

SVEUČILIŠTE U MOSTARU
FAKULTET STROJARSTVA, RAČUNARSTVA I ELEKTROTEHNIKE
PREDDIPLOMSKI STUDIJ RAČUNARSTVA

OPERACIJSKI SUSTAVI

VJEŽBE

Nastavnik: prof.dr.sc. Sven Gotovac
gotovac@fesb.hr

Asistent: doc.dr.sc. Željko Šeremet
zeljko.seremet@fsre.sum.ba

MOSTAR, TRAVANJ 2024.

LJUSKA



SADRŽAJ

- Pojam ljuske
- Načini rada ljuske
- Varijable okruženja
- Povijest naredbi
- Skripte



POJAM LJUSKE (1)

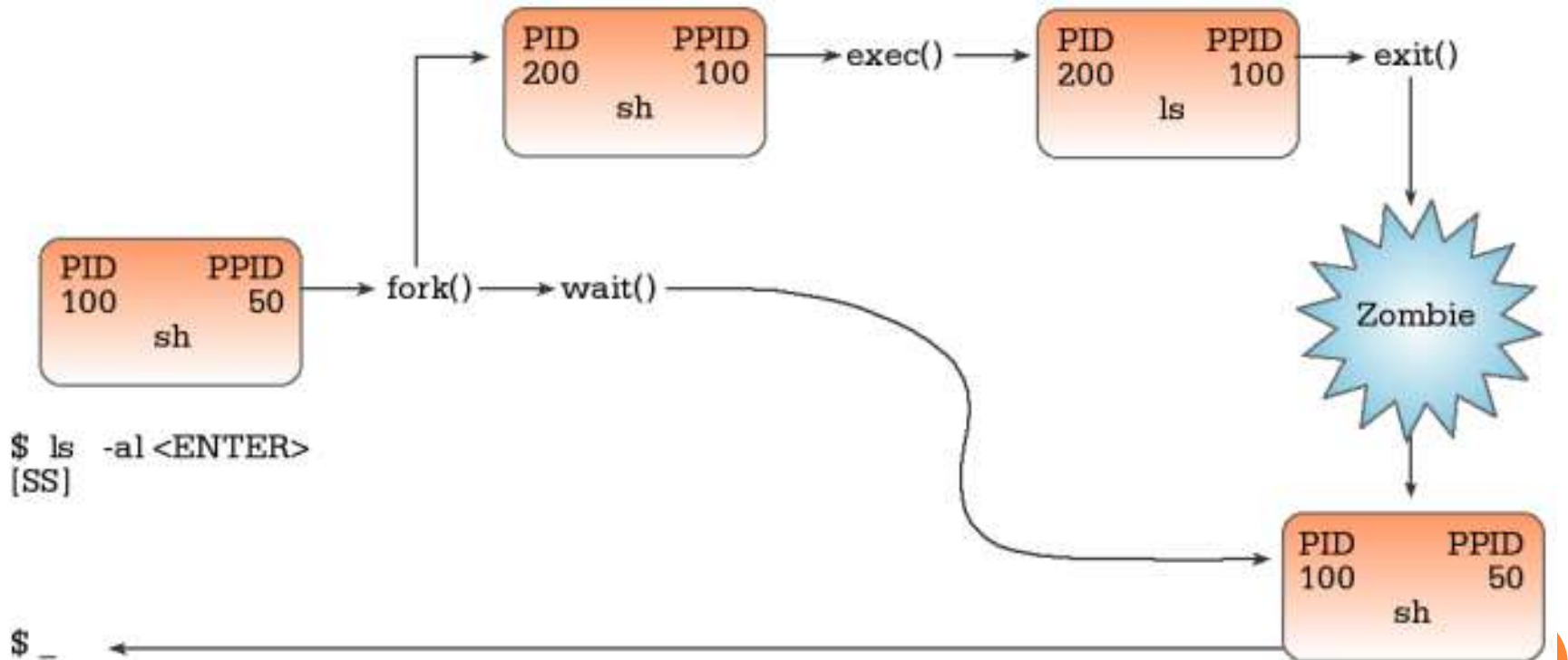
- Ljuska je program
- Služi za lakšu komunikaciju sa jezgrom
 - Pokreće procese za korisnike
 - Brine se za preusmjeravanje ulaza i izlaza
 - Moguća je automatizacija naredbi
 - ...
- Sve zadatke je moguće obaviti pomoću ljuske



POJAM LJUSKE (2)

○ Primjer

- Pokretanje procesa



POJAM LJUSKE (3)

- Nakon pokretanja terminala pokrenuta je ljuska sa izlazom i ulazom vezanim za terminal
- Zadane naredbe interpretira ljuska, ne terminal

[illegible]

VRSTE LJUSKI

- Postoji nekoliko implementacija ljuski
 - sh, bash, csh, ksh...
- Korisnik ima izbor koju će koristiti
 - Naredba chsh mijenja ljusku korisnika
 - Podatak o ljusci korisnika zapisan je u datoteci /etc/passwd
- Popis dostupnih ljuski nalazi se u datoteci /etc/shells



BASH LJUSKA

- Najkorištenija ljuska na Linux sustavu je bash - Bourne again shell
 - Nasljednik sh ljuske (Bourne shell) i kompatibilna s njome skripte pisane u sh mogu biti pokretane sa bash ljuskom ali ne obratno
 - Neka dodatna svojstva
 - Upravljanje poslovima (jobs, bg, fg)
 - Neograničena povijest naredbi
 - Pseudonimi (alias)



VARIJABLE OKRUŽENJA (1)

- **engl. environment variables**
- Slične varijablama u programskim jezicima
 - Spremaju vrijednosti u varijable koje ostali programi ili sama ljuska mogu čitati
- Naredba **env** izlistava varijable okruženja trenutne ljuske
- Zadatak
 - Izlistajte varijable okruženja



VARIJABLE OKRUŽENJA (2)

- Vrijednost varijable dohvaća se dodavanjem “\$” ispred imena varijable

- Primjer

```
$ echo PATH  
PATH
```

```
$ echo $PATH
```

```
/usr/local/bin:/usr/bin:/bin:/usr/games
```



BITNIJE VARIABLE (1)

○ **PATH**

- Popis direktorija koji sadrže naredbe
- Zbog ove varijable možemo upisati ime naredbe bez navođenja apsolutne staze
- Postavljena u jednoj od skripti kod pokretanja ljuške

○ **Zadatak**

- Ispisati varijablu **PATH**
- Dodati direktorij **/opt** varijabli **PATH**
 - `PATH=$PATH:/opt`



BITNIJE VARIABLE (2)

○ SHELL

- Trenutna ljuska korisnika

○ HOME

- Informacija koji je direktorij matični direktorij trenutnog korisnika
- Moguće je koristiti skraćenice poput “~”

○ Zadatak

- Što ispisuje “ls -l ~”



POSTAVLJANJE VARIJABLI (1)

- Varijabla je postavljena dodjeljivanje vrijednosti
 - Praksa je da se nazivi varijabli sastoje od velikih slova
- Primjer

```
$ echo $OS
```

```
$ OS=D305
```

```
$ echo $OS
```

```
D305
```



POSTAVLJANJE VARIJABLI (2)

- Ako otvorimo novu ljusku, varijabla neće biti postavljena
- Ako želimo da sve novo pokrenute podljuske imaju postavljene neku varijablu, koristimo **export**
- **export <varijabla> ili**
export <varijabla>=<vrijednost>
- Samo podljuske mogu naslijediti varijable
- Brisanje varijable ostvaruje se naredbom **unset**



POSTAVLJANJE VARIJABLI (3)

○ Zadatak

- Osigurati da je varijabla OS postavljena
- Otvoriti podljusku trenutne ljuske
- Ispisati varijablu OS
- Izaći iz podljuske
- Izvršiti “export OS”
- Pokrenuti podljusku i ispisati varijablu OS
- Pokrenuti podljusku podljuske, ispisati varijablu OS



NAČINI RADA

- Bash ljusku je moguće pokrenuti na različite načine
 - Interaktivno bez prijave
 - Interaktivno sa prijavom
 - Neinteraktivno
 - Preko **sh** naredbe
- Svaki način utječe na ponašanje ljuske



INTERAKTIVNI NAČIN RADA (1)

- Korisnik može unositi naredbe
- Ako se korisnik pri pokretanju ljuske prijavio (upisao lozinku i korisničko ime) ljuska izvršava datoteke
 - **/etc/profile**
 - **~/.bash_profile, ~/.bash_login or ~/.profile**
 - pročitana je prva datoteka koja postoji
 - **~/.bash_logout**



INTERAKTIVNI NAČIN RADA (2)

- Ako se korisnik pri pokretanju ljuske nije prijavio (upisao lozinku i korisničko ime) ljuska izvršava datoteke
 - `~/.bashrc`
 - `/etc/bash.bashrc`
 - `bashrc` je obično pozvana u `bash_profile`
`if [f ~/.bashrc]; then . ~/.bashrc; fi`
- Zadatak
 - Ispisati varijablu “-”



SKRIPTE (1)

- Skripte služe za izvršavanje skupa naredbi
 - Izvršena skripta napraviti će sve kao da je utipkano u terminal
 - Rade u neinteraktivnom modu
 - Ne čitaju iste skripte kao interaktivni mod
- Bash nije samo ljuska već i skriptni jezik (vidljivo po navedenoj if konstrukciji)



SKRIPTE (2)

- Postoji mnoštvo vrsta skripti
 - Python, Perl, Bash, PHP...
- Prva linija određuje koja naredba će interpretirati skriptu
- Za Bash skriptu to je
`#!/bin/bash`
 - **`#!`** znači da slijedi lokacija programa koji će interpretirati skriptu (sa ili bez razmaka)



SKRIPTE (3)

- Skripte NISU interaktivne
 - Mogu izvršiti samo predefinirani skup naredbi
- Skripta mora imati dopuštenja da bude izvršena!
- Ako direktorij u kojem se nalazi nije u PATH varijabli, mora se navesti puna staza
- Zadatak
 - Napisati Bash skriptu koja ispisuje “Hello World!”
 - Izvršiti skriptu



SKRIPTE (4)

- Pokrenuta skripta stvori podljusku, izvrši naredbe i vrati se u izvornu ljusku
 - Korisno ako želimo podesiti okruženje nove ljuske
 - Može stvoriti probleme ako skripta pretpostavlja okruženje trenutne ljuske
- Ako želimo da se skripta izvrši u trenutnoj ljusci, koristimo `source` ili `.` (točka)



SKRIPTE (5)

○ Zadatak

- Dodati varijablu OS prethodnoj skripti
export OS=D305
- Izvršiti skriptu
- Izvršiti ponovno u trenutnoj ljusci
- Svaki put ispisati OS
- Što se dogodilo kod drugog ispisa i zašto?
- Je li se promjena propagirala na ljusku roditelja korištenjem export?



NADOPUNJAVANJE NAREDBI

- Bash omogućava brže pisanje naredbi njihovim nadopunjavanjem unutar komandne linije
 - Pretražuje naredbe i korisniku ponudi jednu od mogućnosti na temelju dosadašnjeg unosa znakova
- Koristi se tipka TAB
 - Jedan TAB nadopunit će naredbu ako postoji samo jedna mogućnost
 - Dvaput TAB će korisniku ispisati sve mogućnosti



POVIJEST NAREDBI (1)

- Datoteka `~/.bash_history` sadrži popis svih naredbi koje su izvršene
 - Nove naredbe su zapisane tek nakon terminiranja ljuske
 - Ljuska učitava povijest naredbi tijekom pokretanja
 - Dvije ljuske nemaju istu povijest naredbi ako su korištene konkurentno, ali nakon ponovno pokretanja obiju ljusaka, povijest je ista



POVIJEST NAREBI (2)

○ Zadatak

- Pokrenuti istovremeno dvije zasebne ljuske
- Pročitati zadnju naredbu u `~/.bash_history`
- Upisati različite naredbe u ljuske
- Pročitati zadnju naredbu u `~/.bash_history`
- Terminirati jednu ljusku
- Pročitati zadnju naredbu u `~/.bash_history`
- Terminirati drugu ljusku
- Pročitati zadnju naredbu u `~/.bash_history`



POVIJEST NAREDBI (3)

- Za izvršavanje posljednje korištene naredbe koristi se “!!”
 - Isto tako **!<znakovi>** će izvršiti posljednju naredbu koja je tog oblika npr. **!himBH**
- Primjer

```
$ cat /etc/shadow
cat: /etc/shadow: Permission denied
$ sudo !!
```



PRETRAŽIVANJE POVIJESTI NAREDBI

- Što ako znamo otprilike kako je glasila naredba ali se ne možemo sjetiti u potpunosti?
- Pretraživanje se vrši sa `^CTRL+R`
- Zadatak
 - Pronađite zadnju naredbu koja u sebi sadrži **`/etc/shadow`**



PSEUDONIMI

- Pseudonimi su korisni kod skraćivanja naredbi
 - Najčešće se dodaju u jednu od skriptu koju kod svoje inicijalizacije izvršava ljuska
 - Npr. u `~/.bashrc` izravno ili u datoteku `~/.bash_aliases` koja je pozvana iz `~/.bashrc`
- Primjer

```
$ alias ll="ls -l"
$ ll
total 0
```



LITERATURA

- <http://superuser.com/questions/7414/howcani-searchthebashhistoryandrerun>
- <http://tldp.org/LDP/BashBeginnersGuide/html/>



KRAJ

