

## Istorijat Unix-a

### Multics

Operativni sistem Multics je sredinom 60-ih bio zajednički pokušaj najvećih kompanija i univerziteta u to vreme (MIT, General Electric i Bel laboratorija) da naprave operativni sistem u koji bi bila implementirana sva najbolja, do tada poznata, rešenja. Učesnici u projektu su želeli da za moćne računare kreiraju operativni sistem koji će biti u stanju da radi sa velikim brojem terminala. Oni su budućnost računarstva videli tako da u svakom gradu postoji moćan centralni računar, a da građani u svojim domovima poseduju terminale uz pomoć kojih (modemom) pristupaju glavnom računaru.

Na razvoju Multics sistema, od 1964. godine, bili su zaposleni veliki timovi stručnjaka i neki od najboljih programera tog vremena. U ovaj operativni sistem implementirani su svi najbolji do tada poznati algoritmi. Rezultat nije bio onoliko dobar koliko se očekivalo. Multics se u praksi pokazao veoma loše jer je puno procesorskog vremena trošio na donošenje odluka (na svoj rad), a malo ostavljao korisnicima. Osim toga, ovaj projekat je bio i veoma skup. Iz tih razloga Multics nikada nije ozbiljno zaživeo na računarima. Iako je u krajnjoj liniji bio neuspešan, ovaj operativni sistem je ostavio dubok trag u kasnijoj evoluciji operativnih sistema. Imajući u vidu osnovnu koncepciju ovog sistema, on se može smatrati pretečom Unix-a.

U osnovi je bila modularna organizacija hardvera i softvera. Ideja je bila da se sistem može jednostavno proširivati dodavanjem novih komponenti (procesora, memorije, diskova itd). Ovo je bilo moguće uraditi bez prekida rada sistema. MIT je koristio ovo da u nekom trenutku odvoji deo hardvera, sklopi drugi računar koji će obaviti neki novi zadatak, i po obavljenom zadatku vrati delove u glavni računar koji je u međuvremenu normalno radio.

Multics je bio prvi veći sistem koji je omogućio hijerarhijski sistem datoteka. Napisan je na višem programskom jeziku (PL/1). Odvojena lista za kontrolu pristupa za svaku datoteku omogućuje fleksibilno deljenje podataka i potpunu privatnost kada je to potrebno. Svi ovi koncepti uticali su na razvoj Unix-a.

### Počeci Unix-a

Bell Labs, nezadovoljan što Multics postaje sve veći i kompleksniji a ne postiže zacrtane ciljeve, polako se povlači sa projekta. Poslednji istraživači Bell Labs-a, Ken Thompson i Denis Riči, koji su napustili projekat odlučili su da probaju da ponovo naprave sistem mnogo manje veličine. Osnovni koncepti novog sistema bili su višeproceno izvršavanje, interaktivni rad, višekorisnički sistem i jedinstvena hijerarhija sistemskih datoteka. Projekat je dobio ime Unics po uzoru na Multics. Unics je kasnije podržao rad sa više korisnika pa mu je ime promenjeno u Unix. Tokom 1970. godine Unix je napisan na assembleru i tada je prvi put zvanično pokrenut. Unix Programmer's Manual je objavljen 1971. godine.

### Prebacivanje na c

Ken Tompson je kreirao programski jezik B na osnovu kojeg je Denis Riči osmislio programski jezik C kako bi na njemu bio isprogramiran deo operativnog sistema. Naime, kritični delovi sistema su bili isprogramirani na assembleru, a ostatak na programskom jeziku C. U drugoj iteraciji 1973. godine kompletan Unix je napisan na C-u što je bilo u suprotnosti sa tadašnjom idejom da kompleksan softver kao operativni sistem, koji strogo vodi računa o efikasnosti, mora biti napisan na assembleru. To je značajno doprinelo njegovoj prenosivosti na različite sisteme, pa je Unix mogao da se instalira na različitim računarima (arhitekturama).

## Širenje Unix-a

U početku su Unix koristili odeljci AT&T za obradu dokumenata i štampanje, pa su prvi alati na Unix-u imali upravo tu namenu. AT&T je u to vreme imala zabranu prodaje softvera (zbog monopola nad telefonijom), pa je Unix počeo slobodno da se deli zajedno sa izvornim kodom i knjigama sa odgovarajućim uputstvom. Univerziteti su bili veoma zainteresovani, pogotovo za kurseve operativnih sistema. Sistem se brzo širio, korisnici su uočavali šta ne valja, prijavljivali greške i sami pravili zakrpe. Ovaj korak je verovatno bio presudan kako za sam Unix tako i za razvoj operativnih sistema uopšte. Najpoznatije verzije Unix-a implementirane su na Univerzitetu Berkli, sa oznakom BSD koje su date na korišćenje američkim univerzitetima radi daljeg usavršavanja. U ovim verzijama operativnog sistema Unix prvi put je implementirana podrška za umrežavanje.

## Nekompatibilnost Unix-a

Početakom 80-ih PC računari počinju da se šire velikom brzinom. Prodaju se po znatno nižoj ceni od dotadašnjih računara i namenjeni su za kućnu upotrebu. Verovalo se da će Unix sa svojom prenosivošću i mogućnostima postati standardan sistem na ovim računarima. Mnoge firme počinju da nude komercijalne verzije Unix-a za mini kompjutere i radne stanice.

Međutim, u to vreme su prodavci različitih Unix sistema želeli da vežu kupce za svoj proizvod, pa su namerno pravljene nekompatibilne varijacije sistema. Ovo je znatno umanjilo prenosivost Unix programa. Zbog toga se osamdesete godine zovu godine ratova Unix-a. Problem koji je nastao u razvoju Unix-a bio je taj što sistemi više nisu bili besplatni i njihov kod više nije bio javno raspoloživ, što su upravo bili razlozi prvobitnog uspeha Unix-a.

U to vreme je AT&T ponovo dobio pravo prodaje softvera, pa je firma pokrenula niz sudskih procesa koji su zamalo uništili Unix. Microsoft je imao dobru početnu poziciju i sistemom DOS kome je kasnije dodao grafički korisnički interfejs zauzima vodeću poziciju na tržištu operativnih sistema.

## Standardizacija Unix-a

Krajem osamdesetih godina prošlog veka počinju razgovori o standardizaciji. Cilj je da se napravi standard kako bi različiti sistemi zasnovani na Unix-u bili kompatibilni. U okviru IEEE (Udruženje inženjera elektronike i elektrotehnike) formira se telo za standardizaciju. Razvijen je standard POSIX 1988 (Portable operating system interface, X u imenu jer je standard namenjen za Unix). POSIX propisuje koji su sistemski pozivi neophodni da postoje, koji shell programi, koji header fajlovi za C itd., da bi se sistem zvao Unix.

Organizacija koja se danas bavi standardizacijom naziva se Open group, a standard koji se koristi se zove SUS – Single Unix Specification. Postoji i poseban standard koji se bavi Linuxom i zove se Linux Standard Base (LSB).

## Noviji razvoj Unix-a

Sredinom 80-ih Richard Stallman pokrenuo je GNU projekat pod okriljem organizacije Free Software Foundation, za koju je napisao i licencu GNU General Public License. Napisao je i editor emacs kao i GNU c kompajler, ali je najveća prepreka za razvoj GNU operativnog sistema bila nepostojanje kernela (kernel je program koji povezuje aplikativni softver i hardver računara). Linus Torvalds je početkom 90-ih napravio kernel i nazvao ga Linux. Tako je sklopljen sistem GNU/Linux (kraće se zove Linux) koji je bio namenjen za PC računare. Linux nije pravi Unix već je Unix-nalik (Unix-like) operativni sistem.

U to vreme razvijeno je i više verzija BSD-a, tako da su dominirale dve nezavisne verzije Unix-a. U BSD-u su svi delovi sistema činili celinu, dok je ideja bila da u korisnici Linux-a sami prikupe pakete i sklope sistem koji može da funkcioniše. Pojavile su se firme sa idejom da prikupe pakete, prekompiraju,

zapakuju i distribuiraju tako sastavljen sistem. Jedna od prvih takvih firmi je Slackware. Tako nastaju Linux distribucije.

Danas su popularne distribucije Mint, Ubuntu, Red Hat, Debian, Suse, Fedora, itd. a od verzija Unix-a koriste se još Solaris, Mac OS i BSD. Linux je danas vodeći operativni sistem kada su u pitanju serveri, radne stanice i superračunari.

## Najčešće korišćene komande

Generalno, Unix komande se sastoje iz tri dela (poslednja dva su opciona):

1. ime komande
2. opcije komande (slova ili reči kojima prethodi -)
3. argumenti (nazivi datoteka, direktorijuma, proizvoljni tekst)

Na primer: **ls -l /tmp**

### Osnovne naredbe

strelica gore dole	– za ponavljanje prethodno zadatih komandi
Ctrl+R niska	– traženje prethodno izvršenih komandi koje sadrže niska
history	– pregledanje istorije korišćenih komandi
tab	– za napisanu nisku lista sve komande koje počinju tom niskom
tab	– dovršava ime fajla; ako postoji više fajlova sa napisanim početkom, ispisaće se i onaj deo imena tih fajlova koji je zajednički za sve njih.
Ctrl+c	– prekid za većinu programa
sudo -s	– dobijanje administratorskih privilegija na sistemu

copy se u terminalu radi obeležavanjem levim tasterom, a paste klikom na srednji taster miša.

### Naredbe za pomoćne stranice

man 2 write	– pomoćna stranica za write sistemski poziv
man scanf	– pomoćna stranica za funkciju scanf
man man	– pomoćna stranica za komandu man
man -k <string>	– prikazuje sve komande koje u svojim kratkim opisima sadrže string
info gcc	– dobija se opis gcc naredbe
man ls	– pomoćna stranica za komandu ls

Kada se pri pregledanju man stranice otkuca /examples onda se na toj stranici markiraju sva pojavljivanja niske examples. Ovo se odnosi i na bilo koju drugu nisku i omogućava brzu pretragu man stranice za tom niskom.

## Naredbe za kretanje, kreiranje, listanje, brisanje

- pwd – štampa tekući direktorijum
- cd direktorijum – promena tekućeg direktorijuma
- cd .. – vraćanje u prethodni (roditeljski) direktorijum
- cd /home/mr02685 – prelazak u direktorijum čija je putanja navedena
- cd ~ – prelazak u home direktorijum
- ls opcija direktorijum – listanje sadržaja direktorijuma uz korišćenje opcija.

### Opcije:

- R – rekurzivno listanje svih poddirektorijuma navedenog direktorijuma
- l – listanje sa dodatnim informacijama
- a – listanje svih fajlova, tj. i onih koji počinju sa '.', a inače se ne prikazuju
- s – ispis datoteka i njihove veličine u kilobajtima
- lh – ispis datoteka i čitljiv ispis njihove veličine

du -sch direktorijum – ispis veličine navedenog direktorijuma

cp fajl lokacija\_nove\_kopije – kopiranje datoteka

mkdir direktorijum – kreiranje direktorijuma

Pri brisanju biti naročito oprezan!!! U Unix-u ne postoji recycle bin tako da je preporučljivo praviti rezervne kopije podataka.

rm fajl – brisanje datoteke

rmdir direktorijum – brisanje praznog direktorijuma

rm -r direktorijum – brisanje direktorijuma koji nije prazan

mv fajl nova\_lokacija – pomeranje datoteke

## Dobijanje informacija o korisnicima

users – spisak korisnika ulogovanih na sistem

w – izveštaj u vidu spiska o trenutnom statusu sistema i spisak korisnika ulogovanih na sistem zajedno sa procesima koji im pripadaju

w user – kao i prethodno, ali se prikazuju samo podaci za zadatog korisnika

who – spisak korisnika ulogovanih na sistem

who am i – ispis ličnih podataka za tekućeg korisnika

- finger – traži zadate podatke o korisniku i prikazuje informacije iz inicijalne konfiguracije korisnikovog naloga, kao i ostale sadržaje iz korisnikovih .plan i .project fajlova (ukoliko postoje). Prikazuje se i da li je korisnik prijavljen i koliko dugo traje sesija korisnika.
- finger user – kao i prethodno, ali se prikazuju samo podaci za zadanog korisnika
- last – lista ko je kad radio na serveru počev od nekog datuma.
- last user – isto to samo sa određenu osobu.
- groups user – prikazuje grupe kojima pripada korisnik

### Komunikacija sa serverom i drugim korisnicima

- telnet – pristup udaljenom računaru, nepreporučljivo koristiti zbog slabije sigurnosti
- ssh računar – pristup udaljenom računaru, koristi kriptovanje pa se smatra sigurnim. Npr:  
ssh [mi11002@alas.matf.bg.ac.rs](mailto:mi11002@alas.matf.bg.ac.rs) – prijavljivanje na alas; odjavljivanje se vrši komandom exit
- hotssh [mi11002@alas.matf.bg.ac.rs](mailto:mi11002@alas.matf.bg.ac.rs) – prijavljivanje na alas (možda se program hotssh mora dodatno instalirati); može se koristiti grafičko okruženje za prebacivanje fajlova
- scp fajl nova\_lokacija – skraćeno od secure copy, kriptovano kopiranje podataka. Npr:  
scp -r mi11002@alas.matf.bg.ac.rs:BB /home/ja – kopiranje direktorijuma BB sa alasa u direktorijum /home/ja (-r se koristi za rekurzivno kopiranje svih poddirektorijuma)
- scp mi11002@alas.matf.bg.ac.rs:datoteka /home/ja/ – kopiranje jedne datoteke
- pine – provera pošte
- write – slanje poruke drugom korisniku, npr:  
write mi11002 tekst\_poruke (ctrl+c za kraj)

Na Matematičkom fakultetu postoje 2 servera: alas (studentski) i poincare (nastavnički).

### Naredbe za rad sa datotekama i tekstem

- stat fajl – prikazuje informacije o navedenoj datoteci
- emacs r.& – pokretanje editora emacs; & označava da se proces pokreće u pozadini, tj. shell se može koristiti za druge namene dok se proces izvršava
- less – pregled teksta stranu po stranu. Akcije u komandi less:  
/rec Osvetljava sve pojave niske rec. Ako se rec ne navede, to znači find next.  
q Izlaz.
- cat – povezuje fajlove i koristi se za ispis sadržaja fajlova  
cat -b lista\_fajlova numerisanje svih ne-blanko linija izlaza naredbe cat i to počev od 1  
cat -n lista\_fajlova numerisanje svih linija izlaza naredbe cat i to počev od 1

cat -s lista_fajlova	zamena svih višestrukih blanko linija jednom linijom
cat > fajl	upis karaktera koji se unose sa tastature u fajl do pritiska Ctrl+C, ali se pritom prethodni sadržaj fajla gubi
cat >> fajl	dopisivanje karaktera koji se unose sa tastature na kraj fajla
wc lista_fajlova	– brojač redova, reči i karaktera u fajlovima ili standardnom ulazu (ako nije navedena lista)
wc -L	ispisuje dužinu najduže linije
wc -l	broji prelaske u novi red
wc -w	broji reči
wc -c	broji samo karaktere
head fajl	– ispisuje početak fajla (prvih 10 redova)
head -num fajl	– ispisuje prvih num redova fajla
tail fajl	– ispisuje kraj fajla (poslednjih 10 redova)
tail -num fajl	– ispisuje poslednjih num redova
tail -f fajl	– ispisuje kraj fajla, kao i redove koji se naknadno (npr. iz drugog procesa) dopisuju u fajl
sort lista_fajlova	– prikazuje sortirane redove jednog ili više fajlova; ako nema zadatih fajlova, onda se sortiraju podaci sa standardnog ulaza i prikazuju se na standardni izlaz
sort f lista_fajlova	– pri sortiranju se ne razlikuju velika i mala slova (ignore case)
find direktorijum -name ime_fajla	– pronalazi rekurzivno sve fajlove u direktorijumu sa navedenim imenom
locate	– efikasna pretraga fajla u sistemu

### Naredba grep

Grep pretražuje navedene datoteke i ispisuje sve linije koje odgovaraju navedenom obrascu.

Primeri:

grep opcija uzorak lista\_fajlova – ispisuje sve linije u kojima se nalazi uzorak. Opcije:

-c	– ispisuje broj pronađenih linija u kojima se nalazi uzorak
-h	– ispred linije ispisuje i naziv datoteke gde je nađen uzorak
-i	– zanemaruje se veličina slova pri pretrazi
-E regularni_izraz	– prikazuje sve linije koje sadrže regularan_izraz

Regularan izraz može da sadrži:

^	početak linije
---	----------------

\$	kraj linije
.	bilo koji karakter
[lista_karaktera]	bilo koji karakter iz skupa u zagradama, npr [1-2abc9-D]
[^lista_karaktera]	tačno jedan karakter koji nije među navedenim
izraz{n}	izraz se pojavljuje n puta
izraz*	izraz se pojavljuje 0, 1 ili više puta
izraz+	izraz se pojavljuje bar jednom
izraz?	izraz se pojavljuje 0 ili jednom

### Još neke česte naredbe

clear	– brisanje ekrana
ps	– prikazivanje procesa koji se trenutno izvršavaju u ime korisnika koji je pozvao ovu komandu
ps aux	– prikazivanje svih procesa na sistemu
pstree	– prikazivanje procesa koji se trenutno izvršavaju u obliku drveta
top	– spisak procesa sortiran po procentu zauzeća procesora; spisak se stalno ažurira (izlaz sa q)
top -u mi11002	– spisak procesa samo korisnika mi11002
id	– identifikacija
echo poruka	– ispisuje na ekran poruka
echo poruka > fajl	– piše poruku u fajl; prethodni sadržaj fajla biva obrisano.
echo poruka >> fajl	– dopisuje poruku na kraj fajla.
date	– ispisuje trenutni datum i vreme
cal	– prikazuje kalendar za tekući mesec
cal -y	– prikazuje kalendar za tekuću godinu
alias	– koristi se za pravljenje skraćenice za neku komandu koja se koristi; skraćenice su aktivne dok se korisnik ne izloguje
alias listaj="ls -l"	– pravimo skraćenicu listaj ako koristimo shell bash.
listaj	– sad je ovo isto kao da smo otkucali ls -l.
alias	– spisak svih alijasa.
unalias skraćenica	– brisanje definicije za skraćenicu
df opcije fajl_sistemi	– ispisuje slobodan i zauzet prostor na montiranim particijama
/etc/fstab	– datoteka koja sadrži trenutno montirane sisteme datoteka

<code>which</code>	– određuje lokaciju na disku gde se nalazi uneta komanda
<code>tar</code>	– naredba za komprimovanje (dekomprimovanje) podataka
<code>tar -pczf ime.tar.gz .</code>	– pravljenje arhive <code>ime.tar.gz</code> koja sadži sve iz tekućeg direktorijuma
<code>tar -xvzf ime.tar.gz</code>	– raspakivanje arhive <code>ime.tar.gz</code> u tekući direktorijum
<code>tar -xvzf ime.tar.gz -C direktorijum</code>	– raspakivanje arhive <code>ime.tar.gz</code> u navedeni direktorijum
<code>cat /etc/*version*</code>	– ime i verzija distribucije koja se koristi
<code>uname</code>	– informacije o verziji Linuxa kao i o procesorima
<code>uname -r</code>	– prikaz verzije kernela
<code>lsb_release -a</code>	– informacije o standardu kao i o samoj verziji sistema
<code>cat /proc/cpuinfo</code>	– ispisuje informacije o procesoru i jezgrima procesora
<code>cat /proc/meminfo</code>	– ispisuje informacije o memoriji

### Nadovezivanje naredbi

Prednost Unix-a u odnosu na Windows je da se neki zadaci mogu lakše i brže obaviti, bez instalacije novih programa. Osnovna ideja je da se pravi veliki broj malih programa i da se njihovim nadovezivanjem obavljaju složeniji zadaci. Za nadovezivanje naredbi koristi se `|` (pipe), izlaz jedne naredbe prosleđuje se kao ulaz druge naredbe.

PIPE primeri:

<code>w   sort</code>	– sortira izlaz naredbe <code>w</code>
<code>man -k copy   grep string</code>	– nalaženje komandi koje sadrže <code>copy</code> i <code>string</code> u svojim kratkim opisima
<code>man 2 write   grep return</code>	– prikazuje sve linije u pomoćnoj stranici <code>write</code> sistemskog poziva koje sadrže <code>return</code>
<code>last mi11002   grep Mon</code>	– kada je korisnik ulogovan na <code>alas</code> , prikazuje sva logovanja ponedeljkom korisnika sa korisničkim imenom <code>mi11002</code>

### Zadaci

0. Napraviti direktorijum `Os/1` pomoću komande `mkdir`. Sve što se nadalje radi na ovom času smestiti u ovaj direktorijum.

Skripte imaju ekstenziju `.sh` i potrebno je da prvi red u njima bude `#!/bin/bash`

1. Napisati skriptu koja prvo cisti ekran, zatim štampa datum i kalendar za tekući mesec i na kraju štampa broj korisnika trenutno logovanih na sistem.

2. Napisati skriptu koja prvo prikazuje sve funkcije koje u svom kratkom opisu sadrže reci `print` i `char`. Zatim prikazati zadnjih 5 redova sortirane datoteke `linux.txt`. Potom prikazati sve fajlove u tekućem direktorijumu koji sadrže tekst `/bin/bash` (koristiti `grep`, tako da mu se prosledi `string` i fajlovi u kojima se pretražuje).



3. Napisati shell skriptu koja brise sadržaj datoteke hello.txt (kreira je ako ne postoji) i u toj datoteci ostavlja poruku Zdravo! (koristiti echo). Zatim prikazati red koji sadrži vreme poslednje modifikacije datoteke hello.txt. Potom prikazati sve procese koji se izvršavaju u ime korisnika koji je pozvao komandu i čiji pid počinje dvojkom i ima bar 3 cifre. Na kraju prikazati sve procese na sistemu koji se izvršavaju u ime root korisnika korišćenjem komande ps aux (potrebno je da niska root bude na početku reda).

4. Napisati shell skriptu koja prikazuje lokaciju komande du, prikazuje sve linije datoteke neki\_tekst.txt koje se završavaju sa , i potom prikazuju praznu liniju na ekran. Iz iste datoteke prikazati:

- Sve linije koje počinju slovom A i ispisati praznu liniju

- Sve linije koje počinju slovom A, B ili C i potom ispisati praznu liniju

- Sve linije koje počinju slovom A, M ili X i potom ispisati praznu liniju

- Sve linije u kojima je barem jedan od brojeva 10, 11, 12, 20, 21 ili 22 i potom praznu liniju.

5. Napisati shell skriptu koja kreira direktorijum "a" i u njemu kreira datoteke 1.txt, 2.txt, 3.txt koje sve sadrže poruku "Zdravo". Skripta potom kreira direktorijum "b", pozicionira se u direktorijum "a", kopira sadržaj "a" u sadržaj "b", izvrši pomeranje natrag u roditeljski direktorijum i brisanje direktorijuma "a". Prikazati sve fajlove sa ekstenzijom .sh u tekućem direktorijumu kao i njihove veličine.

6. (Pokrenuti na alasu) Napisati shell skriptu koja ispisuje kada je korisnik sa korisničkim imenom mi09004 na alasu poslednji put primio mail (ako je ta informacija trenutno dostupna). Ispisati imena svih fajlova u direktorijumu ~/public\_html/ i svim rekurzivno u svim njegovim poddirektorijumima koji imaju ekstenziju .html. Prvo iskopirati direktorijum na drugo mesto kako prilikom naredne naredbe ne bi došlo do presnimavanja već urađenih zadataka usled greške.

7. Pomoću komande tar spakovati sve prethodne fajlove u arhivu 5.tar.

8. Ispisati broj linija koji trenutno zauzima fajl history. NAPOMENA: sa ovom naredbom može doći do problema pri korišćenju u skriptima.

9. Iskopirati fajl 1.sh pomoću komande scp u direktorijum public\_html na nalog na alasu. Ulogovati se pomoću ssh i proveriti da li je fajl ispravno iskopiran.