

# Komunikacija sa Operativnim Sistemom

POGLAVLJE 16

# Programski nivo komunikacije sa OS

---

- Komunikacija sa operativnim sistemom na programskom nivou se ostvaruje pozivanjem **sistemskih operacija**.
- Znači da bi se u toku izvršavanja korisničkog programa dobila neka **usluga** od operativnog sistema, potrebno je pozvati odgovarajuću **sistemsku operaciju**.
- Na primer, da bi se **preuzeo znak** sa tastature, u korisničkom programu je neophodno navesti poziv **odgovarajuće sistemske operacije**.

# Interaktivni nivo komunikacije sa OS

---

- **Interaktivni** nivo korišćenja operativnog sistema se ostvaruje pomoću **komandi** komandnog jezika.
- One, na primer, omogućuju rukovanje **datotekama** i **procesima**.
- **Najjednostavniju** komandu komandnog jezika predstavlja **putanja izvršne datoteke**.
- Kao operand ovakve komande se može, opet, javiti **putanja datoteke**, ako je komanda namenjena za rukovanje datotekama.

# Interaktivni nivo komunikacije sa OS

---

- Prema tome, na prethodni način oblikovana komanda započinje **operatorom** u obliku **putanje izvršne datoteke**, koja opisuje rukovanje, a završava **operandom (ili operandima)** u obliku **putanja datoteka**, kojima se rukuje. Tako:

**–kopiraj godina1.txt godina2.txt**

- predstavlja primer prethodno opisane komande znakovnog komandnog jezika.

# Interaktivni nivo komunikacije sa OS

---

- U ovom primeru se pretpostavlja da radni imenik obuhvata **izvršnu datoteku** sa imenom **kopiraj.bin** i **tekst** datoteku **godina1.txt**.
- Takođe se pretpostavlja da izvršavanje programa iz izvršne datoteke **kopiraj.bin** dovodi do stvaranja datoteke **godina2.txt**, koja je po sadržaju **identična** datoteci **godina1.txt** i koja pripada radnom imeniku.
- Prethodna komanda opisuje korisnu operaciju, ako svi studenti prve godine studija upisuju drugu godinu studija.

# Interaktivni nivo komunikacije sa OS

---

- Prethodno opisani način zadavanja komandi odgovara **znakovnom** komandnom jeziku.
- Komandni jezik može **olakšati** zadavanje komandi, ako omogući korisniku da **operator** komande bira u **spisku operatora (menu)**, umesto da ga **pamti** i u celosti navodi.

# Interaktivni nivo komunikacije sa OS

---

- **Spisak** operatora se prikazuje na ekranu, a izbor operatora se vrši pomoću namenskih dirki **tastature** ili **miša**.
- Nakon izbora operatora sledi, po potrebi, **dijalog** u kome korisnik navodi (ili opet bira) **operand** (**operande**) komande.
- Ovakvi komandni jezici se nazivaju **grafički komandni jezici** (**menu driven user interface**, **graphical user interface - GUI**).

# Interaktivni nivo komunikacije sa OS

---

- Oni još više **pojednostavljaju** zadavanje komandi, ako korisniku omogućuju da ne bira **operator**, nego samo **operande** komandi.
- U ovom slučaju, izbor operanda se svodi na izbor nekog od **imena datoteka**, prikazanih na ekranu, a operator se **podrazumeva** ili na osnovu tipa odabrane datoteke, ili, eventualno, na osnovu upotrebljene namenske dirke.



# Interaktivni nivo komunikacije sa OS

---

- U svakom slučaju, zadatak komandnog jezika je da omogući korisniku da zada **komandu**, koja precizno određuje i **vrstu rukovanja** i **objekat rukovanja**, a zadatak **interpretiranja komande** je da **pokrene proces**, u okviru čije aktivnosti usledi **rukovanje**, zatraženo komandom.

# Znakovni komandni jezici

---

- Izgled, način rada i mogućnosti **interpretera znakovnog** komandnog jezika (**command language interpreter, shell**) zavise od ciljeva, koje komandni jezik treba da ostvari.
- Ciljevi znakovnih komandnih jezika obuhvataju:
  - 1)omogućavanje **izvršavanja** pojedinih (korisničkih) programa
  - 2)omogućavanje **kombinovanja izvršavanja** više (korisničkih) programa
  - 3)omogućavanje **pravljenja komandnih datoteka** (**command file, shell script**).

# Znakovni komandni jezici

---

- Interpreter znakovnog komandnog jezika ostvaruje prethodne ciljeve tako što sa standardnog ulaza **prima niz znakova**, koji obrazuju **komandu**, prepoznaje u tom nizu znakova **operator komande** (i, eventualno, **njene operande**) i preduzima zahtevanu **akciju**.
- Rezultat svoje akcije ovaj interpreter prikazuje na **standardnom izlazu**.

# Znakovni komandni jezici

---

- Prilikom preduzimanja zahtevane akcije, interpreter znakovnog komandnog jezika se oslanja na **sistemske operacije**.
- Pri tome, on koristi **delove komandi**, odnosno njen **operator** i njene **operande**, kao **argumente sistemskih operacija**.
- Na primer, do izvršavanja pojedinih korisničkih programa dolazi tako što interpreter znakovnog komandnog jezika poziva **sistemske operacije stvaranja procesa**, a kao njene argumente upotrebi **operator** i **operande** komande, odnosno **putanju izvršne datoteke** i **putanje datoteka sa obrađivanim podacima**.

# Znakovni komandni jezici

---

- Ovi argumenti su namenjeni **stvaranom** procesu.
- Tako, na primer, kod programskog jezika C parametri funkcije **main()** omogućuju preuzimanje broja argumenata (**argc**) iz **komandne linije**, kao i **stringova pojedinih argumenata** (**argv**), jer su ovi smešteni na **stek procesa** prilikom njegovog stvaranja.

# Znakovni komandni jezici

---

- Interpreter znakovnog komandnog jezika obavlja obradu znakova komande, pre nego što ih iskoristi kao **argumente** neke systemske operacije.
- Zahvaljujući tome, u okviru operanada komandi se mogu javiti **specijalni znakovi** (**magic character, wild cards**), kao što je, na primer, znak **\***.
- Njegova upotreba je vezana, pre svega, za **imena datoteka** i namenjena je za **skraćeno** označavanje grupa datoteka. Tako, na primer:

# Znakovni komandni jezici

---

## **\*.obj**

•označava sve objektne datoteke u radnom imeniku, a

## **d\*1.txt**

•označava sve tekst datoteke u radnom imeniku, čiji prvi deo imena **započinju znakom d**, a **završava cifrom 1**.

•Zahvaljujući specijalnim znakovima, moguće je, na primer, **jednom komandom uništiti** sve objektne datoteke iz radnog imenika:

**unisti \*.obj**

# Znakovni komandni jezici

---

- ili odštampati sve tekst datoteke iz radnog imenika, čiji prvi deo imena započinje znakom **d**, a završava znakom **1**:

**stampaj d\*1.txt**

- Za obavljanje ovakvih akcija, neophodno je **pretraživanje** imenika i **provera imena** datoteka.
- Zahvaljujući obradi znakova komande, moguće je interpreteru znakovnog komandnog jezika saopštiti i da **preusmeri (redirect)** standardni ulaz i standardni izlaz sa tastature i ekrana na proizvoljno odabrane datoteke.



# Znakovni komandni jezici

---

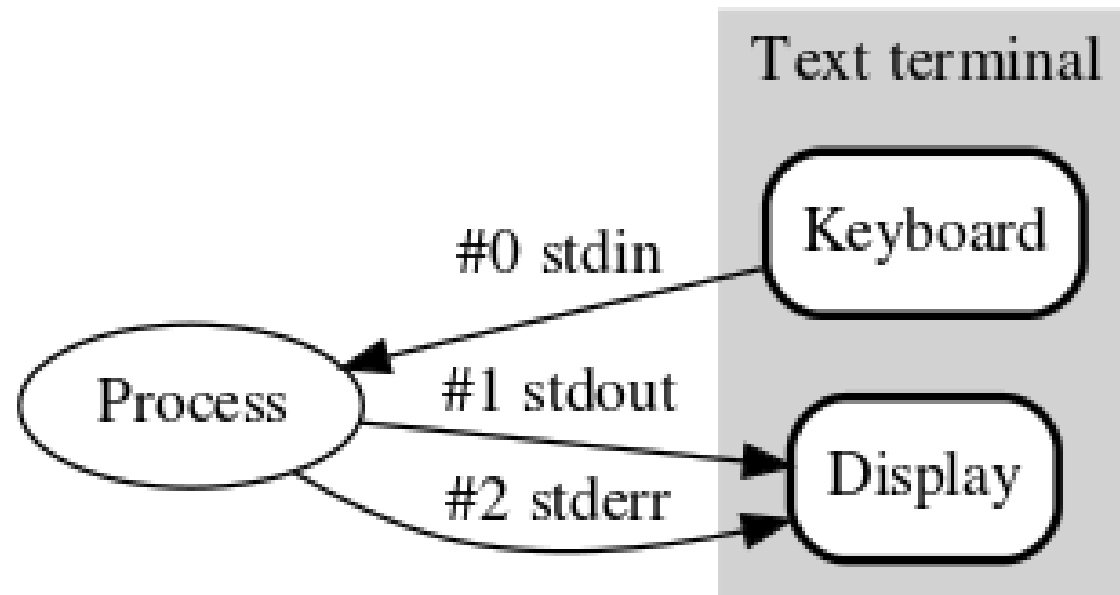
- Ovo je važno za **pozadinske** procese, koji nisu u interakciji sa korisnikom.
- Zahvaljujući **preusmeravanju**, pozadinski proces ne ometa **interaktivni** rad korisnika, jer, umesto tastature i ekrana, koristi **odabrane datoteke**.
- Međutim, da bi se korisnik upozorio na **greške** u toku aktivnosti pozadinskog procesa, uz standardni izlaz se uvodi i **standardni izlaz greške**.

# Znakovni komandni jezici

---

- Pošto je standardni **izlaz greške** namenjen, pre svega, za prikazivanje **poruka o greškama**, kao podrazumevajući standardni izlaz greške služi **specijalna datoteka**, koja odgovara ekranu.
- Ova datoteka se otvara za vreme **stvaranja procesa**, a kao njen **indeks** za tabelu otvorenih datoteka može da služi vrednost **2**.
- I standardni izlaz greške se može preusmeriti na **proizvoljnu datoteku**, čiji sadržaj tada ukazuje na eventualne greške u toku aktivnosti pozadinskog procesa.

# Znakovni komandni jezici



Izvor: <https://upload.wikimedia.org/wikipedia/commons/7/70/Stdstreams-notitle.svg>

# Znakovni komandni jezici

---

- Uobičajeno je da preusmeravanje standardnog ulaza najavljuje znak `<`, a da preusmeravanje **standardnog izlaza** (kao i standardnog izlaza greške) najavljuje znak `>`. Tako, na primer, komanda:

**kompiliraj < program.c > program.obj**

- saopštava interpreteru znakovnog komandnog jezika da stvori proces na osnovu izvršne datoteke **kompiliraj.bin**, pri čemu kao standardni ulaz procesa služi datoteka **program.c**, a kao njegov standardni izlaz datoteka **program.obj**.

# Znakovni komandni jezici

---

- U ovom primeru **ekran** i dalje služi kao **standardni izlaz greške**.
- Izvršavanje prethodne komande omogućuje kompilaciju C programa, sadržanog u datoteci **program.c**.
- Rezultat kompilacije se smešta u datoteku **program.obj**, a eventualne poruke o greškama kompilacije se prikazuju na ekranu.

# Znakovni komandni jezici

---

- **Preusmeravanje** standardnog ulaza i izlaza predstavlja osnovu za **kombinovanje** izvršavanja više korisničkih programa.
- Važnost pomenutog **kombinovanja** se može pokazati na primeru **uređivanja (sortiranja)** reči iz nekog rečnika po kriterijumu **rimovanja**.
- Nakon **sortiranja** reči po ovom **kriterijumu**, sve reči, koje se rimuju, nalaze se **jedna uz drugu**.

# Znakovni komandni jezici

---

- Umesto pravljenja **posebnog** programa za **sortiranje** reči po kriterijumu **rimovanja**, jednostavnije je napraviti program za **obrtanje redosleda znakova** u rečima (tako da prvi i poslednji znak zamene svoja mesta u reči, da drugi i preposlednji znak zamene svoja mesta u reči i tako redom) i kombinovati izvršavanje ovog programa sa izvršavanjem postojećeg programa za sortiranje:

**obrni < recnik.txt > obrnuti\_recnik.txt**

**sortiraj < obrnuti\_recnik.txt > sortirani\_obrnuti\_recnik.txt**

**obrni < sortirani\_obrnuti\_recnik.txt > rime.txt**

# Znakovni komandni jezici

---

- Umesto preusmeravanja standardnog ulaza i standardnog izlaza, moguće je **nadovezati** standardni izlaz **jednog** procesa na standardni ulaz **drugog** procesa i tako obrazovati **tok procesa (pipe)**.
- Nadovezivanje u tok se označava pomoću znaka |. Za prethodni primer ovakav tok bi izgledao:

**obrni < recnik.txt | sortiraj | obrni > rime.txt**



# Znakovni komandni jezici

---

- U ovom primeru su u tok nadovezana tri procesa. Prvi je nastao na osnovu komande:  
**obrni < rechnik.txt**

- drugi je nastao na osnovu komande  
**sortiraj**

- a treći je nastao na osnovu komande:  
**obrni > rime.txt**

- Ako bi se reči zadavale sa tastature, a po rimama sortirani rečnik prikazivao na ekranu, prethodni tok bi izgledao:  
**obrni | sortiraj | obrni**

# Znakovni komandni jezici

---

- Razmena podataka između dva procesa, koji su povezani u **tok**, se ostvaruje posredstvom posebne **specijalne** datoteke.
- Njoj odgovara **bafer** u radnoj memoriji.
- Ova **baferovana** specijalna datoteka služi prvom od ovih procesa kao standardni **izlaz**, a drugom od njih kao standardni **ulaz**.
- Znači, prvi proces samo **piše** u ovu datoteku, a drugi samo **čita** iz nje.

# Znakovni komandni jezici

---

- Prilikom obrazovanja toka procesa, interpreter znakovnog komandnog jezika stvara procese, koji se povezuju u **tok**.
- Pri tome on koristi istu posebnu **specijalnu** datoteku kao standardni **izlaz** i **ulaz** za svaki od **parova** ovih procesa.
- Zatim interpreter znakovnog komandnog jezika čeka na kraj aktivnosti **poslednjeg** od ovih procesa.

# Znakovni komandni jezici

---

- **Pozadinski** procesi se razlikuju od **običnih (interaktivnih)** procesa po tome što **interpreter** znakovnog komandnog jezika, nakon stvaranja pozadinskog procesa, **ne čeka kraj** njegove aktivnosti, nego nastavlja interakciju sa korisnikom.
- Zato su pozadinski procesi u principu **neinteraktivni**.
- Na primer, komanda:  
**kompiliraj < program.c > program.obj &**

# Znakovni komandni jezici

---

- Omogućuje stvaranje **pozadinskog procesa**, koji obavlja **kompilaciju** programa, sadržanog u datoteci **program.c**, i rezultat kompilacije smešta u datoteku **program.obj**, a eventualne greške u kompilaciji prikazuje na ekranu.
- U prethodnom primeru znak **&** sa kraja komande je naveo interpreter znakovnog komandnog jezika na stvaranje **pozadinskog** procesa.

# Znakovni komandni jezici

---

- Svaka komanda, upućena interpreteru znakovnog komandnog jezika, **ne dovodi** do stvaranja procesa.
- Komande, koje se **često** koriste, pa je važno da budu **brzo** obavljene, interpreter znakovnog komandnog jezika obavlja **sam**.
- Za **ostale** komande, za koje se stvaraju **procesi**, interpreter znakovnog komandnog jezika čeka **kraj** aktivnosti stvorenog procesa da bi od njega dobio, kao povratnu informaciju, **završno stanje** stvorenog procesa.

# Znakovni komandni jezici

---

- Ovo stanje se obično kodira **celim brojem**.
- Ako interpreter znakovnog komandnog jezika protumači ovaj broj kao logičku vrednost (**0 - tačno, različito od 0 - netačno**), tada on može da podrži uslovno izvršavanje programa. Tako, na primer, komanda:  
**if kompiliraj < program.c > program.obj**  
**then**  
**linkuj < program.obj > program.bin**  
**fi**

# Znakovni komandni jezici

---

- označava da do linkovanja dolazi samo nakon **uspešne kompilacije**.
- Pri tome su **if**, **then** i **fi** rezervisane reči za interpreter znakovnog komandnog jezika.
- Prva **najavljuje** komandu, na osnovu koje interpreter znakovnog komandnog jezika **stvari proces**.



# Znakovni komandni jezici

---

- Za vreme aktivnosti ovog procesa usledi kompilacija programa, sadržanog u datoteci **program.c**.
- Ako kompilacija prođe **bez grešaka**, stvoreni proces vraća interpreteru znakovnog komandnog jezika vrednost **0** kao svoje završno stanje.

# Znakovni komandni jezici

---

- U ovom slučaju, interpreter znakovnog komandnog jezika interpretira **komandu (komande)** između rezervisanih reči **then** i **fi** i stvara proces, čija aktivnost dovodi do **linkovanja** datoteke **program.obj** sa potprogramima iz sistemske biblioteke.
- Ime sistemske biblioteke se ne navodi, jer se **podrazumeva**.
- U suprotnom slučaju, ako je bilo **grešaka** u kompilaciji, pa je završno stanje procesa, zaduženog za kompilaciju, bilo **različito od vrednosti 0**, interpreter znakovnog komandnog jezika ne interpretira **komandu (komande)** između rezervisanih reči **then** i **fi** i ne stvara proces zadužen za linkovanje.

# Znakovni komandni jezici

---

- Pored prethodno opisane komande za uslovno izvršavanje programa, interpreteri znakovnih komandnih jezika podržavaju komandu za **ponavljanje** izvršavanja programa, ali i druge komande, tipične za procedurene programske jezike, koje omogućuju rukovanje **promenljivim, konstantama, ulazom, izlazom** i slično.
- To dozvoljava pravljenje **komandnih datoteka (BASH files)**.
- One opisuju okolnosti pod kojima se izvršavaju korisnički programi, a sadržaj komandne datoteke preuzima na interpretiranje **interpreter** znakovnog komandnog jezika.

# Znakovni komandni jezici

---

- Zato komandne datoteke imaju **poseban tip**, da bi ih interpreter znakovnog komandnog jezika mogao prepoznati.
- Zahvaljujući tome, ime svake komandne datoteke, uostalom, kao i ime svake izvršne datoteke, predstavlja ispravnu komandu znakovnog komandnog jezika.
- Iako broj i vrste ovakvih komandi nisu ograničeni, jer zavise samo od kreativnosti i potreba korisnika, ipak je moguće napraviti njihovu klasifikaciju i navesti neke neizbežne **grupe komandi**.

# Znakovni komandni jezici

---

- Najgrublja podela komandi je na:

- 1) korisničke komande

- 2) administratorske komande

- Korisničke komande, između ostalog, omogućuju:

- 1) rukovanje datotekama

- 2) rukovanje imenicima

- 3) rukovanje procesima

- 4) razmenu poruka između korisnika

# Znakovni komandni jezici

---

• Standardne komande za rukovanje datotekama omogućuju:

1) izmenu imena (kao i atributa) datoteke (**mv**)

2) poređenje sadržaja datoteka (**diff**)

3) kopiranje datoteka (**cp**)

4) uništenje datoteka (**rm**)

# Znakovni komandni jezici

---

• U komande za rukovanje imenicima spadaju:

- 1) komande za stvaranje i uništenje imenika (**mkdir**)
- 2) komanda za promenu radnog imenika (**cd**)
- 3) komanda za pregledanje sadržaja imenika (imena datoteka i imena imenika, sadržanih u njemu) (**ls**)
- 4) komande za izmenu imena i ostalih atributa imenika. (**mv**)

# Znakovni komandni jezici

---

•Administratorske komande omogućuju:

1)pokretanje i zaustavljanje rada računara,

2)spašavanje (backup) i vraćanje (restore) datoteka rukovanje vremenom

3)rukovanje vremenom

4)sabijanje (compaction) datoteka

5)ažuriranje podataka o korisnicima računara i njihovim pravima

6)generisanje izveštaja o korišćenju računara (o korišćenju procesorskog vremena ili o korišćenju prostora na disku)



# Znakovni komandni jezici

---

7) rukovanje konfiguracijom računara (određivanje načina rada uređaja i programa koji ulaze u njegov sastav)

8) proveru ispravnosti rada računara

1) pripremu diskova za korišćenje (ovo obuhvata pronalaženje oštećenih blokova i njihovo isključivanje iz upotrebe, pronalaženje izgubljenih blokova i njihovo uključivanje u evidenciju slobodnih blokova, formiranje skupa datoteka na disku i njegovo uključivanje u skup datoteka računara).

# Grafički komandni jezici

---

- Interpreteri grafičkih komandnih jezika omogućuju pozivanje bilo koje od prethodnih komandi, a da pri tome **ne zahtevaju** od korisnika da znaju **napamet** imena komandi, niti da zadaju komande posredstvom tastature, uz obavezu strogog poštovanja sintakse znakovnog komandnog jezika.
- Umesto toga, grafički komandni jezici uvode grafičku predstavu komandi (**icon**), ili spiskove sa imenima komandi (**menu**), dozvoljavajući korisnicima da pozovu komandu izborom njene grafičke predstave, ili izborom njenog imena sa spiska imena komandi.

# Grafički komandni jezici

---

- Grafički komandni jezici dozvoljavaju i da se komanda **automatski pokrene** izborom nekog od prikazanih imena datoteka.
- Pretpostavka za ovo je da izabrana datoteka predstavlja podrazumevajući operand date komande.

# Grafički komandni jezici

---

- Za komunikaciju sa grafičkim komandnim jezicima potreban je pokazivački uređaj kao što je **miš**.
- On omogućuje pokazivanje **tačke ekrana**.
- Zadatak interpretera grafičkog komandnog jezika je da, na osnovu **pozicije (koordinata)** pokazane tačke i pritisaka na odgovarajuću dirku pokazivačkog uređaja, odredi šta korisnik želi.

# Grafički komandni jezici

---

- Na primer, ako pokazana **tačka** pripada **skupu tačaka** zone ekrana koja sadrži grafičku predstavu komande ili njeno ime, tada dva uzastopna pritiska na odgovarajuću dirku pokazivačkog uređaja izazivaju obavljanje odabrane komande.