

SRCE

SVEUČILIŠNI RAČUNSKI CENTAR
Zagreb

Uvod u
operacijski sustav UNIX

Pripremio: Dobriša Dobrenić

Copyright (C)
Sveučilišni računski centar, Zagreb
Studen 1994., Verzija 1.0

UVOD U OPERACIJSKI SUSTAV UNIX

Sadržaj:

OSNOVNI POJMOVI	1
KRATKA POVIJEST OPERACIJSKOG SUSTAVA UNIX.....	1
OPERACIJSKI SUSTAV UNIX	1
ULAZ I IZLAZ IZ UNIX SUSTAVA.....	1
DOKUMENTACIJA OS UNIX.....	2
POSTAVLJANJE PARAMETARA TIPKOVNICE.....	2
SUSTAV DATOTEKA OS UNIX	3
SUSTAV DATOTEKA.....	3
HIJERARHIJSKA STRUKTURA SUSTAVA DATOTEKA.....	3
RADNI DIREKTORIJ.....	3
PREGLED DATOTEKA	3
TIP DATOTEKE	4
ISPIS SADRŽAJA TE STVARANJE DATOTEKA.....	5
MIJENJANJE RADNOG DIREKTORIJA	7
STVARANJE DIREKTORIJA	7
KOPIRANJE DATOTEKA.....	8
POVEZIVANJE DATOTEKA.....	9
PREMJEŠTANJE I PREIMENOVANJE DATOTEKA.....	9
BRISANJE DATOTEKA I DIREKTORIJA.....	10
PRAVA PRISTUPA DATOTEKAMA.....	11
ULAZ I IZLAZ NAREDBI	13
PREUSMJERAVANJE ULAZA I IZLAZA NAREDBI.....	13
FILTERI.....	14
ULANČAVANJE (<i>pipelining</i>).....	17
UREĐIVAČI TEKSTA	18
VI EDITOR (<i>visual editor</i>).....	18
KOMUNIKACIJA S DRUGIM KORISNICIMA	20
ELEKTRONIČKA POŠTA (<i>E-mail</i>).....	20
IZRAVNA KOMUNIKACIJA S DRUGIM KORISNICIMA	21
PROCESI U OS UNIX	22
STRUKTURA OPERACIJSKOG SUSTAVA UNIX.....	22
LJUSKA	22
PODACI NA VANJSKIM JEDINICAMA	27
PROGRAM tar (<i>Tape archive utility</i>).....	27
KOMPRESIRANE DATOTEKE	28
PISAČI (<i>printers</i>)	29
<i>Dodatak:</i>	
PRIJENOS DATOTEKA S UNIX FORMATOM ZAPISA NA MSDOS (PCDOS) FORMAT I OBRATNO.....	30

OSNOVNI POJMOVI

KRATKA POVIJEST OPERACIJSKOG SUSTAVA UNIX

Odnos OS UNIX i OS U*X (U**X)

UNIX 1969., prva verzija, Ken Thompson, Bell Laboratories
 1973., Version 6, prvi puta raspoloživ izvan Bellovih laboratorija
 1978., BSD UNIX, Berkeley Software Distribution
 1984., System V, Bell Laboratories

ULTRIX (DEC), **HP-UX** (HP), **SunOS** (Sun), **Irix** (SGI), **AIX** (IBM),
Linux (PC), ...

OPERACIJSKI SUSTAV UNIX

OS UNIX omogućuje istovremeno odvijanje više procesa, pokrenutih od više korisnika (*multiuser, time-sharing system*).

Počiva na hijerarhijskom sustavu datoteka koji omogućuje zaštitu procesa i podataka na nekoliko razina. Jedna od značajnijih karakteristika **OS UNIX** je dohvat svih uređaja kroz datoteke - princip **sve je datoteka**.

Napomena: **OS UNIX** razlikuje velika i mala slova!

ULAZ I IZLAZ IZ UNIX SUSTAVA

Ulaz (*logging in, login*):

- korisnička oznaka (*login, username*)
- korisnička lozinka (*password*)
 (mijenjanje naredbom `'passwd'`)
- oznaka javljanja sustava (*prompt*)
 (*primjer: '>'*)
- korisnički naredbeni interpreter - ljuska sustava (*shell*)
 neke ljuske: `sh, csh, sh5, ksh, tcsh, bash, ...`
 (početna ljuska mijenja se naredbom `'chsh'`)

Izlaz (*logging out, logout*):

- naredba `'exit'`

DOKUMENTACIJA OS UNIX

- dokumentacija na samom sustavu:

- naredba 'man' (*manual pages*)

primjer:

```
> man passwd          # opis naredbe passwd  
...
```

- naredba 'apropos' (isto i 'man -k')

primjer:

```
> apropos password   # podaci koje sustav ima o pojmu password  
...
```

POSTAVLJANJE PARAMETARA TIPKOVNICE

- naredba 'stty' (*set TTY*)

primjer:

```
> stty all           # podaci o parametrima tipkovnice  
...  
> stty everything   # svi podaci o parametrima tipkovnice  
...  
> stty erase '^f'   # postavi parametar erase na CTRL/f  
> stty all          # uočiti promjenu parametra erase  
...  
> stty erase 'undef' # parametar erase neka je nedefiniran  
> stty all          # što je s parametrom erase?  
...  
> stty erase `tipka` # postavi erase na pritisnutu tipku  
> stty all          # postavljen  
...
```

SUSTAV DATOTEKA OS UNIX

SUSTAV DATOTEKA

- princip **sve je datoteka**
- ime datoteke može biti do 255 karaktera, bilo kojih

HIJERARHIJSKA STRUKTURA SUSTAVA DATOTEKA

- obične datoteke
- specijalne datoteke (počinju sa točkom - '.')
- direktoriji - osnovni direktorij zove se *root*

RADNI DIREKTORIJ

- naredba 'pwd' (*print working directory*)

primjer:

```
> pwd # ispis radnog direktorija
/usr/users/student
```

PREGLED DATOTEKA

- naredba 'ls' (*list*)
- opcije: -a, -l, -g, -d, -F, -R, ...
- parametri: ništa, ili jedna, ili više datoteka

- pojam apsolutnog i relativnog puta do datoteke:
apsolutni put uvijek počinje od *root* direktorija;
relativni put počinje od dir. u kom se trenutno nalazimo (radni, tekući);
'.' je radni direktorij,
'..' je roditeljski direktorij radnog direktorija.

- zamjenski znakovi za datoteke:
'~' zamjenjuje matični direktorij ('~pero' zamjenjuje perin matični direktorij)
'*' zamjenjuje nula ili više znakova
'?' zamjenjuje točno jedan znak
'[' , ']' za selekciju znakova

primjer:

```
> ls -alF          # pregled svih datoteka radnog dir. u širem obliku
...
> ls -alF .        # isto
...
> ls -alF ../student
                    # pregled svih dat. dir. ../student u širem obliku
...
> ls -alF /usr/users/student
                    # pregled svih dat. direktorija /usr/users/student ...
...
> ls -alF ~student # pregled svih dat. matičnog dir. korisnika student
...
> ls -alF ~        # pregled svih datoteka mog matičnog dir. ...
...
> ls -alF ../../../../bin/c??*[p-t] # selektivni ispis ...
...
> ls -alF /bin/c??*[p-t]             # ispis, drugi put ... (isti rezultat)
...
> ls -alF ~root/bin/c??*[p-t]       # ispis, treći put ... (isto)
...
```

TIP DATOTEKE

- naredba 'file'

primjer:

```
> file * .*        # ispiši tipove svih datoteka radnog direktorija
...
```

ISPIS SADRŽAJA TE STVARANJE DATOTEKA

- naredba 'cat' (*concatenate*)

- 1) ispisivanje / upisivanje sadržaja
- 2) datoteka = 1. kopij
- 3) spajanje dviju dat. u treću
- 4) dodavanje na kraj datoteke
- 5) kopiranje

primjer:

```
> ls -al .login      # ispis podataka o dat. .login u širem obliku
...
1) > cat .login      # ispis sadržaja dat. .login
...
2) > cat > prva.datoteka.txt    # upis sadržaja u dat. prva.datoteka.txt
ovo je moja prva datoteka
student
^D
> ls -al            # postoji li sad i dat. prva.datoteka.txt?
...
> cat prva.datoteka.txt    # ispis sadržaja dat. prva.datoteka.txt
ovo je moja prva datoteka
student
3) > cat >> prva.datoteka.txt    # dodavanje na kraj datoteke
prva.datoteka.txt
dodatak: uspjelo je!
^D
> cat prva.datoteka.txt    # provjera je li zaista dodan tekst
ovo je moja prva datoteka
student
dodatak: uspjelo je!
3) > cat prva.datoteka.txt .login > zajedno
# spajanje dviju dat. u treću
> cat zajedno
# što je sadržaj dat. zajedno?
...
4) > cat zajedno zajedno >> zajedno.dva
# spajanje dviju dat. i dodavanje u treću
> cat zajedno.dva
# sadržaj dat. zajedno.dva
...
5) > cat zajedno > zajedno.jos_jednom
# ispiši dat. zajedno u dat. zajedno.jos_jednom
> cat zajedno.jos_jednom    # kao kopiranje?
...
```

```
> cat > zajedno.jos_jednom      # upis u postojeću dat.
sto ce sada biti?
^D
> cat zajedno.jos_jednom        # a što je sa starim sadržajem?
...
> set noclobber
      # varijabla koja brine o prepisivanju sadržaja (csh)
> cat > zajedno.jos_jednom      # pokušaj upisa u postojeću dat.
...
> cat >! zajedno.jos_jednom     # nasilni upis u postojeću dat. (csh)
sad mogu prebrisati...
^D
> cat zajedno.jos_jednom        # provjera novog sadržaja
...
> cat > "+ ovo nije poželjno "   # zašto?
bilo sto
^D
```

- naredbe 'head' (početne linije datoteke) i
 'tail' (završne linije datoteke)

primjer:

```
> head zajedno.dva              # ispis prvih 10 linija dat.
...
> head -8 zajedno.dva          # ispis prvih 8 linija dat.
...
> tail zajedno.dva             # ispis posljednjih 10 linija dat.
...
> tail -8 zajedno.dva          # ispis posljednjih 8 linija dat.
...
> tail +8 zajedno.dva          # preskoči prvih 8 linija i ispiši do kraja
...
```

- naredba 'more' (tipka 'h' ili '?' za popis naredbi unutar naredbe 'more')

(kao i cat služi za ispis teksta na zaslonu terminala)

*primjer: samo što ispisuje završni po završetku i - završava se /
 SPACE BAR - (tako) obično*

```
> more zajedno.dva              # more je moćna naredba
...
> more "+ ovo nije poželjno "   # zašto se more bunji?
...
> more ./"+ ovo nije poželjno " # ovo je već bolje (česti trik)
...
```


MIJENJANJE RADNOG DIREKTORIJA

- naredba 'cd' ili 'chdir' (*change directory*)

primjer:

```
> pwd                # ispis radnog direktorija
/usr/users/student
> cd bin             # promijeni radni direktorij u bin
> pwd                # koji je sad radni direktorij?
/usr/users/student/bin
> cd .               # promijeni radni dir. u ovaj u kome se nalaziš!
> pwd                # ništa se nije promijenilo
/usr/users/student/bin
> cd ..             # promijeni radni direktorij u roditeljski
> pwd                # !
/usr/users/student
> cd /bin            # promijeni radni direktorij u /bin
> pwd                # koji je to bin?
/bin
> cd                 # promijeni radni direktorij u moj matični dir.
> pwd                # gdje sam?
/usr/users/student
> cd ~/bin           # promijeni radni direktorij u ~/bin
> pwd                # moj bin
/usr/users/student/bin
> cd /usr/users/student
                        # promijeni radni direktorij u /usr/users/student
> pwd                # doma je ipak najljepše
/usr/users/student
```

STVARANJE DIREKTORIJA

- naredba 'mkdir' (*make directory*)

primjer:

```
> cd                 # promijeni radni dir. u matični
> mkdir radni        # stvori dir. radni
> ls -al             # je li tu?
...
> ls -al radni       # što je u njemu?
...
> ls -ald radni      # koji su njegovi parametri?
...
```

```
> mkdir /usr/users/student/radni/primjeri # stvori dir. ...
> ls -alR # pregled svih dat. u i ispod radnog dir.
...
> mkdir smece/primjeri # stvori dir. smece/primjeri (zašto neće?)
...
> mkdir -p smece/primjeri # a sada?
> ls -alR # tako je
...
```

KOPIRANJE DATOTEKA

- naredba 'cp' (*copy*)
- opcije: *-i, -r, ...*

primjer: *sigurnosno kopiranje (*)*

```
> cd # idi doma
> cp prva.datoteka.txt prva # kopiraj prva.datoteka.txt u dat. prva
> ls -l pr* # sve dat. koje počinju sa pr
...
* { > cp zajedno prva # kopiraj dat. zajedno u dat. prva
  > cat zajedno prva # što je sa starim sadržajem?
...
  > cp -i zajedno.dva prva # sigurnosno kopiranje
  ... ("n")
  > ls -l z* p* # sve dat. koje počinju sa z ili p
...
  > cp prva smece # kopiraj dat. prva u dir. smece
  > ls -l smece # sadržaj dir. smece, postoji li prva?
...
  > cp zajedno.dva smece/novo_ime # kopiraj dat. zajedno.dva u dat. smece/novo_ime
  > ls -l smece # sad ima novo ime
...
  > cp -r smece smece.staro # kopiraj dir. smece u dir. smece.staro
  > ls -lR # ispis svih dat.; usporedi sadržaje kopiranih dir.
...
  > cp -r smece radni/primjeri # kopiraj dir. smece u dir. radni/primjeri
  > ls -lR # što smo u stvari uradili?
...
```

POVEZIVANJE DATOTEKA

str 63 UNIX)

- naredba 'ln' (*link*)
- opcije: -i, -f, -s, ...

osnovna ideja povezivanja datoteka (link)
 - postojeće jednog sadržaja - više i
 različiti nazivi ili polazivača na taj
 sadržaj
 čvrsta veza (hard link): (stvaranje dv)

primjer:

više → simbolična veza (soft link): (stvaranje ln -s)

```
> cd # idi doma
> ln smece/prva pr.1 # poveži postojeću dat. smece/prva sa pr.1
> ls -l pr.1 smece/prva # usporedi parametre povezanih dat.
...
> cat pr.1 smece/prva # usporedi sadržaje
...
> ln smece/novo_ime pr.1 # dat. pr.1 već postoji!
...
> ln -i smece/novo_ime pr.1 # a ovako?
... ("n")
> ls -l pr.1 # sve je u redu
...
> ln -s smece/prva pr.2 # napravi meku vezu od dat. smece/prva na dat. pr.2
> ls -lF pr* # usporedi parametre meko povezanih dat.
...
```

PREMJEŠTANJE I PREIMENOVANJE DATOTEKA

- naredba 'mv' (*move*)
- opcije: -i, ...

primjer:

```
> cd # idi doma
> mv prva ime # preimenuj dat. prva u ime
> ls -l # provjera
...
> mv ime smece # premjesti dat. ime u dir. smece
> ls -l smece # pod kojim imenom se nalazi u dir. smece?
...
> mv -i smece/ime zajedno # za slučaj da već postoji ciljna dat.
... ("n")
> ls -l # sve je u redu
...
> mv smece smece.novo # preimenuj dir. smece u smece.novo
> ls -lR sme* # sadržaj dir. je ostao isti
...
```

PRAVA PRISTUPA DATOTEKAMA

primjer:

```
> ls -al
...
```

1) ruzičnastije
2) crno bijelo
3) stonje = dopuštanje

- jedan znak za tip datoteke ("- ", *plain file*; "d", *directory*; "l", *link*; ...)
- tri znaka za prava pristupa vlasnika datoteke (*user*)
- tri znaka za prava pristupa grupe kojoj pripada vlasnik datoteke (*group*)
- tri znaka za prava pristupa ostalih korisnika (*world*)

- "r" pravo čitanja (*read*)
- "w" pravo pisanja (*write*)
- "x" pravo izvođenja (*execute*)

Promjena prava pristupa

- naredba 'chmod' (*change mode*)

primjer:

```
> ls -al zajedno.dva      # parametri dat. zajedno.dva
...
> chmod u-r zajedno.dva  # samom sebi ne dopuštam čitanje
> ls -al zajedno.dva     # tako je
...
> cat zajedno.dva        # a tko mi je kriv?
...
> chmod u=r zajedno.dva  # dopuštam si samo čitanje i ništa više
> ls -al zajedno.dva     # u redu
...
> cat >> zajedno.dva     # ne mogu upisivati ...
...
> zajedno.dva            # ... ni izvoditi
> chmod u+x zajedno.dva  # dodajem si pravo izvođenja
> ls -al zajedno.dva     # to sam htio
...
> zajedno.dva            # sad mogu izvesti (iako dat. nije program)
...
> chmod -r zajedno.dva   # oduzmi svima pravo čitanja
> ls -al zajedno.dva     # nitko ne može pročitati ovu dat.
...
> chmod ug+r zajedno.dva # dodaj meni i grupi pravo čitanja
> ls -al zajedno.dva     # u redu
...
```

```

> chmod =rw zajedno.dva # stavi svima samo pravo čitanja i pisanja
> ls -al zajedno.dva    # svi imaju samo rw
...
> chmod = zajedno.dva  # što to radim?
> ls -al zajedno.dva   # eto ...
...
> chmod =r,u+w zajedno.dva # svima samo r, a meni još i w
> ls -al zajedno.dva    # provjera
...
> chmod u-rw,g+rw,o=rx zajedno.dva # oduzimam, dodajem, ...
> ls -al zajedno.dva    # to je to
...
> chmod 755 zajedno.dva # može se i ovako - kraće i nerazumljivije
> ls -al zajedno.dva    # rezultat isti kao i =rx,u+w
...

```

- osmični (*oktalni*) brojevni sustav

```

rwx r-x r-x
111 101 101
 7  5  5

```

```

> chmod 500 zajedno.dva # kao i =,u=rx ili u=rx,go= ili ...
> ls -al zajedno.dva    # dobro
...
> chmod 644 zajedno.dva # isto kao =r,u+w
> ls -al zajedno.dva    # treba se samo priviknuti ...
...
> chmod 111 bin         # svima samo x?
> ls -ald bin           # dir. bin zaštićen od čitanja i pisanja

```

ULAZ I IZLAZ NAREDBI

Vrlo moćna mogućnost OS UNIX je preusmjeravanje ulaza i izlaza naredbi u datoteke (time i na sve vanjske jedinice), kao i međusobno spajanje ulaza i izlaza više naredbi.

Razlikujemo:

- standardni ulaz (*stdin*; 0)
- standardni izlaz za rezultate (*stdout*; 1)
- standardni izlaz za pogreske (*stderr*; 2)

PREUSMJERAVANJE ULAZA I IZLAZA NAREDBI

- simboli za preusmjeravanje izlaza za rezultate (*stdout*):
 - '>' (upis u stvorenu datoteku),
 - '>>' (dodavanje na kraj postojeće datoteke)
- simboli za preusmjeravanje izlaza za rezultate i pogreške (*stdout* i *stderr*):
 - '2>&1' (*sh*, *ksh*, ...) (*stderr* je sad kopija *stdout*);
 - '>&' (*csh*) (upis u stvorenu datoteku),
 - '>>&' (*csh*) (dodavanje na kraj postojeće datoteke)
- simbol za preusmjeravanje ulaza (*stdin*):
 - '<' (čitanje iz postojeće datoteke)
- simboli za nasilni upis (ako je postavljena varijabla 'noclobber'):
 - '>!' i '>>!' odnosno '>&!' i '>>&!' (*csh*)
 - '>|' i '>>|' (*ksh*)

primjer:

```
> bc # procesor aritmetičkog programskog jezika ...
2*3+4-7/6
9
^D
> bc -l # uključi i biblioteku za decimalno računanje
2*3+4-7/6
8.2333333333333333333333333333334
^D
> cat > za_bc # u dat. za_bc upiši podatke koje će koristiti bc
2*3+4-7/6
^D
> bc < za_bc # ulazne podatke uzmi iz dat. za_bc
9
```

```
> bc -l < za_bc      # s decimalnim računanjem je ipak točnije ...
8.8333333333333333333333333334
> cat >> za_bc      # dodajem još neke izraze
365*24
^D
> cat za_bc         # kako sad izgleda dat. za_bc?
2*3+4-7/6
365*24
> bc -l < za_bc     # jos jedno računanje s podacima iz dat. za_bc
8.8333333333333333333333333334
8760
```

FILTERI

Programi koji čitaju podatke sa standardnog ulaza i ispisuju rezultat na standardni izlaz.

primjer:

```
> cat               # i cat je filter
...
> cat < za_bc      # važe sva preusmjerenja
...
```

- 'wc' (*word count*) filter
- opcije: -l, -w, -c

primjer:

```
> wc               # brojanje linija, riječi i znakova
prva linija
druga
treca
^D
          3          4          24
> wc zajedno.dva za_bc # postoji i krajnji izvještaj
...
> wc -l * .*      # korisno
...
```

- 'sort' filter
- opcije: -f, -n, -r, -u, -o, +pos, -pos, ...

primjer:

```
> cat > telefoni          # dat. koju ćemo sortirati
Pero Grdic      123-565
Goran Gubis    145-465
marko kovac    25-654
Petar Krajnik  123-565
Zlatko Krajnik 123-562
goran Gubis    145-465
^D
> sort telefoni          # sortiraj dat. telefoni
...
> sort -f telefoni      # sortiranje ne razlikuje velika i mala slova
...
> sort -r telefoni      # reverzno sortiranje
...
> sort -u telefoni      # izbacijednake linije
...
> sort -fu telefoni     # izbacij linije koje su iste po ključusortiranja
...
> sort +1 telefoni     # ključsu polja od broja 1 do kraja linije
...
> sort +1 -2 telefoni   # ključsu polja od broja 1 do broja 2
...
> sort +2 telefoni      # ključsu polja od broja 2 do kraja linije
...
> sort +2 -n telefoni > tel_po_redu
                        # numeričkosortiranje, rezultat u dat.
...
> sort +2 -nu -o telefoni telefoni
                        # rezultat u dat. koja se upravo sortira
...
```


- 'grep' (*global regular expression printer*) filter
- opcije: -v, -w, -i, -n, -c, ...
- karakteri sa specijalnim značenjem: [] . * \$ | ^ \ ...

primjer:

```

> fgrep g tel_po_redu      # izdvoji linije koje u sebi sadrže znak g
...
> fgrep -i g tel_po_redu   # ne razlikuj velika i mala slova
...
> fgrep -v g tel_po_redu   # linije koje ne odgovaraju traženom
...
> grep -i '^g' tel_po_redu # znak g (ili G) na početku
...
> grep '5$' tel_po_redu    # linije u kojima je znak 5 na kraju
...
> grep -in '^p..o' tel_po_redu
                        # obrojči linije sa zadanim uzorkom
...
> grep '^[A-KZ]' tel_po_redu
                        # linije koje počinju sa slovima od A do K ili Z
...
> grep '^[^A-K]' tel_po_redu
                        # linije koje ne počinju sa slovima od A do K
...
> grep -v '^[A-K]' tel_po_redu      # isti rezultat
...
> egrep 'kov|gor' tel_po_redu      # linije sa uzorcima kov ili gor
...
> grep nar                          # izdvoji linije sa uzorkom nar
naramak
nar
mornar
^D
> grep -w nar                        # linije koje sadrže riječ nar
naramak
nar
mor.nar
^D
> grep 'nar*k'                      # uzorak ima specijalni znak *
naramak
nark
narrrrk
^D
> grep 'nar.*k'                    # uzorak ima specijalne znakove . i *
nar
naramak
^D

```

ULANČAVANJE (*pipelining*)

Ulančavanje označava međusobno povezivanje standardnih ulaza i izlaza više naredbi - simbol za ulančavanje je '|' (odnosno '|&' za *stdout* i *stderr*).

primjer:

```
> ls -al | wc                # broji rezultat naredbe ls
...
> ls -al | sort > moje_dat   # spremi u dat. sortiran rezultat naredbe ls
...
> grep -i a tel_po_redu | grep -i k | sort -u +1
                           # više ulančavanja
...
> sort zajedno.dva | more    # spriječimo da rezultat ne protrči
...
```

UREĐIVAČI TEKSTA

VI EDITOR (*visual editor*)

Način rada *vi editora* određuje smisao pojedinih operacija.

- načini rada:

naredbeni način (*command mode*),

način unosa teksta (*text-input mode*),

način posljednje linije (*last line mode*).

- prijelaz iz načina unosa teksta u naredbeni način *escape* (*ESC*) tipkom (CTRL/[)

primjer:

```
> vi nesto          # poziv programa vi s dat. nesto
...
```

naredbeni način, odnosno način posljednje linije

- micanje unutar teksta:

gore (*up*) - 'k' ili CTRL/P

dolje (*down*) - 'j' ili CTRL/N

lijevo (*left*) - 'h' ili CTRL/H ili "*backspace*"

desno (*right*) - 'l' ili "*razmaknica*"

na početak sljedećeg reda - RETURN

na kraj tekućeg reda - '\$'

na početak tekućeg reda - '^'

stranica (ekran) ispred - CTRL/B (CTRL/D)

stranica (ekran) dalje - CTRL/F (CTRL/U)

kraj teksta - 'G'

n-ta linija - 'nG'

u liniju sa uzorkom *uzo*, tražeći naprijed - '/uzo'

u liniju sa uzorkom *uzo*, tražeći unatrag - '?uzo'

...

- unos teksta:

umetanje teksta ispred značke - 'i'

dodavanje teksta iza značke - 'a'

umetanje teksta na početak linije - 'I'

dodavanje teksta na kraj linije - 'A'

umetanje prazne linije ispod tekuće linije - 'o'

umetanje prazne linije iznad tekuće linije - 'O'

- brisanje teksta:

brisanje znaka - 'x' ('5x')

brisanje riječi - 'dw' ('5dw')

brisanje linije - 'dd' ('5dd')

umetanje izbrisane linije ispod tekuće - 'p'

umetanje izbrisane linije iznad tekuće - 'P'

...

- korekcija teksta:

"osvježavanje" ekrana - CTRL/L

poništanje posljednje promjene u tekućoj liniji - 'u'

poništanje svih promjena tekuće linije - 'U'

prijepis pojedinog znaka - 'r'

prijepis više znakova - 'R'

...

- operacije s datotekama tekstova

izlaz sa upitom o snimanju teksta - ':q'

izlaz bez snimanja teksta - ':q!'

izlaz sa snimanjem teksta - ':wq' ili 'ZZ'

izlaz sa snimanjem teksta ako je bilo promjena - ':x'

ponovno učitavanje tekuće datoteke - ':e!'

učitavanje datoteke *dat* - ':e dat'

dodavanje datoteke *dat* na kraj tekuće datoteke - ':r dat'

...

- izvođenje naredbi ljuske UNIX-a

pregled datoteka radnog direktorija - ':! ls'

...

KOMUNIKACIJA S DRUGIM KORISNICIMA

ELEKTRONIČKA POŠTA (*E-mail*)

Elektronička pošta omogućuje komunikaciju među korisnicima računalnih mreža, neovisno o tipu operacijskog sustava koji koriste na svojim računalima.

- naredba 'mail'

primjer:

```
> mail pero@srce.hr      # slanje pošte na adresu pero@srce.hr
...
> cat > poruka          # upis teksta u dat. poruka
Zdravo Pero.
^D
> mail pero@srce.hr < poruka
                        # šaljem dat. poruka na adresu pero@srce.hr
```

- opcije unutar linijskog uređivača teksta (*escape functions*)

prikaz mogućih opcija: '~?'

pozivanje uređivača teksta: '~v'

prosljeđivanje postojeće poruke pod brojem 2: '~f 2'

uključivanje datoteke *dat* unutar poruke: '~r *dat*'

snimanje poruke u datoteku *dat* : '~w *dat*'

...

- primanje poruka

u datoteku *.login* upisati
set mail=/usr/spool/mail/pero
za sinkrono obavještanje.

u datoteku *.login* upisati
biff y
ako želimo asinkrono obavještanje o pristiglim porukama.

pozvati program *mail*.

- naredbe unutar programa *mail*
- pokazivanje mogućih naredbi: 'help' ili '?'
- pokazivanje zaglavlja svih poruka: 'h'
- čitanje treće poruke: '3'
- slanje poruke korisniku: 'm pero@srce.hr'
- odgovaranje na pročitano poruku: 'R'
- dodavanje treće poruke na kraj datoteke *dat*: 'co 3 dat"
- brisanje treće poruke: 'd 3'
- pokazivanje svih *mail* varijabli: 'set'
- izlazak iz programa *mail* sa spremanjem promjena: 'q'
- izlazak iz programa *mail* bez spremanja promjena: 'x'
- ...

IZRAVNA KOMUNIKACIJA S DRUGIM KORISNICIMA

- naredba 'write' (korisnici na istom sustavu)
- naredba 'talk' (korisnici više *U*X* sustava)

primjer:

```
> write root          # povezivanje s korisnikom root na našem sustavu
...
> talk root@srcapp.srce.hr
                        # povezivanje s korisnikom root@srcapp.srce.hr
...
```

PROCESI U OS UNIX

STRUKTURA OPERACIJSKOG SUSTAVA UNIX

- ljuska (*shell*)
- korisnički programi (*utility programs*)
- jezgra (*kernel*)

LJUSKA

Ljuska sustava je naredbeni interpreter preko kojega korisnik komunicira s korisničkim programima odnosno jezgrom operacijskog sustava.

Neke ljuske:

- sh, csh, sh5, ksh, tcsh, bash, ...

Mijenjanje ljuske koja se postavlja nakon ulaska u sustav: naredba 'chsh'

Specijalne datoteke:

- .login, .profile, .cshrc, .kshrc, .logout, ...

Grupiranje naredbi:

- razdvajanje naredbi sa znakom ';'
- produžavanje naredbe u sljedeći red s kombinacijom '\'[RETURN]'
- određivanje redoslijeda izvođenja s oblim zagradama '(', ')'
- uvjetno izvođenje naredbi:
 - '&&' (*AND*) - u slučaju uspjeha, i
 - '||' (*OR*) - u slučaju neuspjeha prve naredbe.

Vremenska kontrola naredbi:

- izvođenje naredbi s odgodom: 'at' naredba
- čekanje: 'sleep' naredba

Korisne naredbe...:

- 'w', 'who', 'who am i', 'whoami', 'users', 'finger' ('f'),
'which', 'whereis', 'clear', 'date', 'echo'

primjer:

```
> f ivan@srcapp.srce.hr # tko je ivan u sustavu srcapp.srce.hr
...
> f @gandalf.zel.etf.hr # tko sve trenutno radi na gandalf.zel.etf.hr
...
> echo Dobar dan # ispis na ekran (stdout)
...
```

Popis procesa sustava:

- naredba 'ps' (*process status*)
- opcije: -l, -x, -a, -u, ...

primjer:

```
> ps -axu # popis svih procesa sustava u širem obliku
...
```

Procesi u prednjem planu i pozadini:

- zaustavljanje procesa sa ^Z
- stavljanje procesa u pozadinu: 'naredba &'
- naredba 'kill' za uništavanje procesa

manipulacija procesima u C ljusci (csh):

- naredba 'jobs' za popis procesa u pozadini
- naredba 'fg' (*foreground*) za nastavak zaustavljenog procesa - u prednjem planu
- naredba 'bg' (*background*) za nastavak zaustavljenog procesa - u pozadini
- naredba 'stop' za zaustavljanje procesa u pozadini
- naredba 'notify' za obavještanje o završetku rada procesa u pozadini

primjer:

```
> ls -alR /usr > usr_svedat
# popis dat. u dir. /usr spremi u dat. usr_svedat
> more usr_svedat # pregled rezultata
...
> ls -alR /usr > usr_svedat # ista naredba - zaustavljamo s ^Z
^Z
...
> jobs # popis svih procesa (sa stanjem) u pozadini
...
> bg %1 # zaustavljeni proces br. 1 nastavi u pozadini
...
> jobs # proces br. 1 sad je u stanju izvođenja
...
> stop %1 # zaustavi proces br. 1
...
> fg %1 # nastavi proces br. 1 u prednjem planu
^Z
...
> jobs # u kom je stanju moj proces?
...
> kill -l # popis mogućih signala naredbe kill
...
> kill %1 # uništavanje procesa br. 1
...
> jobs # nema ga više
...
> ls -alR /usr > usr_svedat &
# proces se može odmah pokrenuti u pozadini
...
> jobs -l # procesi u pozadini i njihov PID broj
...
> ps # moji procesi u sustavu s PID brojem
...
> kill -9 1403 # sigurno uništavanje procesa s brojem 1403
...
> (ls -alR /usr > usr_svedat) >& usr_error &
# kombinirani primjer ...
...
> more usr_svedat # rezultat: sve dat. u dir. /usr
...
> more usr_error # rezultat: sve pogreške pri izvođenju naredbe
...
> (ls -alR > moje_dat) >& /dev/null &
# uobičajen način gubljenja stderr
...
```

Pokretanje više ljuski

Svaka ljuska pokreće se "iznad" prethodne.

Izlaskom iz neke ljuske vraća se u prethodnu, a ako je to početna ljuska izlazi se iz sustava.

primjer:

```
> ps                # popis mojih procesa
...
> csh               # pokreni csh ljusku iznad tekuće
...
> ps               # što je u popisu procesa?
...
> sh               # na csh pokreni sh
...
$ ps               # sad je tu i sh
...
$ ^D               # povratak iz sh ljuske u csh ljusku
...
> ps               # nema više procesa sh
...
> cd /             # promijeni radni dir. u /
> pwd              # provjera
...
> exit             # povratak u početnu ljusku
> ps               # samo je jedna ljuska
...
> pwd              # gdje sam? kako to?
...
```

Neke naredbe unutar C ljuske (csh):

'set', 'setenv', 'alias', 'history', ...

```
> history          # popis mojih prijašnjih naredbi
...
> !!              # ponavljanje posljednje naredbe
...
> !7              # ponovno izvođenje 7. naredbe iz popisa
...
> !mo             # izvedi naredbu koja je posljednja počinjala sa mo
...
> alias hi history # uvođenje zamjene hi za history
> alias           # popis svih zamjena
...
> hi              # zamjena hi radi kao i history
...
```

Programi koji koriste naredbe ljuske (*shell scripts*)

primjer (csh):

```
> csh # prelazimo u csh
> set # koje su varijable sustava?
...
> cd # radit ću u matičnom dir.
> cat > jutro # stvaram dat. jutro
#! /bin/csh -f
#
clear
echo "Dobro jutro `whoami`\"
echo ""
set sat=`date | cut -c12-19`
echo "Sada je točno $sat sati."
echo ""
#
set history=100
alias hi 'history | tail -20'
alias re 'tail -r \!^ | more'
alias li 'ls -alF \!* | more ; \
echo "`ls -a \!* | wc -l` file(s)'"
alias cd 'cd \!* && set prompt="! $cwd > "' ; cd .
^D
> jutro # pokušavam izvesti program jutro
...
> csh jutro # pokrećem posebnu ljusku za jutro
...
> set # ali varijabla history je još uvijek 20!
...
> csh -v jutro # izvođenje liniju po liniju
...
> csh -x jutro # izvođenje naredbu po naredbu
...
> chmod +x jutro # dodavanje prava na izvođenje
> jutro # sad je u redu
...
> set # ali opet ništa ...
...
> source jutro # može i ovako
...
> set # to je ono pravo!
...
```

PODACI NA VANJSKIM JEDINICAMA

PROGRAM tar (*Tape archive utility*)

- naredba 'tar'
- opcije: H, c, r, u, x, t, v, V, w, f, ...

primjer:

```

> tar H                                # popis opcija programa tar
...
> cd                                    # radim u matičnom dir.
> tar cvf sve_a.tar *a*                # skupi sve dat. *a* u dat. sve_a.tar
...
> ls -al                                # sve_a.tar postoji; veličina dat. je...
...
> tar tvf sve_a.tar                    # pregled dat. skupljenih u dat. sve_a.tar
...
> mkdir novi                           # stvori novi dir. novi
> cd novi                               # promijeni radni dir. u dir. novi
> tar xVf ../sve_a.tar                 # raspakiraj dat. ../sve_a.tar
...
> ls -al                                # sve dat. iz sve_a.tar su tu
...
> cd ..                                 # vrati se gore
> tar cvf sve_a2.tar ~/*a*             # skupi sve dat. ~/*a* u sve_a2.tar
...
> tar tvf sve_a2.tar                   # koje su sad dat. u sve_a2.tar?
...
> rm zajedno.dva                       # brišem jednu dat. koja sadrži znak a
> cd novi                               # idem u novi
> rm zajedno.dva                       # i tu brišem dat. zajedno.dva
> tar xVwf ../sve_a2.tar               # raspakiraj ../sve_a2.tar uz pojedinačne upite
... ("Y")
> ls -al                                # ovog puta dat. zajedno.dva nema!
...
> cd                                    # a gore?
> ls -al                                # evo je ovdje! zašto?
...
> fddisk -fmt /dev/rfd0c               # formatiranje savitljivog diska (floppy disk)
...
> tar cvf /dev/rfd0c novi              # skupi cijeli dir. novi na disketu
...
> tar tvf /dev/rfd0c                   # što se nalazi na disketi?
...

```

```

> rm -r novi          # brišem cijeli dir. novi
> cd /                # idem u root dir.
> tar xVf /dev/rfd0c  # i ovdje raspakiram dat. s diskete
...
> cd ~                # moram ipak doma; zašto?
> tar xVf /dev/rfd0c  # pokušaj ovdje
...
> ls -al              # dir. novi postoji
...
> ls -al novi         # i sve dat. u njemu
...
> tar cVwf /dev/rfd0c ~/novi # ako novi stavim na disketu ovako?
...
> tar tVf /dev/rfd0c  # u čemu je razlika?
...
> rm -r novi          # izbriši dir. novi
> cd /                # još jedan pokušaj
> tar xVf /dev/rfd0c  # to je to!
...
> cd                  # doma
> ls -al              # provjera
...
> ls -al novi         # sve je u redu
...

```

KOMPRESIRANE DATOTEKE

- naredbe 'compress' i 'uncompress'

primjer:

```

> ls -al *.tar        # popis svih dat. koje završavaju s tar
...
> compress -v *.tar   # kompresiraj dat. uz obavijest o kompresiji
...
> ls -al              # koje su sad veličine tih dat.?
...
> uncompress -v *.Z   # odkompresiraj dat. koje završavaju sa Z
...
> ls -al              # veličine dat. iste kao i prije
...

```

PISAČI (*printers*)

- naredbe 'lpr', 'print', 'lpq', 'lprm', ...
- podaci o pisačima: dat. /etc/printcap
- korisne naredbe za pripremu dokumenata: 'nroff', 'pr', ...

primjer:

```
> lpr telefoni           # ispis dat. telefoni na standardni pisač
> lpq                   # koji su dokumenti u redu za pisač?
...
> lprm 6                 # briši onog pod brojem 6
...
> lpr -P lp1 telefoni   # ispis na pisač lp1 (pogledati /etc/printcap)
> lpq                   # ispis čeka u redu
...
> lprm 7                 # briši ispis pod brojem 7
...
```

Dodatak:

**PRIJENOS DATOTEKA S UNIX FORMATOM ZAPISA
NA MSDOS (PCDOS) FORMAT I OBRATNO**

- javni programski paket 'mtools' (izvediv na većini UNIX-a):
najvažnije naredbe MSDOS-a emuliraju se na UNIX-u;
glase kao u MSDOS-u, samo s prefiksom 'm'

Prijenos UNIX -> MSDOS

ako je datoteka tekstualna (*ASCII* formata) koristiti opciju '-t'

primjer:

```
> fddisk -fmt /dev/rfd0c      # UNIX formatiranje diskete
...
> mformat a:                 # formatiranje diskete u A: na MSDOS format
...
> mcopy telefoni a: # kopiraj dat. telefoni na disketu u A:
...
> mcopy -t tel* a:telefoni.dos # kopiraj na disketu u A:
...
> mdir a:                    # ovdje su obje
...
> mtype a:telefoni # ispiši dat. s UNIX formatom
...
> mtype a:telefoni.dos # i onu s MSDOS; koja je razlika u ispisu?
...
```

Prijenos MSDOS -> UNIX

primjer:

```
> mdir a:                    # ispiši dat.
...
> mcopy a:telefoni.dos novi
                        # kopiraj dat. s MSDOS formatom u dir. novi na UNIX
...
> mcopy -t a:telefoni.dos novi/telefoni.unix # kao tekst
...
> ls -l novi                # tu su
...
> cd novi                   # idi u dir. novi
> more telef*               # razlika?
```

- komercijalni program 'SoftPC' (na ULTRIX-u):
potpuna emulacija MSDOS-a

Prijenos UNIX -> MSDOS

primjer:

unix terminal:

paziti da datoteke koje se prebacuju imaju **MSDOS** oblik imena;
ako je datoteka **ASCII** formata koristiti '/usr/lib/SoftPC/unixtodos';

```
> /usr/lib/SoftPC/unixtodos telefoni telefoni.dos
                                     # konverzija formata
> more telef*                         # u čemu je razlika?
...
```

SoftPC:

namjestiti **E:** disk na radni direktorij (*MB3, Setup Devices, Disk Drives*);

```
C:\> format a:           # formatiranje diskete u A: na MSDOS format
...
C:\> e:                  # idi na E: disk
E:\> dir                 # što je na E: disku - moj radni dir. na UNIXU
...
E:\> copy telef*.* a:    # kopiraj dat. telef* na disketu u A:
...
E:\> dir a:              # ovdje su obje
...
E:\> type a:telefoni     # ispiši dat. s UNIX formatom
...
E:\> type a:telefoni.dos
                           # pa onu s MSDOS; koja je razlika u ispisu?
...
```


Prijenos **MSDOS** -> **UNIX**

primjer:

SoftPC:

namjestiti *E:* disk na radni direktorij (*MB3, Setup Devices, Disk Drives*);

```
C:\> a:                # prebaci na A: disk
A:\> dir                # ispiši dat.
...
A:\> copy telefoni.dos e:\novi
                        # kopiraj dat. s MSDOS formatom u dir. novi na E:
...
A:\> dir e:\novi\telefoni.dos    # tamo je
...
```

unix terminal:

ako je datoteka *ASCII* formata koristiti `'/usr/lib/SoftPC/dostounix'`;

```
> cd novi                # idi u dir. novi
> /usr/lib/SoftPC/dostounix telefoni.dos telefoni.unix
                        # konverzija formata
> more telef*           # razlika?
...
```