RAČUNARSTVO U GEOFIZICI

Zoran Pasarić
Ivan Guettler

zimski semestar, 2021/2022

O kolegiju

- Motivacija
  - Geofizičar treba relativno dobro ovladati mnogim popratnim vještinama (matematika, statistika, računala, terenski rad, instrumentacija)

- Cilj
  - Naučiti dovoljno računarstva (da bi mogli učinkovito raditi)
  - Naučiti osnove programiranja (sa stanovišta geofizike)

- Način rada
  - Praktični rad s čvrstim osloncem na teoriju
  - Samostalna izrada domaćih zadaća

- Ocjenjivanje
  - Putem domaćih zadaća
  - Korekcija moguća na kraju semestra putem projektnog zadatka

O kolegiju, nast.

- Sadržaj:
  - Osnove osnova
  - GNU-Linux
  - FORTRAN 90
  - PYTHON

- Naglasak na praktičnoj upotrebi
- Ne samo 'kuharica'
- Koncepti i principi

Uvod

- Cilj
  - sustavno objasniti temeljne računalne pojmove i njihove međusobne odnose
  - stvoriti preduvjete za učinkovit rad

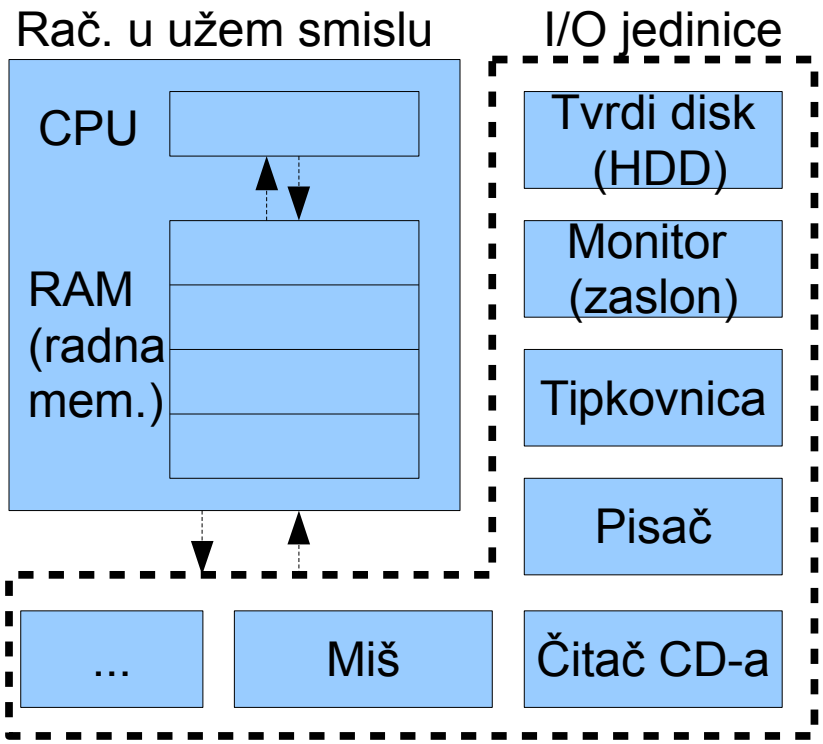
- Literatura
  - Robins &Beebe: Classic Shell Scripting, O'Reilly, 2005.
  - Newham & Rosenblatt: Learning he bash Shell, O'Reilly, 1998.
  - https://www.srce.unizg.hr/files/srce/docs/edu/osnovni-tecajevi/d105_polaznik.pdf
  - http://www.ee.surrey.ac.uk/Teaching/Unix/
  - Kalafatić i dr.: Python za znatiželjne, Element, 2016.
  - VanderPlas: Python Data Science Handbook, O'Reilly Media, Inc., 2016.
  - Johansson: Numerical Python, A Practical Techniques Approach for Industry, Appress, 2015.
  - Fortran 90 Programming, T.M.R. Ellis, Ivor R. Philips, Thomas M. Lahey:Addison-Wesley, 1994,1995,1996., ISBN 0-201-54446-4
  - Mnogi priručnici (manuali)

Osnove osnova

- hardver i operacijski sustav (OS)
- datotečni sustav
- izvršavanje naredbi i programa
- editor
- osnovni poslovi

Hardver

- CPU izvršava naredbe (instrukcije)
- RAM sadrži instrukcije (programe) i podatke s kojima programi rade
  - sastoji se od riječi i stranica
  - svakoj riječi moguć je **direktan** pristup putem **adrese**
  - duljina riječi je fiksna za zadanu arhitekturu (za PC 32 ili 64 bita)
  - svaka riječ sadrži niz nula i jedinica koje predstavljaju **podatak** ili **naredbu** (u logičkom smislu, ništa drugo ne postoji u računalu)

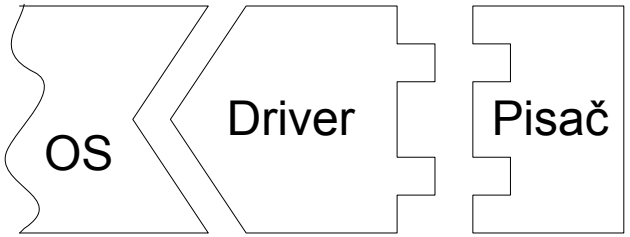


- Ostatak su **ulazno/izlazne** jedinice. Neke su samo ulazne (tipkovnica, miš) neke samo izlazne (zaslon, pisač) a neke i jedne i druge (HDD, ...)

Operacijski sustav (OS)

- Glavni (vrhovni) program koji se brine za skladan i učinkovit rad svih dijelova računala. Pod nadzorom OS-a izvršavaju se ostali programi (aplikacije) te se dodjeljuju resursi potrebni za njihovo izvršavanje

- OS vidi I/O jedinice na jedinstven način, bez obzira na njihovu fizičku prirodu. O fizičkim detaljima brine se odgovarajući DRIVER (pogonitelj), tj. poseban program koji s jedne strane komunicira s OS-om, a s druge strane s konkretnim uređajem.



Dva (tri) dominantna OS-a

- MS WINDOWS, OS tvrtke Microsoft. Dominantno jedno korisnički (single user), više zadaćni (multi tasking)

- GNU/LINUX, Slobodni OS. Više korsnički (multi user), viče zadaćni (multi tasking)
  - izgrađen po uzoru na UNIX
  - učinkovit i ugodan
  - **standard** u znanstvenoj i akademskoj zajednici, a i šire
  - **slobodan** za upotrebu (free as in speech)

- MAC OS, OS tvrtke Apple
  - temeljen na UNIX-u
  - dorađen, naročito grafičko sučelje

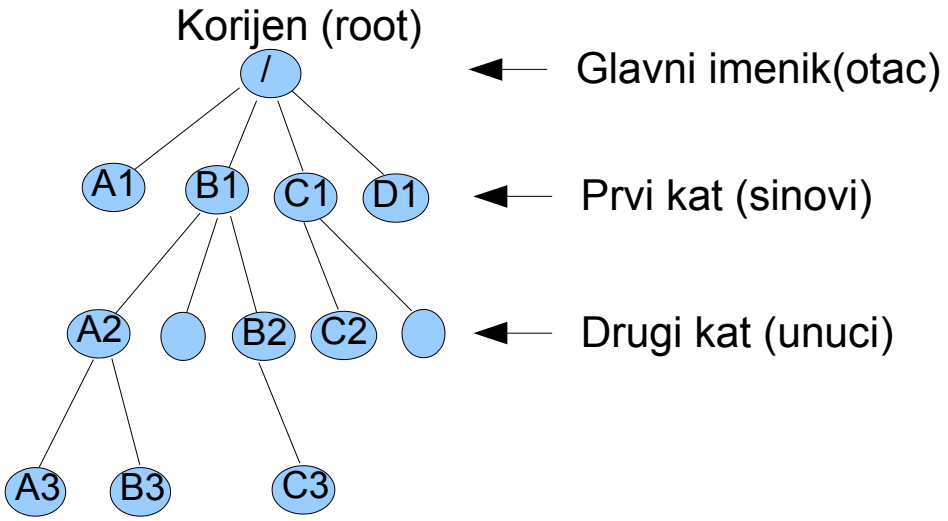
Datotečni sustav

- Datotečni sustav (file system) je logički organizirana skupina datoteka na nekom mediju.
- Prciznije, ali manje zorno, datotečni sustav je skup propisa koji određuju kako se podaci pohranjuju u elektronskom obliku.
- Ovdje mislimo na datotečne sustave koji koriste neku I/O jedinicu, tipično tvrdi disk (uređaj koji služi za “trajnu” pohranu podataka), iako fizička jedinica nije nužna (npr. network file system).

Logička struktura

- Datotečni sustav se logički dijeli na **imenike** (direktoriji, mape) i **datoteke** (file-ovi)
  - DATOTEKA je najmanji IMENOVANI skup podataka. OS pristupa datoteci kao cjelini.
  - IMENIK je logički organizirana skupina datoteka i drugih imenika. Također ima svoje ime i može mu se pristupiti kao cjelini. Dobra je analogija s knjižnicom.
- Imenici i datoteke organizirani su hijerarhijski u STABLO (engl. tree).

Logička struktura, nastavak



- Oznaka za korijen:
  \ (WIN, C:\, D:\ ),
  / (UNIX)
- RADNI ili AKTIVNI imenik je onaj **"u kojem se trenutno nalazimo"**, tj. koji nam je trenutno pridjeljen
- Svaki imenik sadrži DVA pokazivača:
  .. pokazuju na oca,
  . pokazuje na samoga sebe

"Kretanje" kroz imenike: Naredba **cd** mijenja radni imenik, npr. cd A3 (ima smisla ako smo u A2)

- WINDOWS
  cd \
  cd \B1\A2\A3

  cd ..\..\B2
  (iz A3 u B2)

- UNIX
  cd /
  cd /B1/A2/A3

  cd ../../B2
  (iz A3 u B2)

} **apsolutna staza**

} **relativna staza** (u odnosu na trenutni RADNI imenik)

- POTPUNO ime datoteke: /B1/A2/B3/pero.txt

apsolutna STAZA | IME u užem smislu | NASTAVAK (govori o vrsti datoteke)

Vrste datoteka

- **Tekstualne** (ASCII)
  - sadrže znakove koji se mogu ispisati, običan tekst (slova, brojke i nešto specijalnih znakova)
  - organizirane su u retke
- **Binarne**
  - sve ostale (čitljive samo strojno, tj. pomoću odgovarajućih programa)
  - prikazane na zaslonu monitora izgledaju kao 'smeće'

Izvršavanje programa/naredbi

- PROGRAM je skup naredbi ili instrukcija koje čine logičku cjelinu
- IZVRŠNI PROGRAM je binarna datoteka koja sadrži niz STROJNIH instrukcija (koje se sastoje od nula i jedinica)
  - učitava se u memoriju u trenutku izvršavanja
  - tijekom izvođenja, CPU preuzima jednu po jednu instrukciju iz memorije i izvršava je (tzv. Von Neumann-ova arhitektura)
  - redoslijed izvršavanja je (u principu) poznat unaprijed
- STROJNI JEZIK je posve neprikladan za programiranje. Zato:
  strojni jezik → ASSEMBLER → VIŠI PROGRAMSKI JEZICI

Viši programski jezici

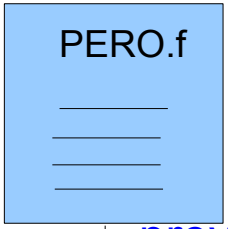
- prilagođeni određenom području primjene
- implementacija na računalu (uglavnom) nevidljiva za korisnika
- primjeri: Fortran, C, Python, Matlab, Java, Basic

- Program u višem programskom jeziku **prevođenje** → izvršni program (strojne instrukcije)
- Ovisno o načinu prevođenja razlikujemo **prevođene** (c*ompilirane*) i **interpretirane** jezike
- Primjer: FORTRAN naspram PYTHON-a

Prevođenje naspram interpretiranja

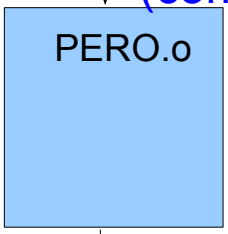
- FORTRAN (compilirani)

Izvorni (source) kod (običan tekst)



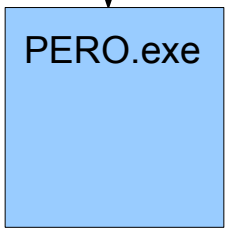
prevođenje (compiler)

objektna (binarna) datoteka; sadrži RELATIVNE adrese



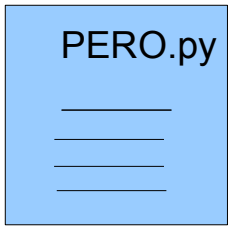
povezivanje (linker)

Izvršni program; sadrži APSOLUTNE adrese



- Python (interpretirani)

Izvorni kod (običan tekst)



Učitava se instrukcija po instrukcija, prevodi u strojni jezik i izvršava.

- prednost: BRZINA
- mana: svaka promjena traži ponavljanje cijelog ciklusa

- prednost: svaka promjena u programu se može odmah provesti/ispitati
- mana: RELATIVNA sporost

Uređivač teksta (editor)

- Računalo (u širem smislu): Skupina smisleno organiziranih datoteka
- Datoteke: Tekstualne (čitljive ČOVJEKU i stroju), binarne (čitljive samo stroju)

- Prirodan način komunikacije sa strojem: **Putem tekstualnih datoteka**!!!
- Glavi alat za rad na računalu: TEKSTUALNI EDITOR (ne tekst procesor)

- EDITOR je program (alat) za učinkovit rad s tekstualnim datotekama.
  Među ostalim to podrazumijeva:
  kreiranje, kopiranje, pisanje, pretraživanje, mijenjanje, ...
- Dobivene datoteke su najčešće namijenjene 'stroju', za razliku od ...
- teksta namijenjenog ljudima koji se obrađuje **tekst procesorima** (pazi se na izgled: vrsta i veličina slova, prijelom stranica, ...)

- Editira se **kopija** datoteke, a nikada sama datoteka; zato na kraju rada biramo
  - **save** ako želimo sačuvati promjene; izvorna datoteka (npr. pero.py) tada ostaje pod drugim imenom (pero.py~, pero.py$ i slično)
  - **quit** ako želimo odbaciti promjene i ostaviti datoteku nepromjenjenu

Osobine dobrog editora

- napredna 'navigacija'
- bojanje sintakse ovisno o vrsti datoteke
- napredno pretraživanje i mijenjanje (regularni izrazi)
- neograničeni “undo” i “redo”
- makro naredbe
- ...

- Zaključak: Odabrati **jedan** dobar editor i njime dobro ovladati!!!
- Pod UNIX-om za sve poslove se koristi (odabrani) jedan te isti editor
- Za ugodan i učinkovit rad: izbjegavati pretjeranu upotrebu miša, pohoditi tečaj daktilografije

Osnovni poslovi: Računanje, crtanje i pisanje

WINDOWS	!	LINUX
FORTRAN, R(C)	!	FORTRAN, R(C)
PYTHON, RC	!	PYTHON, RC
R, RC	!	R, RC
MATLAB*, RC	!	MATLAB*, RC
MS-Office*, P(R)(C)	!	LibreOffice*, P(R)(C)
LATEX, P	!	LATEX, P
COREL, C	!	INKSCAPE*, C
	!	XMGRACE*, C(R)

* Program je 'klikabilan'

GNU/Linux

- Zašto GNU/Linux?
- Gruba struktura
- Ljuska (shell)
- Korisnici, vlasništvo i prava
- Rad s tekstualnim datotekama

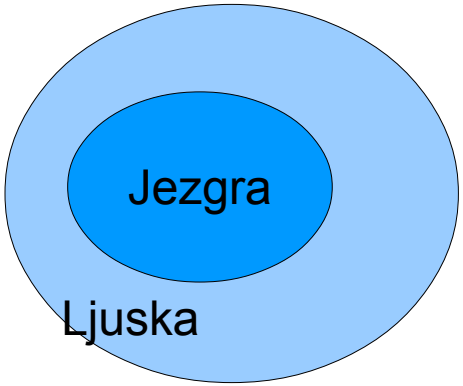
Zašto GNU/Linux

- izgrađen po uzoru na UNIX
- omogućava učinkovit i ugodan rad
- standard u akademskoj zajednici, a i (puno) šire
- slobodan za korištenje (GNU licenca, free as in speach)

Gruba struktura

- OS je vrhovni program koji se birne za učinkovito i usklađeno upravljanje resursima računala. Kako to rješava Linux (Unix)?
  - **jezgra** (*kernel*)
  - **ljuska**, **ovojnica** (*shell*)

} **Potpuno razdvojeno !!**



Jezgra

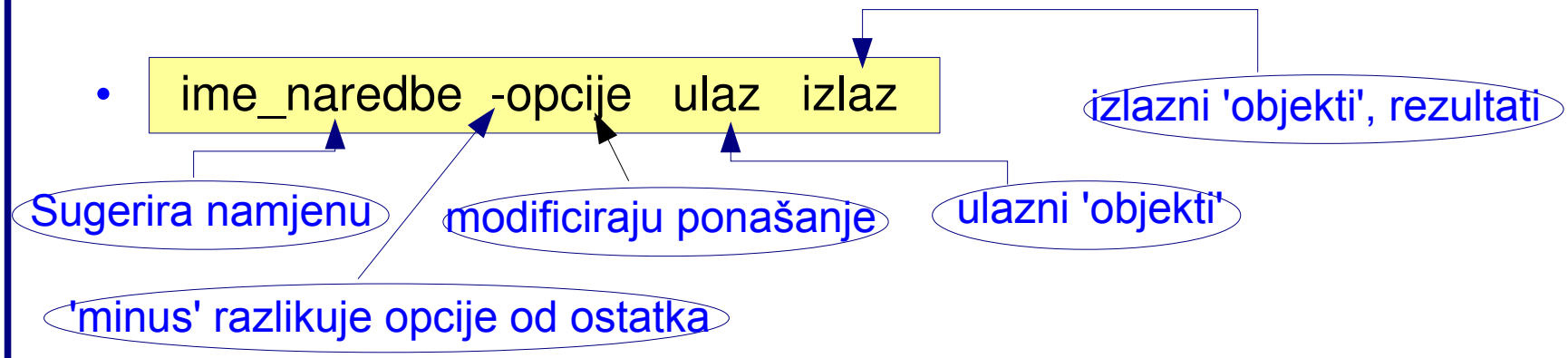
- središnji dio OS-a - učinkovita i beskonfliktna dodjela hardverskih resursa
- poseban (specifičan) program koji upravlja procesima te I/O-om (što uključuje i drivere)
- pristup hardveru moguć je isključivo kroz jezgru putem sistemskih poziva (system calls) ---> STABILNOST
- jezgra stoji između korisnika i hardvera (što god da želimo napraviti moramo pitati jezgru i to na standardan način!!)

Ljuska (shell)

- najgrublje: ljuska je **program koji izvodi naredbe** (*runs commands*), ali i puno više:
  - i**nteraktivno sučelje** između korisnika i jezgre
  - **radno okruženje** (environment)
  - **programski** (skriptni) **jezik**
- omogućava korištenje gotovo svih mogućnosti sustava **bez** upotrebe sistemskih poziva, tj. bez (sistemskog) programiranja

- rad u ljusci se odvija putem terminala (danas prozor, nekad tekstualni ekran)
- rad je interaktivan: korisnik kuca naredbe i reagira na rezultate
  - mana: potrebno predznanje (npr. sintakse naredbi) i praksa
  - prednost: ugodan i iznimno efikasan rad

Opći oblik naredbe pod Linux-om



- Na primjer:

ls	lista sadržaj trenutnog imenika (direktorija)
ls -l	radi isto, ali daje detaljni (long) ispis
ls -lR pero	lista detaljni ispis sadržaja direktorija pero i svih njegovih poddirektorija
ls -1 ..	ispisuje sadržaj oca-imenika (..) i to u jednom stupcu

- **Napomena**: SVI ulazni parametri UVIJEK moraju biti definirani. Ako ih ne odredi korisnik, sustav uzima UNAPRIJED određene, tzv. DEFAULT-ne vrijednosti. To vrijedi u SVIM situacijama, a ne samo kod naredbi.

Osnovne naredbe: Rad s imenicima

• cd dir1	postavlja imenik dir1 kao radni imenik
cd	postavlja kućni imenik (*home directory*) kao radni
• mkdir dir1 dir2 ... dirN	kreira (prazne) imenike dir1, dir2, ... dirN
• pwd	ispisuje punu stazu (trenutnog) radnog imenika

Osnovne naredbe: Rad s imenicima i datotekama

• ls	već napravili
• cp file1 file2	**kopira** datoteku file1 u datoteku file2
cp file1 file2 ... fileN dir	kopira datoteke file1, file2, ... fileN u imenik dir
cp -r dir1/ dir2	kopira sadržaj imenika dir1 (sa svim poddirektroijima) u imenik dir2

• mv file1 file2	vrši **preimenovanje** datoteke file1 u file2
mv file1 file2 ... fileN dir	**premješta** datoteke file1, file2, ... fileN u imenik dir
mv dir1 dir2	premješta imenik dir1 (sa svim poddirektroijima) u imenik dir2

• rm file1 file2 ... fileN	**briše** datoteke file1, file2, ... fileN
rm -r dir	briše imenik dir i sve što je u njemu (-r = *recursive*)
rm -rf dir	isto kao gore, ali **bez ikakvih pitanja** (-f = *forced*), **OPASNA** naredba !!!!!

Osnovne naredbe: Rad s datotekama

- head file1 — **ispisuje prvih 10 linija** (tekstualne) datoteke file1
  head -n 30 file1 file2 — ispisuje prvih 30 linija datoteka file1 i file2, redom
- tail — radi isto kao i head, ali s kraja datoteke
- cat file1 file2 ... fileN — **spaja (nadovezuje)** zadane datoteke i šalje ih na stdout (ispisuje na zaslon)

- diff file1 file2 — **uspoređuje** dvije datoteke (tekstualne) datoteke po linijama
  xxdiff file1 file2 — GUI verzija prethodne naredbe

- wc -c file1 — **ispisuje broj byte-ova** u datoteci file1
  wc -m file1 — ispisuje broj znakova ...
  wc -l file1 — ispisuje broj linija ...
  wc -w file1 — ispisuje broj riječi ...

Osnovne naredbe: komprimiranje

• gzip file1 file2 ... fileN	**komprimiranje** u gzip format; svaka datoteka file1, file2, ..., fileN se **zamijeni** dgovarajućom datotekom s dodatnim nastavkom **.gz**
gzip -d file1 file2	raspakirava datoteke dobivene pomocu gzip

• zip file.zip file1 file2 ... fileN	**kreira zip** arhivu file.zip s datotekama file1, file2, ... fileN
unzip file.zip	**raspakirava zip** arhivu, file.zip
unzip -v file.zip	**lista sadržaj** zip arhive, file.zip

• tar -zcvf file.tgz dir1	imenik dir1 sprema u komprimiranu **tar arhivu** file**.tgz**
tar -tzvf file.tgz	**lista sadržaj** komprimirane tar arhive file.tgz
tar -xzvf file.tgz	**raspakirava** komprimiranu tar arhivu file.tgz

Osnovne naredbe: Razno

- which prog_file — **ispisuje stazu** izvršne datoteke prog_file
- who — daje **popis korisnika** trenutno prijavljenih na sustav; ima **puno opcija**
- echo neki_text — **ispisuje tekst** neki_text na ekran (tj. stdout)
- date — ispisuje ili postavlja **datum i vrijeme** na sustavu

- ssh — *Secure Shell*, omogućava **rad** na **udaljenom** računalu
- ftp — *File Transfer Protocol*, omogućava **prebacivanje datoteka** sa i na **udaljeno** računalo.
- sftp — Sigurna verzija ftp-a
- gftp — ftp i sftp s grafičkim sučeljem

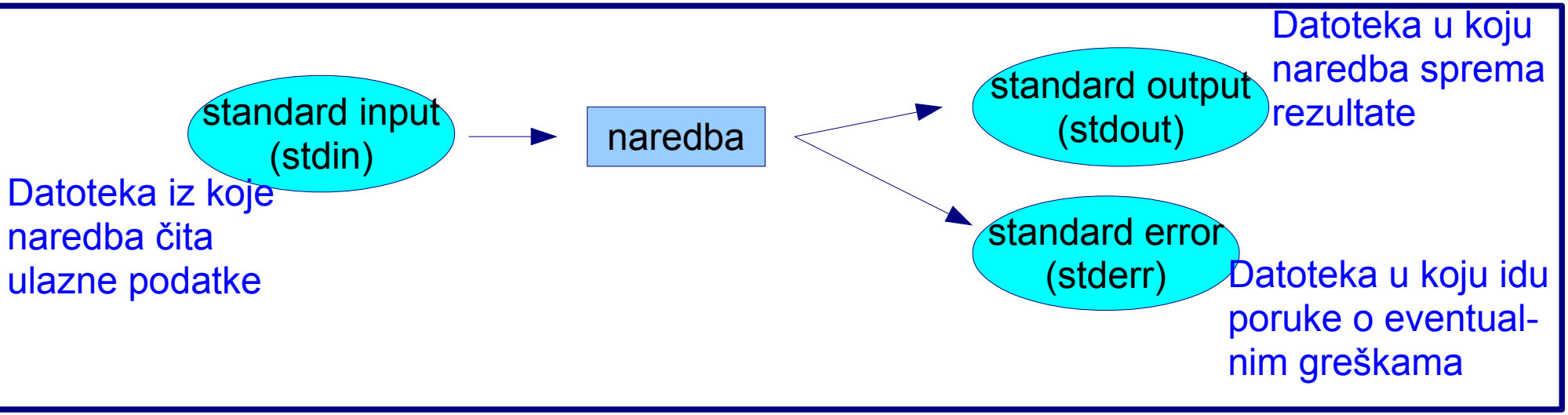
- mount — Uključuje novi datotečni sistem u glavno stablo; u pravilu se vrši automatski

Sustav pomoći

- **kratki opis** tražene funkcije: opcija --help, npr.
  ls --help
- **detaljniji opis**: naredba man (od manual), npr.
  man ls
- naredba **apropos**: pretražuje man stranice pomoću ključnih riječi, npr.
  apropos password (password je ovdje **ključna riječ**)

Standardni I/O (Standard I/O)

- Fundamentalni koncepti: Standardni **ulaz**, standardni **izlaz** te standardni **izlaz grešaka** (*standard input*, *standard output*, *standard error*)
- Omogućavaju **povezivanje** programa



- Napomene:
  - Pod Linux-om svaki objekt je **datoteka** ili **proces** (o procesima kasnije)
  - Ljuska kod pokretanja povezuje stdin s tipkovnicom, a stdout i stderr s ekranom monitora (prozorom terminala)
  - Sama naredba **ne zna** (nije joj niti bitno) odakle dolazi ulaz te kamo ide izlaz

Preusmjeravanje i cjevovodi

- **Preusmjeravanje ulaza i izlaza** (*I/O redirection*)
  - naredba > file
  - naredba >> file

  } stdout se preusmjerava u datoteku file
  - naredba 2> file — radi isto, ali za stderr
  - naredba > fileA 2> fileB — 'rezultati' idu u fileA, 'pogreške' u fileB
  - naredba < file — sadržaj datoteke file se preusmjerava naredbi putem stdin

- **Cjevovodi** (*pipes*)
  - nastaju preusmjeravanjem standardnog izlaza jedne naredbe na standardni ulaz druge naredbe:
    naredba1 | naredba2
  - na primjer: Kako ispisati 7 najnovijih datoteka u imeniku pero?
    ls -1t pero | head -n 7

Digresija: vrste datoteka

- sa stanovišta korisnika
  - **tekstualna** (sadrži znakove koji se mogu ispisati, organizirane u retke)
  - **binarna** (npr. programi u strojnom jeziku, neki formati za slike (jpg, tif), zvuk)

- sa stanovišta računala
  - obična (ordinary)
  - imenik (directory)
  - link (link)
  - uređaj (device)

Razvoj imena datoteka (*file name expansion*, *globbing*)

- pomoću posebnih znakova mogu se zadati **skupine** datoteka
- npr. naredba wc *pero.txt
  - **kreira listu** svih datoteka čija imena završavaju na pero.txt (dakle ivopero.txt, antepero.txt, ali ne i antepero1.txt)
  - **primjenjuje naredbu** wc na svaku od tih datoteka

- **posebni znakovi**: * mijenja bilo koju **skupinu** znakova
  ? mijenja bilo koji, **jedan** znak
  [atz1u] mijenja **jedan od znakova** a,t,z,1 ili u
- ako je neki od posebnih znakova dio imena datoteke, treba ga '**zaštititi**' s \, tj. ivo*ante.txt je datoteka (ne odveć pametnog imena) ivo*ante.txt
- pod Linux-om, znak \ se koristi za **poništenje** specijalnog značenja sljedećeg znaka. *, \[, \?, \\ znače *, [, ?, \, redom.
- problem s \ pod DOS-om

Radno okruženje (*environment*)

- **varijable okruženja** (*environment variables*)
  - npr. PATH, USER, HOME, ...
  - dostupne su korisniku te programima koji se izvršavaju unutar ljuske
  - naredba **env** ispisuje sve trenutne varijable i njihove vrijednosti
  - **$** ispred imena varijable znači (daje) **vrijednost varijable**, npr. ako se korisnik zove pero, $HOME isto je što i /home/pero {npr. echo $HOME}
  - posebno važna varijabla je **PATH**; sadrži popis svih staza koje sustav pretražuje u potrazi za **izvršnim** programom prilikom pokušaja njegovog pokretanja

- **nadopuna sintakse** (*completion*)
  - najvažnija tipka pod Linux-om je **TAB**
  - na jedan tab, TAB, ljuska pokušava (pametno) **nadopuniti** započetu naredbu
  - na dva taba-a, TAB TAB ljuska ispisuje **sve mogućnosti** (ako ih je više)
  - POUKA: **Kad ne znamo što ćemo, pritisnemo TAB**

Radno okruženje, nastavak

- **konfiguracijske datoteke** (među ostalima)
  - .bash_profile sadrži naredbe koje se izvode **prilikom prijave** na sustav (pokretanje tzv. login shell-a); ako ne postoji datotekaecho

    .bash_profile izvršava se datoteka .profile
  - .bashrc sadrži naredbe koje se izvode kod pokretanja *non-login shella*; tu korisnik **prilagođava** okolinu sebi; npr. prompt, aliasi, PATH, dodatne varijable, ...

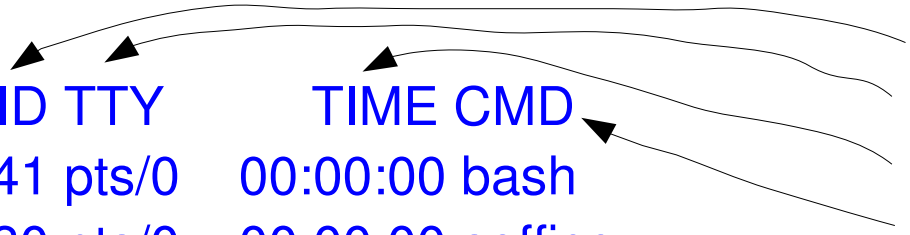
- **history**
  - ljuska **pamti** prethodno korištene naredbe u datoteci **.bash_history**, te ...
  - omogućava njihovu **re-upotrebu** editiranjem komandne linije
  - **komandna linija** prikazuje jedan redak iz datoteke .bash_history i ...
  - **editira se** zadanim editorom (istim koji se inače koristi)
  - pretraga unatrag pomoću **Ctrl-r** (a'la emacs)
  - navigacija kroz history je moguća i tipkama sa strelicama

Upravljanje procesima

- Proces je svaki **aktivni program** (running program)
- **Informacije** o procesima se dobivaju naredbom ps
  - ps — Lista procese koji pripadaju trenutnom korisniku i terminalu (ps – Process Status)
  - ps -e — Lista sve procese
  - ps -ef — Proizvodi detaljni ispis svih procesa
  - ps -u imeKorisnika — Lista sve procese koji pripadaju korisniku 'imeKorsnika'

- Na primjer:

```
pasaric@geo74:~$ ps
  PID TTY          TIME CMD
 8141 pts/0    00:00:00 bash
 8180 pts/0    00:00:00 soffice
 8191 pts/0    00:07:27 soffice.bin
22204 pts/0    00:00:00 ps
```



Identifikacijski broj procesa (PID)
Oznaka terminala
Ukupno potrošeno CPU vrijeme
Naredba koja je aktivirala proces

Upravljanje procesima, nastavak

- Neki **signali** koji se mogu uputiti procesu: STOP, CONT, TERM, KILL
- Za slanje signala koristi se naredba **kill**, npr.

kill -STOP pid	Zaustavlja proces s brojem pid (**Ctrl-z**)
kill -CONT pid	Ponovo pokreće rečeni proces
kill -TERM pid	Trajno prekida proces (**Ctrl-c** – česta upotreba!!)
kill -KILL pid	Bezuvjetno prekida proces (korisno ako se neki program smrzne) (**Ctrl-**\)
kill -9 pid	Isto kao -KILL

- Program pokrenut iz ljuske, dok ne završi, **blokira** unos novih naredbi (u pravilu)
- Takve programe je moguće poslati u **pozadinu** (*background*), tj. '**otkačiti**' od ljuske i nastaviti interaktivni rad

npr:	evince pero.pdf	Pokreće evince reader na datoteci pero.dat i pri tom blokira unos novih naredbi ljuske
	evince pero.pdf &	Radi isto, ali ne blokira daljni rad u ljusci

- Program se iz pozadine može vratiti u **prvi plan (*foreground*)**, tj. opet ga se može '**vezati**' za terminal → naredbe **fg** i **bg** u kombinaciji s **Ctrl-z**

Upravljanje procesima, nastavak

- **Tipkovničke kratice**
  - **Ctrl-c** Šalje trenutno aktivnom procesu signal TERM
    (**Kad želimo prekinuti program kucamo Ctrl-c !!!**)
  - **Ctrl-z** Šalje signal STOP, tj. stavlja proces u stanje mirovanja
- Proces zaustavljen s **Ctrl-z** može se poslati u pozadinu naredbom **bg**
- Proces poslan u pozadinu s **bg**, ili pomoću **&** može se pozvati u prvi plan naredbom **fg**

- Naredba **top** prikazje sliku 'živog' sustava u realnom vremenu:
  - stanje procesa po raznim kriterijima
  - sumarni pregled korištenih resursa
  - npr. top
    top -u pero

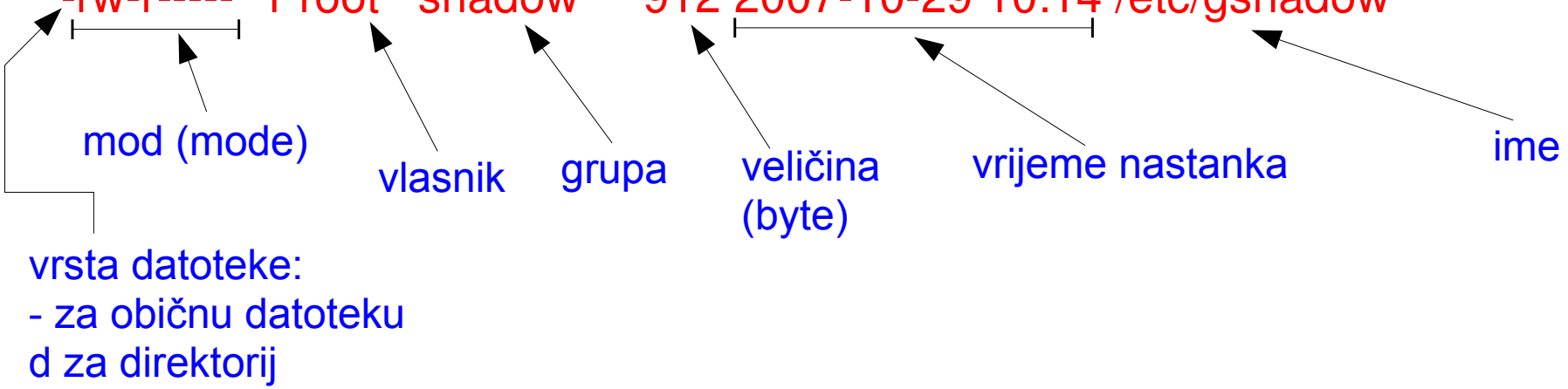
Korisnici, vlasništvo i prava pristupa (*permissions*)

- Svaki korisnik ima **korisničko ime** (*username*), npr. pero i **lozinku** (*password*), npr. k@3.Ta8%
- Korisniku pripada **kućni imenik** (*home directory*); u gornjem primjeru to je /home/pero

- **Privatnost** korisnika (a dijelom i **sigurnost** sustava) počiva na **vlasništvu** i **pravima pristupa** datotekama i imenicima
- Svaka datoteka (imenik) ima **vlasnika** i **grupu** kojoj pripada. **Grupa** je imenovana skupina korisnika – na prirodan način određuje tko smije pristupiti datoteci (imeniku)
- Prava pristupa se odnose na **čitanje (r), pisanje (w), izvršavanje (x)**
- Vlasništvo, prava pristupa i druge informacije se vide naredbom **ls -l**

Na primjer

- >> ls -l /etc/gsha*



- >> ls -ld ument?

  drwxr-xr-x 6 pasaric pasaric 4096 Sep 14 10:34 Documents/

  >> ls -l /bin/l*s

  -rwxr-xr-x 1 root root 165704 Feb 28 2016 /bin/less

  -rwxr-xr-x 1 root root 204248 Jan 5 2016 /bin/loadkeys

  -rwxr-xr-x 1 root root 130736 Feb 22 2017 /bin/ls

Rad s tekstualnim datotekama

- Pregledavanje
  - može se koristiti naredba **cat** u kombinaciji s **head** i **tail**, no
  - pravi preglednik je **less**
  - za masovno pretraživanje služi **grep**

- Editiranje – najvažniji i najčešći posao
  - nužno je nučiti rad u editoru
  - **Vi**, **Emacs**
  - najteži dio

Preglednik less

- Ime *less* je igra riječi. **Less** je značajno poboljšana verzija klasičnog UNIX-ovog preglednika *more*
- **Pokretanje**

less ime_datoteke	Otvara datoteku 'ime_datoteke' za pregledavanje
less -S ime_datoteke	Isto što i gore, ali bez prelamanja dugih linija
less -i ime_datoteke	Isto, ali kod pretraživanja ne razlikuje velika i mala slova

- **Unutar** preglednika koristimo **razne naredbe** (većina vrijedi i za *man pages)*

enter	Pomakni se jednu liniju napred
space	Pomakni se jedan ekran naprijed
g	Skoči na početak
G	Skoči na kraj
156	Idi na liniju 156
/kGb	Traži tekst 'kGb' u datoteci
n	Skoči na iduću (**n**ext) pojavu nađenog teksta
N	Skoči na prethodnu pojavu nađenog teksta
=	Ispiši ime datoteke, broj linija i položaj u datoteci
q	Napusti less (**q**uit)

- Za dohvaćanje pronađenog teksta može se koristiti miša.

Naredba grep

- Služi za **masovno pretraživanje** tekstualnih datoteka
  - grep neki_obr popis_datoteka — Ispisuje sve linije koje sadrže tekst 'neki_obr' iz datoteka sa zadanog popisa
  - grep -i ... — Radi isto, ali bez da razlikuje mala i velika slova
  - grep -v ... — Radi isto, ali ispisuje linije koje NE sadrže rečeni tekst (obrazac, pattern)
- Napomena: Obrazac se može zadati vrlo općenito korištenjem '**regularnih izraza**'
  Kod popisa datoteka koristi se razvoj (ekspanzija) imena

Regularni izrazi (digresija)

- Skup pravila (notacija) za opis teksta **za pretragu prema nekom kriteriju** (npr. ‘ono što počinje slovom a nakon čega slijedi broj’
- RegExp se sastoji od ‘običnih’ znakova i ‘specijalnih’ znakova
- Specijalni znakovi Značenje

\	Uklanja specijalno značenje idućeg znaka.
.	Odgovara bilo kojem **pojedinačnom** znaku.
[…]	Odgovara **jednom od znakova** navedenih u zagradama.
[^…]	Odgovara bilo kojem znaku koji **nije** naveden u zagradama.
*	Odgovara bilo kojem **nizu realizacija** (nula ili više) pojedinačnog znaka koji neposredno prethodi.
+	Kao i prethodno, ali za **jednu ili više** realizacija.
?	Kao i prethodno, ali za **nula ili jednu** realizaciju.
{m}	**Točno** m realizacija
{m,}	**Barem** m realizacija
{m,n}	Između m i n realizacija
^	Odgovara slijedećem regularnom izrazu, ali **na početku retka.**
$	Odgovara prethodnom regularnom izrazu, ali **na kraju retka.**

Regularni izrazi (digresija, nast.)

Primjeri:

	Izraz	Odgovara
•	pero	4 slova u nizu, ‘pero’, bilo gdje u liniji
•	^pero	4 slova u nizu, ‘pero’, na početku linije
•	pero$	4 slova u nizu, ‘pero’, na kraju linije
•	^pero$	linija koja sadrži samo ‘pero’ i ništa više
•	p[eE]ro	ili 4 slova u nizu, ‘pero’, ili 4 slova u nizu pEro, bilo gdje u liniji
•	pe.o	dva slova ‘pe’, bilo koji znak i slovo ‘o’, bilo gdje u liniji, npr. PeZo, pe1o, pe&o, …
•	pe.*o	dva slova ‘pe’, bilo koji niz od nula ili više znakova i slovo ‘o’, bilo gdje u liniji, npr. PeZ12o, peSRE4o, peo, …
•	pe\.o	4 slova u nizu, ‘pe.o’, bilo gdje u liniji
•	p[eE]{2}ro	slovo p, zatim točno dva slova, e ili E, te slovo o

Editor Vi

- Editor Vi (zapravo Vim) je poboljšana verzija klasičnog UNIX-ovog editora Vi
- Traži apsolutno **najmanje** kucanja od svih editora; radi u terminalu

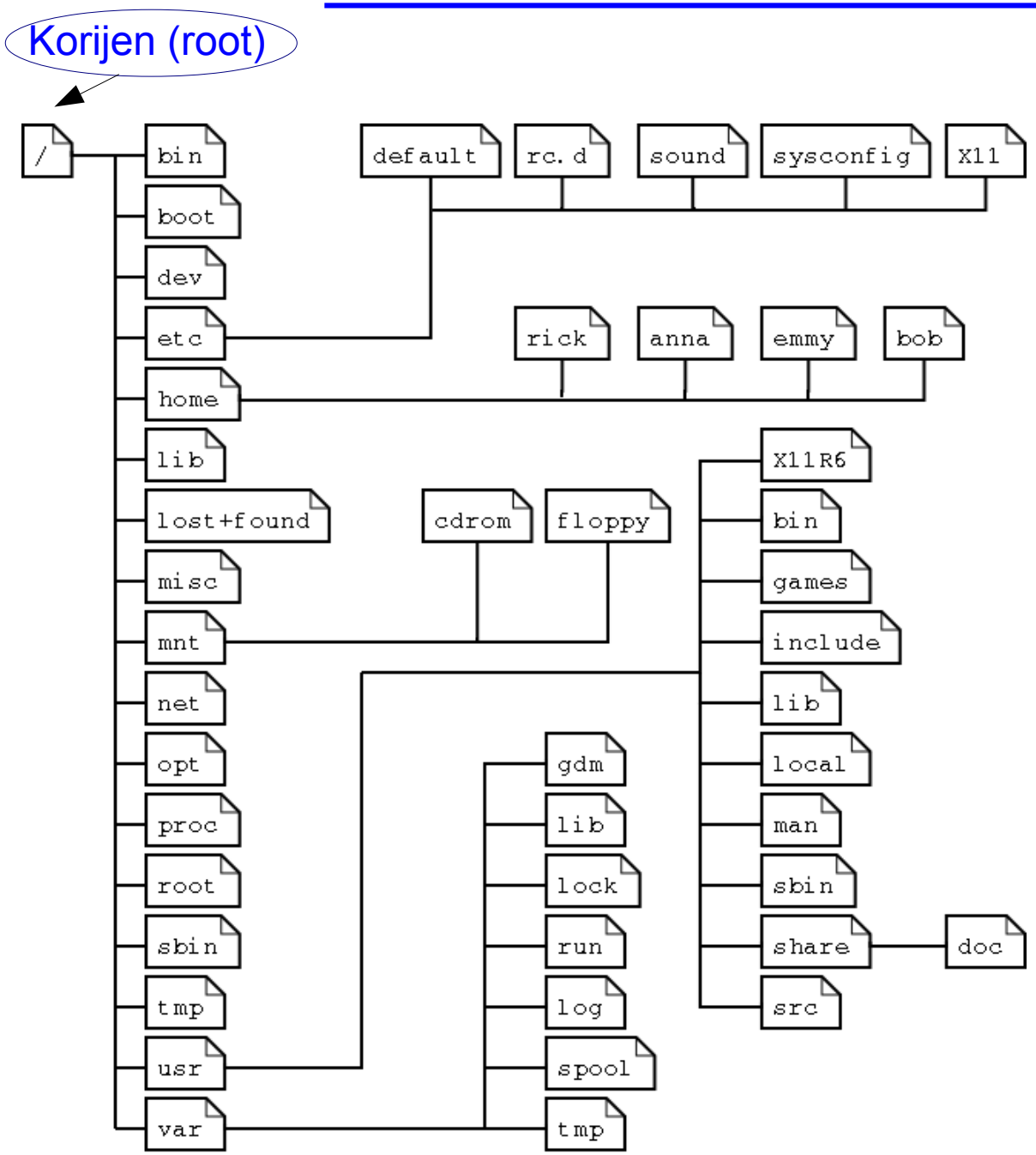
- Tri **moda**:
  - Mod za **unos teksta** (*input mode*): Sve što se kuca je tekst, a rade i strelice za micanje kursora
  - Mod za **unos naredbi** (*command mode*): Sve što se kuca je naredba
  - Mod **zadnje linije** *(last line mode*): Zadaju se složenije naredbe

- **Značenje** nekih naredbi/tipki
  - esc iz moda **unosa** prelazi u mod **naredbi**
  - i prelazi u mod unosa (insert). Istu svrhu imaju i naredbe o,a, ...
  - h,l,k,j micanje kursora, lijevo, desno, gore, dolje
  - x briše znak ispod kursora
  - dd briše liniju na kojoj je trenutno kursor

Editor Vi, nastavak

- **Znak dvotočke (:)** na početku naredbe inicira prelazak u mod zadnje linije, npr.:
  - :wq — Zapiši promjene (write) i zatvori editir (quit)
  - :q — Zatvori vi (quit)
  - :q! — Forsiraj izlaz (bez provjere da li su promjene spremljene)
  - :w ime_dat — Zapiši editiranu datoteku pod imenom ime_dat
  - :5,20w ime_dat — Zapiši linije 5 do 20 u datoteku ime_dat
  - :-1,$s/trazi_txt/zamijeni_txt/gc — Počevši od trenutne linije pa do kraja, pronađi sve pojave od 'trazi_txt' i uz pitanje ih zamijeni tekstom 'zamijeni_txt'
- Vi je pogodan za rad na **udaljenom** računalu jer je svugdje prisutan i potpuno tekstualan

Datotečni sustav Linux-a, pregled



/etc Razne konfiguracijske datoteke i sistemski skripti

/home Korisnički imenici

/mnt Mjesto za uključenje vanjskih jedinica i njihovih datotečnih sustava. U novije doba tomu služi i /media

/tmp Radni prostor dostupan svim korisnicima i programima

/usr Razne stvari

/var Razne sistemske datoteke privremenog karaktera (npr. logovi)

Pravila za softwerske alate (filozofija UNIX-a)

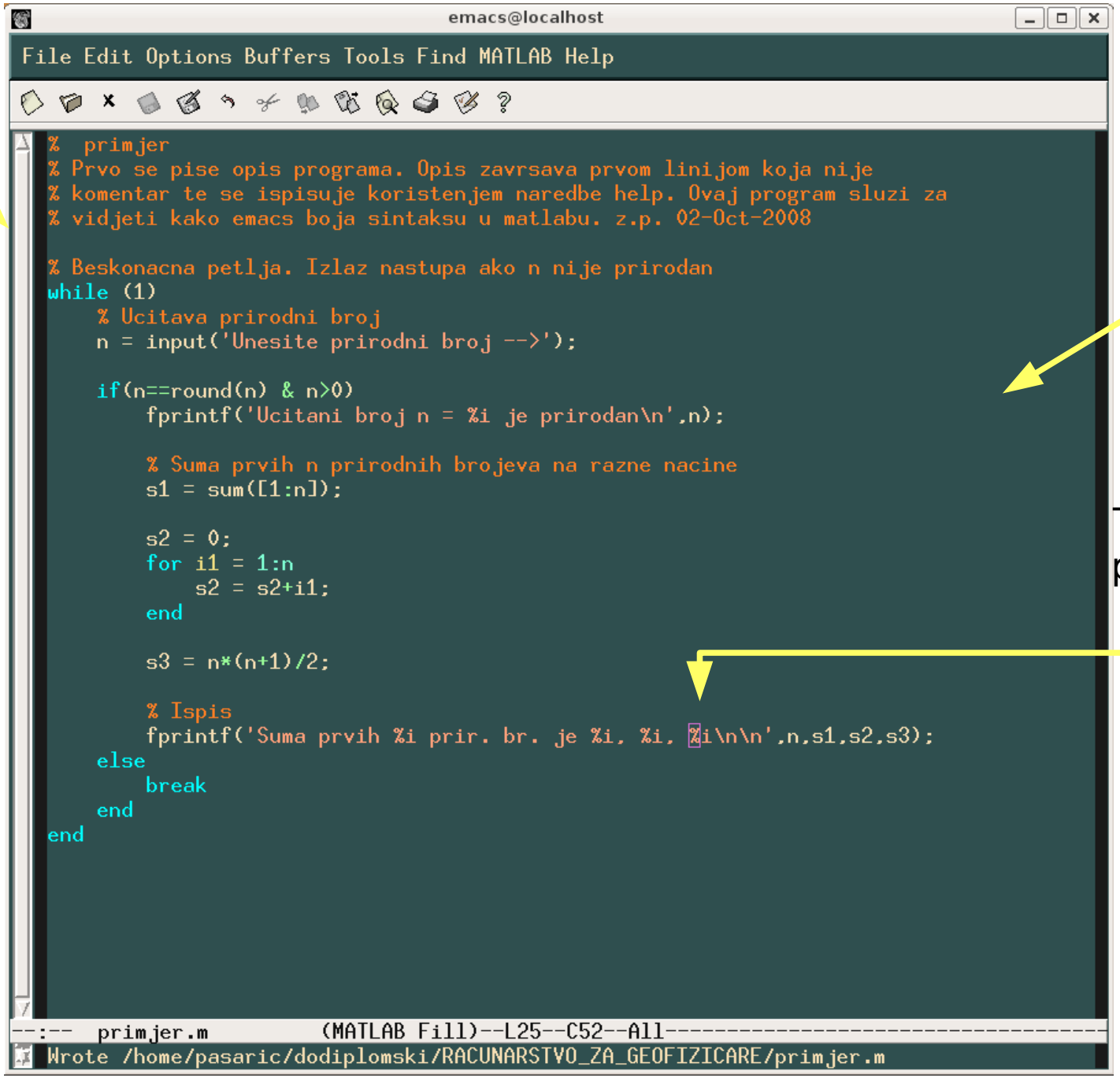
- Radi jednu stvar, ali dobro
- Radi s tekstom (linije), a ne binarno
- Koristi regularne izraze
- Podrazumijevani I/O treba biti standardni I/O
- Ne šalji suvišne poruke
- Izlaz neka bude istog formata kao i ulaz
- Ako je moguće prepusti drugome najteži dio
- Ako nije, napiši općeniti program poštujući gornja paravila, tako da može služiti i drugima!

Editor Emacs

- Uz Vi, Emacs je *de facto* **standard**
- Emacs-om možemo napraviti **praktički sve** (pisati i uređivati tekst, upravljati datotekama i imenicima, pisati i čitati e-poštu, raditi u ljusci, kompilirati programe, organizirati podsjetnik, igrati igre, posjetiti psihijatra, ...)

- potpuno je i podesiv i proširiv (EMACS = *Editor MACroS*, tj. makro naredbe za editiranje); u pozadini stoji programski jezik LISP
- pogodan je za bilo kakve zadaće putem **modova** (*modes*); svaka vrsta 'editiranja' ima odgovarajući, posebno prilagođeni mod (npr. Fortran, Matlab, Latex, Bash, C, Dired, ...)
- u doba tekstualnih terminala Emacs je pružao funkcionalnost današnjih prozora

Editor Emacs, kako izgleda



Editor Emacs, osnovni pojmovi

- **Okvir** (*frame*) - cijeli X-prozor koji Emacs koristi
- **Prozor** (*window*) - prostor namijenjen za editiranje; može se razbiti na više manjih prozora
- **Područje odziva** / **mini spremnik** (*echo area / mini buffer*) -
  - prostor za ispis znakova i naredbi koje kuca korisnik
  - prostor za poruke Emacs-a korisniku
  - prostor za unos parametara za pojedine naredbe (npr. ime datoteke);
- **Linija moda** – informacije o spremniku (bufferu) koji je prikazan u prozoru
  - da li u spremniku ima promjena koje nisu spremljene
  - ime spremnika, glavni i pomoćni mod spremnika, položaj točke
- **Točka** (*point*) – položaj kursora u prozoru (svaki prozor ima svoju točku)

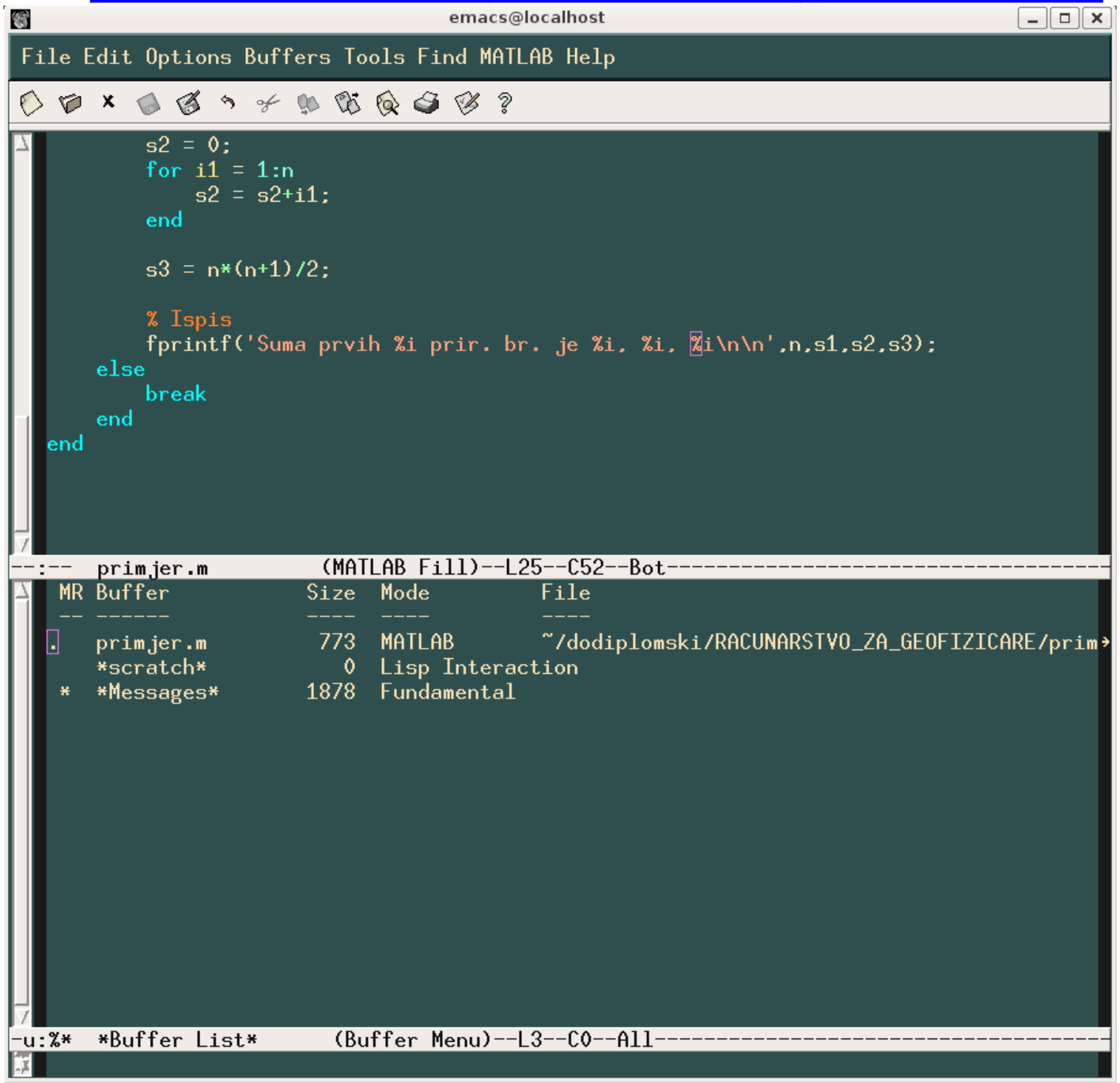
Editor Emacs, kako radi

- Emacs stalno operira s **nekoliko spremnika** (*buffers*)
- Svakoj **datoteci se pridruži spremnik** istog imena i u njega se datoteka **kopira.**
- Svaki spremnik se prikazuje u svom prozoru (može isti spremnik u više prozira).

- Za razne informacije koriste se posebni spremnici; oni se **prikazuju po potrebi**;
  - nemaju pridruženu datoteku;
  - često nisu za pisanje;
  - imena im počinju sa * (npr. *Messages* , *Buffer List*, *scratch*, itd.).
- Ako se pojavi (prikaže) neki spremnik, sadržaj se pogleda, a pripadni prozor potom zatvori; time ostajemo s prvobitnim prozorom.

- **Aktivan** je onaj prozor koji trenutno sadrži kursor (dočim svaki ima točku).
- Naredbe se odnose ili na **cijeli prozor** ili na **zadano područje** (region)
- Područje je prostor između **znaka** (mark) i **točke** (point)

Editor emacs, kako radi, nastavak



Prozor spremnika primjer.m

Prozor spremnika *Buffer List*

Editor Emacs, naredbe

- Emacs posjeduje vrlo mnogo naredbi.
- **Nekim** naredbama pridružene su kratice; **ostale** naredbe se kucaju kako se i zovu; ipak u svim razumnim situacijama radi **nadopuna sintakse,** tipkom **Tab**.
- U **kratice** ulaze tipka Control (**Ctr**) i Meta tipka (na PC-u **Alt**); umjesto Alt najčešće radi i tipka **Esc**.
- Pazi: Ctr-x znači **drži stisnuto** Ctr **i pritisni** x;
  Alt-x znači drži stisnuto Alt i pritisni x;
  Esc x znači stisni Esc, pusti ga i stisni x;

- Najvažnije naredbe:
  - Ctr-g **Opoziv naredbe** čije je izvršavanje u tijeku
  - Ctr-_ ili Ctr-x u **Poništenje** učinka prethodne naredbe (UnDo)
  - Ctrl-x Ctr-c **Izlazak** (zatvaranje) Emacsa (uz eventualna pitanja, Exit)
  - Ctr-x Ctr-s **Spremanje** sadržaja svih spremnika (Save)

Editor Emacs, naredbe, nastavak

- **Exit / save / read / write**
  - Exit (Cancel) Ctr-x Ctr-c
  - Save File Ctr-x Ctr-s
  - Read File Ctr-x Ctr-f
  - Write File Ctr-x Ctr-w
- **Error recovery**
  - Abort command Ctr-g
  - Undo Ctr-_ ili Ctr-x u
- **Buffers**
  - New buffer Ctr-x b
  - Buffer list Ctr-x Ctr-b
  - Kill buffer Ctr-x k

- **Windows**
  - Close active Ctr-x 0
  - Leave active, only Ctr-x 1
  - Split horiz. Ctr-x 2
  - Cursor to next win. Ctr-x o
- **Move**
  - Begining of buffer Esc < ili Alt-<
  - End of buffer Esc > ili Alt->
  - Beginig of line Ctr-a
  - End of line Ctr-e

Editor Emacs, naredbe, nastavak

- **Mark / kill / yank (paste)**
  - Begining of region Ctr-Space
  - End of region -> move kursor
  - Kill buffer Ctr-x k
  - Kill to EOL Ctr-k
  - Kill to BOL Alt-0 Ctr-k
  - Kill region Ctr-w
  - Copy region Alt-w
  - Yank (paste) Ctr-y

- **Search**
  - Forward Ctr-s
  - Backward Ctr-r
  - Repeat last search Ctr-s ili Ctr-r
  - Previous string Alt-p
  - Next string Alt-n
  - Exit search RET
  - Abort Ctr-g
- **Replace**
  - interactively Alt-% ili Esc %

Editor Emacs, naredbe, nastavak

- **Razno**
  - Autom. prelom mode — Alt-x auto-fill-
  - Fill paragraf — Alt-q
  - excange pt. and mark — Ctr-x Ctr-x
  - mark ring — Ctr-u Ctr-spac
- **Printing**
  - Alt-x print-buffer/region
  - Alt-x ps-print-buffer/region

- **Help**
  - key — Ctr-h c
  - function — Ctr-h f
  - mode — Ctr-h m
  - apropos — Ctr-h a