



Otvoreni operativni sistemi - zadaci i pitanja 2 -

Marko Dimitrijević

Napsiati kombinaciju komandi koja će kreirati prazan fajl **epel.repo** u direktorijumu **/etc/yum.repos.d** i ukoliko je kreiranje fajla uspešno, dodeliti mu privilegije **čitanje/pisanje** za vlasnika, **čitanje** za grupu i ostale korisnike.

```
touch /etc/yum.repos.d/epel.repo && chmod 644 /etc/yum.repos.d/epel.repo
```

Napisati komandu koja će pretražiti kompletno stablo direktorijuma i pronaći sve regularne fajlove sa nazivom **core**, koji su veći od **1MB**. Pretragu izvršiti kao proces u pozadini, a standardni i dijagnostički izlaz preusmeriti u fajl **/tmp/core.info**.

```
find / -name core -size +1M -type f &> /tmp/core.info &
```

Napišite komandu kojom ćete napraviti bekap/imidž druge particije prvog diska u fajl **efi.back** u direktorijumu **/backup**. Ukoliko je bekap kreiran, fajl zaštititi tako da samo vlasnik ima pravo čitanja. Proces izvršiti u pozadini.

```
dd if=/dev/sda2 of=/backup/efi.back && chmod 400 /backup/efi.back &
```

Napišite komandu kojom ćete napraviti bekap/imidž druge particije prvog diska u fajl **efi.back** u direktorijumu **/backup**. Ukoliko se prilikom izvršavanja pojavila greška, komanda treba da ispiše izlazni kod.


```
dd if=/dev/sda2 of=/backup/efi.back || echo $?
```

Korisnik želi da promeni svoj default shell tako da prilikom sledećeg logovanja koristi **/bin/tcsh**. Kojom komandom može da proveriti da li ovaj shell postoji na sistemu i ukoliko postoji, da ga postavi kao default shell za svoj nalog?

```
ls /bin/tcsh && chsh -s /bin/tcsh
```

Želimo da prikupimo informacije o fajlu **/etc/dhcpd.conf** tako što ćemo sukcesivno izvršiti komande `ls -la`, `file` i `wc -l` nad ovim fajlom. Napišite kombinaciju komadni kojom ćete sve informacije skupiti u jedan fajl **dhcp.info**.

```
(ls -la /etc/dhspd.conf; file /etc/dhspd.conf; wc -l /etc/dhspd.conf) > dhcp.info
```

Šta je izlazni kod (exit code) aplikacije/komande i kako može da se ustanovi? Koji je izlazni kod ukoliko je aplikacija/komanda izvršena bez greške?

Izlazni kod je sistemska varijabla koja ukazuje na status prethodno izvršene aplikacije/komande.

Izlazni kod prethodno izvršene aplikacije se može videti komandom:

echo \$?

Ukoliko je aplikacija izvršena bez greške, izlazni kod je jednak 0.

Kojom komandom je moguće videti procese koji se izvršavaju u pozadini ili su suspendovani? Kako je moguće pokrenuti suspendovani proces?

Procese koji se izvršavaju u pozadini ili su suspendovani moguće je videti komandom **jobs**.

Suspendovani proces je moguće pokrenuti u pozadini komandom

bg %n

odnosno u aktivnom shell-u komandom

fg %n

gde je **n** broj procesa.

Koji signal se šalje aplikaciji/komandi koja se trenutno izvršava u shell-u simultanim pritiskom tastera **CTRL+Z**?

Šalje se signal **SIGTSTP** (signal 20).

Koji signal se šalje aplikaciji/komandi koja se trenutno izvršava u shell-u simultanim pritiskom tastera **CTRL+C**?

Šalje se signal **SIGINT** (signal 2).

Koji signal se šalje aplikaciji/komandi koja se trenutno izvršava u shell-u simultanim pritiskom tastera **CTRL+D**?

Šalje se signal **SIGQUIT** (signal 3).

Šta sve mogu biti komande shella? Kojim komandama se može videti path/lokacija komande ukoliko je ona skript ili izvršni fajl?

1. interne komande shell-a (shell built-in commands),
2. rezervisane reči shell-a (shell reserved keywords),
3. funkcije shell-a (shell functions),
4. izvršni programi (file system commands, executable files),
5. simbolički linkovi na izvršne programe,
6. aliasi.

Lokacija skripta ili izvršnog fajla se može videti komandoma

which komanda

type komanda

Šta predstavljaju alaiasi? Kako možemo videti definisane aliase u shell-u koji koristimo?

Alias omogućavaju zamenu komandi ili grupe komandi, uključujući opcije i argumente jednim stringom.

Definisane aliase u shell-u koji koristimo možemo izlistati komandom

alias

Šta su izvršni programi (file system commands)? Koje vrste izvršnih programa imamo i kako ih možemo izvršiti?

Izvršni programi mogu biti **binarni izvršni programi** ili **skriptovi** (tekstualni izvršni programi), koji postoje u fajl sistemu i mogu se izvršiti nezavisno od shell-a koji se koristi. Izvršni program se može izvršiti ukoliko korisnik ima **execute** pravo pristupa nad njim.

Izvršni programi se izvršavaju navođenjem njihove **apsolutne** ili **relativne putanje**. Izvršni programi se mogu izvršiti direktno, samo navođenjem imena bez apsolutne ili relativne putanje, ukoliko se nalaze u **sistemskejoj putanji** (*path*).

Šta je sistemska putanja, kojom komandom se može prikazati?

- Sistemska putanja je skup direktorijuma koji se sukcesivno pretražuju kada se komanda izvrši. Ukoliko u sistemske putanji **postoji izvršni program** ili **simbolički link ka njemu** sa imenom koje odgovara otkucanoj komandi, on će biti izvršen. U suprotnom, shell će prijaviti grešku, tj. da program nije pronađen. Sistemska putanja je definisana u varijabli **\$PATH**:

echo \$PATH

Koji su modovi rada VI editora i čemu služe? U kom modu je editor prilikom pokretanja?

Postoje tri moda rada VI editora:

- **komandni** – svi znaci otkucani sa tastature se tretiraju kao komande,
- **input** – sluzi za unos teksta, tasteri imaju normalno značenje,
- **mod zadnje linije** – sluzi za unos slozenih komandnih linija.

Prilikom pokretanja VI editora, aktivan je komandni mod.

Koja je razlika između imenovanih i neimenovanih
bafera VI editora?

- **Neimenovanih bafera** ima 9 a referenciranje na njih se obavlja navadjanjem njihovog rednog broja. Prilikom upisivanja u bafer novi sadržaj se uvek upisuje u prvi a pri tome se stari sadržaj bafera pomera iz prethodnog u naredni (rade kao pomerači registri, FIFO) pri čemu se sadržaj poslednjeg (devetog) bafera gubi.
- **Imenovanih bafera** ima 26, imena bafera odgovaraju slovima engleske abecede. Može im se direktno pristupiti. Malim i velikim slovom se pristupa istom baferu, u prvom slučaju se briše njegov prethodni sadržaj (*overwrite* operacija), dok se u drugom upisuje tekst u isti bafer ali se novi sadržaj dodaje prethodnom (*append* operacija).