



# Programiranje za UNIX

---

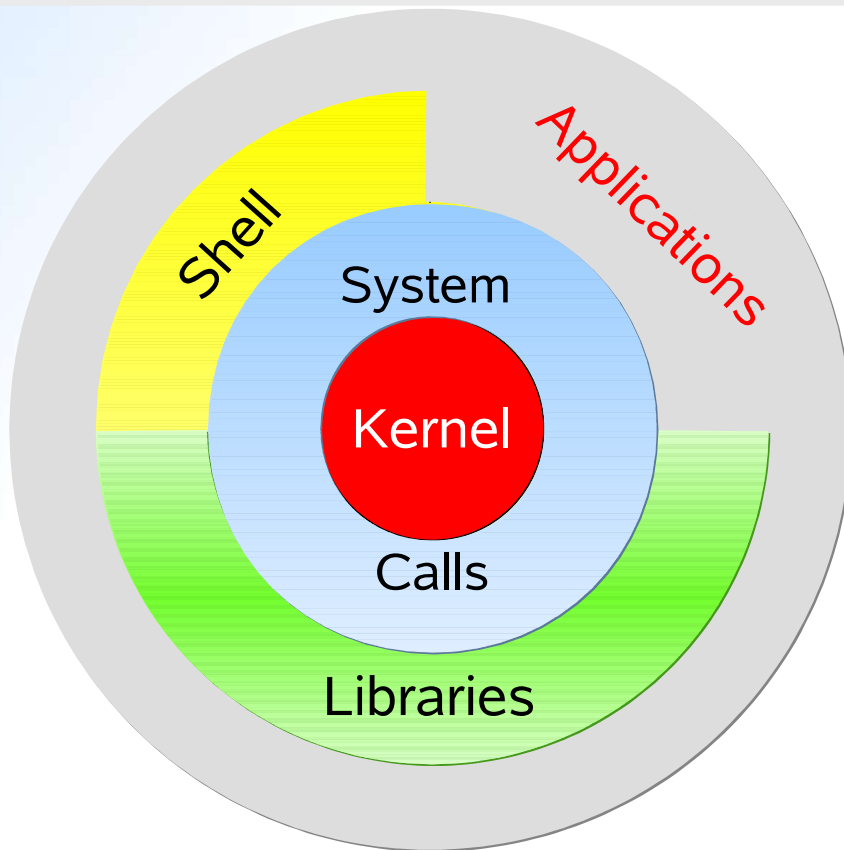
**Uvod**

## Sadržaj

---

- UNIX Arhitektura
- X-Windows
- Programska okruženja
- UNIX standardi





3

## UNIX slojevi

---

➤ **Kernel(Jezgra)** – centralni dio operacijskog sustava:

- Upravlja računalnim sklopovljem, dijeli resurse između više korisnika i procesa
- Sučelje za programiranje koje generalizira pristup sklopovlju

➤ **Sistemske pozivi** – Skup funkcija za pristup servisima jezgre

➤ **Libraries** – Datoteke srodnih funkcija više razine koje dodatno olakšavaju upravljanje operacijskim sustavom

4

## ➤ Shell (ljuska) – interpreter naredbenog retka

- Primanje korisničkih naredbi na standardnom ulazu
- Upravljanje funkcijama jezgre i pokretanje korisničkih programa
- Prikaz rezultata na standardnom izlazu

## ➤ Aplikacije

## ➤ Datotečni sustav

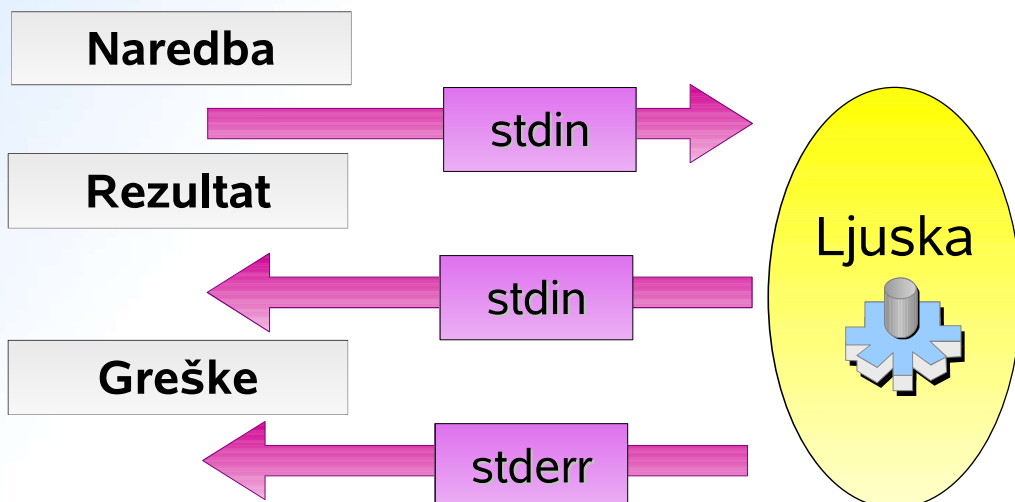


5

## Ljuska (shell)

### ➤ Format UNIX naredbe

**/> naredba [opcije] [argumenti]**



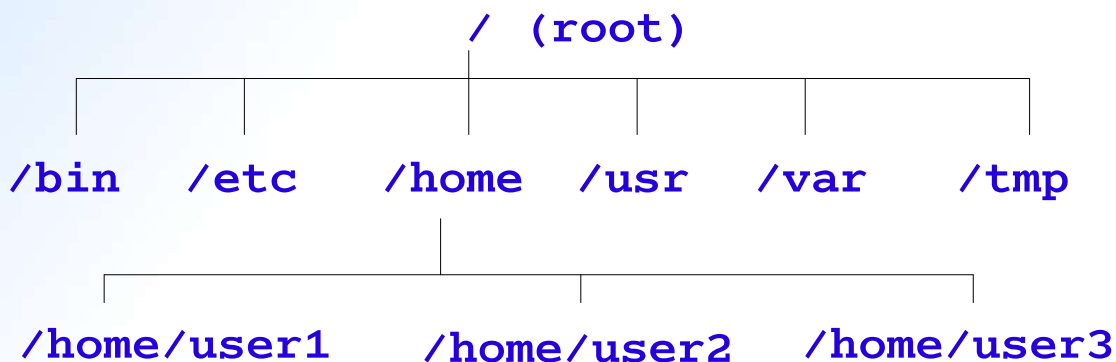
**man** – pomoć za unix naredbe



6

# Datotečni sustav (file system)

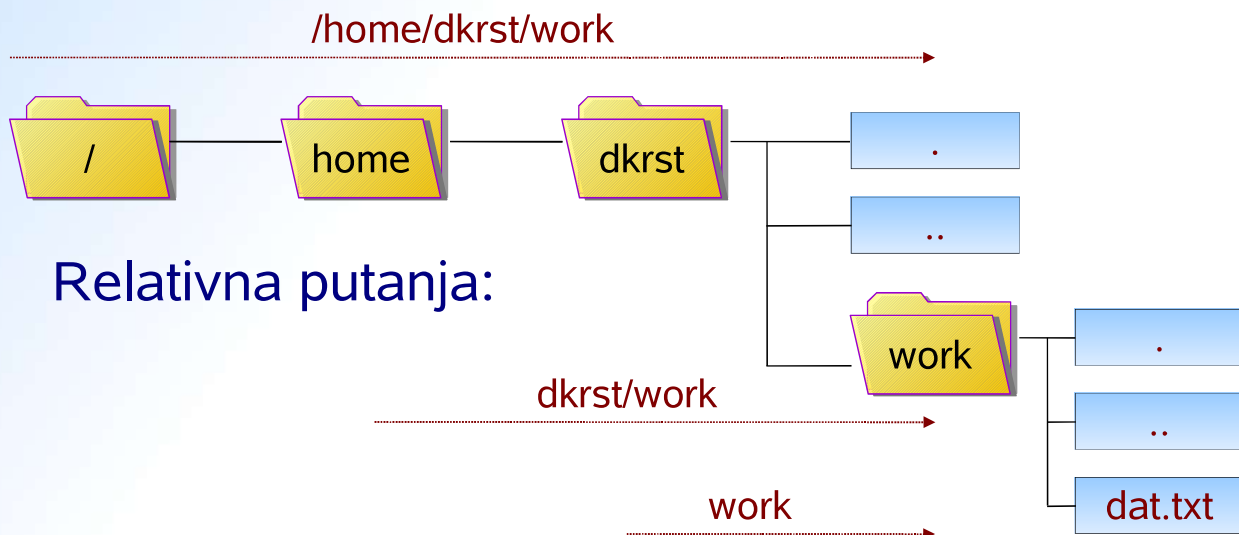
➔ UNIX datotečni sustav organiziran je u obliku stabla



7

## Struktura datotečnog sustava

Apsolutna putanja (path):



Relativna putanja:

**cd** – promjena radnog direktorija

8

# Struktura direktorija

➔ Svaki direktorij može sadržavati datoteke raznog tipa, uključujući i druge direktorije

```
/home/dkrst/work> ls -al
```

drwx-----	3	dkrst	users	512	2008-03-10	.
drwx-----	4	dkrst	users	512	2008-03-10	..
-rwx-----	7	dkrst	users	4608	2008-03-11	dat.txt

roditeljski direktorij

radni direktorij

Svaki direktorij sadrži samog sebe  
i svoj roditeljski direktorij

datoteka

**ls** – pregled sadržaja direktorija

9

## Skraćene oznake direktorija

- . - radni direktorij
- .. - roditeljski direktorij
- ~ - home direktorij korisnika

```
/> cd ~  
/home/dkrst> pwd  
/home/dkrst  
/home/dkrst> cd ..  
/home> cd dkrst/work  
/home/dkrst/work> pwd  
/home/dkrst/work  
/home/dkrst/work> cd .  
/home/dkrst/work> cd /home  
/home> pwd  
/home
```

**pwd** – ispis trenutnog radnog direktorija

10

## Tipovi datoteka

---

### ➤ Obične datoteke

- Pohranjene informacije bilo kojeg tipa (tekst, slike, programi, ...)

### ➤ Direktoriji

- Točke grananja datotečnog stabla, sadrže druge datoteke bilo kojeg tipa

### ➤ Simbolički linkovi

- Datoteke koje pokazuju na druge datoteke bilo kojeg tipa



11

## Tipovi datoteka

---

### ➤ Specijalne datoteke

- Točke u datotečnom sustavu koje predstavljaju računalno sklopovlje (diskovi, ulazno/izlazne jedinice, ...)

### ➤ Socket

- Koriste se za komunikaciju među procesima, lokalno (UNIX domain) ili preko mreže (network domain)

### ➤ Pipes (cjevovodi)

- Koriste se za lokalnu komunikaciju među procesima



12

# SVE JE DATOTEKA!

---

- Normalne datoteke
- Direktoriji
- Komunikacijske pristupne točke
- Sklopovlje računala
- ...

➔ Svi dijelovi UNIX operativnog sistema imaju sučelje koje funkcionira po principu datoteke



13

## Vlasništvo nad datotekama

---

### ➔ Vlasnik (user)

- Jedan točno određen korisnik sustava koji određuje prava na datoteci

### ➔ Grupa (group)

- Svaka datoteka u skupnom je vlasništvu grupe korisnika
- Datoteka pripada samo jednoj grupi
- Korisnici sustava mogu biti u jednoj ili više grupa
- Članovi grupe mogu se mijenjati i nakon kreiranja datoteke



## Prava pristupa na datotekama

### ➤ Prava pristupa određuje vlasnik datoteke:

- čitanje **r**
- pisanje **w**
- izvršavanje **x**

### ➤ Prava pristupa zasebno se definiraju za vlasnika, grupu i ostale korisnike sustava

### ➤ Iznimno, prava pristupa na svim datotekama može zadavati i **root** korisnik (administrator sistema)



<number>

## Prava pristupa datotekama

```
-rw-r--r-- 1 dkrst users  
_ r w x r w x r w x  
| vlasnik | grupa | ostali |
```

### Primjer:

```
-rw-r--r--
```

- vlasnik: pravo čitanja (**r**) i pisanja (**w**)
- grupa: pravo čitanja (**r**)
- ostali: Pravo čitanja (**r**)



**chmod** – promjena prava pristupa

<number>



# Apsolutno zadavanje prava

---

## Apsolutna prava pristupa:

= r w x r w x r w x  
1/0 | 1/0 | 1/0 | 1/0 | 1/0 | 1/0 | 1/0 | 1/0 | 1/0  
4 | 2 | 1 | 4 | 2 | 1 | 4 | 2 | 1

- Apsolutna prava pristupa dobivaju se pojedinačno za vlasnika, grupu i ostale korisnike zbrajanjem vrijednosti tri binarne znamenke koje određuju prava pristupa

`chmod 0644 dat1.txt`  
**vlasnik:** `rw`; **grupa:** `r`; **ostali:** `r`



<number>

## Simbolička prava pristupa

---

- Dodavanje ili oduzimanje prava pristupa na osnovi simboličke oznake

`u` – vlasnik  
`g` – grupa  
`o` – ostali  
`a` – svi korisnici

`chmod ug+w dat1.txt`  
dodavanje prava pisanja za vlasnika i grupu  
`chmod o-w dat1.txt`  
skidanje prava čitanja za ostale korisnike sustava  
`chmod a+r dat1.txt`  
davanje prava čitanja za sve korisnike



<number>

## Prava pristupa za direktorije

---

➤ Što znače pravo čitanja, pisanja i izvršavanja primijenjena na direktorij?



<number>

## Prava pristupa za direktorije

---

➤ Što znače pravo čitanja, pisanja i izvršavanja primijenjena na direktorij?

**r** (čitanje) – pravo pregledavanja sadržaja direktorija

**w** (pisanje) – pravo izmjene sadržaja direktorija

**x** (izvršavanje) – pravo otvaranja datoteka u direktoriju



<number>

# Programi i procesi

## ➤ Program je izvršna datoteka:

- Izvorni kod preveden i povezan u naredbe strojnog koda koje se direktno izvršavaju
- Skup naredbi programskog jezika (npr. **shell skripte**) koji se interpretira u trenutku pokretanja

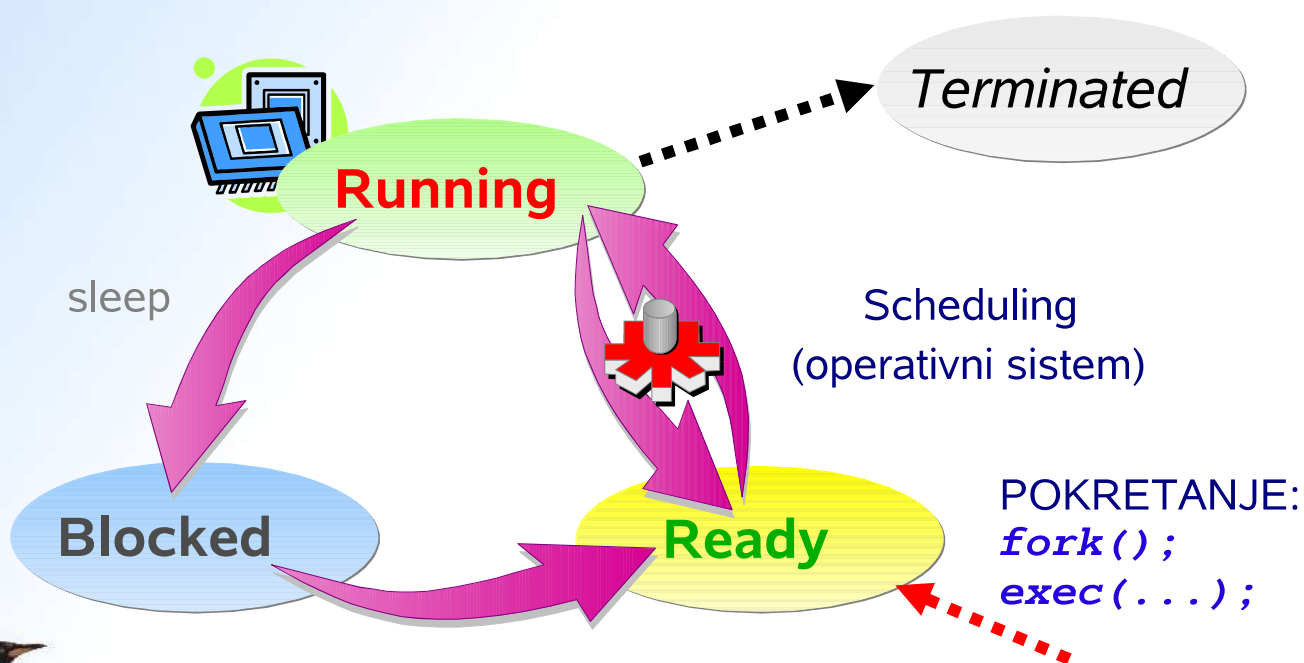
## ➤ Aktivni entitet u memoriji računala koji se izvršava na računalnom sklopovlju

- **PID, PPID** – JMBG procesa i njegovog roditelja
- **UID, GID** – stvarno pojedinačno i grupno vlasništvo
- **EUID, EGID** – efektivno vlasništvo
- **Prioritet procesa**

**ps** – pregled aktivnih procesa

<number>

## Životni ciklus procesa



**top** – aktivni procesi u stvarnom vremenu

<number>

# Signali

- Obavijesti o asinkronim događajima koje jezgra šalje procesu
  - Proces može instalirati *signal handler* (funkciju koja se poziva kada stigne signal) ili blokirati većinu signala signal
- Najjednostavniji način komunikacije među procesima
  - Proces može zatražiti slanje signala drugom procesu (sistemski poziv **kill**)

**kill** – slanje signala procesu

<number>

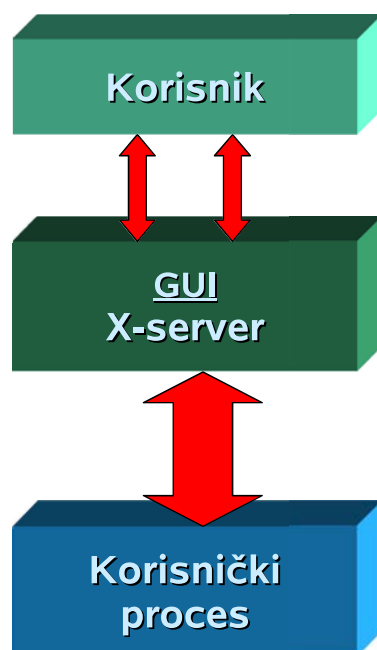


## X windows klijent/server arhitektura

### ➤ X-server

- Upravlja sklopovljem za grafički prikaz
- Sučelje između korisnika i njegovih procesa
- Standardni set funkcija (za programera) za kreiranje grafičkog sučelja

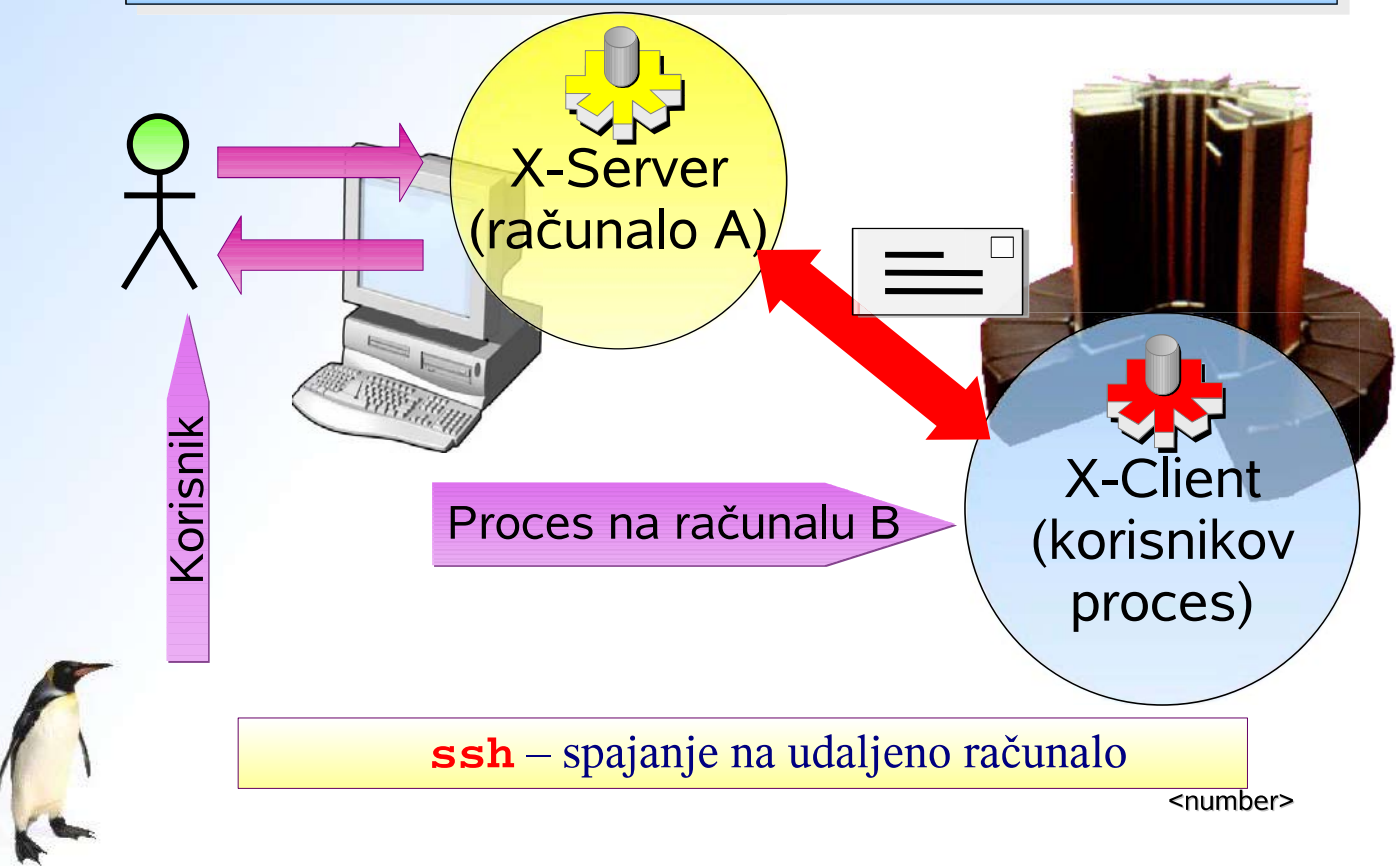
➤ Proces i sučelje procesa prema korisniku ne moraju nužno biti na istom računalu!



<number>

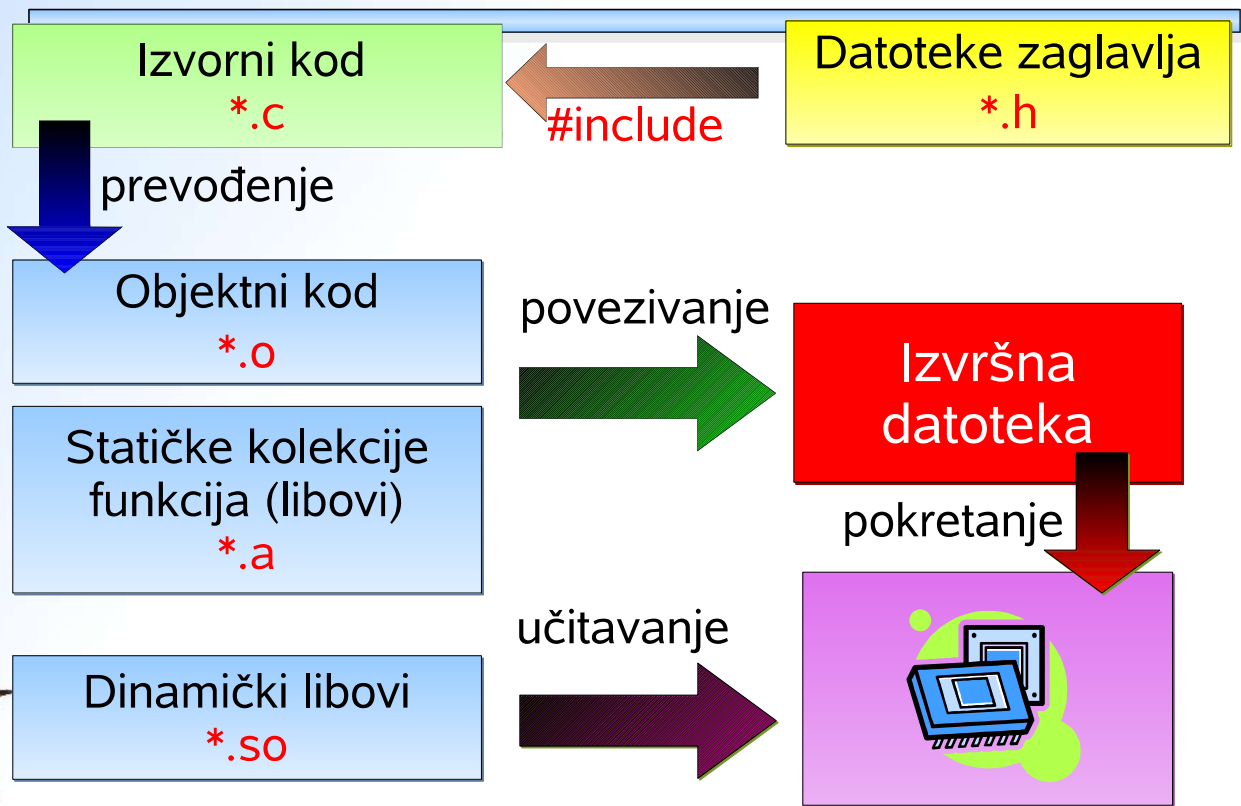


# X windows klijent/server arhitektura



<number>

## Prevođenje i povezivanje programa



<number>

- **gcc** – GNU C kompajler i linker
- Besplatan, otvoreni kod, prenosivost

Sintaksa:

**gcc [-opcije] ulazne\_datoteke**

- Ulazne datoteke: **\*.c, \*.o, \*.a, ...**

**Prevođenje i povezivanje dva su odvojena procesa!**



<number>

## gcc

---

- Osnovne opcije:

- **-C** – samo prevođenje (rezultat je objektna datoteka)
- **-Olevel** – razina optimizacije (1-3)
- **-g** – uključi informaciju za debugger
- **-I**dir**** – uključi **dir** u listu direktorija sa datotekama zaglavlja
- **-L**dir**** – uključi **dir** u listu direktorija sa kolekcijama funkcija (libovima)
- **-Wall** – prikaži sva upozorenja
- **-o **ime**** – ime izlazne datoteke



<number>

## Kolekcije funkcija - libovi

---

- Arhiva objektnih funkcija (library) je jedna datoteka koja u sebi sadrži više funkcija prevedenih u objektni kod na način da je svaku od njih moguće izdvojiti u izvornom obliku

**Lib se kao ulaz u gcc koristi na isti način kao bilo koja druga objektna datoteka**

- Prilikom povezivanja (linkanja) **gcc** iz liba izdvaja potrebne funkcije i koristi ih na isti način kao i druge (direktno zadane) datoteke objektnog koda

**ar** – manipuliranje libovima

<number>



## **make** - automatiziranje prevođenja i povezivanja

---

- **make** rutina čita skup pravila za prevođenje i povezivanje opisan u datoteci

**make -f <datoteka> <opcije>**

- Ako ime datoteke sa pravilima nije navedeno, učitava se datoteka **Makefile**
- Provjerava se vrijeme zadnjeg pokretanja postupka prevođenja i povezivanja
- Izvršavaju se samo oni koraci koji su zastarjeli



<number>

## Struktura **Makefile** datoteke

<tab>

**target : dependencies**  
**commands**

- **target** - naziv pravila (ime programa) koje želimo pozvati
- **dependencies** - datoteke o kojima ovisi prevođenje
- **commands** - akcija koja se izvršava za generiranje datoteke

**Redak sa naredbama obavezno započinje sa <tab>**

<number>

## **Makefile** varijable

- Varijabla je simboličko ime definirano u **Makefile** datoteci, a predstavlja tekstualni string

**FLAGS = -O3 -Wall**

ime varijable

vrijednost (string)

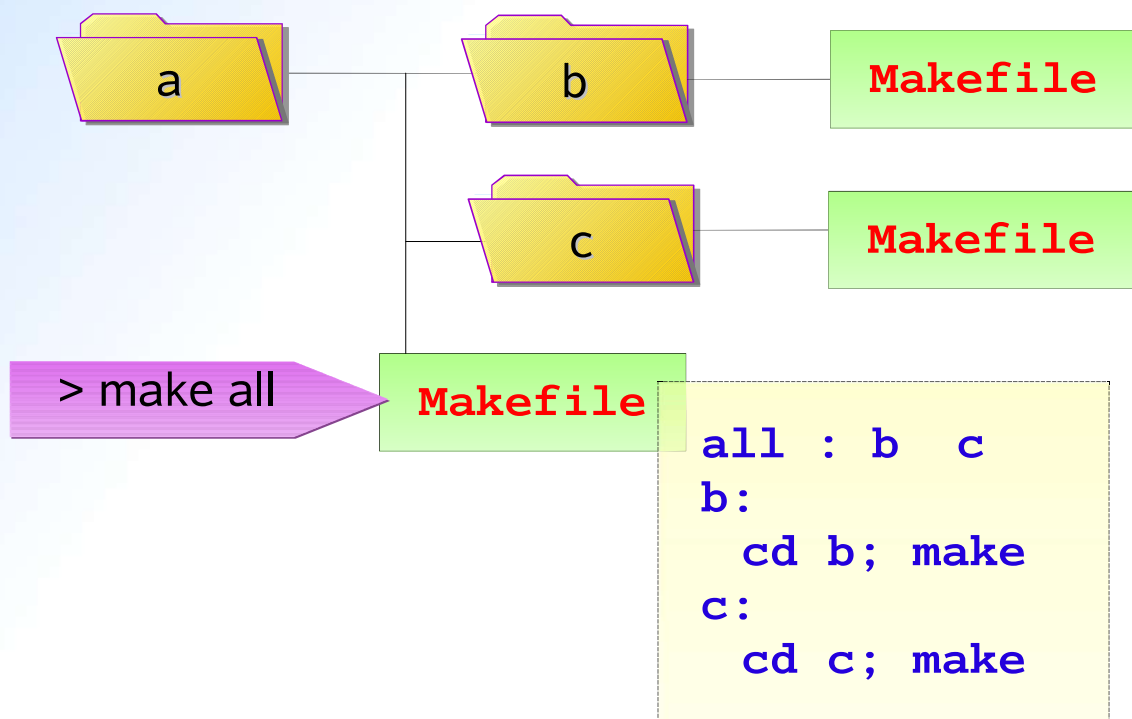
- Vrijednost varijable referencira se korištenjem oznake iza koje slijedi ime varijable u zagradama

**\$(FLAGS)**

<number>



# Rekurzivni **Makefile**



<number>

## Primjer **Makefile** datoteke

```
CC = /usr/bin/gcc
CFLAGS = -Wall -O3
TARGETS = hello count
default: count
all: $(TARGETS)

hello: hello.o
    $(CC) $(CFLAGS) hello.o -o hello

count: count.o fn.o
    $(CC) $(CFLAGS) count.o fn.o -o count

clean :
    rm -f *.o *~ $(TARGETS)

.c.o:
    $(CC) $(CFLAGS) -c $<
```

<number>

## ISO C

---

- Osigurava **prenosivost C programa** na različite platforme
- Kolekcije funkcija (libovi) podijeljeni su u 24 područja bazirana na zaglavljima
  - **stdio.h** – standard IO library
  - **stdlib** – standard utility functions
  - **string.h** – standard string operations
  - **limits.h** – implementation constants
  - **errno.h** – error codes
  - ...

<number>



## IEEE POSIX

---

- 1003.1 OS interface standard – POSIX.1
  - Promiče prenosivost aplikacija na različite UNIX platforme
  - Uključuje ISO C i dodatna opcionalna zaglavlja
- Neka obavezna POSIX zaglavlja:
  - **unistd.h** – symbolic constants
  - **sys/socket.h** – socket interfaces
  - **sys/types** – primitive system data types
- Opcionalna POSIX zaglavlja
  - **pthread.h** – threads

<number>

