

Gimnazija „Jovan Jovanović Zmaj“
Novi Sad

Maturski rad iz Operativnih Sistema

Operativni sistem Linux

Profesor mentor:
Saša Tošić

Učenik:
Aleksa Siriški IV-6

Novi Sad, maj 2022. god.

1 Predgovor

Za ovu temu sam se opredelio iz više razloga. Prvenstveno zbog ljubavi prema informacionim tehnologijama, koju sam stekao zahvaljujući mojim roditeljima. Drugi razlog je to što smatram da je ovo veoma fascinantna tema, jer obuhvata kompleksnost koje se može postići kada na jednom projektu radi čitav svet. Na kraju, ono što me je privuklo da izaberem baš ovu temu, jeste činjenica da je budućnost IT-a slobodan i besplatan kod.

U ovom radu analiziraću osnovne komponente jednog izuzetnog operativnog sistema, istoriju njegove kreacije kao i filozofski pogled na isti.

Sadržaj

1	Predgovor	2
2	Istorija[1]	4
3	Distribucije	5
4	Komponente	6
4.1	Kernel	6
4.1.1	Kategorije kernela[5]	6
4.1.2	Osnovni delovi Linux kernela[6]	7
4.2	Bootloader	8
4.2.1	Bootloader-i sa podrškom za Linux	8
4.3	Daemons	8
4.4	initramfs	8
4.4.1	Init sistemi sa podrškom za Linux	8
4.5	Shell	9
4.5.1	Shell-ovi sa podrškom za Linux	9
4.6	Display server	9
4.6.1	Display serveri sa podrškom za Linux	9
5	Komande	10
5.1	Osnovne komande u Linux terminalu	10
6	Hijerarhija sistema datoteka	11
6.1	Glavni direktorijumi Linux sistema datoteka	11
7	Dozvole i vlasništvo	12
8	Okruženja radne površine	12
8.1	Desktop okruženja sa podrškom za Linux	12
9	Kontejnerizacija	13
9.1	Vrste kontejnerskih okruženja	13
10	Zaključak	14
	Literatura	15
11	BIOGRAFIJA MATURANTA	16

2 Istorija[1]

MIT u saradnji sa drugim kompanijama su napravili eksperimentalni operativni sistem zvan Multiplexed Information and Computing Service (Multics) 1960. godine. Unics (danas UNIX) su kasnije stvorili i objavili Ken Thompson i Dennis Ritchie (Slika a) iz AT&T 1970. godine kao sopstvenu verziju Multics-a. Kasnije je prekućan u C programskom jeziku i time postao veoma fleksibilan i izmenljiv. Razni fakulteti i univerziteti su pravili svoje verzije Unix-a, npr. BSD koji je poput Linuxa i dan danas u upotrebi.

Richard Matthew Stallman (Slika b), osnivač GNU projekta[2], je uz svoj tim započeo izradu kompletnog operativnog sistema čija je glavna namena bila da bude otvorena i slobodna alternativa za Unix. Jedina stvar koja je falila jeste kernel, sto je jezgro operativnog sistema koje služi da poveže sve druge komponente u zajednicu.

Linus Torvalds (Slika c) koji je bio isprovociran nedostatkom kernela za potpuno slobodan i otvoren operativni sistem odlučuje da napiše svoj sopstveni. Već je bio upućen u Minix i GNU softver te je tokom studija u Finskoj započeo projekat nazvan "Linux" (Slika d). U početku je jedini radio na njemu, ali do danas se priključilo preko 15000 programera u kreiranju kernela koji se sastoji iz više od 17 miliona linija koda[3].



(a) Ken Thompson i Dennis Ritchie, tvorci Unix-a



(b) Richard Matthew Stallman, tvorac GNU-a



(c) Linus Torvalds, tvorac Linux-a



(d) Linux logo

3 Distribucije

GNU/Linux operativni sistem se može samostalno napraviti kompajliranjem svih potrebnih programa (npr. Linux From Scratch), ali se najčešće koriste već gotove distribucije. Sve distribucije dele jednu stvar, Linux kernel, ali po svemu ostalom se mogu potpuno razlikovati, doduše većina distribucija ima neke zajedničke osnovne komponente.

Postoje distribucije pravljenе za servere:

- Debian (Slika a)
- Ubuntu server
- CentOS
- Turnkey

Kao i distribucije pravljenе za desktop korisnike:

- Ubuntu (Slika b)
- Linux Mint
- PopOS
- Fedora Workstation (Slika c)

Takođe postoje minimalne distribucije pravljenе za kontejnere:

- Alpine Linux (Slika d)
- Fedora CoreOS
- openSUSE MicroOS



(a) Debian



(b) Ubuntu



(c) Fedora



(d) Alpine

4 Komponente

4.1 Kernel

Kernel[4] je najvažniji deo svakog operativnog sistema. Služi da pokrene svaku komponentu koja je potrebna za korišćenje OS-a kao i da dozvoli komunikaciju softvera sa hardverom, ili jednog softvera sa drugim. Kernel je neophodan da bi operativni sistem funkcionisao i kao takav je ključno da je minimalan i efikasan.

4.1.1 Kategorije kernela[5]

- Monolitski - ovde se nalaze svi servisi operativnog sistema se nalaze i izvršavaju se u prostoru samog kernela (eng. *kernel space*). Ima zavisnosti (eng. *dependencies*) između sistemskih komponenti i ogromne količine linija koda što ga čini veoma kompleksnim.
 - Unix
 - Linux
 - Open VMS
 - XTS-400
- Mikrokernel - minimalistički, ima virtualnu memoriju i raspoređivanje niti (eng. *thread scheduling*). Samim tim što je manji je i stabilniji, ali zato ima mnogo više sistemskih poziva jer stavlja sve druge procese u korisnički prostor (eng. *user space*).
 - GNU Hurd
 - Mach (MacOS)
 - L4
 - AmigaOS
 - Minix
 - K42
- Hibridni - mešavina monolitiskog i mikro kernela. Ima brzinu i dizajn monolitiskog ali modularnost i stabilnost mikrokernela.
 - Windows NT
 - Netware
 - BeOS
- Egzokernel - prati end-to-end princip, ima najmanju apstrakciju hardvera i direktno dodeljuje fizičke resurse aplikacijama.
- Nanokernel - samo apstrakcija hardvera, bez sistemskih servisa. Veoma sličan mikrokernelu.

4.1.2 Osnovni delovi Linux kernela[6]

- Menadžer procesa - kao što mu ime implicira, služi da rukuje procesima tj. programima. Svaki proces je zapravo svoj virtualni procesor, nazvan nit (eng. *thread*). Programi mogu alocirati više niti, takozvanih "radnika" (eng. *workers*). Termin "proces" i "nit" su istog značenja, mada je "proces" više popularan u Desktop aplikacijama.
- Menadžer memorije - memorija u Linuxu je programima predstavljena kao gomila stranica koje su potom multiplicirane i dodeljivane po zahtevu tog programa. Standard su stranice od 4KB. Takođe, stranice je moguće prebaciti sa sistemske RAM memorije na hard disk, takav postupak se naziva *swapping*.
- Drajveri - kod koji čini najveći deo linux kernela, omogućava korišćenje određenog hardvera.
- Virtualni sistem datoteka (VFS) - apstrahuje osnovne komande za manipulaciju podataka čime omogućava korišćenje raznovrsnih sistema datoteka.

4.2 Bootloader

Bootloader je program koji se pokreće pre samog operativnog sistema. Omogućava izbor između više operativnih sistema (dual boot), kao i izbor između različitih kernela na istom operativnom sistemu. Najpre se pokreće kernel, zatim mikrokod procesora (npr. intel-ucode, amd-ucode) i na kraju inicijalni memorijski disk (**initial ram filesystem**). Takođe, u bootloader-u se definišu dodatne i zamenjive opcije za kernel.

4.2.1 Bootloader-i sa podrškom za Linux

- **GR**and **U**nified **B**ootloader - najpopularniji zbog velike podrške za svakakve sisteme, jedini podržava enkriptovanu boot partciju. Nastao kao deo programa za GNU operativni sistem.
- rEFInd - nastoji da ima što više opcija kao GRUB ali sa manjim i jednostavnijim konfiguracijama.
- systemd-boot - deo systemd, veoma jednostavan i minimalan.
- EFISTUB - metoda umetanja Linux kernela kao samostalnog bootloadera u novije UEFI BIOS sisteme.

4.3 Daemons

Demoni[7], naziv nastao od Maxwell-ovog demona koji u pozadini sortira molekule, služe da rade baš to: u pozadini postoje bez korisničkog interfejsa i čekaju na neki događaj (eng. event) i/ili da nude usluge (eng. services).

4.4 initramfs

Init je prvi program koji se pokreće nakon kernela, automatski postaje roditelj svakog narednog programa i poslednji se zaustavlja pri gašenju računara. Kao takav, služi da omogući pristup sistemima datoteka (eng. mount filesystems), da reguliše demone i snabdeva informacije o njima.

4.4.1 Init sistemi sa podrškom za Linux

- systemd - najpopularniji ali i najobimniji, ne prati KISS (*keep it simple, stupid*) jer integriše mnogobrojne dodatne funkcionalnosti kojim init sistem ne bi trebao da se bavi.
- OpenRC - prvi izbor za one koje ne vole systemd, minimalan i brz.
- GNU Shepherd - Nastao kao deo programa za GNU operativni sistem.
- BusyBox - nastao za korišćenje na ugrađenim slabim uređajima (eng. embedded devices).
- runit - UNIX init sistem, zamena za SysVinit.
- SysVinit - tradicionalni init sistem.

4.5 Shell

Shell (eng. *ljuska*) služi da korisniku pruži isključivo tekstualni interfejs, naziv "ljuska" upućuje da je to spoljni deo operativnog sistema. Korisnik unosi komande i dobija odgovore. Osim pokretanja programa, shell može da koristi izlaz jedne komande kao ulaz druge, što se naziva *piping* (prvo uveden u UNIX-u). Takođe može da se koristi i kao programski jezik pri pisanju shell skripti.

4.5.1 Shell-ovi sa podrškom za Linux

- `bash` - najpopularniji i standard za većinu distribucija, brz i jednostavan. Nastao kao zamena za `sh`.
- `zsh` - obimniji i sporiji, najpre se koristi radi automatskog završavanja pri unosu komandi i sugestije istih.
- `fish` - sličan `zsh`-u, jednostavniji za konfiguraciju ali nije POSIX (neke osnovne komande se razlikuju od većine drugih shell-ova)
- `sh` - UNIX shell
- `dash` - minimalan i efikasan, nastao isključivo da pokreće POSIX skripte. Četiri puta brži od `bash`-a, ali nije pravljen za direktno korišćenje u terminalu.

4.6 Display server

Displej server služi da omogući prikaz GUI-a korisnicima koristeći WIMP (prozori, ikonice, meniji, pokazivač kursora) paradigmu za korisnički interfejs. Svaka aplikacija dobija svoj prozor izmenljive veličine, najčešće pravougaonog oblika.

4.6.1 Display serveri sa podrškom za Linux

- X, X11 ili XOrg - prvi i najkorišćeniji displej server za UNIX-oidne sisteme. Nastao pre više od 30 godina i još uvek u upotrebi zbog spore adopcije drugih protokola. Zahteva dodatan kompozitor (eng. *compositor*) za bilo kakve efekte (npr. providnost prozora) i VSync.
- Wayland - nastao u cilju da bude moderna alternativa za X pre više od 10 godina, ali je tek u proteklih par godina počeo eksponencijalno da se razvija i danas je standard na samo par distribucija (Fedora, Ubuntu, openSUSE, Debian). Drastično povećava bezbednost time što dodaje kontrolu dozvola aplikacijama, za razliku od X gde svaka aplikacija može da vidi jedna drugu. Wayland nema dodatne kompozitore, već je svaka primena Wayland protokola sama svoj kompozitor.

5 Komande

Jedna od velikih prednosti Linux operativnog sistema jeste terminal, kojim je omogućena upotreba kompletnog korisničkog interfejsa. Tek je naknadno ubačena podrška za grafički korisnički interfejs.

5.1 Osnovne komande u Linux terminalu

- `pwd` - ispisuje trenutni direktorijum u kom se nalazi i u kom će se izvršavati sve komande. Inicijalni terminal je kućni folder korisnika koji ga pokreće.
- `ls` - ispisuje sve foldere i fajlove u zadatom direktorijumu. U slučaju skrivenih foldera i fajlova koristi se **`ls -a`**.
- `cd` - menja trenutnu lokaciju tj. folder.
- `mv` - pomera fajl.
- `cp` - kopira fajl.
- `mkdir` i `rmdir` - pravljenje i brisanje foldera.
- `touch` i `rm` - pravljenje i brisanje fajlova.
- `cat` - ispis svega što se nalazi unutar fajla.
- `man` - ispisuje uputstva jedne komande.
- `sudo` - **S**uper **U**ser **DO** - pokreće komandu kao root korisnik.

6 Hijerarhija sistema datoteka

Na Linux operativnom sistemu sve je predstavljeno kao fajl - jezgra procesora, sistemski diskovi, čak i tastatura i miš. Hijerarhija sistema datoteka (FSH) definiše strukturu foldera. Sve se nalazi unutar glavnog root foldera i pristupa mu se kosom crtom.

6.1 Glavni direktorijumi Linux sistema datoteka

- bin - neophodni programi za pokretanje sistema - mount, ls, cp.
- boot - neophodni fajlovi za pokretanje sistema - kernel, initramfs, mikrokod procesora.
- dev - uređaji predstavljeni kao fajlovi - hard diskovi, procesor, grafička kartica.
- etc - konfiguracioni fajlovi za sistemske programe.
- home - folderi i fajlovi svih korisnika na sistemu sa izuzetkom glavnog korisnika na /root.
- lib - biblioteke potrebne za pokretanje programa iz /bin.
- media - mesto za priključne medije, npr. usb flash, CD.
- mnt - slično /media, namenjeno za pristup privremenim uređajima.
- proc - informacije o sistemskim procesima i kernelu u obliku fajlova
- tmp - privremeni fajlovi koji se brišu pri gašenju računara
- usr - primarni direktorijum za izvršavajuće programe na sistemu
- var - promenljivi fajlovi koji zapisuju promene i podatke tokom izvršavanja programa

7 Dozvole i vlasništvo

Linux operativni sistem za svaki objekat, tj. fajl ili folder, ima određene dozvole. Moguće ih je videti komandom **ls -l**. Postoje tri vrste dozvola: čitanje, pisanje i pokretanje. Takođe postoje tri vrste vlasništva: vlasnik, grupa i svi ostali. Na ta tri vlasništva se određuje koje dozvole imaju, koristeći sledeće brojeve:

- 0 - nema nikakvih dozvola
- 1 - dozvola za pokretanje
- 2 - dozvola za pisanje
- 3 - dozvola za pokretanje i pisanje
- 4 - dozvola za čitanje
- 5 - dozvola za čitanje i pokretanje
- 6 - dozvola za čitanje i pisanje
- 7 - ima sve dozvole

Dozvole nekog fajla ili foldera se menjaju komandom **chmod** a vlasništvo komandom **chown vlasnik:grupa**.

8 Okruženja radne površine

Okruženja radne površine su skupovi programa koji omogućavaju korišćenje grafičkog interfejsa. Čine ih razne komponente kao što su menadžer prijava, zaključavajući ekran, tekstualni editor i najvažnije menadžer prozora. Moguće imati samo menadžer prozora i zasebno skidati sve druge potrebne programe za korišćenje grafičkog interfejsa ali korišćenje kompletnog okruženja drastično pojednostavljuje taj proces.

8.1 Desktop okruženja sa podrškom za Linux

- GNOME[8] - GNU okruženje sa podrškom za Wayland, najpopularniji izbor za distribucije. Nastao kao deo programa za GNU operativni sistem.
- KDE Plasma[9] - drugo najpopularnije okruženje, takođe sa podrškom za Wayland. Nastao kao više izmenljivo okruženje koje je meri sa GNOME koji ima specifični filozofski pogled na način upotrebe računara.
- Xfce[10] - modularno i nezahtevno okruženje.
- LXQt[11] - okruženje koje koristi najmanje resursa.
- MATE[12] - započet kao klon GNOME 2 kada je GNOME prešao na verziju 3, ali sada i sam koristi tu verziju.
- Cinnamon[13] - standardno okruženje za Linux Mint, nastoji da olakša prelazak sa Windows okruženja.

9 Kontejnerizacija

Kontejnerizacija je jedna od jačih strana Linux operativnog sistema. Mogućnost izolacije programa od ostatka sistema sa minimalnim gubitkom performansi je razlog zašto je Linux izbor za najveći procenat servera. Za razliku od virtualnih mašina, kontejneri dele kernel sa domaćim operativnim sistemom (eng. host OS) i time štede na veličini.

9.1 Vrste kontejnerskih okruženja

- Linux kontejneri[14] - najprostiji vid korišćenja kontejnera na Linuxu, pomoću cgroups iz kernela omogućeno je kreiranje više kontejnera na jednom domaćinu.
- Docker[15] - komercijalizovan i besplatan softver za manipulaciju kontejnera. Najpopularniji na tržištu.
- Podman[16] - RedHat-ova alternativa za docker, nastoji da bude sintaksno identična ali ne zahveta root dozvole za razliku od Docker-a.
- Flatpak[17] - RedHat-ova jednostavna distribucija desktop aplikacija korišćenjem kontejnera. Developeri mogu predvidljivo da prave kontejner pored sopstvene aplikacije i tako osiguravaju da će se ona identično ponašati ne zavisno od domaćinskog operativnog sistema. Usput pruža ogromna poboljšanja za bezbednost u vidu dozvola (slično Android-u).
- Snap[18] - Canonical-ova distribucija aplikacija, slična flatpak-u ali trenutno mnogo sporija. Serverska strana im je centralizovana i zatvorenog koda.
- AppImage[19] - nastoji da olakša distribuciju aplikacije kao flatpak i snap, ali nema jedno mesto na kom se traže aplikacije, već developeri sami prave i distribuiraju nalik običnim pokretnim datotekama (.exe, .deb, .rpm). Svaka aplikacija dolazi uz svoj skup biblioteka i programa potpuno izolovanih, što bespotrebno troši memoriju, ali je lakše za održavanje.

10 Zaključak

Linux operativni sistem je prvi izbor za najveći broj servera, ali je takođe i korišćen u raznim naučnim i obrazovnim institucijama. Uprkos tome što se bazira na besplatnom modelu postojanjem kompanija kao što su RedHat i mnoštvom sponzora (Microsoft, Google, Facebook/Meta) Linux nastavlja dalje da se ekspanzionalno razvija. Ljudi koji se bave programiranjem Linux-a imaju platu kao kad bi radili na identičnom proizvodu zatvorenog koda, što dokazuje da je etika slobodnog koda uspešna i da ne postoji razlog za zatvorenim kodom osim prikupljanja personalnih podataka bez javnog priznanja o tom postupku. Važno je napomenuti da ako je nešto "otvoren" kod ne znači i da je "slobodan" kod. Otvoren kod dozvoljava svakome da vidi tačno šta se izvršava u programu ali mu ne daje na pravo da radi sa tim kodom šta god poželi, dok slobodan kod omogućuje svakome da ponovo koristi taj kod za nešto potpuno drugo ako će takođe dozvoliti ponovnu upotrebu tog novonastalog koda što drastično smanjuje potrebnu za pisanjem istog koda iznova. Da bi se sprečila krađa koda i zloupotreba Stallman je osmislio licence za otvoren i slobodan kod (npr. GNU General Public License[20]).

Najčešće neuočljivo ali Linux je prisutan u velikom broju svakodnevnih uređaja. Najpopularniji je Android operativni sistem za pametne telefone koji koristi Linux kernel kao i pametni televizori (npr. Samsung-ova odluka da nove modele televizora preinstalira sa Tyzen-om, Linux distribucijom, umesto AndroidTV-a). U današnje vreme i prosečni korisnik koji nije i ne želi da bude upućen u kompleksne tokove rada računara može vrlo jednostavno da koristi Linux upotrebom gotovih distribucija i usput pokrene većinu Windows programa zahvaljujući sloju kompatibilnosti WINE i VALVE-ovim PROTON. Brzinom razvoja i rasprostranjenosti podrške hardvera i softvera vrlo je verovatno da će u bližoj budućnosti Linux zameniti i trenutno glavne Desktop operativne sisteme.

Literatura

- [1] J.T.S. Moore. Revolution os, 2001. <http://revolution-os.com>.
- [2] Richard Stallman. Prva najava gnu projekta, 1983. <https://www.gnu.org/gnu/initial-announcement>.
- [3] The Linux Foundation. Godišnji izveštaj o izradi linux kernela, 2017. <https://www.linuxfoundation.org/press-release/linux-foundation-releases-annual-kernel-development-report>.
- [4] The Linux Information Project. Definicija kernela, 2004. <http://www.linfo.org/kernel>.
- [5] Geeks for Geeks. Kategorije kernela, 2022. <https://www.geeksforgeeks.org/kernel-in-operating-system>.
- [6] Anatomija linux kernela. <https://developer.ibm.com/articles/l-linux-kernel>.
- [7] The Linux Information Project. Definicija demona, 2005. <http://www.linfo.org/daemon>.
- [8] Gnome. <https://www.gnome.org>.
- [9] Kde plasma. <https://kde.org/plasma-desktop>.
- [10] Xfce. <https://xfce.org>.
- [11] Lxqt. <https://lxqt-project.org>.
- [12] Mate. <https://mate-desktop.org>.
- [13] Cinnamon. https://linuxmint.com/rel_una_cinnamon.php.
- [14] Linux kontejneri. https://wiki.archlinux.org/title/Linux_Containers.
- [15] Docker. <https://www.docker.com>.
- [16] Podman. <https://podman.io>.
- [17] Flatpak. <https://www.flatpak.org>.
- [18] Snap. <https://snapcraft.io>.
- [19] Appimage. <https://appimage.org>.
- [20] Gpl. <https://www.gnu.org/licenses/gpl-3.0.en.html>.

11 BIOGRAFIJA MATURANTA

Aleksa Siriški rođen je 10. jula 2003. godine u Novom Sadu. Pohađao je Osnovnu školu „Svetozar Marković Toza“ do šestog razreda. Tamo stiče interesovanje za matematiku, informatiku, fiziku i jezike. Septembra 2016. upisuje Osnovnu školu pri Gimnaziji „Jovan Jovanović Zmaj“, kako bi intenzivnije radio u oblastima matematike, fizike i informatike. Istovremeno pohađa programerski kurs „Centar za mlade talente“. Školoвање nastavlja u istoj gimnaziji i opredeljuje se za smer „Učenici sa posebnim sposobnostima za računarstvo i informatiku“. Naredne četiri godine, uporedo sa školom, pohađa i kurseve engleskog i nemačkog jezika. Planira da više obrazovanje stekne na Prirodno matematičkom fakultetu, smer Informacione tehnologije.

