

ELEKTROTEHNIČKI FAKULTET UNIVERZITETA U BEOGRADU



OSNOVE LINUX-A
-Diplomski rad-

Kandidat:

Zora Mihaljčić 2012/461

Mentor:

doc. dr Zoran Čiča

Beograd, Avgust 2016.

SADRŽAJ

SADRŽAJ.....	2
1. UVOD.....	4
2. UVOD U UNIX I LINUX OPERATIVNE SISTEME	5
2.1. ISTORIJAT UNIX OPERATIVNOG SISTEM	5
2.2. LINUX	6
2.3. LINUX DISTRIBUCIJE.....	6
3. PREGLED LINUX SISTEMA.....	7
3.1. LINUX KERNEL	7
3.2. UPRAVLJANJE PROCESIMA	10
3.3. PROCESI I NITI.....	10
3.3.1. Dodeljivanje procesora procesima	10
3.3.2. Komunikacija između procesa	10
3.3.3. Upravljanje memorijom	10
3.4. IZVRŠAVANJE KORISNIČKIH PROGRAMA	11
3.5. ULAZNO-IZLAZNI SISTEM	11
3.6. SISTEMI DATOTEKA I AKTIVNO UNIX STABLO	11
3.7. MREŽNE STRUKTURE.....	12
3.8. INIT	12
3.9. PRIJAVLJIVANJE SA TERMINALA	12
3.10. SYSLOG.....	12
3.11. PERIODIČNO IZVRŠAVANJE KOMANDI.....	12
3.12. GRAFIČKI KORISNIČKI INTERFEJS	13
3.13. MREŽNI RAD.....	13
3.14. PRIJAVLJIVANJE SA MREŽE	13
3.15. DELJENJE DATOTEKA I MREŽNI SISTEMI DATOTEKA	14
3.16. ELEKTRONSKA POŠTA.....	14
3.17. ŠTAMPANJE	14
4. SHELL PROGRAM I NJEGOVE VRSTE	15
4.1. BASH SHELL	16
4.1.1. Opšte osobine.....	16
4.1.2. Bash shell - specifikacija i detalji implementacije	17
4.2. OSNOVNE KOMANDE USHELL-U.....	17
4.2.1. Komanda ls	18
4.2.2. Komanda cd	18
4.2.3. Komanda rm	19
4.2.4. Komanda mkdir.....	19
4.2.5. Komanda mv	19
4.2.6. Komanda cp	19
4.2.7. Komanda less.....	19
4.2.8. Komanda uname	19
4.2.9. Komanda tar	19
4.3. SKRIPT FAJLOVI.....	20
4.4. PROMENJIVE	20
4.5. REGULARNI IZRAZI.....	21
4.6. GREP KOMANDA.....	22
5. LINUX REČNIK.....	23

6. ZAKLJUČAK.....	29
-------------------	----

1. UVOD

Linux je prvobitno predstavljen kao slobodan operativni sistem za računare zasnovane na 32-bitnoj Intel x86 seriji mikroprocesora. Od tada je prebačen na više računarski platformi nego bilo koji drugi operativni sistem. Linux je glavni operativni sistem na serverima, mejnfrejm računarima i superračunarima. Više od 90% superračunara koristi neku verziju Linuxa, uključujući 10 najbržih. Linux takođe radi na mikrosistemima u kojima je operativni sistem obično ugrađen u uređaj. To su, na primer, mobilni telefoni, tablet računari, ruteri, televizori i konzole za igru. Operativni sistem Android je takođe izgrađen na osnovu Linux jezgra.

Cilj rada je da se na osnovu integracije podataka iz savremene literature stekne osnovno znanje o radu na Linux operativnom sistemu, njegova rasprostranjenost i upotreba u savremenoj tehnologiji. Uprošćena forma omogućava početnicima da se lakše snađu u radu sa ovim operativnim sistemom, a takođe čitajući istorijat prošire svoje teorijsko znanje. Pored toga, rad čitaocu daje i uvid u shell okruženje i njegove osnovne komande.

Rad se sastoji iz pet delova. Nakon kratkog uvoda sledi istorijat Linux operativnog sistema. U narednim poglavljima dat je opšti pregled Linux sistema, značajnih delova jezgra i osnovnih servisa Linux operativnog sistema, shell okruženja i njegovih osnovnih komandi. Poslednje poglavlje predstavlja svojevrstni rečnik pojmova, komandi i aplikacija koje prate Linux operativni sistem.

2. UVOD U UNIX I LINUX OPERATIVNE SISTEME

U ovom poglavlju opisan je kraći istorijat UNIX i Linux operativnih sistema.

2.1. Istorijat UNIX operativnog sistem

Unix je stabilan, moćan i fleksibilan operativni sistem visokih performansi pogodan za izvršavanje kritičnih aplikacija od visoke važnosti. Čvrsto je povezan sa mrežnim servisima TCP/IP protokola. Umesto servera sa klasičnim serijskim terminalima UNIX server se nalazi u mreži, pri čemu sa radnim stanicama ostvaruje vezu preko LAN/WAN mreže i TCP/IP skupa protokola.

Većina UNIX sistema, poput Sun Solaris i IBM AIX je komercijalna, što znači da korisnik mora da plati licencu za korišćenje, a izvorni kod nije raspoloživ. To je jedan od glavnih razloga nastanka i popularnosti Linux operativnog sistema, koji zadržava većinu dobrih osobina UNIX sistema, a dodatno se odlikuje raspoloživim izvornim kodom i praktično besplatnim korišćenjem.

Razvoj UNIX operativnog sistema počeo je sredinom 1960-tih godina u AT&T Bell laboratorijama u saradnji sa kompanijom General Electric i tehnološkim institutom Massachusetts. Projekat, odnosno operativni sistem MULTICS (*Multiplexed Information and Computing Service*) predstavljao je interaktivni operativni sistem namenjen da opslužuje veliki broj korisnika čiji su terminali direktnim serijskim ili modemskim komunikacionim kanalima povezani na centralizovani server. Takav koncept operativnog sistema bio je preambiciozan za tadašnji stepen razvijenosti hardvera, pri čemu se prvenstveno misli na procesorsku snagu i količinu systemske memorije. MULTICS nije doživeo svoju praktičnu primenu jer se posle nekoliko godina razvoja pokazao kao preskup i preambiciozan projekat od koga su AT&T Bell laboratorije odustale. Konkretno, MULTICS je preteča UNIX sistema koji se smatra jednim od najkvalitetnijih i najrasprostranjenijih operativnih sistema, na čijem se razvoju radi preko 30 godina, sa tendencijom dalje egzistencije i usavršavanja.

Dva fundamentalna imena vezana za razvoj UNIX operativnog sistema su Ken Thompson, MULTICS sistemski programer u Bell laboratorijama, i Dennis Ritchie, poznatiji kao tvorac programskog jezika C. Godine 1969, Ken Thompson je započeo razvoj novog operativnog sistema za DEC PDP-7 racunar, napravivši redukovani MULTICS, odnosno UNICS (*Uniplexed Information and Computing Service*). Radi lakšeg izgovora i pisanja, ime UNICS je kasnije evoluiralo u UNIX. UNIX je prvobitno napisan u asemblerskom jeziku, a samim tim bio je potpuno zavisano od klase procesora za koji se realizuje. Godine 1971, Ritchie je napisao programski jezik C kao viši programski jezik koji omogućava systemsko programiranje, a zatim sa Thompsonom preveo kod UNIX sistema na C, što se može smatrati prekretnicom i jednim od najznačajnijih poteza u razvoju operativnih sistema.

Zahvaljujući C jeziku UNIX je mogao biti prenešen na razne računarske arhitekture sa vrlo malo programskih modifikacija, što je svakako bio ključ uspeha i popularnosti UNIX operativnog sistema. Nakon prevođenja na C jezik, autori su u cilju daljeg unapređenja prosledili izvorni kod UNIX sistema univerzitetima širom Amerike, pri čemu su programeri sa Berkeley univerziteta u Kaliforniji dominantno uticali na dalji razvoj. Danas postoji veliki broj različitih vrsta UNIX

sistema koje su zasnovane ili na industrijskom standardu SVR4 (nova varijanta UNIX sistema - *System V Release 4*) ili na BSD (*Berkeley Software Distribution*) distribuciji.

2.2. Linux

Jedna od poslednjih varijanti UNIX operativnih sistema, čiji je razvoj započeo Linus Torvalds 1991. godine na Univerzitetu u Helsinkiju, je Linux. Torvalds je svoj operativni sistem koji objedinjuje oba standarda, SRV4 i BSD, objavio na Internetu i podsticao druge programere širom sveta da se priključe njegovom daljem razvoju. Linux je svojom koncepcijom stabilnog, a jeftinog operativnog sistema doživeo veliku ekspanziju i popularnost. Simbol Linux sistema je mali pingvin (Tux). Linux je prvobitno namenjen 32-bitnim Intel x86 mikroprocesorima (počevši od 80386), na kojima može funkcionisati kao radna stanica (*workstation*) ili kao server. Jezgro Linux sistema je kasnije modifikovano i prilagođeno procesorima koji ne pripadaju Intel x86 klasi, među kojima treba istaći Intel IA-64, DEC Alpha, SUN SPARC/UltraSPARC, Motorola 68000, MIPS, PowerPC i IBM mainframe S/390. Može se konstatovati da današnji Linux u odnosu na bilo koji operativni sistem podržava najširi spektar procesora i računarskih arhitektura.

2.3. Linux distribucije

Brojne profitne i neprofitne organizacije čine Linux raspoloživim u formi distribucija, odnosno različitih kombinacija kernela, sistemskog softvera i različitih korisničkih aplikacija. Većina distribucija sadrži kolekciju CD/DVD medijuma na kojima se nalaze operativni sistem, izvorni kod, detaljna dokumentacija, kao i štampana uputstva za instalaciju i upotrebu sistema. Cene ovakvih distribucija su u većini slučajeva simbolične, osim ako se u distribuciji nalazi komercijalan softver ili je distribucija specifične namene.

Osnovna komponenta svake Linux distribucije je kernel operativnog sistema. Osim kernela i sistemskog softvera u distribuciji se nalaze i instalacioni alati, softver za podizanje operativnog sistema (boot loader), razne korisničke aplikacije (kancelarijski paketi - office suite, softver za manipulaciju bit-mapiranih slika) i serverski paketi.

U poznatije Linux distribucije spadaju: Debian GNU/Linux, Linux Mandrake, Red Hat Linux, Slackware Linux i SuSE Linux.

3. PREGLED LINUX SISTEMA

Linux je višekorisnički, višeprocetni operativni sistem sa potpunim skupom UNIX kompatibilnih alata, projektovan tako da poštuje relevantne POSIX (*Portable Operating System Interface*) standarde. Linux sistemi podržavaju tradicionalnu UNIX semantiku i potpuno implementiraju standardni UNIX mrežni model.

Linux operativni sistem sastoji se od kernela, sistemskog softvera, korisničkih aplikacija, programskih prevodilaca i njihovih odgovarajućih biblioteka (GCC - GNU C Compiler i C biblioteka za Linux) i dokumentacije. Kernel je jezgro operativnog sistema - on omogućava konkurentno izvršavanje procesa, dodeljuje im memoriju i druge resurse i obezbeđuje mehanizam za ostvarivanje usluga operativnog sistema. Kernel štiti korisničke procese od direktnog pristupa hardveru - procesi pristupaju hardveru korišćenjem sistemskih poziva kernela, čime se obezbeđuje jedna vrsta zaštite između samih korisnika. Sistemski programi koriste kernel u cilju implementacije različitih servisa operativnog sistema. Svi programi, uključujući i sistemske, funkcionišu na nivou iznad kernela, što se naziva korisnički režim rada, dok se sistemske aktivnosti poput pristupa hardveru obavljaju na nivou kernela, odnosno u sistemskom režimu rada (supervisory mode). Razlika između sistemskih i aplikativnih programa je u njihovoj nameni, npr. aplikacije su namenjene za razne korisne aktivnosti (kao što su obrada teksta i slike), dok su sistemski programi namenjeni za rad sa sistemom i administraciju.

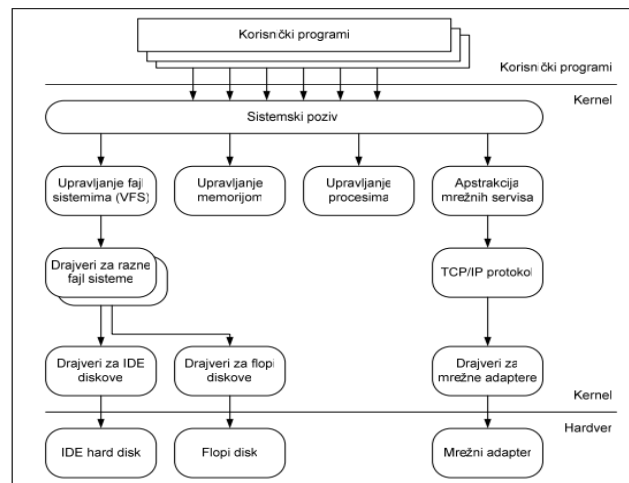
3.1. Linux kernel

Tri osnovne verzije Linux kernela su početna verzija, verzija 1.x i verzija 2.x. Početna verzija 0.01, koju je 1991. godine kreirao Linus Torvalds, podržavala je samo Intel 80386 kompatibilne procesore, mali broj hardverskih uređaja i Minix sistem datoteka. Mrežni servisi nisu imali kernelsku podršku. Verzija 1.0, nastala u martu 1994. godine, uključivala je podršku za standardne TCP/IP mrežne protokole, BSD-kompatibilni socket interfejs za mrežno programiranje i drajversku podršku za mrežne kartice. Ova verzija je dodatno podržavala ext i ext2 sisteme datoteka, široku klasu SCSI disk kontrolera, kao i brojne hardverske uređaje. Verzija 1.2 (mart 1995) je poslednja verzija Linux kernela namenjena isključivo PC arhitekturi. U verziji 2.0 (jun 1996) uvedena je podrška za više arhitektura (Motorola i Intel procesori, Sun Sparc i PowerMac sistemi), kao i podrška za višeprocetorsku arhitekturu (SMP). Dodatno, poboljšano je upravljanje memorijom i uvećane su performanse TCP/IP protokol steka, a ugrađena je i podrška za unutrašnje kernelske niti (internal kernel thread).

Kernel je odgovoran za najznačajnije funkcije operativnog sistema. Dve osnovne karakteristike kernela su:

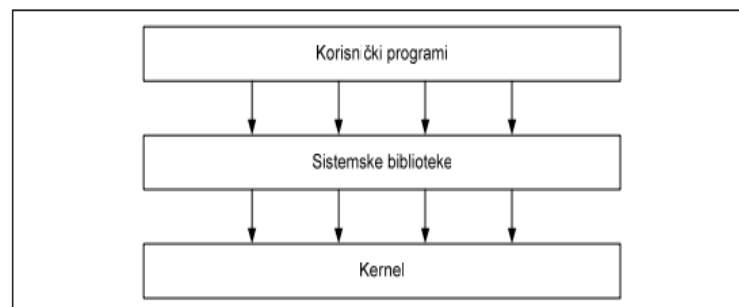
- kernel kod se izvršava u kernel modu u kome je jedino moguće pristupati svim komponentama hardvera,
- kompletan kernel kod i sve kernel strukture podataka čuvaju se u istom adresnom prostoru (monolithic).

Kod većine UNIX sistema aplikacije se preko sistemskog poziva direktno obraćaju kernelu, kao što je prikazano na slici 3.1.1.



Slika 3.1.1. Jezgro Linuxa.

Kod Linux sistema sistemski pozivi se upućuju kernelu preko sistemskih biblioteka koje definišu standardni set funkcija preko kojih aplikacije komuniciraju sa kernelom. Ovaj metod komunikacije sa kernelom prikazan je na slici 3.1.2.



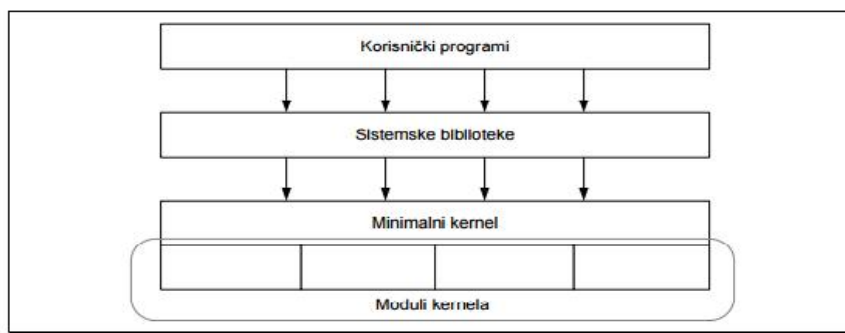
Slika 3.1.2. Komunikacija između aplikacija i kernela.

Sistemski programi izvršavaju specifične upravljačke poslove, kao što je konfigurisanje mrežnih uređaja i protokola, punjenje kernelskih modula itd.

Moduli kernela su delovi kernelskog koda koji mogu da se prevedu, napune u memoriju ili izbace iz memorije nezavisno od ostatka kernela. Kernelski moduli implementiraju drajvere za hardverske uređaje, novi sistem datoteka, mrežne protokole, itd. Moduli omogućavaju raznim programerima da napišu i distribuiraju drajvere koji ne moraju da prođu GPL licencu. Moduli kernela omogućavaju micro-kernel arhitekturu, odnosno realizaciju minimalne stabilne konfiguracije kernela bez dodatnih drajvera.

Module Linux kernela čine tri komponente:

- Upravljanje modulom, koja omogućava punjenje modula u kernelsku memoriju i komunikaciju modula sa ostatkom kernela, proveru da li je modul u memoriji i da li se koristi i izbacivanje modula iz memorije (pod uslovom da se modul ne koristi),
- registracija drajvera, koja omogućava modulu da objavi ostatku kernela da je novi drajver u memoriji i da je raspoloživ za korišćenje. Kernel održava dinamičku tabelu drajvera koji se pomoću posebnog seta programa mogu napuniti ili izbaciti iz memorije u svakom trenutku,
- rezolucija konflikata, odnosno mehanizam koji služi da spreči hardverske konflikte tako što omogućava drajveru da rezerviše hardverske resurse (IRQ, DMA, ports) i time spreči druge drajvere ili autoprobe funkciju da ih koriste. Na slici 3.1.3. prikazana je struktura modularnog Linux kernela.



Slika 3.1.3. Struktura modularnog Linux kernela.

Linux kernel čini nekoliko značajnih komponenti:

- upravljanje procesima
- upravljanje memorijom
- upravljanje sistemima datoteka (VFS)
- apstrakcija mrežnih servisa
- podrška za hardverske uređaje
- podrška za različite sisteme datoteka
- podrška za TCP/IP

Kritične komponente Linux kernela su upravljanje procesima i upravljanje memorijom. Komponenta za upravljanje memorijom kontroliše dodeljivanje memorije i swap prostora procesima, kernelskim komponentama kao i bafersko keširanje. Komponenta za upravljanje procesima kreira procese i omogućava višeprocetni rad (multitasking) dodeljujući procesor procesima po odgovarajućem algoritmu.

3.2. Upravljanje procesima

Linux koristi standardni UNIX proces mehanizam (fork) koji razdvaja kreiranje procesa i njegovo izvršenje u dve različite operacije:

- sistemski poziv fork, koji kreira novi proces,
- sistemski poziv exec, koji izvršava program u resursima novostvorenog procesa.

Pod UNIX sistemom sve informacije koje operativni sistem mora čuvati da bi kontrolisao jedan proces predstavljaju kontekst tog procesa. Pod Linux operativnim sistemom, svaki proces je u potpunosti opisan identitetom, okolinom, i kontekstom.

3.3. Procesi i niti

Linux koristi istu internu reprezentaciju za procese i niti. Nit (thread) je jednostavno novi proces koji deli adresni prostor roditelja. Za razliku od novog procesa koji pomoću sistemskog poziva *fork* formira novi kontekst sa jedinstvenim adresnim prostorom, nit nastaje pomoću sistemskog poziva *clone* koji kreira novi kontekst, ali dozvoljava novom procesu da deli adresni prostor roditelja.

3.3.1. Dodeljivanje procesora procesima

Linux koristi dva algoritma za dodelu procesora procesima (*process-scheduling algorithms*):

- Time-sharing algoritam za korektno raspoređivanje između procesa (*fair preemptive scheduling*). Dodela se vrši na osnovu prioriteta procesa koji definiše korisnik i kredita (efektivni prioritet) koji raste s porastom vremena čekanja na processor.
- Real-time algoritam za procese gde su apsolutni prioriteti mnogo značajniji od ravnomerne raspodele. Linux je ipak soft real-time operativni sistem.

3.3.2. Komunikacija između procesa

Komunikacija između procesa obuhvata obaveštavanje procesa o događaju i prenos podataka s jednog procesa na drugi. Kao i UNIX sistem, Linux informiše procese u korisničkom režimu o događaju putem signala. Procesi u kernel modu umesto signala koriste specijalnu vrstu deljive memorije (wait.queue struktura) za interprocesnu komunikaciju. Za prosleđivanje podataka između procesa koristi se *pipe* mehanizam, koji omogućava jednosmernu razmenu podataka putem komunikacionog kanala koji proces nasleđuje od roditelja, i deljiva memorija, koja je brza i fleksibilna, ali zahteva sinhronizaciju.

3.3.3. Upravljanje memorijom

Upravljanje memorijom obuhvata upravljanje operativnom (RAM) memorijom i upravljanje virtuelnom memorijom. Upravljanje operativnom, odnosno fizičkom memorijom obuhvata dodeljivanje (alokaciju) i oslobađanje stranica (pages, normal extent), grupe stranica (large extent) i malih memorijskih blokova (small extent).

Fizička memorija se deli na udružene blokove čije su veličine stepeni broja 2. Blokovi se prema potrebi alokacije dalje razbijaju na manje blokove ili se parovi udružuju u veće celine.

Sistem virtuelne memorije povećava ukupan adresni prostor koji je dostupan procesima. Sistem kreira stranicu virtuelne memorije na zahtev, upravlja punjenjem te stanice u fizičku memoriju sa diska i povratkom stranice na disk u swap prostor.

3.4. Izvršavanje korisničkih programa

Linux podržava brojne formate za punjenje i izvršavanje programa. Među njima svakako treba istaći stari UNIX format *a.out* i novi *elf* format koji je maksimalno prilagođen konceptu virtuelne memorije.

Zaglavlje ELF formata opisuje sekcije programa. Sekcije programa su po veličini prilagođene veličini stanice virtuelne memorije. Program kod kog su funkcije iz sistemske biblioteke direktno ugrađene u kod programa je program sa statičkim povezivanjem. Glavni nedostatak ovakvog načina povezivanja je povećanje veličine koda, jer svaki poziv funkcije iz biblioteke kopira celu funkciju u kod. Takođe, sa veličinom koda raste i količina memorije koja je potrebna za njegovo izvršavanje. Na drugoj strani, dinamičko povezivanje je efikasnije u smislu iskorišćenja memorije - sama funkcija se ne kopira u kod, tako da je za izvršenje potrebna manja količina memorije, ali se programi po nepisanom pravilu izvršavaju sporije.

3.5. Ulazno-izlazni sistem

Linux deli uređaje u tri klase: blok uređaje (poput diskova i CD-ROM ureaja), karakter uređaje (poput štampa) i mrežne uređaje. Svaki uređaj je predstavljen specijalnom datotekom (device node, device file) koja se nalazi u direktorijumu `/dev` root sistema datoteka. Kada korisnik upisuje podatke u datoteku koja predstavlja neki uređaj ili čita iz te datoteke, vrši se neka ulazno-izlazna operacija, odnosno sistem šalje ili prima podatke sa uređaja koji je predstavljen tom datotekom. Time se ukida potreba za postojanjem posebnih programa (samim tim i posebnom metodologijom programiranja ulazno-izlaznih operacija) neophodnih za rad sa uređajima.

3.6. Sistemi datoteka i aktivno Unix stablo

Linux sistemi datoteka koriste hijerarhijsku strukturu stabla i semantku UNIX sistema datoteka. Interno, kernel sakriva detalje i upravlja različitim sistemima datoteka preko jednog nivoa apstrakcije koji se naziva virtuelni sistem datoteka VFS.

Aktivno Linux stablo datoteka čini jedan ili više sistema datoteka koji su montirani na odgovarajuće direktorijume preko kojih im se pristupa. Osnovu aktivnog stabla datoteka čini korenski sistem datoteka (root filesystem), čiji koren (root) direktorijum ujedno predstavlja i koren direktorijum aktivnog stabla datoteka. Zavisno od hardverske konfiguracije i odluke administratora sistema, struktura aktivnog Linux stabla može biti jednostavna (aktivno stablo realizovano jednim sistemom datoteka), ili složena (aktivno stablo realizovano većim brojem sistema datoteka - root, /boot, /var, /usr, /home ...).

3.7. Mrežne strukture

Umrežavanje je ključno područje funkcionalnosti Linux sistema. Linux koristi standardni TCP/IP protokol stek kao osnovni komunikacioni protokol, a dodatno podržava i brojne druge protokole koji nisu uobičajeni za komunikaciju dva UNIX sistema (AppleTalk, IPX, Samba). Interno, umrežavanje pod Linux sistemom obuhvata tri softverska nivoa: socket interfejs, protokol drajvere i drajvere za mrežne kartice.

3.8. Init

Proces *init* se pokreće kao prvi proces na svakom Linux sistemu i to je poslednja akcija koju kernel obavlja prilikom podizanja sistema. Kada se pokrene, *init* nastavlja proces podizanja operativnog sistema, obavljajući razne inicijalne procedure kao što su provera i akitviranje sistema datoteka i pokretanje servisa (daemons).

3.9. Prijavljivanje sa terminala

Funkciju prijavljivanja preko serijskih linija i sistemske konzole (bez grafičkog okruženja) obezbeđuje proces *getty*. Proces *init* kao proces roditelj pokreće posebnu instancu programa *getty* za svaki terminal na kome je dozvoljeno prijavljivanje na sistem. Dalje, *getty* čita korisničko ime i o tome izveštava proces login, koji proverava unešeno korisničko ime i lozinku. Ako su korisničko ime i lozinka korektni login pokreće proces shell koji prihvata i izvršava korisničke komande. Proces *init* će otpočeti novu instancu procesa *getty* kada proces komandnog interpretera završi aktivnost, odnosno kada se korisnik odjavi sa sistema, ili ako se login proces završi nekorektnim prijavljivanjem.

3.10. Syslog

Kernel i mnogi sistemski programi generišu razna upozorenja i poruke o greškama koje se upisuju u datoteke, tako da se mogu pregledati kasnije. Program koji obavlja funkciju upisivanja poruka u datoteke je *syslog*. Program se može konfigurisati da poruke aranžira u različite datoteke na osnovu stepena značaja i procesa koji je poruke generisao.

3.11. Periodično izvršavanje komandi

Većina korisnika, uključujući sistem administratore često ima potrebu za periodičnim izvršavanjem neke komande. Na primer, sistem administrator može da zakaže periodično izvršenje

komande koja čisti direktorijume sa privremenim datotekama (/tmp i /var/tmp), čime sprečava prepunjenje diska.

Funkciju periodičnog izvršenja komandi obezbeđuje servis *cron*. Svaki korisnik može u svojoj crontab datoteci definisati komande koje želi periodično da izvršava i vreme kada te komande treba izvršiti. *Cron demon* vodi računa o izvršenju komandi u specificiranom vremenu. Servis koji je sličan cron servisu je at, ali za razliku od *cron* servisa komandu pokreće samo jednom, u specificiranom trenutku.

3.12. Grafički korisnički interfejs

Linux ne ugrađuje korisnički interfejs u kernel - korisnički interfejs se implementira na korisničkom nivou. Koriste se dve vrste interfejsa, alfanumerički i grafički, čime se postiže veća fleksibilnost sistema. Nedostatak ovakvog uređenja je relativno složena implementacija različitih korisničkih interfejsa za svaki program, što operativni sistem čini težim za učenje i korišćenje. Primarno grafičko okruženje koje se koristi u Linux sistemima je X Window System (skraćeno - X). X ne implementira korisnički interfejs već obezbeđuje alate pomoću kojih grafički korisnički interfejs može da se implementira. Najčešće korišteni window menadžeri na Linux sistemima su: KDE i Gnome.

3.13. Mrežni rad

Umrežavanje je čin povezivanja dva ili više računara relativno visokog stepena autonomije u cilju omogućavanja njihove međusobne komunikacije. Iako proces povezivanja računara u mrežu i konfigurisanja mrežnog okruženja može biti komplikovan, krajnji rezultat je veoma koristan. Korisnici mogu koristiti deljene mrežne resurse poput štampača, direktorijuma i up-linkova, a takođe se mogu omogućiti i mehanizmi centralizovane autentifikacije i administracije. Većina osnovnih servisa Linux sistema, poput sistema datoteka, štampanja i arhiviranja podataka može se realizovati pomoću mrežnih funkcija operativnog sistema.

3.14. Prijavljivanje sa mreže

Procedura prijavljivanja sa mreže razlikuje se od klasične procedure prijavljivanja na sistem, koja zahteva postojanje posebne fizičke serijske linije za svaki terminal na kom je potrebno omogućiti prijavljivanje i rad.

Za svakog korisnika koji se prijavljuje na sistem preko mreže postoji posebna virtuelna mrežna konekcija. Broj tih konekcija teorijski može biti neograničen, tako da nije moguće izvršavati poseban getty proces za svaku virtuelnu konekciju. Za potrebe prijavljivanja sa mreže, umesto gomile getty procesa, postoji glavni demon proces (master daemon, wrapper) koji osluškuje sve nadolazeće zahteve. Ukoliko se pojavi zahtev za prijavljivanje na sistem, wrapper kao proces roditelj pokreće novi proces (telnet ili rlogin demon), sličan getty procesu, koji dalje upravlja pokušajem prijavljivanja, dok glavni demon nastavlja da osluškuje nove zahteve sa mreže.

3.15. Deljenje datoteka i mrežni sistemi datoteka

U značajnije servise UNIX i Linux sistema spada i deljenje datoteka na mreži. Mrežni sistemi datoteka funkcionišu na sledeći način: zahtevi za operacijama nad datotekama i direktorijumima šalju se po mreži na računar na čijim se diskovima sistem datoteka nalazi, a korisnik ima utisak da se sve datoteke nalaze na lokalnom sistemu datoteka. Na ovaj način omogućava se deljenje datoteka na krajnje jednostavan način, pošto se ne zahteva nikakva modifikacija korisničkih programa. Najčešće korišćeni tip mrežnih sistema datoteka je NFS (*Network File System*), koji je razvila kompanija Sun Microsystems.

3.16. Elektronska pošta

Elektronska pošta je najpopularniji metod komunikacije putem računara. Svaki korisnik ima prijemno poštansko sanduče (incoming mailbox) u vidu datoteke u specijalnom formatu koja se po pravilu nalazi na direktorijumu /var/spool/mail. Kada neko šalje poštu specijalan program, koji omogućava prijem i slanje pisama, locira prijemno poštansko sanduče i dodaje pismo u mailbox datoteku. Ukoliko se prijemno poštansko sanduče nalazi na drugom računaru, pismo se najpre šalje preko mreže ka destinacionom računaru, čiji će mejl program dalje pismo distribuirati u odgovarajuće sanduče.

Poštanski sistem čine dve osnovne vrste programa: MTA - Mail Transfer Agent (na primer sendmail) koji vrše isporuku pošte u lokalno poštansko sanduče i prosleđuju poštu udaljenim računarima, i MUA - Mail User Agent (pine, mutt ili elm) koje služe korisnicima za čitanje i slanje pošte.

3.17. Štampanje

Jedan štampač u jednom vremenskom trenutku može da koristi samo jedan korisnik, ali je krajnje neekonomično ne dozvoliti upotrebu štampača u različitim vremenskim trenucima većem broju korisnika. Zato štampačima upravlja program koji implementira red čekanja za dati štampač (printer queue): svi zahtevi za štampu postavse se u red čekanja i kada štampač obavi jedan posao upravljački program mu automatski šalje sledeći.

Program koji implementira red čekanja čuva kopiju koju treba štampati na disku print servera, čime se omogućava aplikativnom programu da preda zahtev za štampu u red i nakon toga nastavi svoje aktivnosti. Disk je relativno brz uređaj u odnosu na štampač, tako da predaja zahteva u red traje kratko. Nakon predaje zahteva aplikativni program ne mora da čeka da zahtev bude odštampan, a takođe ne mora ni da kontroliše proces štampe, tako da nastavlja sa daljim radom.

4. SHELL PROGRAM I NJEGOVE VRSTE

Shell je interfejs između korisnika i kernela, odnosno jezgra operativnog sistema. Shell prihvata komande koje korisnik zadaje, zatim ih interpretira i potom ih izvršava, pri čemu po potrebi pokreće odgovarajuće programe. Na UNIX sistemima postoji više različitih komandnih interpretera, a korisnici u toku rada po potrebi mogu preći iz jednog u drugi.

Komandni interpreter je proces koji obavlja sledeće funkcije u cilju obezbeđivanja interfejsa između korisnika i operativnog sistema:

- interpretaciju komandne linije,
- pokretanje programa,
- redirekciju ulaza i izlaza,
- povezivanje komandi u pipeline,
- zamenu imena datoteka,
- rukovanje promenljivim i kontrolu okoline,
- shell programiranje.

Shell program interpretira korisničke komande bilo da se iste unose pojedinačno od strane korisnika ili da se čitaju liniju po liniju iz fajla koji se zove shell skript. Neke vrste shell-ova koje su dostupne na UNIX i GNU/Linux sistemima su:

- sh - Bourne shell - shell koji se inicijalno koristio na UNIX operativnim sistemima. Ima veoma mali broj osobina u poređenju sa shell-ovima nove generacije, ali sve osobine su zadržane sa ciljem održanja kompatibilnosti sa UNIX sistemima.
- bash - Bourne Again Shell - shell koji predstavlja super skup sh shell-a. Danas jedan od najkompletnijih shell-ova koji postoje.
- csh - C shell - shell koji ima sintaksu preuzetu iz programskog jezika C. Shell koji je favoriziran od strane programera.
- tcsh - Turbo C shell - super skup csh.
- ksh - Korn shell

Unutar file shells (/etc/shells) nalazi se spisak svih shell-ova dostupnih na sistemu na kojem radimo.

```
/bin/ash
/bin/bash
/bin/bash1
/bin/csh
/bin/false
/bin/ksh
/bin/sh
/bin/tcsh
/bin/true
/bin/zsh
/usr/bin/csh
/usr/bin/ksh
/usr/bin/passwd
/usr/bin/bash
/usr/bin/rbash
/usr/bin/tcsh
/usr/bin/zsh
/etc/shells lines 1-17/17 (END)
```

Slika 4.1. Spisak shell-ova dostupnih na sistemu.

Promena samog shell-a realizuje se samo navođenjem imena samog shell-a.

```
neon@cable-89-216-161-5:~> sh
sh-3.00$ bash
neon@cable-89-216-161-5:~> csh
hostname: Unknown host
hostname: Unknown host
/home/neon> tcsh
hostname: Unknown host
hostname: Unknown host
/home/neon> zsh
neon@:~> bash
neon@cable-89-216-161-5:~> █
```

Slika 4.2. Promena shell-a.

4.1. Bash shell

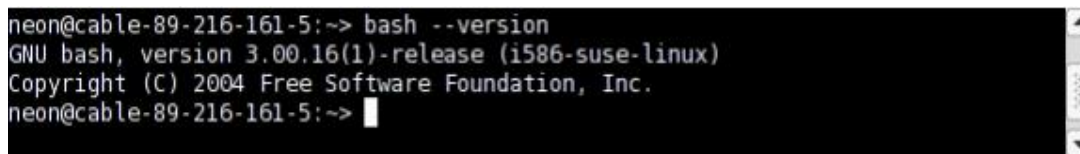
4.1.1. Opšte osobine

Bash shell nastao je proširenjem sh shell-a. Takođe u sam bash dodane su i neke osobine csh i ksh tako da je bash jedan od najkompletnijih shell programa danas.

Bash korisniku omogućava:

- izvršavanje komandi iz komandne linije (edit, execute)
- podesiva dubina pamćenja prethodno izvršenih komandi (history command)
- kontrola poslova (job control)
- upotrebu shell funkcija i aliasa
- upotrebu indeksiranih nizova bez ograničenja (fizička ograničenja su jedini limit)
- aritmetičke operacije

Za ispravnu upotrebu shell-a na samom startu neophodno je da znamo verziju shell-a u kojoj radimo. Provera verzije realizuje se komadnom **bash --version**.



```
neon@cable-89-216-161-5:~> bash --version
GNU bash, version 3.00.16(1)-release (i586-suse-linux)
Copyright (C) 2004 Free Software Foundation, Inc.
neon@cable-89-216-161-5:~> █
```

Slika 4.1.1. Komanda bash --version.

4.1.2. Bash shell - specifikacija i detalji implementacije

Da bi što bolje i jasnije shvatili ulogu samog shell-a kao i da bi na pravi način mogli da posmatramo stvari detaljno ćemo se pozabaviti samim načinom implementacije shell-a kao i načinom funkcionisanja shell programa. Bash je moguće pokrenuti na nekoliko različitih načina:

1. Poziv preko interaktivnog login shell-a ili sa --login (shell se dobija nakon što se korisnik autorizuje na sistem unosom korisničkog imena i lozinke). Ovakav način pokretanja shell-a realizuje i čitanje sledećih konfiguracionih i profil file-ova prilikom pokretanja shell-a:

- /etc/profile
- ~/.bash_profile, ~/.bash_login ili ~/.profile (čita se prvi postojeći fajl)
- ~/.bash_logout

2. Poziv preko interaktivnog ne-login shell-a (nema potrebe za autentikacijom na sistem). Primer ovakvog poziva shell-a je otvaranje shell-a iz grafickog korisničkog interfejsa (klikom na ikonu ili izborom opcije iz menija nakon čega se ne zahteva logovanje) Ovakav način pokretanja shell programa čita ~/.bashrc file prilikom pokretanja shell-a.

3. Poziv preko neinteraktivnog shell-a (standardan način pokretanja shell-a prilikom izvršavanja skriptova). Fajl koji se čita određen je promjenjivom BASH_ENV .

4. Poziv sa sh komandom (/etc/profile, ~/.profile). Kada se poziva interaktivno ENV GNU Linux Shell promjenjiva može da ukazuje na dodatne startup informacije.

5. POSIX mod (opcija koja je uključena pozivom ugrađene komande **set-o posix** ili pozivom bash programa sa **--posix** opcijom), File koji se čita je određen ENV promjenjivom.

6. Remote poziv (komanda **rshd**) - neophodno je voditi računa o sigurnosti informacija prilikom upotrebe remote alata kao što su rlogin, telnet, rsh i rcp jer se podaci šalju nekriptovani. U slučaju potrebe za remote izvršavanje određenih komadi, transfer fajlova ili neke druge aktivnosti poželjno je upotrebiti Secure SHell (SSH). Razne verzije klijentskih programa su dostupne (putty ili neki klijent koji podržava ssh protokol).

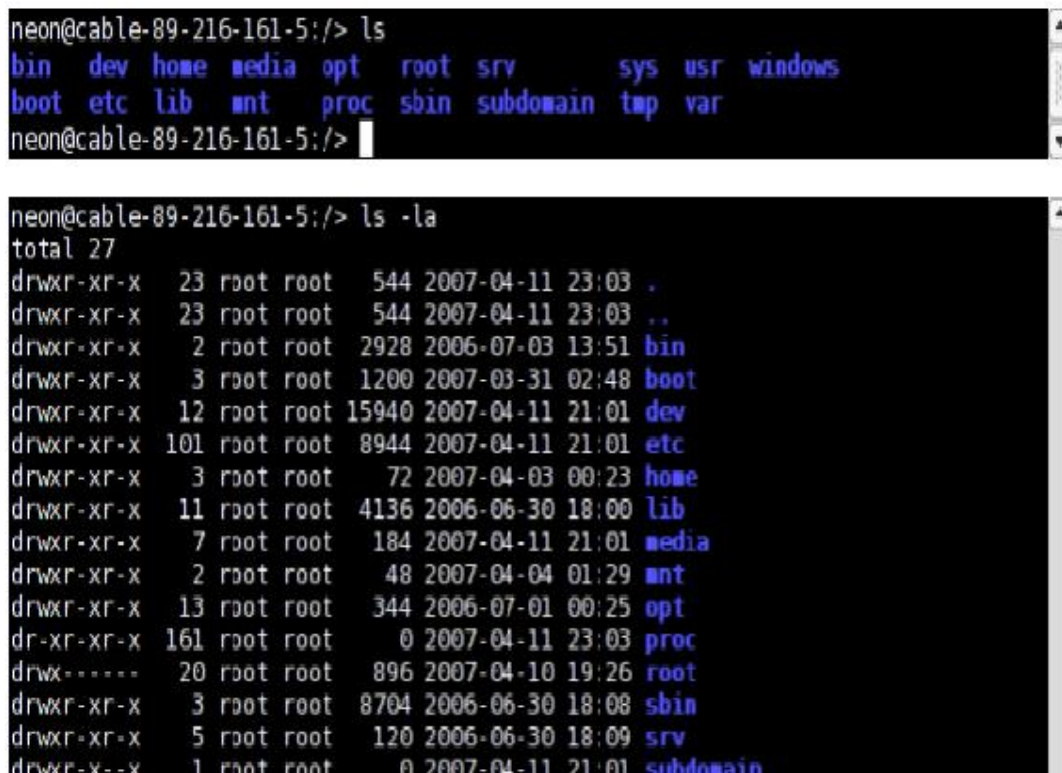
7. Poziv kada UID nije jednak EUID (nikavi fajlovi se ne čitaju u ovom slučaju)

4.2. Osnovne komande ushell-u

Pregled osnovnih komandi koje koristimo u radu sa linux shell-om.

4.2.1. Komanda *ls*

Komandom *ls* izlistavamo sadržaj direktorijuma. Komanda *ls* daje mogućnost korišćenja velikog broja opcija. Komandom *man ls* dobijamo detalje vezane za samu komandu. Prikazaćemo samo dva slučaja upotrebe komande *ls* i to za izlistavanje sadržaja direktorijuma (bez skrivenih fajlova) i za izlistavanje kompletnog sadržaja direktorijuma (sa skrivenim fajlovima).



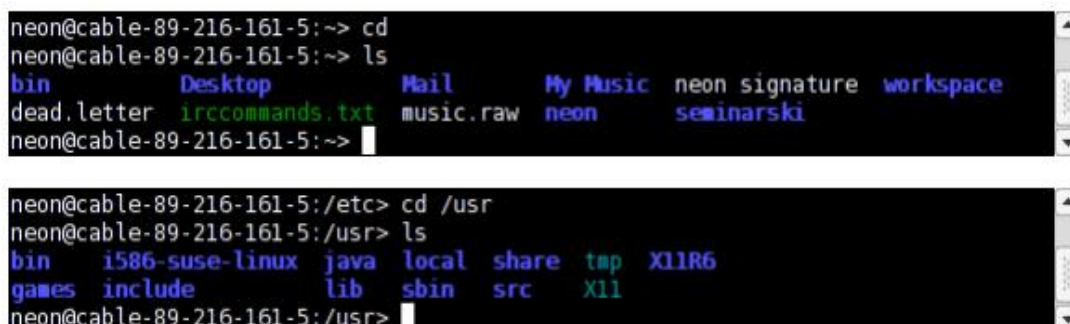
```
neon@cable-89-216-161-5:~/> ls
bin  dev  home  media  opt  root  srv      sys  usr  windows
boot etc  lib  mnt   proc sbin  subdomain tmp  var
neon@cable-89-216-161-5:~/>

neon@cable-89-216-161-5:~/> ls -la
total 27
drwxr-xr-x  23 root root   544 2007-04-11 23:03 .
drwxr-xr-x  23 root root   544 2007-04-11 23:03 ..
drwxr-xr-x   2 root root  2928 2006-07-03 13:51 bin
drwxr-xr-x   3 root root  1200 2007-03-31 02:48 boot
drwxr-xr-x  12 root root 15940 2007-04-11 21:01 dev
drwxr-xr-x 101 root root  8944 2007-04-11 21:01 etc
drwxr-xr-x   3 root root    72 2007-04-03 00:23 home
drwxr-xr-x  11 root root  4136 2006-06-30 18:00 lib
drwxr-xr-x   7 root root   184 2007-04-11 21:01 media
drwxr-xr-x   2 root root    48 2007-04-04 01:29 mnt
drwxr-xr-x  13 root root   344 2006-07-01 00:25 opt
dr-xr-xr-x 161 root root     0 2007-04-11 23:03 proc
drwx----- 20 root root   896 2007-04-10 19:26 root
drwxr-xr-x   3 root root  8704 2006-06-30 18:08 sbin
drwxr-xr-x   5 root root   120 2006-06-30 18:09 srv
drwxr-xr-x   1 root root     0 2007-04-11 21:01 subdomain
```

Slika 4.1.2. Dva načina pokretanja bash-a.

4.2.2. Komanda *cd*

Promena trenutnog direktorijuma. Komandom *cd* bez navođenja putanje do određenog direktorijuma skaćemo na default direktorijum koji je određen sa HOME promenljivom. Ukoliko se pored komande navede i putanja skaćemo direktno u navedeni direktorijum. Prikazaćemo navedena dva slučaja.



```
neon@cable-89-216-161-5:~/> cd
neon@cable-89-216-161-5:~/> ls
bin      Desktop  Mail      My Music  neon signature workspace
dead.letter  irccommands.txt music.raw  neon      seminarski

neon@cable-89-216-161-5:~/etc> cd /usr
neon@cable-89-216-161-5:~/usr> ls
bin  i586-suse-linux  java  local  share  tmp  X11R6
games  include          lib  sbin  src    X11
neon@cable-89-216-161-5:~/usr>
```

Slika 4.2.2. Dva načina promene trenutnog direktorijuma pomoću komande *cd*.

4.2.3. Komanda *rm*

Komandom *rm* ime_dokumenta brišemo fajl odnosno fajlove u zavisnosti od načina zadavanja komande. Ukoliko želimo obrisati neki folder i rekurzivno njegov kompletan sadržaj onda koristimo komandu **rm -r ime_direktorijuma**.

4.2.4. Komanda *mkdir*

Komandom *mkdir* kreiramo novi direktorijum.

4.2.5. Komanda *mv*

Komandom *mv* prebacujemo sadržaj sa jedne fizičke lokacije na drugu (source -> destination). **mv source destination** (komanda u svom osnovnom obliku).

4.2.6. Komanda *cp*

Komandom *cp* kopiramo sadržaj sa jedne lokacije na drugu (source -> destination). **cp source destination** (komanda u svom osnovnom obliku).

4.2.7. Komanda *less*

Komanda koju koristimo za prikaz fajlova. Komanda *less* izvršava se brže i kod pregledanja većih fajlova što joj daje prednost u odnosu na upotrebu editora vi za pregledanje fajlova. Nakon korišćenja *less* komande vraćamo se u komandnu liniju izborom tastera q.

4.2.8. Komanda *uname*

Komandu *uname* koristimo za dobijanje osnovnog skupa informacija o sistemu na kojem radimo. Osnovni skup informacija podrazumeva verziju kernela (naziv), verziju procesora, naziv mašine, hardversku platformu, operativni sistem, itd.

4.2.9. Komanda *tar*

Komanda *tar* koristi se kao alat za rad sa arhivama. Ova komanda veoma je rasprostranjena i svaki korisnik linux operativnih sistema dosta često je u kontaktu sa arhivama (izvorišni kodovi, dokumenti koji se šalju mejlovima poželjno je da budu arhivirani i kompresovani). Za sve detalje i mogućnosti komande *tar* korisnik se upućuje na manual stranice ove komande (man tar). Pomoću *tar* komande možemo da arhiviramo fajlove kao i da raspakujemo već postojeće tar archive.

4.3. Skript fajlovi

Skript fajlovi predstavljaju skupove komandi organizovane u okviru celina poznatih pod naziv skript fajlovi ili skriptovi. Skript fajl sadrži niz sukcesivno navedenih komandi koje se izvršavaju onim redosledom kojim su navedeni u samom skript fajlu. Osobine kvalitetnog skript fajla:

- pokretanje bez grešaka
- realizacija zadatka za koji je predviđen
- programska logika je jasno definisana i jednoznačna
- skript ne radi stvari koje nisu predviđene
- skript treba da bude upotrebljiv (*reusable*)

4.4. Promenjive

Prilikom rada sa shell-om razlikujemo globalne promenjive, lokalne promenjive, a takođe postoji podela promenjivih prema sadržaju. Globalne promenjive ili promenjive okruženja dostupne su u svim shell-ovima. Da bi prikazali sve promenjive okruženja koristimo komandu *printenv*. Analogno komandi *printenv* moguće je koristiti i komandu *env*. Rezultat je identičan u oba slučaja.

Lokalne promenjive moguće je prikazati pomoću komande *set* međutim tada prikazujemo i sve promenjive okruženja kao i funkcije.

Promenjive prema sadržaju možemo podeliti na:

- string
- integer
- array
- constant

Postavljanje lokalnih promenjivih realizuje se navođenjem imena promenjive, a zatim se navodi vrednost promenjive iza znaka jednakosti (prema konvenciji lokalne promenjive se pišu malim slovima). `variable_name = "value"` Lokalne promenjive vidljive su samo u shell-u u kojem su i postavljene.

Globalne promenjive postavljamo komandom `export`. `export VARIABLE_NAME = "value"` (prema konvenciji velikim slovima se označavaju globalne promenjive). Svaki shell može da ima i rezervisane promenjive.

```
GNOME2_PATH=/usr/local:/opt/gnome:/usr
XKEYSYMDB=/usr/X11R6/lib/X11/XKeysymDB
DESKTOP_STARTUP_ID=
HOST=
TERM=xterm
SHELL=/bin/bash
PROFILEREAD=true
HISTSIZE=1000
XDM_MANAGED=/var/run/xdmctl/xdmctl-:0,maysd,mayfn,sched,rsvd,method=classic
STYLE=Industrial
GNOME_PATH=/opt/gnome:/usr
GTK_RC_FILES=/etc/opt/gnome/gtk/gtkrc:/home/neon/.gtkrc-1.2-gnome2
WINDOWID=73404812
QTDIR=/usr/lib/qt3
XSESSION_IS_UP=yes
GROFF_NO_SGR=yes
JRE_HOME=/usr/lib/jvm/java/jre
USER=neon
DESKTOP_LAUNCH=gnome-open
LIBGL_DRIVERS_PATH=/usr/X11R6/lib/modules/dri
LD_LIBRARY_PATH=/usr/X11R6/lib
LS_COLORS=no=00;fi=00;di=01;34:ln=00;36:pi=40;33:so=01;35:do=01;35:bd=40;33;01:c
d=40;33;01:or=40;31:ex=00;32:*.cmd=00;32:*.exe=01;32:*.com=01;32:*.bat=01;32:*.b
tm=01;32:*.dll=01;32:*.tar=00;31:*.tbz=00;31:*.tgz=00;31:*.rpm=00;31:*.deb=00;31
:*.arj=00;31:*.taz=00;31:*.lzh=00;31:*.zip=00;31:*.zoo=00;31:*.z=00;31:*.Z=00;31
:*.gz=00;31:*.bz2=00;31:*.tb2=00;31:*.tz2=00;31:*.tbz2=00;31:*.avi=01;35:*.bmp=0
1;35:*.fli=01;35:*.gif=01;35:*.jpg=01;35:*.jpeg=01;35:*.mng=01;35:*.mov=01;35:*.
mpg=01;35:*.pcx=01;35:*.pbm=01;35:*.pgm=01;35:*.png=01;35:*.ppm=01;35:*.tga=01;3
5:*.tif=01;35:*.xbm=01;35:*.xpm=01;35:*.dl=01;35:*.gl=01;35:*.wmv=01;35:*.aiff=0
0;32:*.au=00;32:*.mid=00;32:*.mp3=00;32:*.ogg=00;32:*.voc=00;32:*.wav=00;32:
GNOME_ICON_PATH=/usr/share/images:/usr/share/YaST2/theme/current/icons/48x48/app
s:/opt/kde3/share/icons/crystalsvg/32x32/apps:/opt/kde3/share/icons/crystalsvg/3
2x32/devices:/opt/kde3/share/icons/crystalsvg/32x32/mimetypes:/opt/kde3/share/ic
ons/crystalsvg/32x32/filesystems:/opt/kde3/share/icons/crystalsvg/32x32/actions:
```

Slika 1.4. Rezultat komandi printenv i env.

4.5. Regularni izrazi

Regularni izrazi su pattern-i koji opisuju skupove stringova. Regularni izrazi konstruišu se analogno aritmetičkim izrazima upotrebljavajući operatore u kombinaciji sa manjim izrazima. Osnovni gradivni element svakog regularnog izraza je karakter.

Takođe, regularni izrazi uključuju u svoj sastav i metakaraktere koji imaju specijalno značenje unutar regularnih izraza.

Da bi postojanje metakaraktera bilo što jasnije navodimo neke metakaraktere sa opisom značenja odnosno sa efektima koje imaju:

- . - označava jedan karakter (tj. džoker koji predstavlja bilo koji karakter)

? - ono što se nalazi ispred je opciono i biće uzeto u obzir najviše jednom

* - ono što se nalazi ispred uzima se 0 ili proizvoljan broj puta

+ - ono što se nalazi ispred uzima se 1 ili proizvoljan broj puta

{N} - ono što se nalazi ispred uzima se u obzir tačno N puta

{N,} - ono što se nalazi ispred uzima se u obzir N ili više puta

{N,M} - ono što se nalazi ispred uzima se u obzir namanje N a najviše M puta

Regularni izrazi imaju svoju primenu u okviru komandi koje koristimo za rad u shell-u, naročito onih koje zahtevaju neke generičke promene. Jedna od komandi koji koriste regularne izraze je i **grep** komanda.

4.6. Grep komanda

Grep komanda pretražuje fajl i upoređuje sa sadržajem koji se prosledi grep komandi. Kao rezultat dobijamo linije unutar fajla koji se pretražuje i koji sadrži tražene paterne. Za detalje vezane za komandu **grep** uvek je dostupna pomoć upotrebom komande `man grep`. npr. Ukoliko želimo da saznamo u kojoj liniji u fajlu `/etc/passwd` nalazi string `"root"` onda unosimo sledeću komandu: **grep -n root /etc/passwd**.

```
neon@cable-89-216-161-5:/etc> grep -n root /etc/passwd
1: root:x:0:0:root:/root:/bin/bash
neon@cable-89-216-161-5:/etc> █
```

Slika 4.6.Prikaz grep komande.

5. LINUX REČNIK

Cilj ovog poglavlja je da obezbedi svojevrsni rečnik pojmova, komandi i aplikacija. Namena mu je da bude posetnik za početnike, koji mogu da ga odštampaju i da ima tako uvek bude pri ruci.

Pre proučavanja rečnika potrebno je znati format linux komandi, koje se zadaju u sledećoj formi:

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

A

acoread - čitač fajlova u PDF formatu

adduser - dodavanje novog korisnika

alias - kreiranje shell aliasa za komande koje koristimo iz shell-a

alias <skraćenica> “<puna komanda>” – Na ovaj način u shellu tesh formiramo skraćenice za određene komande koje kasnije koristimo. Skraćenice su aktivne dok se ne izlogujemo.

alias <skraćenica>=“<puna komanda>” – Isto to samo za shell bash

alias l “ls -l” – pravimo skraćenicu l ako koristimo shell tesh

anacron – izvršavanje komandi periodično, a ne podrazumeva neprekidno pokretanje mašine

aspell - provera unesenog teksta (ima ogroman broj mogućnosti)

B

bash - Bourne Again SHell

bg - pokretanje procesa u pozadini

bzip2 - fajl kompresor

C

cat - spaja i prikazuje fajlove na standardnom izlazu

cd <direktorijum> - promena direktorijuma

cd .. – vraćanje u prethodni direktorijum
cd. – tekući direktorijum
cd – vraćanje u home direktorijum
chgrp - promena grupe chmod - promena prava pristupa fajlu
chown - promena vlasnika i grupe
cp - kopiranje filova crontab - upravljanje crontab fajlovima
csh - C shell

D

date - postavljanje ili prikazivanje datuma
df - prikaz iskorišćenosti diska
diff - razlika između fajlova
du – Disk Usage – Koliko u kilobajtima zauzima koji direktorijum (uključujući i tekući)
du~ – slično, samo za korisnika home direktorijuma
du -b – slično, samo u bajtima
du -a – ovako radi i za fajlove i za direktorijume
df – Disk free – podaci o particijama na hard disk

E

echo “<poruka>” – ispisuje poruke
echo “Neki tekst” ><fajl> – piše poruku u fajl. Prethodni sadržaj fajla biva obrisan.
echo “Neki novi tekst” >><fajl> – dopisuje poruku na kraj fajla
emacs - pokretanje emacs editora teksta
exit - izlaz iz trenutnog shell-a
export - dodavanje funkcije(a) u shell okruženje

F

fdisk - komanda za manipulaciju sa particijama
fetchmail - dohvatanje mejla sa POP, IMAP, ETRN ili ODMR servera
fg - prebacivanje posla u frontend (iz pozadinskog)
file – određivanje tipa fajla
find [<direktorijum>] <expression> – štampaju se direktorijumi, fajlovi i poddirektorijumi
find~ -name “*” – lista sve fajlove u home direktorijumu i njegovim poddirektorijumima

find / -name <fajl> – lista sve pojave fajla u svim direktorijumima hard diska

find [<direktorijum>] -name <ime fajla> -exec <komanda> – izvršava komandu koju navedemo nad fajlovima koje je pronašao kao argumentima (ako imamo pristup tim fajlovima)

find [<direktorijum>] -perm <ovlašćenje> – traži fajlove po ovlašćenju

ftp - servis za transfer fajlova

G

gimp - program za rad sa slikama

grep - pretraga i prikaz određene linije prema određenom paternu

grep -v <uzorak><lista fajlova> – lista sve linije osim onih koje sadrže <uzorak>

grep -F <lista fiksnih karaktera> – pojedini specijalni karakteri . ^ \$ i još neki se tretiraju kao regularni izrazi, i odmah se razvijaju u listu. Da se to ne bi dogodilo koristimo opciju -F

grep -x '<tekst>' – spisak svih linija koje sadrže jedino <tekst>

grep -n <uzorak><lista fajlova> – pored svake linije ispisuje i njen redni broj (indeksira ih)

grep -E '<regularni izraz>' – spisak svih linija koje sadrže <regularan izraz>

grub - shell za podešavanje opcija za butovanje sistema

gv - PostScript i PDF čitač fajlova

gzip - kompresovanje ili proširivanje fajlova

H

head <fajl> – ispisuje početak fajla (prvih 10 slova)

host - DNS alat

httpd - Apache hypertext transfer protocol

I

iptables - administrator ip tabela

J

jar - Java ARchive alat

jobs - lista pozadinskih procesa

K

kill - gašenje određenih procesa (svih procesa)

kill <pid> – ubijanje procesa čiji je pid <pid> (vidi se pomoću ps). Pid je skraćeno od Process ID

kill %<job> – kao i malopre, broj <job> se vidi pomoću jobs

L

ldapmodify - modifikovanje LDAP unosa ldapsearch - LDAP search alat

lilo - Linux Boot Loader locate - lociranje fajlova

logout - zatvaranje trenutne sesije shell-a

ls <lista direktorijuma ili fajlova> – List – pandan DOS-ove naredbe dir

ls * – lista i one fajlove koji su unutar poddirektorijuma

ls d * – ne listaju se direktorijumi

ls a[Bb]* – lista sve fajlove čije je prvo slovo je a, drugo slovo B ili b

ls -l – detaljnije informacije o fajlovima, npr. Atributi

ls -s – lep ispis (brojevi pored fajlova govore koliko je veliki fajl u kilobajtima)

ls -R – ispis sadržaja tekućeg direktorijuma i poddirektorijuma

ls -p – da bi u screenu direktorijumi bili označeni kosom crtom (/)

ls -o – da bi u screenu izvesni fajlovi bili osvetljeni

lynx - veb pretraživač (txt mod) - nema slika, ali veoma brz

M

mail - slanje i primanje mejla

man - prikaz stranica koji sadrže pomoć

mkdir - pravljenje direktorijuma

more - prikaz sadržaja fajla na jednoj strani (korisno za pregled sadržaja ukoliko fajl ima više strana)

mount - mountovanje fajl sistema ili prikaz informacija o već mountovanim fajl sistemima

mv - premeštanje i rename (promjena imena) fajlova

mv <fajl><novo ime fajla> – move – slično kao kopiranje, samo sto se originalni fajl neće sačuvati, tj. ovako se vrši preimenovanje

mv <lista fajlova><direktorijum> – promena mesta boravka fajlova

mv -i <lista fajlova><direktorijum> – za svaki fajl pita da li da se izvrši move

P

passwd - promena lozinke

ping - slanje echo request-a host-u

pwd - prikaz u kome se trenutno direktorijumu nalazimo

ps – Process Status – spisak procesa koji se izvršavaju

ps a – slično, samo za sve procese (all)

ps u <username> – prikaz svih pokrenutih procesa za <username>

ps l – kompletan prikaz procesa koji se izvršavaju (sa prioritetima)

ps ax – prikazuje trenutne procese svih ljudi (i od roota) i procese koji nisu vezani za terminal. Opcija x se najviše koristi za otkrivanje procesa koji su se otrgli kontroli.

Q

quota - prikazuje kapacitet diska i kvote koje su dodeljene za korisnika

R

reboot - zaustavljanje i ponovno pokretanje sistema rm - uklanjanje fajl/ova

rpm - RPM package manager

S

scp - sigurna konekcija na udaljeni računar

sleep - čekanje određenog vremenskog perioda

sort <lista fajlova> - ispisuje fajlove sortirane po prvom polju (koloni) na ekran (stdout)

su - prelaz sa korisnika na korisnika (su - user)

T

tail - prikaz poslednjeg dela fajla

tar - komanda za arhiviranje (najčešće se koristi u kombinaciji sa alatom za arhiviranje)

traceroute - prikaz route paketa koji se uzima od hosta

U

uptime - prikazuje vreme koje je proteklo od pokretanja sistema

update - vraćanje izmenjenih buffer-a od strane kernela na disk

userdel - brisanje naloga za korisnika i odgovarajućih fajlova

V

vim - poboljšani vi editor

W

w – ko je ulogovan

who – koga još ima na sistemu

who am i – ispis ličnih podataka

whoami – ispisuje se samo username

which – ovako se dobija gde se nalazi program za promenu šifre

which<file> – Gde je na disku fajl koji se startuje tim imenom

howrite - pisanje poruke drugom korisniku

Z

zgrep - pretraga kompresovanih fajlova prema određenom paternu

zmore - filter za prikaz kompresovanog teksta

6. ZAKLJUČAK

Linux je računarski operativni sistem sastavljen i razvijen u skladu sa principima slobodnog softvera. Glavni deo Linuxa je jezgro (takozvani kernel) ili centralni deo operativnog sistema čiju je prvu verziju razvio Linus Torvalds 1991.godine. Samo jezgro operativnog sistema postalo je dostupno javnosti 5.oktobra iste godine kada je postavljeno na FTP server Univerziteta u Helsinkiju.

U tekstu koji sledi bice reči o prednostima i manama Linux operativnog sistema u odnosu na njegove glavne konkurente.

1. Linux operativni sistem je napravljen tako da virusa gotovo nema
2. Ima vrlo mali broj programa koji se zbog neke greške zaustave sami od sebe
3. Retko se dešava da neki program zaustavi ceo sistem
4. Ni jedan program ne zaustavlja rad drugih programa. Kada jedan program ne radi dobro, drugi sasvim normalno nastavljaju svoje izvršavanje.
5. Programi su manji, ne zahtevaju mnogo memorije
6. Deinstairanje i instaliranje programa gotovo nikada ne briše vlastite podatke
7. GNU/Linux je odličan za programiranje.Postoje skoro svi programski jezici za Linux,kompajleri za C, C++, ANSI C, Pascal, Fortran, programerski jezici kao Lisp, COBOL, ADA itd.
8. Linux nudi gotovo besplatan softver

Poteškoće kod Linuksa su to što moramo da naučimo veliki broj komandi za rad u GNU/Linuxu i prevedemo veliki broj novih reči. Takođe, instaliranje novih programa nije jednostavno kao u Windowsu.

Najveća prepreka koju Linux treba da savlada jeste nedostatak popularnog softvera, koji će raditi pod njim. Nedostatak Linuxovih verzija se ogleda u nedostatku popularnih poslovnih i ličnih aplikacija kao što su aplikacije za grafičku obradu, grafički dizajn, tehničko crtanje imultimedijalnu obradu.

Linux se i dalje razvija i prilagođava trendovima i potrebama korisnika, a svako ko želi da na bilo koji način učestvuje – dobrodošao je. Alternativa komercijalnim rešenjima koju je svet slobodnog softvera iznedrio višestruko je značajna, ne samo zbog faktora cene već i zbog slobode koju korisnici ovih sistema uživaju, te ne bi bilo pretenciozno reći da je Linux tu odigrao glavnu ulogu.

LITERATURA

- [1] B. Đorđević, D. Pleskonjić, N. Maček: “*Operativni sistemi: UNIX i Linux*”
- [2] Machtelt Garrels, *Bash Guide for Beginners*, version 1.8, 2006
- [3] Mendel Cooper, *Advanced Bash Scripting Guide*, version 4.0, 2006
- [4] openSUSE 10.2 Reference Guide, 2007
- [5] openSUSE 10.2 Start Guide, 2007