezavah, razvoju programja in pisanju znanstvenih programov. Linux je izjemno zmogljiv, vendar pa navadno zahteva nekaj časa, preden se naučite krotiti to moč. Če torej večinoma uporabljate računalnik za pisarniška dela, ali če nimate pravega veselja do učenja novih ukazov in pojmov, potem Linux verjetno ni za vas.

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 instantnih 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 ta spis: predpostavljam, da

- poznate osnovne ukaze in pojme okolja DOS
- je Linux, po možnosti skupaj z X Window System, že pravilno nameščen na vašem osebem računalniku
- je vaša ukazna lupina (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 nastavitvev.

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. Ravno kar ste vnesli svoje uporabniško ime in geslo, in zdaj ste se zazrli v zaslon z mislijo: „No, kaj pa zdaj?“

Ne obupajte. Skoraj ste že pripravljeni, da boste počeli vse stvari, ki ste jih počeli pod 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
- 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 3.3 (Dovolilnice in lastništvo nad datotekami). DOS vam na drugi strani dovoli, da pobrišete celotno vsebino vašega diska.
- Obstaja posebni „superuporabnik“ z imenom „root“. To je upravitelj 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 nastavljalivost; 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 doseg 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.* Resno mislim. 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“ (<http://metalab.unc.edu/mdw/LDP/gs/gs.html>, slovenski prevod je v pripravi), Larry Greenfield „Linux User Guide“ (<<ftp://metalab.unc.edu/pub/Linux/docs/linux-doc-project/users-guide>>), in spisek pogosto zastavljanj 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`. Izčrpnjše 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. Primer:

```
$ 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 <code>undelete</code> ni
DELTREE dirname	rm -R dirname/	prav tako
DIR	ls	različna skladnja

DIR file /S	find . -name file	povsem različna
EDIT file	vi file	ne vem, če vam bo všeč
	jstar file	ta je tak kot edit v DOS
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	kot severni in južni tečaj!

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

3 Datoteke in programi

3.1 Datoteke: uvodni pojmi

Struktura imenikov in datotek na Linuxu je zelo podobna tisti na DOS/Windows. Datoteke imajo imena, za katera veljajo določena pravila, in so shranjena v imenikih. Nekatere od njih so programi, in večina med njimi 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. Pod Linuxom 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 !*\$&.

- Nobenih obveznih pripon kot na primer .COM in .EXE za programe ali .BAT za paketne datoteke ni. Izvodljive 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 izvodljivi, torej programa. V DOS imajo varnostne kopije pripono `.BAK`; v Linuxu se končajo s tilde, `~'`. 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 nasplošno pa ena ali dve črtici, torej `-izbira` ali `-dolga-izbira`. Ukaz `dir /s`, na primer, 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 3.4 (Datoteke: primerjava ukazov), na vašem mestu pa bi bral dalje.

3.2 Simbolne povezave

Unix pozna zvrst datoteke, ki na 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; podobno kot bližnjice na Windows 95. Primera simbolnih povezav sta imenik `/usr/X11`, ki kaže na `/usr/X11R6/` in enota `/dev/modem`, ki kaže bodisi na `/dev/cua0` ali `/dev/cua1`.

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
```

3.3 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.

Pod Unixom ima vsaka datoteka 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 zaenkrat 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 zaenkrat 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 4.2 (Imeniki in dovolilnice). Dovolilnice so razlog, zakaj ne morete pobrisati datoteke /bin/sl, razen če niste 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) ali - (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 izvodljivo.

```
$ 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+rxw 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 zgleda težavno, z nekaj prakse pa boste hitro razumeli koncept.

Edino superuporabnik, root, lahko spreminja dovolilnice za katerokoli datoteko na sistemu, ostali pa le svoje. RMP.

3.4 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

3.4.1 Primeri

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 spodnjo skripto, oglejte si razdelek 8.1 (Skripte ukazne lupine: datoteke .BAT na steroidih) za podrobnosti.

```
#!/bin/sh
```

```
# ren: preimenuj več dat
```



```

if [ $# -lt 3 ] ; then
    echo "usage: ren \"pattern\" \"replacement\" files..."
    exit 1
fi

OLD=$1 ; NEW=$2 ; shift ; shift

for file in $*
do
    new=`echo ${file} | sed s/${OLD}/${NEW}/g`
    mv ${file} $new
done

```

Opozorilo: skripta se ne obnaša enako kot REN v DOS, ampak uporablja tkim. regularne izraze, o katerih še ni tekla beseda. Na kratko: če bi s to skripto radi spremenili pripone več datotek naenkrat, to storite kot v tem zgledu: `ren "htm$" "html" *.htm`. Ne pozabite na znak \$.

- Uporabljajte ukaza `cp -i` and `mv -i`, ki vas opozorita, preden bi kakšna datoteka bila prepisana.

3.5 Poganjanje programov: večopravilnost in seanse

Program poženete tako, da vtipkate njegovo ime, torej enako kot v DOS. Če se imenik (glejte razdelek 4 (Uporaba imenikov)), kjer se program nahaja, nahaja v poti (PATH; razdelek 7.1 (Sistemske inicializacijske datoteke)), boste s tem program pognali. Za razliko od DOS pa Linux ne po poglaj 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 `<ALT-F1> ... <ALT-F6>`
- Znotraj prijavnne seanse 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 prepozna 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 prikličemo 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.

3.6 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 odprto 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.

4 Imeniki

4.1 Imeniki: pripravljalni pojmi

Ogledali smo si že razlike med datotekami pod DOS in pod Linuxom, zdaj pa so na vrsti imeniki. Pod DOS je vrhnji imenik \, medtem ko je pod Linuxom /. Podobno je tudi s podimeniki – pod DOS so nivoji ločeni z obrnjeno poševnico (\), pod Linuxom 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 upravitelj 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.

4.2 Imeniki in dovolilnice

Ne le datoteke, tudi imeniki imajo dovolilnice. Kar smo se v razdelku 3.3 (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 obstoječo 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
```

4.3 Imeniki: vzporedba ukazov

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

4.3.1 Primeri

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 -R`. Na lastno odgovornost.
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 -` prikliče nazadnje izvedeni `cd`.

5 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.

5.1 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`, `mddir`, `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.

5.2 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. `minix`, `vfat` ali `dos`. 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
```

Jasno je, 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/a: ; mkdir /mnt/a ; mkdir /mnt/cdrom
# chmod 777 /mnt/a* /mnt/cd*
# 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/a:	msdos	user,noauto	0	0
/dev/fd0	/mnt/a	ext2	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/a:
$ mount /mnt/a
$ mount /mnt/cdrom
```

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

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

5.3 Varnostne kopije

Več paketov obstaja, s katerimi si lahko pomagata 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 nadaljnjih pa pripravljenih. Podatke lahko v celoti restavrirate tako, da vtaknete prvo disketo v disketnik in natipkate:

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

6 Kaj pa okna?

Ekvivalent okolju Windows je na Linuxu okensko okolje X Window System, ali na kratko X11. Za razliko od Macintosh 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 priključimo 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 ukrožite 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*, *XForms*, *Qt*, *Gtk* ali kaj četrtega. Nekateri, 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 srednjega gumba 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 ukrožiti po svoje. Izgled korenkega 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.

Na disku bi prihranili nekaj prostora, če bi uporabljali programe, ki si delijo isto knjižnico grafičnih elementov. V praksi je to zaenkrat težko izvedljivo.

Projekta K Desktop Environment (KDE) in Gnome sta si zastavila, da napravita X11 v obnašanju enako koherentne kot Windows. Oba sta zaenkrat še v testni fazi, kar je videti, pa obljublja veliko. Prvega najdete na <http://www.kde.org/>, drugega na <http://www.gnome.org/>.

7 Ukrojitev sistema

7.1 Sistemske inicializacijske datoteke

Dve pomembni datoteki pod 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 zažene kakšen program ali skripta. Pod Linuxom je inicializacijskih datotek več. Nekatere od njih so takšne, da se jih raje ne dotikajte, če ne veste zelo dobro, kaj počenjate. Tole je vseeno spisek najpomembnejših:

DATOTEKE	OPOMBE
/etc/inittab	zaenkrat se ne dotikajte!
/etc/rc.d/*	enako

Če je vse, kar potrebujete nastavitev \$PATH ali kakšne druge spremenljivke okolja, če bi radi zamenjali prijavno sporočilo ali ob prijavi avtomatično zagnali kak program, si oglejte naslednje datoteke:

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 primer 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.

7.2 Programske inicializacijske datoteke

Pod Linuxom si lahko takorekoč karkoli ukrožite 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 našteje in vse ostale, na katere boste naleteli: preberite priročnik. Za konec mi dovolite, da vam priporočim branje spisa Configuration HOWTO. Najdete ga na <http://metalab.unc.edu/mdw/HOWTO/Config-HOWTO.html>.

8 Kanček programiranja

8.1 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 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.

Skripta – 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_skripta>`. 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, ker ga pisec tega spisa ne mara. 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 9.4 (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
# zglied.sh
# To je komentar
# Ne spreminjajte prve vrstice!
echo "Sistem: `uname -a`"      # uporabimo izpis zunanlega 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."
```

8.2 C za vas

Na Unixih 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 izvodljiv program z imenom `a.out`. Če želimo izvodljiv program poimenovati kako drugače, lahko to navedemo v ukazni vrstici, npr.:

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

Če bi radi v izvodljiv 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<neka>` pove prevajalniku gcc, naj v izvodljiv program poveže še knjižnico `/usr/lib/lib$<neka>` `$.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 `svglib` za delo v grafičnem načinu. Če ste dovolj pogumni, da se boste lotili programiranja za X11, obstaja že omenjene knjižnice `XForms`, `Qt`, `Gtk` in mnoge druge, ki pisanje programov za X11 znatno olajšajo. Oglejte si na primer še <http://www.xnet.com/~blatura/linapp6.html>.

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.

9 Preostali 1%

Pravzaprav več kot le 1%...

9.1 Arhiviranje datotek: tar in gzip

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

```
$ tar -cvf <arhivska_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 `unrarj`, `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 programu `tar` dodatno podamo še izbiro `-z`:

```
$ tar -ztzf <datoteka.tar.gz> | less
```

9.2 Nameščanje dodatnih programov

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

```
# tar -zxvf <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. Dostikrat gre to enostavno z zaporedjem ukazov `./configure, make in make install`. 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.

Paketi za distribucije Debian, Red Hat in Caldera uporabljajo lasten arhivski format; prva `.deb`, drugi dve `.rpm`. Posebno drug format je precej razširjen; takšne pakete namestimo enostavno z ukazom:

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

9.3 Nasveti brez katerih ne gre

Avtomatična dopolnitev ukazov: pritisk na tipko `<TAB>` pri tipkanju v ukazni lupini avtomatično dopolni vrstico. Na primer, če prevajate program, `gcc precej_dolgo_ime_datoteke.c`, je dovolj, če natipkate začetek imena in pritisnete `<TAB>`: `gcc prec<TAB>` (začetek mora biti dovolj dolg, da enolično določa datoteko; v nasprotnem primeru bo ukazna lupina prikazala vse datoteke, ki ustrezajo pogoju).

Pomikanje nazaj po izpisu: pri izpisovanju na zaslon se dostikrat zgodi, da je tisto zanimivo ravno kar 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.

9.4 Kje najdemo programe

Če vas skrbi, kje boste našli nadomestila za svoje stare programe z DOS/Windows, vam svetujemo, da se najprej obrnete na enega od glavnih arhivov programja za Linux: `<ftp://metalab.unc.edu/pub/Linux>`, `<ftp://tsx-11.mit.edu/pub/linux>`, ali `<ftp://ftp.funet.fi/pub/Linux>` (vsi so zrcaljeni širom sveta, tako

da poiščite najbližje zrcalo). Drug imeniten kraj je stran „Linux Applications and Utilities Page“ na <http://www.xnet.com/~blatura/linapps.shtml>.

9.5 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 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 4.3 (Imeniki: vzporedba ukazov)) je eden najmočnejših in najuporabnejših. 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 izvodljive datoteke.

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

poišče vse datoteke, katerih lastnik je `root`. Možnosti je še veliko, glejte 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
```

- **Regularni izraz** so navidez zapleteni, vendar hudirjevo močan način za iskanje po besedilih. Naslednji regularni izraz, `^[^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 priklopjanje diskov. Preberite priročnik za dodatne informacije.
- Ukaz `uname -a` izpiše osnovne informacije o sistemu.
- 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.

9.6 Pripone imen datotek in programi, ki jih uporabljajo

Pri imenih datotek ste morda naleteli na celo kopico različnih pripon. Če izpustimo nekatere bolj eksotične (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. Nekaj časa je krožila okrog distribucija NTeX s pokvarjenimi pisavami, pazite nanjo.
- `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`.

9.7 Pretvorba datotek

Pri izmenjavi besedilnih datotek med okolji DOS/Windows in Linux pazite na dejstvo, da sistema zaključujeta vrstice na različen način. V DOS se vsaka vrstica zaključí z dvema znakoma, CR/LF (Carriage Return/Line Feed; pomik na začetek vrstice, skok v novo vrstico), pod Linuxom pa samo z LF. Datoteke z DOS zato na Linuxu izgledajo okranceljane z znaki `^M` 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 različnih kodnih razporedov. Pod DOS se največ uporablja CP852 ali stari sedembitni standard JUS I.B1.002, pod Windows CP1250, na Linuxu pa ISO 8859-2 (Latin 2). Med različnimi kodnimi nabori lahko pretvarjate z orodjem recode.

Pretvorba iz formatov Word ali WordPerfect v navadno besedilo je malenkost zahtevnejša, je pa možna. Potrebujete eno od orodij, ki jih najdete na primer na `<ftp://ftp.tex.ac.uk>`. Poskusite s paketom word2x iz imenika `/pub/tex/tools/`, ali pa katerega od paketov iz imenika `/pub/tex/support/`.

10 Konec, zaenkrat

Čestitamo! Zdaj imate že nekaj pojma o Unixu in lahko začnete z delom. Zavedajte pa se, da je vaše poznavanje sistema zaenkrat 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 na 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.

10.1 Copyright

Če ni drugače povedano, so spisi Linux HOWTO pravno zaščiteni s strani svojih avtorjev. Spisi Linux HOWTO se smejo razmnoževati in razširjati v delu ali v celoti, v tiskani ali elektronski obliki, pod pogojem, da je to sporočilo v vseh kopijah ohranjeno v nespremenjeni obliki. Redistribuiranje s pridobitnimi nameni je dovoljeno in vzpodbujano, vendar pa avtor želi biti obveščen o takšnih distribucijah.

Vsi prevodi, izpeljana dela ali zbirke, ki vključujejo katerikoli spis Linux HOWTO se morajo razširjati pod enakimi pogoji. Iz spisov HOWTO tako ne morete izvesti izpeljanega dela in ga razširjati pod pogoji, ki so bolj omejujoči od navedenih. Izjemo pod določenimi pogoji lahko dovoli urednik projekta Linux HOWTO; naslov je naveden nižje v besedilu.

Na kratko, pomagati hočemo, da se ta informacija razširi skozi čimveč kanalov, vendar pa se ne odrekamo svojim avtorskim pravicam, in želimo biti obveščeni o načrtih za razširjanje spisov HOWTO.

Če imate vprašanja, prosim stopite v stik s Timom Bynumom, urednikom spisov Linux HOWTO. Prek e-pošte je dosegljiv na naslovu `linux-howto@metalab.unc.edu`.

10.2 Pravni poduk

“From DOS to Linux HOWTO” je napisal Guido Gonzato, `guido@ibogfs.cineca.it`. Prevod v slovenščino z nekaj priredbami je delo Primoža Peterlina, `primoz.peterlin@biofiz.mf.uni-lj.si`. 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.

Ta spis je ponujen takšen, kot je. Veliko napora je bilo vloženega v to, da bi bila zapisana informacija čim točnejša, 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.

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

Uživajte Linux in življenje,

Guido =8-)