



# Operativni Sistemi

II Semestar – 2022/23 – Vježbe

Sedmice 3,4

## Handout za Vježbe

### Agenda:

- Linux fajl sistem
- Fundamentali Koncepti Upotrebe Linuxa
- Boot Proces
- DigitalOcean - Docker Upotreba kroz praktični dio

### Kontakt:

[Narcisa.hadzajlic@size.ba](mailto:Narcisa.hadzajlic@size.ba) (B Grupa)

[Adin.jahic2019@size.ba](mailto:Adin.jahic2019@size.ba) (A Grupa)

## Linux Fajl sistem

Svaki Linuxov datotečni sistem je preuzeo neke osnovne koncepte od Unix operativnog sistema: datoteke su predstavljene strukturalno putem tzv. **inodes**, direktoriji su datoteke koje sadrže listu unosa, te unutarnjim dijelovima računara i periferijama se može pristupiti putem I/O (ulazno-izlaznih) upita preko neke datoteke.

**Inodes** Svaki fajl je predstavljen strukturalno, tj. putem tzv. *inodes*-a. Svaka pojedina *inode* sadrži podatke o datoteci: tip datoteke, pristupne dozvole, vlasništvo, vremenske informacije, veličinu, te direktive do blokova podataka. Adrese blokova podataka dodijeljene datoteci se pohranjuju u njegovoj *inode*-i. Kada korisnik zatraži I/O operaciju na datoteci, kernel konvertira naredbu u blok broj, te koristi taj broj kao indeks u tabeli blok adresa, te sukladno tomu čita ili zapisuje fizički blok.

**Direktoriji** Direktoriji su strukturirani unutar hijerarhijskog stabla i svaki može sadržavati fajlove i pod-direktorije. Sami direktoriji su implementirani kao posebna vrsta fajla, tj. direktori nije ništa drugo no datoteka koja sadrži listu unosa, a svaki unos sadrži *inode* broj i ime datoteke. Kada neki proces zatraži određenu putanju (eng. *path*), kernel pretražuje direktorije u potrazi za određenim *inode* brojem. Kada se na kraju ime konvertira u *inode* broj, onda se *inode* učitava u memoriju i koristi kao izlazna informacija.

**Linkovi** Unix datotečni sistem je implementirao koncept linkova (vezā). To znači da više imena može biti dodijeljeno jednoj *inode*, koja pak sadrži polje u kojem je broj povezan s odgovarajućom datotekom. Dodavanje linka se sastoji od stvaranja direktorija, gdje *inode* broj upućuje do same *inode*, te od povećanja broja linkova unutar *inode*-e. Kada se link izbriše, tj. kada korisnika koristi naredbu *rm*, kernel smanjuje broj linkova, te odjeljuje *inode*-u ako joj broj postane 0. Ovakva vrsta linka se zove *hard link* i može se koristiti samo unutar jednog datotečnog sistema, odnosno ne mogu se stvoriti multi-sistemske *hard link*ovi. Također, linkovi mogu samo voditi do datoteka, jer se linkovi za direktorije ne mogu kreirati da bi se izbjeglo utvaranje (stvaranje nepostojećeg) kruga u stablu direktorija. Druga vrsta linkova koja postoji kod Unix datotečnih sistema su simbolički linkovi. To su datoteke koje sadrže ime. Kada kernel susretne simbolički link u svojoj potrazi za konvertiranjem *inode*-a, on zamjenjuje ime linka sa njegovim sadržajem, tj. sa imenom datoteke koja se potražuje, te ponovno započinje potragu prema zadanoj putanji. Kako simbolički link ne vodi do *inode*, tako je moguće napraviti multi-sistemske simboličke linkove, pri čemu se misli na datotečne sisteme, a ne operativne. Sami simbolički linkovi mogu voditi čak i do nepostojeći datoteka. Ovi linkovi su veoma korisni, jer nemaju ograničenja kakva imaju *hard link*ovi, ali zato simbolički linkovi koriste prostor na disku (gdje su smještene njihove *inode* i blokovi podataka) i uzrokuju znatna usporenja u konverziji *inode*-a, jer kernel mora uvijek ponovno započeti potragu kada god susretne simbolički link.

**Filesystem Hierarchy Standard** FHS je dokument koji određuje standard u kreiranju fajlova i direktorija, odnosno mjesto i ime velike većine direktorija i fajlova. Usklađenost s ovim dokumentom znači i sprječavanje stvaranja razlika među raznim distribucijama GNU/Linux-a.

Postoje tri vrste fajlova: korisnički, sistemski i izvršni. Prvi su svi oni koje kreiraju korisnici na sistemu, drugi najčešće dolaze u *plain text* formatu, dok su treći fajlovi u stvari programi. Pored njih imamo i tzv. *(un)shareable* ili (ne)djeljive i *variable* ili promjenjive fajlove. Prvi su oni kojima svatko može pristupiti, odnosno ne može, dok su drugi oni koje sistem automatski mijenja bez potrebe za intervencijom korisnika.

Svi fajlovi i direktoriji u Linux sistemu se nalaze unutar direktorija root. Ovo je glavni direktorij koji se označava znakom '/' i sadrži druge pod-direktorije koji se dalje granaju po principu stablastog grananja. Root direktorij nikada ne može biti pod-direktorij. Ispred ovog znaka ne treba pisati root, jer je veoma lako odrediti da li znak '/' označava root direktorij ili dijeli direktorije. Pojam stablasto grananje smo već prije susreli i on označava da se iz jednog direktorija ostali direktoriji i pod-direktoriji granaju kao grane stabla., npr. od inicijalnog '/' direktorija imamo '/bin/' direktorij koji ima svoje pod-direktorije, itd.

Još jedna stvar koju nikako ne smijemo pomiješati je to da Linux koristi pojam 'root' u tri slučaja: root račun, root home direktorij (/root/) i root direktorij za cijeli sistem (/). Kada god spominjemo pojam root, potrebno je naglasiti na koji root mislimo. Za razliku od Windows sistema koji direktorije razdvaja znakom '\' (eng. backslash), Linux je preuzeo staru Unix oznaku '/' (eng. slash).

U Linuxu je obavezno iza imena direktorija staviti znak *slash*, npr. `/usr/bin/`, jer bez toga znaka ne bismo znali da li se radi o direktoriju ili fajlu. Pored root direktorija za cijeli sistem, Linux svakom korisniku, uključujući i administratora, kreira *home* direktorij, gdje korisnici pohranjuju svoje fajlove i druge direktorije, odnosno pod-direktorije. U najvećem broju slučajeva korisnici imaju potpunu kontrolu nad sadržajem cijelog svog home direktorija zato jer u njemu obično nema sistemskih ili fajlova drugih korisnika.

Pod pojmom potpune kontrole se misli na akcije kreiranja, imenovanja, brisanja, itd. Međutim, home direktorij sam po sebi ne nudi nikakvu zaštitu, tj. drugi korisnici mu mogu pristupiti bez problema. Zato je potrebno podesiti dozvole pristupa direktoriju tako da mu samo vi može te pristupiti. Iako vam se iz ovoga može učiniti da je veoma teško kretati se kroz GNU/Linux fajl sistem, nije tako. Za kretanje direktorijima dovoljno je poznavati samo dvije komande: *pwd* (koja vam govori gdje se trenutno nalazite) i *cd* (odlazak u željeni direktorij).

## ■ Direktoriji FHS

Prema već spomenutom FHS dokumentu, Linux sistem ima 15 glavnih direktorija koji se nalaze unutar glavnog root direktorija za cijeli sistem. Jednostavnije rečeno, ovih 15 direktorija predstavlja vašu cijelu GNU/Linux instalaciju. Imena ovih direktorija su: `/bin/`, `/boot/`, `/dev/`, `/etc/`, `/home/`, `/lib/`, `/lost+found/`, `/mnt/`, `/opt/`, `/proc/`, `/root/`, `/sbin/`, `/tmp/`, `/usr/` i `/var/`. Pored imena svakog direktorija, FHS određuje i koje fajlove trebaju sadržavati.

**/bin/** direktorij sadrži komande i alate koje koriste korisnici iz shella. Postoji i direktorij `/usr/bin/` koji sadrži komande određenog korisnika.

**/boot/** direktorij sadrži sve *image*-e kernela, te fajlove koje koristi boot loader.

**/dev/** direktorij sadrži vrijednosti fajl sistema koje predstavljaju uređaje koji su prikopčani na sistem.

**/etc/** direktorij je rezervisan za konfiguracijske fajlove svega što se nalazi na tom računaru. Binarni fajlovi se ne smiju stavljati u ovaj direktorij, jer oni imaju svoje posebno mjesto unutar direktorija `/sbin/`, dok se opcionalno mogu staviti i u `/bin/` direktorij.

**/home/** direktorij sadrži home direktorije svakog korisnika na sistemu pri čemu direktorij dobiva ime prema imenu korisnika. Svi fajlovi koje kreira korisnik, kao i njegovi konfiguracijski fajlovi, se smještaju u ovaj direktorij.

**/lib/** direktorij treba sadržavati samo one library fajlove (skupovi funkcija i potprograma koje koristi glavni program) koji su potrebni za izvršavanje binarnih fajlova u direktorijima `/bin/` i `/sbin/`. Ovi fajlovi su posebno potrebni za bootanje sistema i izvršavanje komandi unutar root fajl sistema.

**/lost+found/** direktorij sadrži fajlove pronađene ili stvorene uslijed greške na disku ili nepravilnog gašenja računara.

**/mnt/** direktorij sadrži privremeno mountovane fajl sisteme kao CD-ROM, Floppy uređaj ili neke druge particije.

**/opt/** direktorij obezbjeđuje skladištenje velikih, statičkih aplikacijskih paketa. Najlakše ga je usporediti s Windows direktorijom `\Program Files\`.

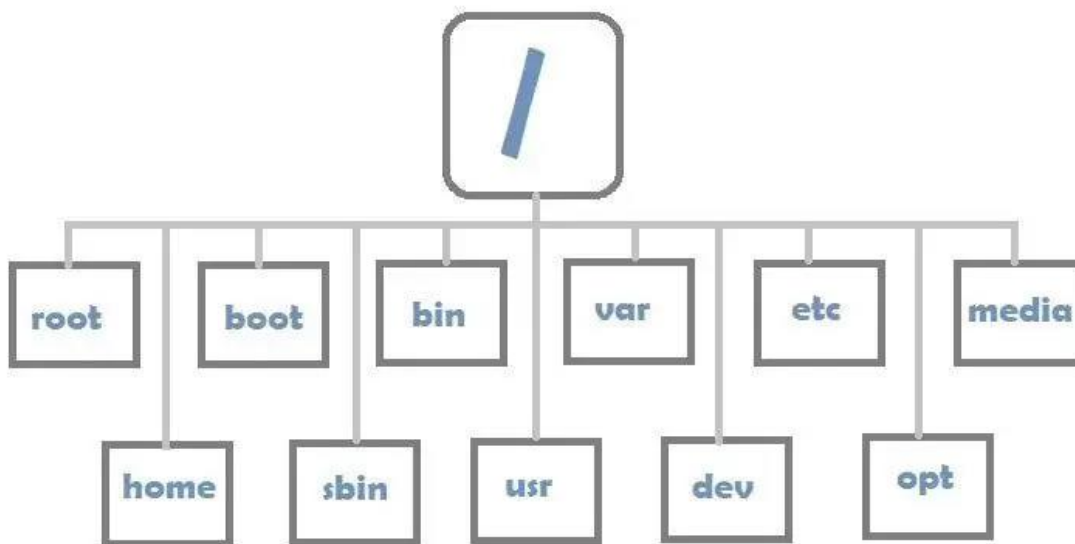
**/proc/** direktorij sadrži specijalne fajlove koji ili izvlače informacije ili ih šalju kernelu.

**/root/** direktorij je administratorov home direktorij.

**/sbin/** direktorij sadrži izvršne fajlove koje koristi samo root korisnik (admin, superuser). Izvršni fajlovi koji se nalaze u ovom direktoriju obavljaju zadatke oko boota, mountovanja

`/usr/` i povrata sistema (system recovery). Ostali binarni fajlovi se nalaze u `/bin/` direktoriju. **/tmp/** direktorij sadrži privremene (temporary) fajlove. **/usr/** direktorij sadrži fajlove kojima svatko u sistemu može pristupiti. Ovdje se nalaze pod-direktoriji koji su namješteni na read-only. Neki od njih su: `dict/`, `games/`, `local/`, `src/`, itd. Detaljnije o njima ćemo neki drugi put.

**/var/** direktorij sadrži promjenjive ili *variable* fajlove. To su fajlovi koji se stalno mijenjaju bez intervencije korisnika kao npr. log fajlovi (`/var/log/`) ili printer *spool* fajlovi (`/var/mail/spool/`).



## Fundamentali Koncepti Upotrebe Linuxa

**mkdir** <NazivDirektorija> - pravi direktorij sa datim nazivom.

**cd** <Path> - kretanje po linux fajl sistemu. Cd .. izlazi iz direktorija,

**cp** <Source> <Destination> - kopiranje fajlova

**mv** <Source> <Destination> - premještanje fajlova

**rm,rmdir** <Naziv> - brisanje fajlova

**ls** Služi za listanje sadržaja direktorija.

**cat** Služi za prikazivanje sadržaja neke datoteke.

**more** <naziv> Naredba služi za prikaz datoteke s mogućnošću kretanja gore/dole. za prikaz datoteke index.html, tipkom f idemo naprijed (forward), tipkom b idemo nazad (backward), tipkom q izlazimo u shell (quit).

**tail -broj <fajl>** - služi za prikazivanje kraja neke datoteke.

**head -broj <fajl>** Ova naredba služi za prikazivanje početka neke datoteke.

**clear** Služi za brisanje sadržaja ekrana.

**find <fajl>** Može poslužiti za traženje datoteka ili direktorija.

**du** Služi za prikazivanje diskovne potrošnje nekog direktorija ili datoteke. Koristi se na slijedeći način. du -sk \* ... prikazuje veličinu radnog direktorija, sumirano, u kilobajtima du -sk fit ... za prikaz veličine fit direktorija, sumirano, u kilobajtima •

**df** Ovom naredbom možemo pogledati koliko je diskovnog prostora zauzeto odnosno slobodno na svim diskovima i particijama.

df -h ... za prikaz trenutno mountanih diskova s human prikazom •

**file** Ova naredba će nam reći o kojoj vrsti datoteke se radi

**wc** Služi za prikazivanje broja linija, riječi i bajtova neke datoteke. wc index.html ... prikazuje broj linija, riječi i bajtova index.html datoteke wc -l index.html ... prikazuje broj linija index.html datoteke

**nano** <fajl> - editovanje fajla

**ps** – prikaz procesa.

**pwd** – trenutna putanja.

**passwd** – promjena lozinke

**zip, unzip** - kompresovanje foldera

**man <komanda>** - uputstvo o korištenju naredbe

**touch <fajl>** - kreiranje fajla

**chown <korisnik> <fajl>** - mijenja vlasnika fajla ili direktorija.

## Chmod (baziran na brojevima)

Ova naredba služi za mjenjanje dozvola na datotekama ili direktorijima. Postoje tri osnovne dozvole: read, write i execute.

- *Read* dozvola nam govori smije li netko čitati i gledati sadržaj neke datoteke ili direktorija.
- *Write* dozvola nam govori smije li netko mjenjati sadržaj neke datoteke ili direktorija.
- *Execute* dozvola nam govori smije li netko izvršavati i pokretati neki program ili skriptu.

Dozvole se dodjeljuju u tri grupe - za vlasnika, članove grupe i ostale. Koristi se u obliku duuugggooo.

1. Prva oznaka govori da li se radi o direktoriju ili datoteci.
2. Slijedeće tri oznake uuu nam govore o vlasniku,
3. slijedeće tri se odnose na grupu i zadnje tri na ostale korisnike.

Pogledajmo to na primjeru dozvole -rwxr-xr-x.

- Prva oznaka, -, nam govori da se radi o datoteci. Da se radi o direktoriju onda bi prva oznaka bila slovo d.
- Oznake od drugog do četvrtog mjesta nam govore o vlasniku koji ima read dozvolu (r), write dozvolu (w) i smije izvršavati program (x).
- Slijedeće tri oznake govore o dozvolama za članove grupe koji imaju dozvolu za čitanje i izvršavanje kao i posljednje tri oznake koje govore da i svi ostali korisnici iste dozvole za čitanje i izvršavanje ali ne i mjenjanje sadržaja datoteke.

Kod korištenja naredbe mogu se koristiti slova ili brojevi. Objasniti ćemo ukratko korištenje brojeva.

- 4 - daje atribut read
- 2 - daje atribut write
- 1 - daje atribut execute

Npr: kad želite dati atribut read i write bez atributa execute, zbrojite brojeve 4 i 2 i dobijete 6. Na primjeru bi to izgledalo ovako:

- `chmod 644 index.html`
  1. Prvi broj 6 određuje dozvolu za vlasnika što je zbroj broja 4 i 2, dakle read i write,
  2. drugi broj je za grupu
  3. treći za ostale korisnike.

Dakle, ovom naredbom smo datoteci index.html dali dozvolu koja izgleda -rw-r--r-- što bi značilo da vlasnik smije čitati i mjenjati sadržaj a svi ostali samo čitati. Ove dozvole, 644, bi trebale imati sve datoteke koje su namjenjene javnosti, dakle html stranice, slike, itd. Kad bi datoteka imala dozvolu -rw-rw-rw- dakle 666, onda bi osim vas datoteku mogli mjenjati članovi grupe i svi ostali korisnici. Zamislite da imate neki dokument koji želite zaštititi tako da nitko ne može čitati njegov sadržaj osim vlasnika odnosno vas. Dat ćemo mu dozvolu 600 što bi značilo da ćete moći čitati i mjenjati sadržaj a svi ostali imaju atribut nula što bi u prijevodu značilo nemaju nikakva prava nad tom datotekom i ne mogu joj pristupiti. To bi mogli riješiti naredbom `chmod 600 dokument.txt`.

```
shum@sol1:~$ ls -l
total 20
drwx----- 2 shum staff 4096 Jan 16 22:04 Mail
drwx----- 3 shum staff 4096 Jan 16 14:15 csc128
drwxr-xr-x  2 shum staff 4096 Jan 13 16:42 public
drwxr-xr-x  2 shum staff 4096 Jan 16 14:07 public_html
-rw-r--r--  1 shum staff 628 Jan 15 20:04 verse
```

Diagram labels and arrows:

- permissions (e.g., drwxr-xr-x) → file type
- number of links (e.g., 2) → number of hard links
- user (e.g., shum) → user (owner) name
- group (e.g., staff) → group name
- size (e.g., 4096) → size
- date/time (e.g., Jan 16 22:04) → date/time last modified
- filename (e.g., Mail) → filename

Permission legend:

**rwx**

- r → readable
- w → writeable
- x → executable

## Linux boot proces

1. Kada se računar uključi, ono prvo provjerava da li su svi hardverski dijelovi ispravni i funkcionalni. Ovo se naziva POST (Power-On Self Test) i obično se prikazuje kratki tekst na zaslonu koji obavještava korisnika o stanju hardvera, poput količine RAM-a koja je instalirana ili broja priključenih uređaja.
2. Nakon završetka POST-a, BIOS čita MBR (Master Boot Record) s prvog sektora hard diska (0. sektora), koji se nalazi na particiji koju BIOS smatra aktivnom. MBR je mali program koji se nalazi na prvom sektoru tvrdog diska i njegova svrha je da omogući računar da pronađe i pokrene bootloader koji će učitati operativni sistem.
3. MBR sadrži informacije o particijama hard diska, vrsti bootladera i mjestu gdje se bootloader nalazi. BIOS čita MBR i koristi te informacije da pronađe bootloader.
4. Nakon što je BIOS pronašao bootloader, on učitava bootloader u memoriju računara.
5. Bootloader je program koji je zadužen za pokretanje operativnog sistema. U slučaju Linuxa, najčešće se koristi GRUB (GRand Unified Bootloader). GRUB omogućava korisniku da odabere koju particiju hard diska želi koristiti za pokretanje operativnog sistema, kao i da učitava kernel koji upravlja svim hardverskim uređajima i softverskim aplikacijama.
6. Nakon što je kernel učitao, on započinje inicijalizaciju hardvera i postavlja osnovnu infrastrukturu za operativni sistem. Kernel zatim preuzima kontrolu nad računarom i pokreće init sistem, koji je zadužen za pokretanje procesa koji će omogućiti pokretanje korisničkih aplikacija.
7. Kada se init sistem pokrene, on izvršava niz skripti koje su namijenjene za pokretanje raznih usluga i procesa, kao što su mrežni servisi, sigurnosne funkcije i korisnička sučelja. Ovaj proces obično traje nekoliko sekundi, nakon čega se sustav prijavljuje kao spreman za upotrebu.

## Praktični dio

### Zadaci:

- Provizionirati VM pomoću DigitalOcean cloud provajdera
- Izvršiti sistemski update sa apt-get update
- Instalirati Docker (vidjeti online manual)
- Pokrenuti Standardni docker „hello world“ container
- Itestirati naredbe linuxa definisane u dosadašnjem handoutu.
- Klonirati zajednicki repozitorij te uraditi commit na „public“ branch sa sistemskim vremenom vašeg operativnog Sistema I vašim imenom I prezimenom.

**Za sve eventualne primjedbe, komentare, sugestije obratiti se na mail:  
adin.jahic2019@size.ba**