Osnove administracije operacijskog sustava 2

Debian

L102

OBRAZOVNI PROGRAMI srca

priručnik za polaznike

srce
Sveučilište u Zagrebu
Sveučilišni računski centar

Ovu su inačicu priručnika izradili:

Autor: mr. sc. Branimir Radić

Recenzent: Darko Culej

Urednik: Dominik Kenđel

Lektor: dr. sc. Jasna Novak Milić

Sveučilište u Zagrebu

Sveučilišni računski centar

Josipa Marohnića 5, 10000 Zagreb

edu@srce.hr

ISBN 978-953-8172-97-7 (meki uvez)

ISBN 978-953-8172-98-4 (PDF)

Verzija priručnika L102-20231114

Ovo djelo dano je na korištenje pod licencom Creative Commons Imenovanje-Dijeli pod istim uvjetima 4.0 međunarodna (CC BY-SA 4.0). Licenca je dostupna na stranici: https://creativecommons.org/licenses/by-sa/4.0/deed.hr.

Sadržaj

Uvod

Trajanje poglavlja:
10 min

Ovaj je tečaj prirodni nastavak uvodnog tečaja L101 koji polaznike uvodi u Linux administraciju. Tečaj služi kao proširenje znanja stečenih u tečaju L101 i zajedno sa njim predstavlja osnovu za početak rada na bilo kojem Linux operacijskom sustavu sa naglaskom na Debian, konkretno Debian 11 za koji su izrađene vježbe.

Tečaj obrađuje širok spektar tema uključujući osnove administrativnog podešavanja i upravljanja operacijskim sustavom linux: Razumijevanje uloge jezgre OS-a, razumijevanje i podešavanje pokretanja linux OS-a, upravljanje korisnicima i grupama u linuxu, rad sa mrežnim postavkama u linuxu i upravljanje mrežnim servisima, razumijevanje i izrada BASH skripti i osnove sigurnosti, ispisa i X grafičke okoline.

Ove teme nude širok uvod u administraciju Linuxa početnicima i podsjetnik za polaznike sa iskustvom u polju.

Nakon pohađanja tečaja polaznici će znati osnovne principe rada u Linux administraciji, osnovne problematike CLI-a na Linuxu kao i osnove mrežne povezanosti komunikacije i nadzora aktivnosti na poslužiteljima.

1. Jezgra operacijskog sustava Linux

Trajanje poglavlja:
50 min

Po završetku ovoga poglavlja moći ćete:

- imenovati što je to jezgra operacijskih sustava zasnovanih na *Unix*u.
- klasificirati jezgru po prostoru odvijanja procesa te po mogućnosti proširenja i smanjenja veličine i funkcionalnosti jezgre tijekom rada
- ispisati aktivne module, pronaći na datotečnom sustavu dostupne module i učitati ih ili ukloniti iz jezgre
- pribaviti izvorni kôd jezgre i prilagoditi ga potrebama (konfigurirati)
- programski prevesti i instalirati jezgru
- podesiti programe za učitavanje operacijskog sustava.

Ova cjelina obrađuje raspoložive vrste instalacije i provedbu instalacije operacijskog sustava Debian. Upoznajemo se sa strukturom Linuxova datotečnog sustava i particijama.

1.1. Jezgra – osnovni koncepti

1.1.1. Uvod

Jezgra operacijskog sustava *Linux* (*Linux kernel*) središnja je komponenta operacijskih sustava zasnovanih na *Unix*-u. *Linux*-ovu jezgru koriste operacijski sustavi *Linux*, neki specijalizirani sustavi poput mrežnih usmjernika i uređaji poput pametnih telefona i tableta koji rabe operacijski sustav *Android*. Postoji pet vrsta jezgri: monolitna jezgra, mikrojezgra, hibridna jezgra, nanojezgra i eksojezgra. Najviše se rabe monolitna i mikrojezgra.

Monolitna jezgra je jezgra gdje su svi servisi (servisi za upravljanje datotečnim sustavom, virtualni sustav datoteka, upravljački programi uređaja itd.) kao i središnje funkcionalnosti (raspoređivanje poslova, upravljanje memorijom, odgovori na zahtjeve i slično) čvrsto povezane u istom prostoru.

U **mikrojezgri** (*microkernel*) prednost se daje pristupu gdje su središnje funkcionalnosti izolirane od servisa i upravljačkih programa uređaja. Na primjer, virtualni datotečni sustav i *minifx* (proces koji upravlja blok uređajima) izdvojeni su iz jezgre i s njom komuniciraju preko **IPC-a** (*inter-process communication*).

Jezgra operacijskog sustava *Linux* je monolitna. Razliku između monolitnih i mikrojezgri opisuje sljedeća rečenica: "To što je u *Linux*u modul, u mikrojezgri je servis (gdje je servis izolirani proces koji komunicira preko **IPC-a**)."

Važno je razumjeti da mogućnost dodavanja i uklanjanja modula ne čini *Linux*-ovu jezgru ništa manje monolitnom, jer je važno samo to što svi moduli djeluju u istom prostoru koji upravlja i sa središnjom funkcionalnosti jezgre.

Prednost je monolitne jezgre u tome što može biti oblikovana za vrlo brzi rad i visoke performanse.

Nedostatak monolitne jezgre je uska povezanost svih komponenti koja rezultira time da pad ili pogrešna konfiguracija pojedinih komponenti jezgre često uzrokuje pad cijelog sustava.

Prednost mikrojezgre je u izolaciji servisa, koji se mogu jednostavno pokrenuti u slučaju pada servisa i koji svojim padom ne uzrokuju pad jezgre i pad sustava. Nedostatak mikrojezgre je asinkrona priroda **IPC** komunikacije koja izuzetno otežava pronalaženje pogrešaka. Pronalaženje pogreške često uključuje pretraživanje zapisa aktivnosti niza servisa i konačno, ako se tako ne nađe pogreška, kontrolnu inspekciju rada **IPC** servisa, što može biti posebno teško kad se pojave i složeni redovi čekanja u komunikaciji.

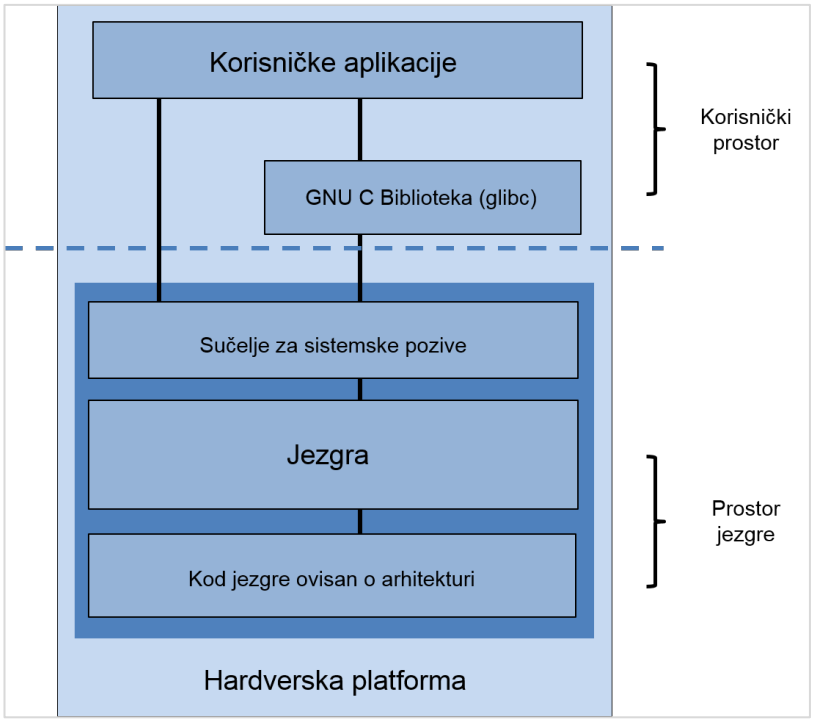
1.1.2. Komponente jezgre

Jezgra operacijskog sustava *Linux* podijeljena je u tri dijela:

- Na vrhu se nalazi **sučelje za sistemske pozive**. To sučelje provodi jednostavne operacije poput čitanja (*read*) i pisanja (*write*).
- Ispod sučelja za sistemske pozive nalazi se **kôd jezgre**, preciznije dio kôda jezgre koji nije ovisan o platformi na kojoj se jezgra nalazi. Taj dio jezgre zajednički je svim procesorskim arhitekturama koje podržava *Linux*.
- Na dnu se nalazi dio jezgre koji je ovisan o arhitekturi i poznat je kao **paket za podršku matičnoj ploči** (*BSP Board Support Package*). Taj dio jezgre odrađuje komunikaciju sa komponentama hardvera preko poruka prilagođenih upravo tom hardveru.

Sistemska memorija u *Linux*u podijeljena je na dvije različite regije:

- korisnički prostor
- prostor jezgre.

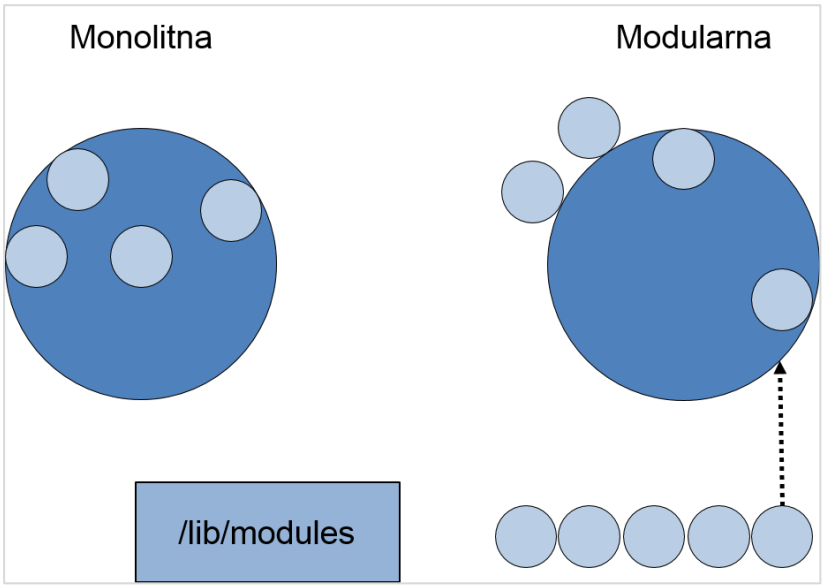


Prostor jezgre je skup memorijskih lokacija na kojima se izvode komponente jezgre i iz kojeg jezgra pruža funkcionalnosti drugim procesima na računalu.

Korisnički prostor je skup memorijskih lokacija na kojima se izvode korisnički procesi odnosno svi procesi koji nisu dio jezgre.

1.1.3. Dinamičko proširenje jezgre

Osim ranije opisane definicije koja jezgre dijeli prema tome u kojem se prostoru odvijaju koji procesi, postoji podjela i prema kriteriju mogućnosti proširenja i smanjenja veličine i funkcionalnosti jezgre tijekom rada. Važno je istaknuti da se i u toj podjeli rabi izraz "monolitan", ali s drugim značenjem.



Kod monolitne jezgre podrška za sav hardver, mrežne funkcije i operacije nad datotečnim sustavom nalazi se u jednoj središnjoj datoteci jezgre (*image file,* u daljnjem tekstu slika jezgre). Prednost je monolitnih jezgri u brzini podizanja sustava.

Kod modularne je jezgre dio podrške preveden i čini sliku jezgre, a dio se podrške dinamički učitava na zahtjev. Prednost je modularne jezgre da dodavanje hardverskih komponenti ne zahtijeva ponovno prevođenje i izradu nove slike jezgre, nego je dovoljno učitati odgovarajući modul.

Prema toj podjeli jezgra operacijskog sustava *Linux* je **modularna jezgra.**

1.1.4. Upravljanje modulima

Moduli koji se mogu učitati nalaze se u poddirektorijima direktorija **/lib/modules/<verzija-jezgre>/**, a učitani moduli vidljivi su u direktoriju **/proc/modules**. Budući da naredba `# uname -r` vraća inačicu trenutne jezgre, u literaturi se često rabi poziv te naredbe za pristup direktoriju na primjer u obliku: `#ls /lib/modules/`uname -r`/`.

Primjer prikaza trenutačno aktivne jezgre pomoću naredbe **uname -r**:

```
root@L102:~# uname -r
5.10.0-16-amd64
```

Naredbe za upravljanje modulima:

Naredba	Opis
lsmod	Naredba za prikaz modula učitanih u jezgru.
modinfo	Naredba za prikaz podatka o danom modulu.
insmod	Dodaje pojedinačni modul (bez provjere ovisnosti).
rmmod	Naredba uklanja pojedinačni modul (bez provjere ovisnosti).
modprobe	Naredba za dodavanje/uklanjanje modula (uz provjeru ovisnosti).

1.1.5. Naredbe lsmod i modinfo

Naredba za prikaz modula učitanih u jezgru je **lsmod**. Primjer izvođenja naredbe **lsmod**:

```
root@L102:~# lsmod

Module                  Size  Used by
vboxvideo              49152  0
intel_rapl_msr         20480  0
intel_rapl_common      28672  1 intel_rapl_msr
intel_pmc_core_pltdrv    16384  0
intel_pmc_core         45056  0
intel_powerclamp       20480  0
ghash_clmulni_intel    16384  0
joydev                 28672  0
aesni_intel           368640  0
hid_generic            16384  0
```

Prvi stupac prikazuje ime modula, drugi stupac veličinu modula (veličina datoteke modula), a treći stupac prikazuje koliko modula ga koristi. Nakon broja modula koji koriste modul slijedi popis modula koji ga koriste. Taj popis je ponekad nepotpun.

Vrijednost -1 u trećem stupcu opisuje da modul sam upravlja svojim uklanjanjem pomoću rutine **can_unload** i ako je to slučaj vrijednost će uvijek biti -1.

Naredba **modinfo** daje podatke o modulu. Primjer je izvođenja bez opcija nad modulom za upravljanje uređajima CDROM:

```
root@L102:~# modinfo cdrom

filename:       /lib/modules/5.10.0-16-amd64/kernel/drivers/cdrom/cdrom.ko
license:        GPL
depends:
retpoline:      Y
intree:         Y
name:           cdrom
vermagic:       5.10.0-16-amd64 SMP mod_unload modversions
sig_id:         PKCS#7
signer:         Debian Secure Boot CA
sig_key:        4B:6E:F5:AB:CA:66:98:25:17:8E:05:2C:84:66:7C:CB:C0:53:1F:8C
sig_hashalgo:   sha256
signature:      25:C1:6D …

parm:           debug:bool
parm:           autoclose:bool
parm:           autoeject:bool
parm:           lockdoor:bool
parm:           check_media_type:bool
parm:           mrw_format_restart:bool
```

Naredba **modinfo** prihvaća opcije za ispis samo nekih podataka o modulima. Opcije su:

Opcija	Opis
-a	podaci o autoru
-d	opis modula
-l	podaci o licenci
-p	podaci o parametrima modula
-n	podaci o datoteci koja sadrži modul

1.1.6. Naredbe insmod, rmmod i modprobe

Naredbe **insmod** i **rmmod** dodaju odnosno uklanjaju pojedini modul iz jezgre i pri tome ne provode provjere o ovisnostima i ne javljaju pogreške. Naredba **rmmod** provjerava ima li aktivnih modula koji se koriste modulom koji se pokušava ukloniti iz jezgre, ali ne provjerava ovisnosti iz datoteke **modules.dep.bin**. Ako se pri izvođenju tih naredbi dogodi pogreška, ona će ipak biti zapisana u datoteku sistemskih zapisa jezgre i može se vidjeti naredbom **dmesg**.

Naredba **modprobe** bez opcija dodaje, a s opcijom **-r** uklanja modul iz jezgre. Razlika i razlog zašto je naredba **modprobe** bolji način za upravljanje modulima je u tome što čita zapise o ovisnostima modula u datoteci **modules.dep.bin** (ili **modules.dep** ako binarni oblik datoteke nije dostupan). Pri učitavanju modula naredbom **modprobe** izvrši se provjera jesu li ispunjene sve ovisnosti modula, a ako nisu, tada se učitaju svi moduli o kojima ciljani modul ovisi. Pri uklanjanju modula naredba javlja poruku o pogrešci ako postoje moduli koji se koriste modulom koji se pokušava ukloniti:

```
# modprobe -r vboxguest
FATAL: Module vboxguest is in use.
```

Osim na korisnički zahtjev, u *Linux*-ovu se jezgru moduli učitavaju i automatski kad se pojavi potreba za njihovim korištenjem. **kmod** (*kernel driver module*) mora biti konfiguriran pri prevođenju jezgre (iduće poglavlje) za automatsko učitavanje modula. Većina standardnih konfiguracija jezgre (kakve se pribavljaju preko paketa ili softvera za upravljanje paketima) imaju uključen **kmod**. Važno je razlikovati **kmod** i **Kmod**. Malim slovom se označava modul jezgre, a velikim slovom paket koji sadržava alate za rad sa modulima (**depmod**, **insmod**, **kmod**, **lsmod**, **modinfo**, **modprobe**, **rmmod** i **libkmod**).

1.2. Prilagodba potrebama i izgradnja jezgre

1.2.1. Pribavljanje izvornog koda

Jezgra je središnji softverski element računalnog operacijskog sustava. Budući da je za rad sustava nužna ispravna i aktivna jezgra, instalacija jezgre u nekim je pogledima drugačija od instalacije *ostalog* softvera. U nastavku ćemo, pri opisu koraka instalacije (nove) jezgre, napomenuti koji su koraci i zašto drugačiji.

Pribavljanje izvornog kôda

Stranica projekta razvoja jezgre je https://www.kernel.org/. Novija se inačica može preuzeti s repozitorija git, a preko repozitorija http (*wget/web* preglednik) ili **rsync** neku od stabilnih inačica.

```
$ wget
https://cdn.kernel.org/pub/linux/kernel/v5.x/linux-5.19.6.tar.xz
$ wget
https://cdn.kernel.org/pub/linux/kernel/v5.x/linux-5.19.6.tar.sign
```

Osim pribavljanja kôda treba i provjeriti **integritet softvera jezgre**. Da bi provjera bila neovisna o kompresiji, datoteka gpg napravljena je za dekompresiranu inačicu jezgre. Tako se jedna datoteka rabi za potpis .gz, .bz2 i .xz komprimiranih inačica.

Najprije treba pribaviti ključ. ID ključa najjednostavnije je pribaviti izvođenjem naredbe za provjeru. Prvo se neuspješno pokuša potvrditi ključ. Zatim se na osnovi ID broja prikupi ključ i zatim se ponovi provjera:

```
$ xz -cd linux-5.9.16.tar.xz | gpg --verify linux-5.9.16.tar.sign -
gpg: Signature made Sun 17 May 2021 06:51:58 PM CEST using RSA key ID 6092693E
gpg: Can't check signature: public key not found

$ gpg --recv-keys 6092693E
gpg: keyring `/root/.gnupg/secring.gpg' created
gpg: requesting key 6092693E from hkp server keys.gnupg.net
gpg: /root/.gnupg/trustdb.gpg: trustdb created
```

```
gpg: key 6092693E: public key "Greg Kroah-Hartman (Linux kernel stable release
signing key) <greg@kroah.com>" imported
gpg: no ultimately trusted keys found
gpg: Total number processed: 1
gpg:               imported: 1  (RSA: 1)

$ xz -cd linux-5.9.16.tar.xz | gpg --verify linux-5.9.16.tar.sign -
gpg: Signature made Sun 17 May 2021 06:51:58 PM CEST using RSA key ID 6092693E
gpg: Good signature from "Greg Kroah-Hartman (Linux kernel stable release
signing key) <greg@kroah.com>"
gpg: WARNING: This key is not certified with a trusted signature!
gpg:          There is no indication that the signature belongs to the owner.
Primary key fingerprint: 647F 2865 4894 E3BD 4571  99BE 38DB BDC8 6092 693E
```

Nakon provjere naredbom tar provodi se ekstrakcija datoteka iz arhive:

```
$ tar xfv linux-5.9.16.tar.xz
```

1.2.2. Konfiguracija

Naredbe za konfiguraciju jezgre pozivaju se u direktoriju jezgre.

```
$ cd linux-5.9.16
```

Za konfiguraciju jezgre rabe se četiri naredbe:

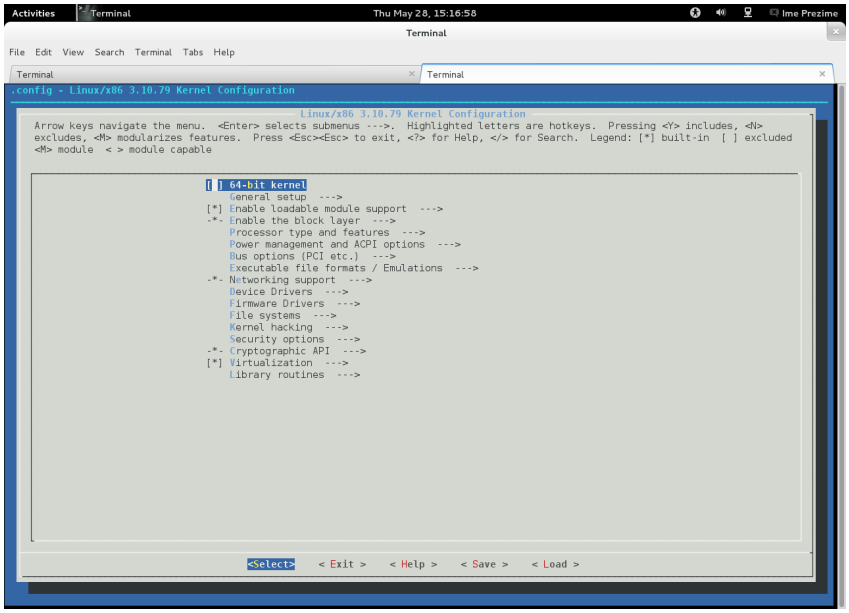
- **make-oldconfig**
- **make-config**
- **make-menuconfig**
- **make-xconfig**.

Naredba **make-oldconfig** čita sadržaj konfiguracije trenutačno aktivne jezgre i postavlja upite za sve opcije koje ne postoje u trenutačnoj konfiguraciji. Cilj je ove naredbe pojednostaviti konfiguraciju pri nadogradnji jezgre na novu inačicu.

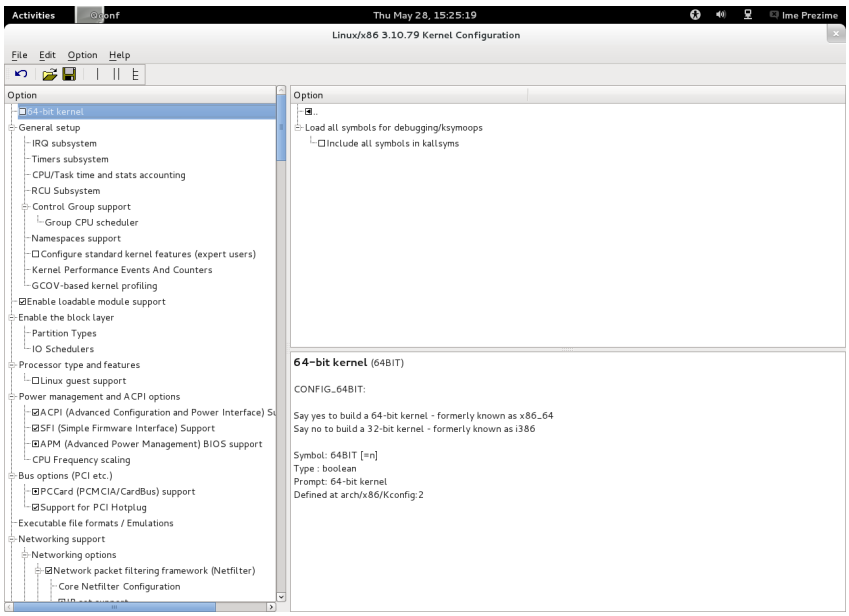
Razlog zašto treba pojednostaviti konfiguraciju je više od 3000 opcija koje se postavljaju pri konfiguraciji jezgre. Pozivom **make-config** pokreće se uređivač teksta u kojem se mogu postaviti sve opcije za konfiguraciju jezgre. Zbog veličine te konfiguracije napravljene su i naredbe **make-menuconfig,** koja konfiguraciju pokreće u **CLI** (*Command Line Interface*) simuliranom GUI-u, i naredba **make-xconfig** koja pokreće XKlijent za konfiguraciju.

Primjeri izgleda naredbi:

```
make menuconfig
```



`make xconfig`



1.2.3. Programsko prevođenje

Programsko prevođenje jezgre (*compiling*) provodi se pozivom naredbe **make** u odgovarajućem direktoriju, dakle:

```
$ make
```

Glavna razlika u odnosu na neki standardni paket je u trajanju. Na virtualnom računalu s dodijeljenom jednom procesorskom jezgrom i 2 GB RAM-a prevođenje je trajalo približno 3,5 sati. Na poslužiteljima, čak i novim sa vrhunskim hardverom prevođenje rijetko traje kraće od 15 minuta.

Nakon prevođenja standardni paket spreman je za instalaciju, ali kod jezgre dodatno treba programski prevesti module naredbom:

```
$ make modules
```

Naredba se izvodi značajno kraće od prevođenja same jezgre, traje manje od 5 minuta na virtualnom računalu. Nakon prevođenja moduli se instaliraju naredbom:

```
$ make modules_install
```

Moduli se smještaju u poddirektorije direktorija **/lib/modules/5.9.16/kernel/**, gdje je **5.9.16** inačica jezgre. Kad se želi napraviti nova prilagođena jezgra, potrebno je pri postavljanju konfiguracije promijeniti inačicu. Inačica jezgre definirana je u datoteci **Makefile** sa sljedećim poljima:

VERSION	=	5
PATCHLEVEL	=	19
SUBLEVEL	=	16
EXTRAVERSION	=	

Postavljene parametre ne bi trebalo mijenjati, jer oni opisuju inačicu jezgre. U parametar **EXTRAVERSION** može se postaviti proizvoljna vrijednost i tad će inačica jezgre biti 5.9.16-**EXTRAVERSION**.

1.2.4. Instalacija

Instalacija jezgre provodi se izvođenjem naredbe `make install`. Izvođenjem te naredbe stvaraju se u direktoriju **/boot** sljedeće datoteke:

Naziv datoteke	Opis
config-5.9.16	Konfiguracijska datoteka jezgre.
initrd.img-5.9.16	Inicijalna RAM datoteka (datoteka inicijalnog (privremenog) root datotečnog sustava potrebnog za proceduru pokretanja sustava).
System.map-5.9.16	Tabela simbola koje koristi jezgra.
vmlinuz-5.9.16	Jezgra.

Instalacijska naredba će također unijeti izmjene u konfiguracijsku datoteku **grub.cfg**. Postavke unesene u **grub.cfg** dodaju u izbornik pri pokretanju sustava dvije dodatne opcije. Za pokretanje sustava sa novom jezgrom i za pokretanje sustava sa novom jezgrom u *recovery mode* (u stvari pokretanje u *single user mod*u). Primjer izmjena (podebljane su razlike između tih dvaju opcija):

```
submenu 'Advanced options for Debian GNU/Linux' $menuentry_id_option 'gnulinux-
advanced-cf0736b3-7e7d-4b65-9a44-64bbb8956973' {
 menuentry 'Debian GNU/Linux, with Linux 5.9.16' --class debian --class gnu-
linux --class gnu --class os $menuentry_id_option 'gnulinux-5.9.16-advanced-
cf0736b3-7e7d-4b65-9a44-64bbb8956973' {
 load_video
 insmod gzio
 if [ x$grub_platform = xxen ]; then insmod xzio; insmod lzopio; fi
```

```
 insmod part_msdos
 insmod ext2
 set root='hd0,msdos1'
 if [ x$feature_platform_search_hint = xy ]; then
 search --no-floppy --fs-uuid --set=root --hint-bios=hd0,msdos1 --hint-
efi=hd0,msdos1 --hint-baremetal=ahci0,msdos1 cf0736b3-7e7d-4b65-9a44-
64bbb8956973
 else
 search --no-floppy --fs-uuid --set=root cf0736b3-7e7d-4b65-9a44-64bbb8956973
 fi
 echo 'Loading Linux 5.9.16 ...'
 linux /boot/vmlinuz-5.9.16 root=UUID=cf0736b3-7e7d-4b65-9a44-64bbb8956973 ro
quiet net.ifnames=0
 echo 'Loading initial ramdisk ...'
 initrd /boot/initrd.img-5.9.16
 }
 menuentry 'Debian GNU/Linux, with Linux 5.9.16 (recovery mode)' --class debian
--class gnu-linux --class gnu --class os $menuentry_id_option 'gnulinux-5.9.16
-recovery-cf0736b3-7e7d-4b65-9a44-64bbb8956973' {
 load_video
 insmod gzio
 if [ x$grub_platform = xxen ]; then insmod xzio; insmod lzopio; fi
 insmod part_msdos
 insmod ext2
 set root='hd0,msdos1'
 if [ x$feature_platform_search_hint = xy ]; then
 search --no-floppy --fs-uuid --set=root --hint-bios=hd0,msdos1 --hint-
efi=hd0,msdos1 --hint-baremetal=ahci0,msdos1 cf0736b3-7e7d-4b65-9a44-
64bbb8956973
 else
 search --no-floppy --fs-uuid --set=root cf0736b3-7e7d-4b65-9a44-64bbb8956973
 fi
 echo 'Loading Linux 5.9.16 ...'
 linux /boot/vmlinuz-5.9.16 root=UUID=cf0736b3-7e7d-4b65-9a44-64bbb8956973 ro
single
 echo 'Loading initial ramdisk ...'
 initrd /boot/initrd.img-5.9.16
 }
}
```

Instalacija softvera iz izvornog koda završava instalacijom odnosno postavljanjem izvršnih i konfiguracijskih datoteka u odgovarajuće direktorije. Kod jezgre to nije zadnji korak jer je jezgra temelj cijelog sustava te promjena jezgre zahtjeva ponovno pokretanje sustava, dakle za naš primjer: `# reboot`

I u izborniku odabir jedne od inačica "Debian GNU/Linux, with Linux 5.9.16 (recovery mode)" ili "Debian GNU/Linux, with Linux 5.9.16". Nakon pokretanja sustava naredbom **uname -r** moguće je prikazati inačicu aktivne jezgre:

```
# uname -r
5.9.16
```

1.2.5. Programi za učitavanje operacijskog sustava i inicijalna RAM datoteka

Program za učitavanje operacijskog sustava (*boot loader*) je prvi softver koji se pokreće na računalu, a pokreće ga **BIOS** (*Basic Input/Output System*) ili **UEFI** (*Unified Extensible Firmware Interface*). Program za učitavanje operacijskog sustava nadležan je za učitavanje jezgre, parametara jezgre i inicijalne **RAM** datoteke (*inititial RAM disk*). Danas standardni pokretač sustava je **GRUB**. Ranije korišteni, danas uglavnom napušteni pokretač sustava je **LILO** koji je prethodio **GRUB**-u.

GRUB je kratica za **GRand Unified Bootloader** i standardan je program za učitavanje operacijskog sustava na *Linux* distribucijama, uključujući i GNU/Debian.

GRUB datoteke nalaze se u direktoriju **/boot/grub**, a središnja konfiguracijska datoteka je **grub.cfg**. Datoteka se generira automatski naredbom **update-grub**, koristeći zapise iz skripta iz direktorija **/etc/grub.d/** i skripte **/etc/default/grub**. Izmjene postavki pokretača sustava provode se mijenjanjem datoteke **/etc/default/grub** i ponovnim pokretanjem datoteke **update-grub**.

Primjer izgleda minimalne datoteke **grub.cfg**:

```
timeout=5

menuentry 'Debian GNU/Linux, with Linux 5.9.16' {
root=hd0,1
linux /boot/kernel-5.9.16 root=/dev/sda3
initrd /boot/initrd-5.9.16.img
}
```

Timeout definira koliko dugo grub čeka u izborniku na nalog korisnika prije automatskog nastavka. **Menuentry** definira opis koji će se za određenu konfiguraciju jezgre prikazati u izborniku. Unutar vitičastih zagradi su parametri jezgre: ishodišni direktorij ("/"), jezgra, datoteka **initrd** i direktorij te uređaj na kojem se nalaze.

Izmjene datoteke **/etc/default/grub** ili **/boot/grub/grub.cfg** neće postati aktivne dok se ne pozove naredba **update-grub** i dok se ponovno ne pokrene sustav.

Inicijalna RAM datoteka (*inititial RAM disk,* skraćeno **initrd**) je datoteka koja sadrži podatke za učitavanje inicijalnog i virtualnog ishodišnog datotečnog sustava (*root file system*) u memoriju. Taj je virtualni datotečni sustav potreban da bi se učitali upravljački programi (*driver*) koji omogućavaju pristup blok uređajima. Tako se omogućava učitavanje pravog ishodišnog datotečnog sustava i jezgre. Nakon toga se **initrd** uklanja i oslobađa se memorijski prostor koji je **initrd** zauzimao. Neke distribucije standardno, a sve distribucije u *rescue modu* rada rabe **initrd** kao konačni (i jedini) ishodišni datotečni sustav.

Initrd je vezan na jezgru i pri pokretanju sustava **initrd** se učitava kao dio učitavanja jezgre.

1.3. Vježba: Upravljanje modulima

1. Prijavite se na računalo kao korisnik l102. U GUI-u pokrenite **Terminal** (*Activities* → **Terminal**).
2. Naredbom `su -` postanite **root** korisnik (lozinka: L102).
3. a) Ispišite aktivne module na ekran.

b) O kojim modulima ovisi modul **usbcore**?

4. Ispišite na ekran sadržaj direktorija u čijem se poddirektoriju nalaze moduli koji **mogu** biti učitani. Uđite u taj direktorij.
5. Iz direktorija iz 4. zadatka uđite u poddirektorij **./kernel/sound/pci/hda/**. U direktoriju se nalazi niz modula – ispišite na ekran sadržaj direktorija.
6. Provjerite koliko je modula učitano u jezgri (`lsmod | wc -l`)?
7. Pojedinačno učitajte module **snd-hda-codec-idt**, **snd-hda-codec** i **snd-hda-codec-hdmi** naredbom **modprobe**. Koliko je sada modula učitano? Zašto?
8. Uklonite module koje ste učitali (pojedinačnim naredbama modprobe). Pokušajte početi s **snd-hda-codec**. Što se dogodilo? Zašto? Dovršite uklanjanje modula.

1.4. Vježba: Programsko prevođenje jezgre

1. Prijavite se na računalo kao korisnik l102. U GUI-u pokrenite **Terminal** (*Activities* → **Terminal**).
2. Naredbom `su -` postanite **root** korisnik (lozinka: L102).
3. Izvršite naredbu za instalaciju paketa potrebnih za programsko prevođenje jezgre:

```
# apt install devscripts equivs libncurses5-dev libncursesw5-dev
wget qt5-qmake pkg-config qtbase5-dev flex bison -y
```

4. Izađite iz *root* okoline (`# exit` ili **CTRL + D**)
5. Napravite direktorij **/tmp/jezgra**.
6. Uđite u direktorij **/tmp/jezgra** i u njega s mreže pribavite sliku jezgre (na stranici kernel.org odaberite jednu od inačica *longterm* tarball, na primjer:

   https://www.kernel.org/pub/linux/kernel/v4.x/linux-5.9.16.tar.xz

```
$ cd /tmp/jezgra
$ firefox www.kernel.org &
$ wget https://cdn.kernel.org/pub/linux/kernel/v5.x/linux-
5.10.142.tar.xz
```

7. Otpakirajte pribavljenu datoteku i uđite u novoizrađeni direktorij. (`$ tar -vxJf`).

Pokrenite `make xconfig` i proučite mogućnosti. Uočite kako su podijeljene kategorije i kako se dolazi do određenih svojstava.

Zatvorite editor i snimite promjene u **.config**. Pregledajte konfiguracijsku datoteku **.config**.

2. Pokretanje sustava Linux

Trajanje poglavlja:
45 min

Po završetku ovoga poglavlja moći ćete:

- razumjeti, locirati, očitati i promijeniti skriptu za pokretanje
- podesiti skriptu da se pokreče standardno sa sustavom
- provesti osnovne operacije upravljanja servisima u **systemd**
- ručno ili pomoću naredbi insserv ili update-rc.d izraditi simboličke poveznice za automatsko pokretanje servisa u određenoj razini izvođenja (runlevel) u SysVinit
- protumačiti ili podesiti LSB zaglavlje u skripti za pokretanje servisa u SysVinit
- definirati ulogu programa za učitavanje operacijskog sustava
- promijeniti postavke u konfiguracijskim datotekama GRUB-a i automatski ih unijeti u grub.cfg
- razlikovati i kronološki poredati korake pokretanja sustava.

Ova cjelina obrađuje pokretanje sustava Linux. U lekciji su obrađeni koraci pokretanja sustava kao i skripte za pokretanje servisa koje su integralni dio pokretanja sustava. U cjelini su dane osnove aktualnog init sustava (systemd) i System V - njegova prethodnika na Debianu.

2.1. systemd

2.1.1. Init sustav systemd

systemd je init sustav koji se koristi u Linux distribucijama za upravljanje korisničkim prostorom i slijedno pokretanje servisa. Ime **systemd** pridržava se Unix konvencije imenovanja *daemona* dodavanjem slova d na kraj imena servisa, budući da upravlja pokretanjem sustava (engl. *system*). **systemd** je konfiguriran na način da je kompatibilan sa SysV init skriptama. Na brojnim sustavima je **systemd** danas standardan sustav.

Novi koncept koji **systemd** donosi je koncept **jedinica** (engl. *unit*). Jedinice su predstavljene konfiguracijskim datotekama smještenim u direktorijima `/lib/systemd/system` i `/etc/systemd/system/`.

Direktorij `/etc/systemd/system/` je prioritetni direktorij i konfiguracijske datoteke jedinica koje su smještene u taj direktorij imaju najveći prioritet. Najveći prioritet znači da će se u slučaju da postoji više istoimenih konfiguracija u različitim direktorijima pri konfiguraciji jedinice koristiti postavke unesene u datoteci koja se nalazi u `/lib/systemd/system`.

Na aktivnom Linux ili Unix sustavu nalazi se u svakom trenutku rada niz aktivnih pozadinskih procesa. Ti pozadinski procesi poznati su i kao servisi. Servisi mogu biti dio rada operacijskog sustava ili se pokretati kao dio aplikacije. Neovisno o njihovoj prirodi ili namjeni servise koji trebaju biti aktivni na sustavu pokreće sustav **systemd**. Središnja naredba za upravljanje servisima je **systemctl**.

2.1.2. Naredba systemctl

systemctl je središnja naredba za upravljanje servisima u **systemd**. Napomenimo da konfiguracija za servis mora postojati u odgovarajućem direktoriju na sustavu kako bi **systemctl** mogao upravljati servisom.

Sintaksa naredbe **systemctl** je:

`systemctl <opcija> <cilj>`.

Cilj naredbe je servis ili neki drugi tip jedinice, a osnovne opcije za upravljanje servisima su navedene i protumačene u donjoj tablici.

Opcija	Cilj
start	Opcija za pokretanje servisa
stop	Opcija za zaustavljanje servisa
restart	Opcija za ponovno pokretanje servisa
reload	Opcija za ponovno učitavanje konfiguracije servisa
status	Provjera stanja servisa - vraća stanje servisa sa detaljima o pokretanju servisa i zadnje zapise o aktivnosti servisa
is-active	Provjera je li servis pokrenut
enable	Postavljanje da se servis pokreće pokretanjem sustava
disable	Postavljanje da se servis ne pokreće pokretanjem sustava

Konfiguracijske datoteke za servise završavaju postfiksom **.service** , ali pri pozivima tog servisa ne mora se koristiti puno ime. Na primjer, za servis **sshd** pozivi za pokretanje mogu biti:

`#systemctl start sshd` i `#systemctl start sshd.service`

Obje naredbe će uspješno pokrenuti sshd servis kako je vidljivo u sljedeća dva primjera:

```
root@debian-1:~# systemctl status sshd.service
● ssh.service - OpenBSD Secure Shell server
Loaded: loaded (/lib/systemd/system/ssh.service; enabled; vendor preset:
enabled)
Active: inactive (dead) since Tue 2017-09-05 10:57:55 CEST; 57s ago
root@debian-1:~# systemctl start sshd.service
root@debian-1:~# systemctl status sshd.service
● ssh.service - OpenBSD Secure Shell server
Loaded: loaded (/lib/systemd/system/ssh.service; enabled; vendor preset:
enabled)
Active: active (running) since Tue 2017-09-05 10:59:02 CEST; 6s ago
```

```
root@debian-1:~# systemctl status sshd.service
● ssh.service - OpenBSD Secure Shell server
Loaded: loaded (/lib/systemd/system/ssh.service; enabled; vendor preset:
enabled)
Active: inactive (dead) since Tue 2017-09-05 11:00:24 CEST; 1s ago
root@debian-1:~# systemctl start sshd
root@debian-1:~# systemctl status sshd.service
● ssh.service - OpenBSD Secure Shell server
Loaded: loaded (/lib/systemd/system/ssh.service; enabled; vendor preset:
enabled)
Active: active (running) since Tue 2017-09-05 11:00:32 CEST; 2s ago
```

U gornjem primjeru dan je samo dio izlaza naredbi za prikaz statusa servisa. Za olakšavanje uklanjanja problema pri radu sa servisima status naredba u izlazu daje i dodatne podatke o procesu servisa i zadnje zapise iz datoteke aktivnosti servisa.

systemd je velika promjena u odnosu na **Sysvinit**, a zbog toga je korisno znati koji su u **systemd** ekvivalenti standardnih operacija iz **sysvinit**.

Sljedeća tablica daje prikaz naredbi za upravljanje servisima zajedno sa ekvivalentima naredbama u Sysvinit.

Naredba Sysvinit	Naredba Systemd	Objašnjenje
service <servis> start	systemctl start <servis>	Trenutačno pokretanje servisa (stanje ne ostaje pri restartu).
service <servis>stop	systemctl stop <servis>	Trenutačno zaustavljanje servisa (stanje ne ostaje pri restartu).
service <servis>restart	systemctl restart <servis>	Slijedno stop pa start.
service <servis>reload	systemctl reload <servis>	Isto što i restart, ali ako je moguće bez zaustavljanja učita novu konfiguraciju.
service <servis>condrestart	systemctl condrestart <servis>	Ponovno pokretanje (*restart*) ali samo servisa koji je već pokrenut.
service <servis>status	systemctl status <servis>	Testiranje stanja servisa.
ls /etc/rc.d/init.d/	systemctl (ili) systemctl list-unit-files --type=service (ili) ls /lib/systemd/system/*.service /etc/systemd/system/*.service	Ispis svih servisa (i više kod systemd) koji mogu biti pokrenuti.
*chkconfig <servis>on	systemctl enable <servis>	Postavljanje da se servis pokreće automatski pokretanjem sustava.
*chkconfig <servis>off	systemctl disable <servis>	Postavljanje da se servis ne pokreće automatski pokretanjem sustava.
*chkconfig <servis>	systemctl is-enabled <servis>	Provjera da li se servis pokreće automatski pokretanjem sustava.

*chkconfig –list	systemctl list-unit-files --type=service (ili) ls /etc/systemd/system/*.wants/	Prikaz na koji su način servisi podešeni (uključeni/isključeni/nešto drugo).
*chkconfig --list | grep 5:on	systemctl list-dependencies graphical.target	Prikaz svih servisa koji se pokreću sa sustavom.
*chkconfig <servis>--list	ls /etc/systemd/system/*.wants/<servis>.service	Prikaz kada se servis pokreće (koja razina izvođenja ili unutar koje mete).
*chkconfig <servis>--add	systemctl daemon-reload	Naredba za obnovu stanja kada se doda servis odnosno njegove konfiguracije.

*U prvom stupcu su sa zvjezdicom označene naredbe za izvršavanje kojih je potrebno instalirati dodatan softver u Debianu. Te naredbe su standardne u RedHat distribuciji i na njemu zasnovanim distribucijama. Kada nije bilo moguće na Debianu instalirati dodatni softver ponekad je bilo potrebno napisati skriptu.

systemd upravlja pokretanjem sustava koristeći novi koncept jedinica. Jedinica može biti servis, utičnica (engl. *socket*), uređaj, montiranje (engl. *mount point*) i drugo. **systemd** podržava čak 12 različitih jedinica. Jedinice se definiraju konfiguracijskim datotekama, a tip jedinice vidljiv je iz sufiksa imena datoteke konfiguracije.

Instrukcije zadane u konfiguracijskoj datoteci jedinice uključuju: kada se jedinica pokreće, o kojim drugim resursima ili jedinicama ova jedinica ovisi, kojom se naredbom pokreće i zaustavlja, kako **systemd** testira je li jedinica aktivna i treba li poduzeti neke akcije ako nije aktivna i brojne druge instrukcije.

Tip jedinice i datoteka koja definira pojedinu jedinicu definiraju ponašanje pri izvođenju **systemctl** naredbe. Za svaku jedinicu mora postojati konfiguracijska datoteka, a datoteke koje definiraju jedinice instaliraju se u direktoriju `/lib/systemd/system`. Datoteke smještene u direktoriju `/etc/systemd/system` smještaju se kako bi se promijenilo ponašanje postojećih servisa ili definiralo ponašanje korisničkih aplikacija. Prioritet koji ovaj direktorij ima omogućava forsiranje ponašanja u skladu sa konfiguracijom u tom direktoriju.

Mete (engl. *target*) su drugi novi koncept koji je uveo **systemd**. **Mete** su skupine jedinica. Na taj se način izrađuju logički okviri sastavljeni od više jedinica koji ostvaruju pojedinu funkcionalnost. Tako, na primjer, meta **graphical.target** poziva sve elemente potrebne za pokretanje grafičkog sučelja. Važno je uočiti da **meta** može sadržavati ne samo jedinice, već i druge **mete**.

2.1.3. systemd - konfiguracija pokretanja sustava

systemd osim izravnog upravljanja pojedinim jedinicama diktira i ponašanje sustava pri pokretanju. **Systemd** je prvi proces koji se pokreće nakon inicijalizacije sustava, a središnja datoteka koja upravlja pokretanjem sustava je konfiguracija mete `default.target`.

Kako je ranije navedeno, ako ta datoteka postoji u direktoriju `/etc/systemd/system/` onda ona definira ponašanje sustava pri pokretanju. Inače se koristi datoteka iz direktorija

`/lib/systemd/system/`. Ta je datoteka najčešće simbolička poveznica na neku drugu metu koja definira pokretanje svih potrebnih servisa za rad željene okoline.

Tako kada je, na primjer, željena okolina grafičko sučelje **default** može biti simbolička poveznica na metu **graphical**. Navedeno je vidljivo u ovom primjeru:

```
root@debian-1:~# ls -all /lib/systemd/system/default.target
lrwxrwxrwx 1 root root 16 Jul 5 22:31 /lib/systemd/system/default.target
-> graphical.target
```

Kada primjerice ne bismo željeli da se pokreće grafičko sučelje zajedno sa sustavom bilo bi dovoljno promijeniti postojeću poveznicu ili izraditi novu u prioritetnom direktoriju **/etc/systemd/system/**.

U nastavku su prikazana oba primjera:

```
root@debian-1:/lib/systemd/system# ln -sf multi-user.target default.target
root@debian-1:/lib/systemd/system# ls -all /lib/systemd/system/default.target
lrwxrwxrwx 1 root root 17 Sep 4 14:11 /lib/systemd/system/default.target ->
multi-user.target
root@debian-1:/lib/systemd/system# ln -s /lib/systemd/system/multi-user.target
/etc/systemd/system/default.target
root@debian-1:/lib/systemd/system# ls -all /etc/systemd/system/default.target
lrwxrwxrwx 1 root root 37 Sep 4 14:13 /etc/systemd/system/default.target ->
/lib/systemd/system/multi-user.target
```

Nakon ovih izmjena dovoljno je ponovno pokrenuti sustav i nakon toga grafičko sučelje neće biti aktivno. Ovakav pristup omogućava neograničeni broj konfiguracija za pokretanje sustava. Tako je moguće postaviti konfiguracije u kojima se sa sustavom pokreće proizvoljan izbor dostupnih servisa. U Sysvinit za to su korištene razine izvođenja, više informacija o razinama izvođenja i njihovom pozivu u systemd nalazi se u poglavlju 2.2.

2.1.4. Skripte za upravljanje jedinicama

Ranije je navedeno da postoji 12 tipova jedinica, a mogućnosti njihove konfiguracije se razlikuju.

Struktura datoteka za konfiguraciju jedinica razbijena je na segmente strogo definiranog imena smještenih u uglate zagrade. Primjerice `[Unit]`, `[Service]`, `[Install]` i slično. Nakon definicije u konfiguraciji se nalaze uređeni parovi Ime=Vrijednost.

Na primjer, za **ssh** servis:

```
[Unit]
Description=OpenBSD Secure Shell server
After=network.target auditd.service
ConditionPathExists=!/etc/ssh/sshd_not_to_be_run

[Service]
```

U gornjem primjeru je dana standardna konfiguracija `[Unit]` detalja o servisu **ssh**. Važno je uočiti da blok konfiguracija završava definicijom novog bloka pa nije potrebno zatvoriti pojedini blok kao kod formata xml.

Standardno je prvi segment konfiguracije `[Unit]`. U tom se segmentu postavlja međusobni odnos sa drugim jedinicama i daje dodatni opis jedinice.

Datoteka završava sa opcionalnim segmentom `[Install]`. Ovaj dio definira ponašanje jedinice kada se njeno stanje mijenja iz ***enabled*** u ***disabled*** i obrnuto. U stanju **enabled** jedinica se pokreće zajedno sa sustavom. To konkretno znači da se jedinica poveže sa nekom jedinicom koja se već pokreće pri pokretanju sustava.

Između ranije navedenih segmenata nalazi se konfiguracija jedinice istoimena imenu jedinice. Za jedinice tipa device, target, snapshot i scope ne postoje specifične direktive pa se zbog toga u definiciji jedinica tog tipa ne nalaze segmenti istoimeni tipu jedinice.

U praksi najčešće korišten tip jedinice je servis. U segmentu `[Service]` se može definirati tip servisa, način pokretanja i zaustavljanja, skripte koje se moraju izvršiti prije/poslije zaustavljanja i pokretanja servisa te mnoge druge opcije.
Na primjer, standardna konfiguracija za **ssh** servis je:

```
[Unit]
Description=OpenBSD Secure Shell server
After=network.target auditd.service
ConditionPathExists=!/etc/ssh/sshd_not_to_be_run

[Service]
EnvironmentFile=-/etc/default/ssh
ExecStart=/usr/sbin/sshd -D $SSHD_OPTS
ExecReload=/bin/kill -HUP $MAINPID
KillMode=process
Restart=on-failure
RestartPreventExitStatus=255
Type=notify

[Install]
WantedBy=multi-user.target
Alias=sshd.service
```

U dijelu `[Unit]` je definiran opis servisa, jedinice i mete koje moraju biti pokrenut prije ssh i direktorij koji ne smije postojati kako bi se ssh pokrenuo.

U dijelu konfiguracije `[Service]` je definirano slijedno: datoteka koja definira okolinu za izvršavanje, naredba za pokretanje i zaustavljanje servisa, način prisilnog zaustavljanja servisa, kada se ponovno pokreće servis i kada se neće ponovno pokrenuti servis te na kraju tip servisa.

U dijelu konfiguracije `[Install]` definirano je koja meta očekuje ovaj servis i alternativni naziv za servis koji se može koristiti pri pozivima naredbe **systemctl**.

2.2. SysVinit

2.2.1. Razine izvođenja

Razina izvođenja (*runlevel*) je specifični način rada računalnih operacijskih sustava koji implementiraju inicijalizaciju *System-V*. Postoji sedam razina izvođenja koje označavaju brojevi od 0 do 6. Katkad se rabi 10 razina izvođenja numeriranih brojevima od 0 do 9. U tablici su prikazane standardne razine izvođenja sa kratkim opisom.

Razina izvođenja	Opis
0	zaustavljanje sustava
1	jednokorisnički
2-5	višekorisnički
6	ponovno pokretanje sustava

Operacijski sustav ulazi u točno jednu razinu pri pokretanju. Prelazak između razina izvođenja provodi se gašenjem postojećeg prije ulaska u novi. U svakom se trenutku operacijski sustav nalazi u točno jednoj razini izvođenja.

Razine izvođenja se u **SysVinit** koriste za stvaranje distinktnih okolina pri pokretanju sustava koje su prilagođene određenoj namjeni. Konfiguracijska datoteka za upravljanje razinama izvođenja je **/etc/inittab**.

Najvažnije linije u konfiguraciji su:

```
id:2:initdefault:
```

Ova linija definira zadanu razinu izvođenja 2 i pri normalnom tijeku pokretanja sustava razina izvođenja će biti 2.

```
l0:0:wait:/etc/init.d/rc 0
l1:1:wait:/etc/init.d/rc 1
l2:2:wait:/etc/init.d/rc 2
l3:3:wait:/etc/init.d/rc 3
l4:4:wait:/etc/init.d/rc 4
l5:5:wait:/etc/init.d/rc 5
l6:6:wait:/etc/init.d/rc 6
```

Ove linije definiraju koji se servisi pokreću u svakoj zadanoj razini izvođenja. Preciznije definiraju da se u razini X pokreće naredba **/etc/init.d/rc** s opcijom X. Naredba **rc** s opcijom X pokreće sve skripte u direktoriju **/etc/rcX.d**. Konfiguracija servisa koji se pokreću pri određenoj razini izvođenja provodi se izvan datoteke **inittab** kroz izmjene servisa u direktorijima **rcX.d** koje pokreće naredba **rc**.

Svakom procesu dodijeljen je broj koji je njihov identifikator procesa. Pri pokretanju sustava prva naredba koja se izvodi nakon učitavanja jezgre i ishodišnog direktorija je `init` u **SysVinit** i `systemd` u **systemd** init sustavu. Procesi init i systemd su prvi procesi koji se izvode u

korisničkom prostoru. Zbog toga je tim procesima dodijeljen identifikator procesa 1 i oni su roditelji svim procesima.

Prelazak iz aktivne razine izvođenja u neki drugi provodi se naredbama **init** i **telinit**.

Sintaksa naredbi je ista - prihvaćaju jednu opciju od mogućih:

Opcija	Objašnjenje
0-6	pokreće se prelazak u zadani *runlevel*
a,b,c	init poziva samo onaj dio /etc/inittab konfiguracije označen zadanim slovom
q ili Q	init provjerava datoteku /etc/inittab za moguće izmjene
s ili S	pokreće se prijelaz u *single user mode*
u ili U	init se ponovno poziva (bez provjere izmjena datoteke /etc/inittab)

U **systemd** postoje (u trenutku pisanja ovog teksta) naredbe **init** i **telinit**, te naredbe pozivaju mete imena runlevel[0-6]. Ovako imenovane mete su samo privremeno rješenje. Sve te mete su samo simboličke poveznice na druge standardne **systemd** mete. Sljedeća naredba na ekran prikazuje kako su to u biti poveznice na mete poweroff, rescue, multi-user, graphical i reboot.

```
# ls -all /lib/systemd/system/|grep runlevel|grep -v wants
lrwxrwxrwx 1 root root 15 Jul 5 22:31 runlevel0.target -> poweroff.target
lrwxrwxrwx 1 root root 13 Jul 5 22:31 runlevel1.target -> rescue.target
lrwxrwxrwx 1 root root 17 Jul 5 22:31 runlevel2.target -> multi-user.target
lrwxrwxrwx 1 root root 17 Jul 5 22:31 runlevel3.target -> multi-user.target
lrwxrwxrwx 1 root root 17 Jul 5 22:31 runlevel4.target -> multi-user.target
lrwxrwxrwx 1 root root 16 Jul 5 22:31 runlevel5.target -> graphical.target
lrwxrwxrwx 1 root root 13 Jul 5 22:31 runlevel6.target -> reboot.target
```

2.2.2. Direktoriji rcX.d

Svaka je razina izvođenja posebna po tome što su neki servisi zaustavljeni ili pokrenuti. Skripte za pokretanje tih servisa smještene su u direktorijima **/etc/rc.d/init.d/** ili **/etc/init.d/**. U distribuciji *Debian/GNU* koristi se direktorij **/etc/init.d/**. Na standardnom *desktop Linux* sustavu u tom se direktoriju nalazi više od 50 skripti:

```
# ls /etc/init.d/
acpid checkroot.sh kbd mountdevsubfs.sh procps saned umountroot alsa-utils
console-setup kdm mountkernfs.sh pulseaudio sendsigs unattended-upgrades anacron
cron keyboard-setup mountnfs-bootclean.sh rc single urandom atd dbus killprocs
mountnfs.sh rc.local skeleton vboxadd avahi-daemon exim4 kmod mtab.sh rcS
speech-dispatcher vboxadd-service binfmt-support gdm3 lm-sensors networking
README ssh vboxadd-x11 bluetooth halt lpd network-online reboot sudo virtualbox-
guest-utils bootlogs hddtemp minissdpd nfs-common rmnologin udev x11-common
bootmisc.sh hdparm motd openvpn rpcbind udev-mtab checkfs.sh hostname.sh
mountall-bootclean.sh postfix rsync umountfs checkroot-bootclean.sh hwclock.sh
mountall.sh pppd-dns rsyslog umountnfs.sh
```

Standardna skripta za pokretanje servisa treba prihvaćati četiri opcije:

Opcija	Opis
start	Pokreće servis.
stop	Zaustavlja servis.
restart	izvršava zaustavljanje (*stop*) za pokretanje (*start*).
status	Vraća status servisa (*running* ili *stopped*).

Pogledajmo sadržaj direktorija **/etc/rc2.d/**

```
$ ls /etc/rc2.d/ -l
README
S01motd -> ../init.d/motd
S01vboxadd -> ../init.d/vboxadd
S02vboxadd-service -> ../init.d/vboxadd-service
S13rpcbind -> ../init.d/rpcbind
S14nfs-common -> ../init.d/nfs-common
S16binfmt-support -> ../init.d/binfmt-support
S16rsyslog -> ../init.d/rsyslog
S16sudo -> ../init.d/sudo
S16virtualbox-guest-utils -> ../init.d/virtualbox-guest-utils
S17acpid -> ../init.d/acpid
S17anacron -> ../init.d/anacron
S17atd -> ../init.d/atd
S17cron -> ../init.d/cron
S17dbus -> ../init.d/dbus
S17exim4 -> ../init.d/exim4
S17hddtemp -> ../init.d/hddtemp
S17lpd -> ../init.d/lpd
S17postfix -> ../init.d/postfix
S17rsync -> ../init.d/rsync
S17speech-dispatcher -> ../init.d/speech-dispatcher
S17ssh -> ../init.d/ssh
S18avahi-daemon -> ../init.d/avahi-daemon
S18bluetooth -> ../init.d/bluetooth
S18network-online -> ../init.d/network-online
S19openvpn -> ../init.d/openvpn
S20gdm3 -> ../init.d/gdm3
S20kdm -> ../init.d/kdm
S20pulseaudio -> ../init.d/pulseaudio
S20saned -> ../init.d/saned
S21bootlogs -> ../init.d/bootlogs
S22minissdpd -> ../init.d/minissdpd
S22rc.local -> ../init.d/rc.local
S22rmnologin -> ../init.d/rmnologin
```

Kao što je vidljivo osim datoteke **README** sve su datoteke simboličke poveznice na skripte u direktoriju **/etc/init.d/** . Sve skripte za pokretanje/zaustavljanje servisa moraju se nalaziti u tom

direktoriju i tada se stvaraju simboličke poveznice u **rc[0-6].d** direktorijima. Broj u imenu direktorija je broj razine izvođenja *(runlevel*) u kojem će se pokrenuti ili zaustaviti skripte čije su simboličke poveznice u direktoriju sa istim brojem.

2.2.3. Naredbe update-rc.d i chkconfig

Tri su načina za dodavanja servisa u proizvoljnu razinu izvođenja u SysVinit:

1. Ručno, izradom simboličke poveznice naredbom **# ln -s.**
2. Korištenjem naredbe **# update-rc.d**
3. Korištenjem naredbe **# insserv**

Ručno dodavanje treba izbjegavati iz više razloga. Prvi je razlog što se pri ručnom podešavanju najlakše može pogriješiti. Nadalje to je najsloženiji način za dodavanje skripte u određene razine izvođenja. Konfiguracija koju provode naredbe **update-rc.d** ili **insserv** ista je kao sedam ručnih dodavanja, a samim time postoji i sedam puta više mogućnosti za pogrešku.

Naredba **update-rc.d** starija je i korištena je do *Debian*ove inačice 6.0. U inačici 6.0 (*squeeze* izdan 6. veljače 2011.) za upravljanje servisima pod različitim razinama izvođenja uvedena je naredba **insserv**, a u inačici 7.0 (*wheezy* izdan 4. svibnja 2013.) ponovno se koristi naredba **update-rc.d,** koja poziva **insserv** kao sistemski poziv koji izvršava akciju koju poziva **update-rc.d**. Za odluku kojom se naredbom treba koristiti na drugim operacijskim sustavima, najbolje je pogledati man stranice tih naredbi.

Sintaksa naredbe **update-rc.d** je:

```
update-rc.d [-n] [-f] name remove
update-rc.d [-n] name defaults [NN | SS KK]
update-rc.d [-n] name disable|enable [ S|2|3|4|5 ]
```

Opcija **-n** koristi se za testiranje i pomoću nje naredba isproba i javi kako bi prošlo provođenje naredbe. Opcija **-f** treba se koristiti svakog puta kad se uklanjaju simboličke poveznice na skriptu koja i dalje postoji u direktoriju **/etc/init.d/** . (Nema smisla, ali tako je.)

Opcija **defaults** opisuje pod kojim se razinama izvođenja odvija koja akcija (start|stop). Bez eksplicitnog navođenja razine izvođenja naredbom će se u svim direktorijima (rc[0-6].d) stvoriti simbolička poveznica start ili stop. Tako na primjer, ove dvije naredbe stvaraju identične simboličke poveznice:

```
update-rc.d foobar defaults
update-rc.d foobar start 20 2 3 4 5 . stop 20 0 1 6.
```

Razlog uvođenja nove naredbe **insserv** bio je uvođenje dodatnog LSB (*Linux Standard Base*) standarda za skripte za pokretanje servisa. Prema novom standardu na početku svake skripte treba biti opisni blok ovog oblika:

```
### BEGIN INIT INFO
# Provides:          ime_skripte
# Required-Start:    $servis_o_kojem_ovisi_1 $servis_o_kojem_ovisi_2
# Required-Stop:     $servis_o_kojem_ovisi_1 $servis_o_kojem_ovisi_2
# Default-Start:     2 3 4 5
# Default-Stop:      0 1 6
# Short-Description: Start daemon at boot time
# Description:       Enable service provided by daemon.
### END INIT INFO
```

Iako je ovo blok komentara i neće se koristiti (niti utjecati na bilo koji način) pri ručnom pozivu skripte, taj blok čitaju i prema njemu pri pokretanju sustava naredbe **update-rc.d**, **insserv** i servis **init** usklađuju ponašanje sustava:

- Naredbe čitaju polja **Default-Start** i **Default-Stop** i postavljaju start (S) poveznice za razine izvođenja navedene u polju **Default-Start** i stop (K) poveznice za razine izvođenja navedene u polju **Default-Stop** .
- Polje **Provides** definira ime skripte.
- Polja **Required-Start** i **Required-Stop** definiraju koje skripte moraju biti pokrenute prije pokretanja ili zaustavljanja te skripte. Poziv se u ovom polju provodi na osnovu vrijednosti **Provides** referenciranih skripti.

Naredba za ručno pokretanje servisa je **service**. Tom se naredbom zamjenjuje izravni poziv skripte pomoću potpune putanje. Naredbom **service** mogu se izvesti sve standardne akcije navedene u poglavlju 2.2.1. Na primjer za pokretanje servisa **ssh** mogu se koristiti ova dva načina:

```
# /etc/init.d/ssh start
```

ili

```
# service ssh start
```

Važno je uočiti da je slijed parametara naredbe **service** suprotan od naredbe **systemctl** gdje prvo ide akcija, a zatim objekt nad kojim se provodi akcija.

Naredba se **chkconfig** koristi za pregled aktivnih servisa. Opcijom **-A** dobije se stanje svih servisa (koji postoje u direktoriju **/etc/init.d/**) u trenutačnoj razini izvođenja, a opcijom **- l** dobije se pregled u svim razinama izvođenja:

```
$ chkconfig -l
acpid                   0:off  1:off  2:on   3:on   4:on   5:on   6:off
alsa-utils              0:off  1:off  2:off  3:off  4:off  5:off  6:off  S:on
anacron                 0:off  1:off  2:on   3:on   4:on   5:on   6:off
atd                     0:off  1:off  2:on   3:on   4:on   5:on   6:off
avahi-daemon            0:off  1:off  2:on   3:on   4:on   5:on   6:off
binfmt-support          0:off  1:off  2:on   3:on   4:on   5:on   6:off
```

```
bluetooth                  0:off  1:off  2:on   3:on   4:on   5:on   6:off
bootlogs                   0:off  1:on   2:on   3:on   4:on   5:on   6:off
bootmisc.sh                0:off  1:off  2:off  3:off  4:off  5:off  6:off  S:on
checkfs.sh                 0:off  1:off  2:off  3:off  4:off  5:off  6:off  S:on
checkroot-bootclean.sh     0:off  1:off  2:off  3:off  4:off  5:off  6:off  S:on
checkroot.sh               0:off  1:off  2:off  3:off  4:off  5:off  6:off  S:on
console-setup              0:off  1:off  2:off  3:off  4:off  5:off  6:off  S:on
cron                       0:off  1:off  2:on   3:on   4:on   5:on   6:off
dbus                       0:off  1:off  2:on   3:on   4:on   5:on   6:off
exim4                      0:off  1:off  2:on   3:on   4:on   5:on   6:off
gdm3                       0:off  1:off  2:on   3:on   4:on   5:on   6:off
hddtemp                    0:off  1:off  2:on   3:on   4:on   5:on   6:off
hdparm                     0:off  1:off  2:off  3:off  4:off  5:off  6:off  S:on
hostname.sh                0:off  1:off  2:off  3:off  4:off  5:off  6:off  S:on
hwclock.sh                 0:off  1:off  2:off  3:off  4:off  5:off  6:off  S:on
kbd                        0:off  1:off  2:off  3:off  4:off  5:off  6:off  S:on
kdm                        0:off  1:off  2:on   3:on   4:on   5:on   6:off
keyboard-setup             0:off  1:off  2:off  3:off  4:off  5:off  6:off  S:on
killprocs                  0:off  1:on   2:off  3:off  4:off  5:off  6:off
kmod                       0:off  1:off  2:off  3:off  4:off  5:off  6:off  S:on
lm-sensors                 0:off  1:off  2:off  3:off  4:off  5:off  6:off  S:on
lpd                        0:off  1:off  2:on   3:on   4:on   5:on   6:off
minissdpd                  0:off  1:off  2:on   3:on   4:on   5:on   6:off
motd                       0:off  1:on   2:on   3:on   4:on   5:on   6:off
mountall-bootclean.sh      0:off  1:off  2:off  3:off  4:off  5:off  6:off  S:on
mountall.sh                0:off  1:off  2:off  3:off  4:off  5:off  6:off  S:on
mountdevsubfs.sh           0:off  1:off  2:off  3:off  4:off  5:off  6:off  S:on
mountkernfs.sh             0:off  1:off  2:off  3:off  4:off  5:off  6:off  S:on
mountnfs-bootclean.sh      0:off  1:off  2:off  3:off  4:off  5:off  6:off  S:on
mountnfs.sh                0:off  1:off  2:off  3:off  4:off  5:off  6:off  S:on
mtab.sh                    0:off  1:off  2:off  3:off  4:off  5:off  6:off  S:on
network-online             0:off  1:off  2:on   3:on   4:on   5:on   6:off
networking                 0:off  1:off  2:off  3:off  4:off  5:off  6:off  S:on
nfs-common                 0:off  1:off  2:on   3:on   4:on   5:on   6:off  S:on
openvpn                    0:off  1:off  2:on   3:on   4:on   5:on   6:off
postfix                    0:off  1:off  2:on   3:on   4:on   5:on   6:off
pppd-dns                   0:off  1:off  2:off  3:off  4:off  5:off  6:off  S:on
procps                     0:off  1:off  2:off  3:off  4:off  5:off  6:off  S:on
pulseaudio                 0:off  1:off  2:on   3:on   4:on   5:on   6:off
rc.local                   0:off  1:off  2:on   3:on   4:on   5:on   6:off
rcS                        0:off  1:off  2:off  3:off  4:off  5:off  6:off
rmnologin                  0:off  1:off  2:on   3:on   4:on   5:on   6:off
rpcbind                    0:off  1:off  2:on   3:on   4:on   5:on   6:off  S:on
rsync                      0:off  1:off  2:on   3:on   4:on   5:on   6:off
rsyslog                    0:off  1:off  2:on   3:on   4:on   5:on   6:off
saned                      0:off  1:off  2:on   3:on   4:on   5:on   6:off
sendsigs                   0:off  1:off  2:off  3:off  4:off  5:off  6:off
speech-dispatcher          0:off  1:off  2:on   3:on   4:on   5:on   6:off
ssh                        0:off  1:off  2:on   3:on   4:on   5:on   6:off
sudo                       0:off  1:off  2:on   3:on   4:on   5:on   6:off
udev                       0:off  1:off  2:off  3:off  4:off  5:off  6:off  S:on
udev-mtab                  0:off  1:off  2:off  3:off  4:off  5:off  6:off  S:on
umountfs                   0:off  1:off  2:off  3:off  4:off  5:off  6:off
umountnfs.sh               0:off  1:off  2:off  3:off  4:off  5:off  6:off
```

```
umountroot                  0:off  1:off  2:off  3:off  4:off  5:off  6:off
unattended-upgrades         0:off  1:off  2:off  3:off  4:off  5:off  6:off
urandom                     0:off  1:off  2:off  3:off  4:off  5:off  6:off  S:on
vboxadd                     0:off  1:off  2:on   3:on   4:on   5:on   6:off
vboxadd-service             0:off  1:off  2:on   3:on   4:on   5:on   6:off
vboxadd-x11                 0:off  1:off  2:off  3:off  4:off  5:off  6:off
virtualbox-guest-utils      0:off  1:off  2:on   3:on   4:on   5:on   6:off
x11-common                  0:off  1:off  2:off  3:off  4:off  5:off  6:off  S:on
```

Na *Red Hat*u i inačicama *Linux*a izvedenim iz njega naredba za upravljanje i pregled servisa je **chkconfig**.

2.3. GRUB i LILO

2.3.1. LILO i njegova uloga u pokretanju sustava

LILO i **GRUB** su programi za učitavanje operacijskog sustava (*bootloader*) - prvi programi koji se pokreću nakon što BIOS prepusti kontrolu učitavanja operacijskog sustava.

LILO (punim nazivom *Linux Loader*) je godinama bio program za učitavanje operacijskog sustava u većini *Linux*-ovih distribucija. Danas se u većini distribucija **GRUB** rabi kao *bootloader*. U mnogim *Linux*-ovim/*Unix*-ovim sustavima danas nije jednostavno naknadno instalirati **LILO**, nego treba pribaviti paket s nekog nestandardnog repozitorija.

Konfiguracijska datoteka **/etc/lilo.conf** kojom se **LILO** koristi ne pristupa podacima na datotečnom sustavu, jer ne posjeduje sposobnost za interpretiranje sadržaja na datotečnom sustavu. Zbog toga se konfiguracija iz **/etc/lilo.conf** mora prevesti u sustav posebno oblikovanih i pozicioniranih uputa na disku (čitljivih i razumljivih **LILO**-u). Naredba za pripremu sustava na osnovi zapisa u **/etc/lilo.conf** je **/sbin/lilo** i treba je izvršiti nakon svake promjene konfiguracije.

Primjer konfiguracije:

```
lba32
boot = /dev/disk/by-id/ata-VBOX_HARDDISK_VB4ab168b2-f1ee0a8a
map = /boot/map
install = menu
menu-scheme = Wb:Yr:Wb:Wb
prompt
timeout = 100
vga = normal

image = /boot/vmlinuz-5.9.16
 label = "Debian GNU/Linux, with Linux 5.9.16"
 root = "UUID=5559b6e6-6f95-4c4c-a7ee-f197870e4950"
 read-only
 initrd = /boot/initrd.img-5.9.16

 image = /boot/vmlinuz-5.9.16.3-amd64
```

```
 label = "Debian GNU/Linux, with Linux 5.9.16.3-amd64"
 root = "UUID=5559b6e6-6f95-4c4c-a7ee-f197870e4950"
 read-only
 initrd = /boot/initrd.img-5.9.16.3-amd64
```

Osim mogućnosti vidljivih u ovoj konfiguraciji **LILO** podržava brojne mogućnosti, ali najvažniji su one koje opisuju lokaciju i datoteku jezgre:

Opcija	Opis
image	Lokacija jezgre (počinje blok koji dalje opisuje jednu moguću jezgru).
label	Definira ime koje se koristi za referenciranje jezgre (dio *image* bloka).
root	Specificira gdje se nalazi *root* particija (/) (dio *image* bloka).
read-only	Specifikacija da se "/" inicijalno montira u načinu rada *read-only* (kasnije u proceduri pokretanja sustava se montira u načinu rada *read-write* (dio *image* bloka).
initrd	Specificira lokaciju datoteke inicijalnog ramdiska za danu jezgru (dio *image* bloka).
boot	Particija na kojoj se nalazi *boot* sektor.
map	Specificira lokaciju map-datoteke. Ako nije specificirano koristi se **/boot/map**.
default	Ime jezgre koja se automatski pokreće.

2.3.2. GRUB i njegova uloga u pokretanju sustava

GRUB (***GR**and **U**nified **B**ootloader*) je *bootloader* koji je ujedno i jednostavna ljuska koja može čitati sadržaj datotečnih sustava. Danas je to najčešće korišten *bootloader* u sustavima zasnovanim na ***Unix***-u. Postoje dvije inačice: inačica **GRUB1** (danas poznat i kao **GRUB Legacy**) i **GRUB2**.

Tri najčešće korištene *Linux*ove ditribucije (*Ubuntu, Fedora i openSUSE*) rabe **GRUB2** kao osnovni *bootloder*:

Distribucija	Od kad je GRUB2 preferirani bootloader
Ubuntu	*Ubuntu* 9.10 u listopadu 2009.
Fedora	*Fedora* 16 u studenom 2011.
OpenSUSE	*OpenSUSE* 12.2 u rujnu 2012.

Inačica GRUB 1

GRUB se za konfiguracijske datoteke i module potrebne za rad koristi direktorijem **/boot/grub**. Konfiguracijske datoteke u **GRUB-u** su **/boot/grub/menu.lst** ili **/boot/grub/grub.conf**. Te se datoteke ručno uređuju pri dodavanju nove jezgre ili kada želimo promijeniti ponašanje *bootloader*a. Konfiguracija je podijeljena u opći dio postavki na početku datoteke i na dijelove koji opisuju pojedine jezgre ili alternativne operacijske sustave koji se mogu pokrenuti iz izbornika:

```
default=0
timeout=10
splashimage=(hd0,0)/grub/splash.xpm.gz

# section to load Linux
```

```
title Debian GNU/Linux, with Linux 5.9.16
        root (hd0,0)
        kernel /vmlinuz-5.9.16 ro root=/dev/sda2
        initrd /initrd-5.9.16.img

# section to load Windows
title Windows
        rootnoverify (hd0,0)
        chainloader +1
```

Mogućnost	Opis
default	Definira koji se zapis rabi (0 znači prvi).
timeout	Definira vrijeme u sekundama koliko dugo GRUB čeka unos od korisnika prije učitavanja mogućnosti odabrane pod *default*.
splashimage	Definira sliku pozadine za korištenje pri učitavanju GRUB-a.
title	Definira naslov prikazan u GRUB-ovu izborniku.
root	Definira particiju na kojoj se nalazi "/boot".
kernel	Definira putanju do datoteke jezgre i mogućnosti pri mountanju te particije kao i samu particiju.
initrd	Definira putanju do inicijalne RAM datoteke.

2.3.3. GRUB2

GRUB2 se koristi istim direktorijem kao i **GRUB** i na površini radi vrlo slično. Ipak, **GRUB2** je nastao kao iz temelja nanovo kodirani *bootloader* s namjerom dodavanja podrške za ne-x86 platforme, lokalizaciju, omogućavanje korištenja ne-ASCII znakova, dinamičko upravljanje modulima, migracija koda specifičnog za platforme u specifične module i implementaciju objektno orijentiranog *framework*a. Konfiguracijska datoteka je **/boot/grub/grub.cfg** i nije primjerena za ručno uređivanje. Ta se datoteka izrađuje naredbom **update-grub** iz datoteke **/etc/default/grub** i datoteka u direktoriju **/etc/grub.d/**.

```
$ ls /etc/grub.d/
00_header 05_debian_theme 10_linux 20_linux_xen 30_os-prober 40_custom 41_custom
README
$ cat /etc/default/grub |grep -v "#"

GRUB_DEFAULT=0
GRUB_TIMEOUT=5
GRUB_DISTRIBUTOR=`lsb_release -i -s 2> /dev/null || echo Debian`
GRUB_CMDLINE_LINUX_DEFAULT="quiet"
GRUB_CMDLINE_LINUX=""
```

Mijenjanje postavki u **GRUB**-u **2** ostvaruje se uređivanjem tih datoteka i ponovnim pozivom **update-grub**.

Na primjeru datoteke **grub.cfg** koja je daleko složenija od ranije u Legacy varijanti vidljivo je koliko se više postavki postavlja kroz konfiguraciju **GRUB**-a **2** nego kroz **GRUB** i ujedno zašto je poželjno razdvojiti te postavke u više konfiguracijskih datoteka.

Mogućnosti su vrlo slične kao i kod **GRUB**-ove konfiguracije, ali važno je naglasiti da ova konfiguracija ima intuitivniju strukturu s logičkim segmentima koji su izdvojeni u vitičastim zagradama. Također je vidljivo da se izravno naredbama *bootloader*a učitavaju određeni moduli.

2.4. Od pokretanja sustava do ljuske

2.4.1. Koraci u pokretanju sustava

Tri su glavna koraka u pokretanju sustava:

1. *bootloader*
2. jezgra
3. INIT.

Korak *Bootloader*

Uspješno pokrenuti program za učitavanje operacijskog sustava prikazuje izbornik za odabir jezgre ili drugih operacijskih sustava.

U ovom se koraku učitava inicijalna **RAM** datoteka. Jezgra se učitava u radnu memoriju.

Korak Jezgra

Jezgra je učitana u prethodnom koraku i sada se provodi dekompresija. Inicijalna **RAM** datoteka je učitana i učitavaju se dodatni moduli.

Jezgra identificira hardverske komponente sustava i učitava odgovarajuće module. Jezgra zatim priprema za rad "**/**" u zaštićenom načinu rada samo za čitanje (*read only mod*) te postaju dostupni direktoriji **/bin** i **/sbin**.

Završni je korak pokretanje procesa **init** u korisničkom memorijskom prostoru.

Korak INIT ili systemd

Kod SysVinit proces **init** čita sadržaj datoteke **/etc/inittab** i na osnovi tih zapisa postavlja postavke za svoje izvršavanje uključujući razine izvođenja. Zatim se pokreće naredba **/etc/init.d/rcS** (prije **/etc/rc.sysinit**) koja pokreće sadržaj direktorija **/etc/rcS.d/**. U tom se koraku provodi inicijalizacija i priprema za rad svih lokalnih datotečnih sustava u skladu sa zapisima u **/etc/fstab**.

Nakon toga se izvršava ista naredba nad direktorijem odgovarajuće (prije imenovane) razine izvođenja.

Zatim se pokreće program **getty** za upravljanje fizičkim i virtualnim terminalima te na kraju **/bin/login**, čime je završena procedura pokretanja sustava.

Kod systemd proces **systemd** traži **default.target** u direktorijima standardnim za **systemd** skripte jedinica počevši od prioritetnog direktorija`/etc/systemd/system`. Kada pronađe traženu metu pokreće ju i time sve servise jedinice prema uputama iz default mete.

2.5. Vježba: Razine izvođenja

1. Prijavite se na računalo kao korisnik l102. U GUI-u pokrenite **Terminal** (*Activities* → **Terminal**).
2. Naredbom „`su -`“ postanite *root* korisnik (lozinka: L102).
3. Pogledajte u aktivnom sustavu koja je meta osnovna za pokretanje sustava.

```
# ls -al /etc/systemd/system/default.target
# ls -al /lib/systemd/system/default.target
```

Pogledajte sadržaj datoteke koja definira standardnu metu pri pokretanju sustava.

`# cat /lib/systemd/system/default.target`

4. Promijenite simboličku poveznicu `/lib/systemd/system/default.target` tako da pokazuje na datoteku `/lib/systemd/system/multi-user.target`
5. Provjerite koliko je servisa trenutno u sustavu u aktivnom stanju (active).

`# systemctl list-units`

---

6. Ponovno pokrenite sustav. Koje su razlike vidljive?

   Radi li grafičko sučelje?

Provjerite ponovno broj aktivnih servisa.

---

7. Izvršite sljedeću naredbu:

   `# systemctl isolate graphical.target`

   Što se dogodilo?

Provjerite koliko je sada aktivnih servisa?

---

8. Kreirajte sada u direktoriju **/etc/systemd/system/** simboličku poveznicu imena **default.target** na **/lib/systemd/system/graphical.target** i ponovno pokrenite sustav.

9. Ponovite provjeru aktivnih servisa i uočite razlike u odnosu na stanje u 7. zadatku

2.6. Vježba: GRUB – program za učitavanje operacijskog sustava

1. Prijavite se na računalo kao korisnik l102. U GUI-u pokrenite **Terminal** (*Activities* → **Terminal**).
2. Naredbom `su -` postanite root korisnik (lozinka: L102).
3. Ispišite na ekran sadržaj direktorija **/boot/**. Koliko je jezgri instalirano na poslužitelj? Koje su to jezgre?

4. Proučite datoteku **/boot/grub/grub.cfg**. Koliko u datoteci ima zapisa koji definiraju proceduru pokretanja jezgre? Je li broj isti kao broj jezgri? Zašto?

5. Pronađite naredbu koja postavlja jezgru koja se standardno pokreće i promijenite je u neku drugu jezgru.

   Promijeniti liniju

   `set default="0"` u

   `set default="1>1"`

6. Pokrenite naredbu `reboot`.
7. Što se promijenilo pri pokretanju sustava?

8. Upišite root lozinku za prijavu na sustav.
9. Provjerite koja je trenutno aktivna jezgra naredbom `uname -r`.

10. Izvršite naredbu `update-grub2`.
11. Ponovite naredbu `reboot`.
12. Provjerite koja je sad aktivna jezgra (uname -r). Objasnite zašto.

3. Upravljanje grupama i korisnicima

Trajanje poglavlja:
55 min

Po završetku ovoga poglavlja moći ćete:

- opisati razliku između naredbi useradd i adduser
- dodati nove sistemske korisnike
- nove osobne korisnike
- naučiti upravljati članstvom grupa korisnika
- izraditi, promijeniti i izbrisati grupe
- izmijeniti postavke korisničkih računa naredbom **usermod**
- izmijeniti postavke valjanosti lozinke naredbom **chage**.

Ova cjelina obrađuje izradu grupa i korisnika. Upoznajemo odnos korisnika i grupa, način konfiguracije i prilagodbe potrebama.

3.1. Stvaranje novih korisnika

3.1.1. Useradd

Za dodavanje korisnika u sustav *Debian* postoje dvije naredbe: **adduser** i **useradd.** Važno je napomenuti da je u *CentOS*u i brojnim drugim granama *RedHat Linux*ovih distribucija naredba **adduser** samo alias za naredbu **useradd**.

Naredba **useradd** jednostavnija je inačica naredbe za dodavanje korisnika. Već prva linija man-stranice naredbe objašnjava da je naredba **useradd** ograničeni oblik naredbe za dodavanje korisnika i da se savjetuje uporaba naredbe **adduser**. Naredba **useradd** primarno se rabi za dodavanje sistemskih korisnika koji su potrebni za izolaciju pri izvođenju određenih servisa.

Sintaksa naredbe je sljedeća:

```
# useradd <korisnik>
```

Naredba dodaje korisnika **korisnik** i ništa nakon toga. Po izvođenju te naredbe samo administrator može izvoditi naredbe u ime korisnika, jer obzirom da nema lozinke, nikom drugom nije omogućeno naredbom **# su - korisnik** postati novi korisnik. Ako želimo omogućiti prijavljivanje u sustav korisnika izrađenog naredbom **useradd** uz useradd, treba izvesti još barem naredbu **passwd**.

```
# passwd <korisnik>
```

Naredbom **passwd** postavlja se lozinka za korisnika. Bez parametra se naredba **passwd** rabi za mijenjanje lozinke trenutačnog korisnika. Prvo pogledajmo kako korisnik ***root*** mijenja lozinku drugim korisnicima:

```
# passwd l102
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
#
```

Dakle korisnik ***root*** ne mora potvrditi svoj identitet niti znati postojeću lozinku korisnika. Korisnika će sustav pitati trenutačnu lozinku kad želi promijeniti vlastitu lozinku (pozivom **passwd** bez dodatnih opcija):

```
$ passwd
Changing password for l102.
(current) UNIX password:
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
$
```

Ispis osnovnih postavki naredbe **useradd** dobije se pomoću opcije **–D**.

```
# useradd -D
GROUP=100
HOME=/home
INACTIVE=-1
EXPIRE=
SHELL=/bin/zsh
SKEL=/etc/skel
CREATE_MAIL_SPOOL=no
```

Te se opcije mogu mijenjati naredbom **useradd -D** s dodatnim opcijama. Za popis opcija treba pogledati man-stranice. Primjer je promjene direktorija **home** novih korisnika:

```
# useradd -D -b /etc
# useradd -D |grep HOME
HOME=/etc
```

3.1.2. Adduser

Naredba **adduser** interaktivno dodaje korisnika i izvodi niz akcija koje naredba **useradd** ne obavlja, a koje su nužne za omogućavanje rada korisnika. Sintaksa je naredbe istovjetna naredbi **useradd**.

```
# adduser <korisnik>
```

Naredba dodaje korisnika **korisnik** i istoimenu grupu, a zatim korisniku dodijeli tu grupu kao matičnu. Zatim u direktoriju **/home** napravi poddirektorij **/korisnik** i kopira datoteke iz **/etc/skel** u njega. Zadnji obavezni korak je unos lozinke za novog korisnika i to dva puta da bi se potvrdila

točnost unosa. Nakon toga se mogu, ako se to želi, dodati podaci o novom korisniku. Primjer je naredbe **adduser**:

```
# adduser ivanhorvat
Adding user `ivanhorvat' ...
Adding new group `ivanhorvat' (1003) ...
Adding new user `ivanhorvat' (1003) with group `ivanhorvat' ...
Creating home directory `/home/ivanhorvat' ...
Copying files from `/etc/skel' ...
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
Changing the user information for ivanhorvat
Enter the new value, or press ENTER for the default
 Full Name []: Ivan Horvat
 Room Number []: 4
 Work Phone []: 0991919191
 Home Phone []: 019090909
 Other []:
Is the information correct? [Y/n] Y
#
```

3.1.3. Konfiguracijske datoteke i osnovne postavke

Konfiguracijske datoteke koje sadrže konfiguracije korisnika su **/etc/shadow** i **/etc/passwd**. Podaci o svim korisnicima u sustavu pohranjeni su u datoteci **/etc/passwd**. Svaka linija u datoteci daje podatke o jednom korisničkom računu, a graničnik između zapisa je dvotočka. Struktura svake linije je:

- korisničko ime (*login*)
- lozinka (u kriptiranom obliku)
- identifikacijski broj korisnika (UID)
- identifikacijski broj matične grupe (GID)
- dodatni podatci o korisniku (pet potpolja)
    - puno ime i prezime
    - broj sobe
    - telefonski broj
    - drugi telefonski broj
    - dodatna napomena
- matični direktorij korisnika
- matična ljuska korisnika.

Primjer linije za korisnika **ivanhorvat**:

```
ivanhorvat:x:1004:1004:IvanHorvat,1,111111,22222,Znanstvenik:/home/ivanhorvat:/bin/bash
```

U polju lozinke nalazi se kriptirana lozinka korisnika ili u specijalnom slučaju znak “x“. Taj znak označava da lozinka nije pohranjena u datoteci **passwd**, nego u datoteci **shadow**. Drugo polje je x te se kriptirana lozinka tog korisnika nalazi u datoteci **/etc/shadow**.

Dobra je politika na poslužiteljima koji imaju veliki broj korisnika pohranjivati njihove lozinke u datoteci na koju je postavljeno ograničenje pristupa na 600 ili čak 400 kako nitko osim administratorskog računa ne bi imao pristup datoteci. Hoće li se rabiti datoteka **shadow** ili **passwd,** manje je važno.

U datoteku **/etc/shadow** sustav pohranjuje podatke o valjanosti lozinke korisnika. Datoteka se sastoji od niza linija gdje jedna linija opisuje jednog korisnika pomoću niza polja odvojenih znakom : (dvotočka).

- korisničko ime (*login*)
- lozinka (u kriptiranom obliku)
- datum zadnje izmjene lozinke
- *minimum* - interval dana između dopuštene izmjene lozinke
- *maximum* - interval dana koliko je lozinka valjana (nakon isteka lozinku treba promijeniti)
- *warn* – interval koliko prije sustav upozorava korisnika da mora promijeniti lozinku
- *inactive* – interval koliko dugo korisnički račun ostaje aktivan nakon isteka maksimalnog razdoblja
- *expire* – datum isteka korisničkog računa.

Primjer linije u datoteci **/etc/passwd** za korisnika **ivanhorvat** koja sadrži kriptiranu lozinku:

```
ivanhorvat:$6$gp0mHdDv$BiuDEBrloXGiARMH0XL3p0beEOHYhZwGtSsmIqovm4XAG/nm1Sz8E4On/
V3KxWdmx.9n9t98y/qGI2Q97sEOk.:16498:0:99999:7:::
```

Naredbe se **pwconv** i **pwunconv** koriste za prebacivanje kriptiranih lozinki iz datoteke **passwd** u **shadow** i iz datoteke **shadow** u **passwd**.

Dakle:

```
#/usr/sbin/pwconv
```

prebacuje kriptirane lozinke iz datoteke **passwd** u datoteku **shadow**, a

```
#/usr/sbin/pwunconv
```

prebacuje kriptirane lozinke iz datoteke **shadow** u datoteku **passwd**.

3.2. Upravljanje grupama

Linux koristi korisničke grupe kao način grupiranja korisnika. Tako je omogućeno jednostavno upravljanje pravima pristupa grupa korisnika. Pomoću korisničkih grupa pojedinom se korisniku jednom naredbom (pridruživanjem odgovarajućoj grupi) može dodijeliti niz prava ili se nizu korisnika može omogućiti pristup do pojedine datoteke (mijenjajući vlasništvo, odnosno prava pristupa nad datotekom).

3.2.1. Naredbe groupadd i groupdel

Identifikacijski broj grupe (*group identifier,* kraće GID) je numerička reprezentacija grupe. GID može biti broj između 0 i 32767, s tim da je GID administratorskog računa 0.

Prilikom dodavanja novog korisnika njemu se dodjeljuje minimalno jedna grupa. Ta početno dodijeljena grupa naziva se **inicijalna** ili **primarna** grupa. Dvije su standardne konvencije dodijele **primarne** grupe:

- Prvi je pristup da se svi korisnici prvo smjeste u istu grupu ***users*** s GID-brojem 100 (*group id*, GID).
- Druga se konvencija naziva **shema korisničke privatne grupe** (*User Private Group scheme UPG*). Svakom se korisniku pri izradi korisničkog računa doda i grupa istog imena kao i korisnikov login i s vrijednosti GID-broja između 500 i 60000 (ako je dostupno taj GID će biti isti kao i korisnikov UID).

Korisnik može pripadati u više grupa, a pregled pripadnosti grupama korisnika dobije se naredbama **groups** ili **id**:

```
$ id
uid=1000(l102) gid=1000(l102)
groups=1000(l102),24(cdrom),25(floppy),29(audio),30(dip),
44(video),46(plugdev),105(scanner),108(bluetooth),110(netdev)
$ groups
l102 cdrom floppy audio dip video plugdev scanner bluetooth netdev
```

Kao što je vidljivo u primjeru naredbe **id,** *Debian* primjenjuje shemu **UPG**. Iako korisnik može pripadati i obično pripada brojnim grupama samo je jedna grupa u bilo kojem trenutku njegova primarna grupa. Primarna grupa određuje vlasništvo svih datoteka koje korisnik izrađuje. Na primjer, datoteke koje je izradio korisnik **l102** dodjeljuju se grupi **l102**.

```
$ touch test_grupa
$ ls test* -l
-rw-r--r-- 1 l102 l102 0 Jun  8 14:07 test_grupa
```

Naredba **newgrp** dodaje korisniku članstvo u novoj grupi i otvara novu korisničku sjednicu (*session*). Korisniku to postaje primarna grupa promijenjena samo u toj sjednici. Korisnik u toj sjednici može dodavati datoteke s drugačijim vlasništvom:

```
$ touch test-grupa1
$ newgrp audio
$ touch test-grupa2
$ ls test* -l
-rw-r--r-- 1 l102 l102 0 Jun  8 14:15 test-grupa1
-rw-r--r-- 1 l102 audio    0 Jun  8 14:15 test-grupa2
```

Naredba **groupadd** rabi se za izradu nove grupe. Naredba prihvaća dodatne parametre poput

GID, lozinke (kriptirane) ili direktorija koji će se rabiti kao "/" (CHROOT_DIR). CHROOT_DIR je direktorij koji će korisnici članovi grupe vidjeti kao ishodišni direktorij, odnosno korisnici će vidjeti samo taj direktorij i njegove poddirektorije.

Naredba **addgroup** rabi se za dodavanje korisnika u postojeću grupu (tada se rabi s dvije opcije) ili za dodavanje novih grupa (tada se rabi sa samo jednom opcijom).

```
# addgroup l102 root
Adding user `l102' to group `root' ...
Adding user l102 to group root
Done.
# addgroup korisnici
Adding group `korisnici' (GID 1002) ...
Done.
```

Naredba **groupdel** briše postojeću grupu. Važno je naglasiti da nije moguće obrisati grupu koja je **primarna** grupa bilo kojem korisniku. Potrebno je ili najprije obrisati korisnika ili (trajno) promijeniti **primarnu** korisnikovu grupu. U slučaju uspješnog izvršavanja naredba ne vraća nikakvu poruku. U slučaju pogreške pri izvršavanju naredbe javlja se ova pogreška:

```
# groupdel root
groupdel: cannot remove the primary group of user 'root'
```

3.2.2. Konfiguracijske datoteke grupa

Datoteke **/etc/group** i **/etc/gshadow** su datoteke u kojima se nalaze informacije o grupama. Te su datoteke ekvivalent datotekama **/etc/passwd** i **/etc/shadow** koje sadrže podatke o korisnicima.

Datoteka **/etc/group** sastoji se od niza linija koje opisuju pojedine grupe. Svaka grupa opisana je nizom od četiri polja koja su odvojena graničnikom ":".

1. ime grupe
2. kriptirana lozinka (x označava da lozinka nije ovdje)
3. GID
4. popis svih članova grupe (odvojena zarezom)

Primjer linije iz **/etc/group**:

```
audio:x:29:pulse,l102
```

Datoteka **/etc/gshadow** koristi se vrlo rijetko jer je sigurnosno loša praksa postavljati i koristiti se lozinkama nad grupama. Datoteka se također sastoji od četiri polja koja su odvojena graničnikom ":".

1. ime grupe
2. kriptirana lozinka (***** označava da lozinka nije ovdje, **!** označava da se lozinka ne koristi)
3. popis administratora grupe

4. popis svih članova grupe (odvojeni zarezom).

Naredba **grpconv** izrađuje datoteku **/etc/gshadow**, a naredba **grpunconv** je briše.

Primjer linije iz **/etc/gshadow**:

```
scanner:!::saned,l102
```

3.3. Izmjene postavki korisničkih računa

3.3.1. Naredbe usermod, groupmod i chage

Naredba za upravljanje postavkama korisničkog računa je **usermod**. Tom se naredbom mogu promijeniti sve postavke postavljene tijekom izrade korisničkog računa. Naredbom **usermod** može se koristiti samo korisnik *root* za promjenu korisničkih računa.

Opcije naredbe **usermod** su:

Opcija	Objašnjenje
-d	postavljanje korisničkog ishodišnog direktorija (*home*)
-g	promjena primarne korisničke skupine
-l	promjena korisničkog logina
-u	promjena UID-a
-s	promjena korisničke ljuske
-G	promjena kojim grupama korisnik pripada
-a	rabi se s -G za dodavanje grupa kojima korisnik pripada

Važno je naglasiti da su promjene napravljene naredbom **usermod** trajne. Dakle, naredbom **usermod** mijenja se sadržaj datoteka kao što su **/etc/passwd** i **/etc/groups,** a za razliku od naredbe **newgrp,** izlazak iz ljuske ne utječe na stanje.

Naredba za upravljanje postavkama grupe je **groupmod**. Opcije naredbe **groupmod** su:

Opcija	Objašnjenje
-g	promjena GID-a
-n	preimenovanje grupe
-R	primjena nad drugim virtualnim direktorijem "/" (CHROOT_DIR)

Naredba kojom se administratori koriste za upravljanje postavkama lozinke je **chage**. Opcije naredbe **chage** su:

Opcija	Objašnjenje
-E	odabir datuma isteka lozinke
-I	definiranje broja dana između prelaska lozinke u neaktivnu i zaključavanja računa
-l	ispis podataka o istjecanju valjanosti lozinke
-m	definiranje minimalnog broja dana između promjena lozinke
-M	definiranje maksimalnog broja dana između promjena lozinke
-W	definiranje broja dana koliko se prije korisniku šalje poruka da će izmjena lozinke biti potrebna

Naredbom **chage** administrator može mijenjati sve postavke koje se ispisuju pomoću opcije za ispis **-l** :

```
# chage -l l102
Last password change : Mar 24, 2015
Password expires : never
Password inactive : never
Account expires : never
Minimum number of days between password change : 0
Maximum number of days between password change : 99999
Number of days of warning before password expires : 7
```

Primjer definiranja novog maksimalnog broja dana između izmjena lozinke**:**

```
# chage -M 999 l102
# chage -l l102
Last password change : Mar 24, 2015
Password expires : never
Password inactive : never
Account expires : never
Minimum number of days between password change : 0
Maximum number of days between password change : 999
Number of days of warning before password expires : 7
```

3.4. Vježba: Upravljanje korisnicima i grupama

1. Prijavite se na računalo kao korisnik l102. U GUI-u pokrenite **Terminal** (*Activities* → **Terminal**).
2. Naredbom `su -` postanite **root** korisnik (lozinka: L102).
3. Napravite sigurnosnu kopiju (*backup*) datoteka **shadow**, **passwd** i **group** tako da ih kopirate u direktorij **/tmp/**.

   ```
   # cp /etc/shadow /etc/passwd /etc/group /tmp/
   ```

4. Naredbom `useradd` izradite korisnika sistem001.

---

5. Naredbom `adduser` izradite korisnika **ivanhorvat** (ispunite sva polja proizvoljnim zapisima), obavezno zapišite lozinku dodijeljenu korisniku **ivanhorvat**.

   Lozinka: _____

6. Usporedite naredbom `diff` postojeće datoteke **shadow**, **passwd** i **group** u direktoriju **/etc/** i prijašnje kopije u direktoriju **/tmp/**.

   ```
   # diff /etc/shadow /tmp/shadow
   # diff /etc/passwd /tmp/passwd
   # diff /etc/group /tmp/group
   ```

   U čemu su razlike između tih datoteka?

---

7. Izvedite naredbu `pwconv` i provjerite koje su se promjene dogodile u datotekama **shadow** i **passwd**.

   ```
   # diff /etc/shadow /tmp/shadow
   # diff /etc/passwd /tmp/passwd
   ```

   Je li došlo do promjene?

---

8. Izvedite naredbu `pwunconv` i provjerite koje su se promjene dogodile u datotekama **shadow** i **passwd**.

   ```
   # diff /etc/shadow /tmp/shadow
   # diff /etc/passwd /tmp/passwd
   ```

   Je li došlo do promjene?

---

9. Kao običan korisnik (ne kao administrator) prebacite se u korisnički račun **ivanhorvat** (`su - ivanhorvat`). Zabilježite trenutačni direktorij.

10. Kao administrator definirajte lozinku za korisnika **sistem001**.

11. Kao običan korisnik (ne kao administrator) prebacite se u korisnički račun **sistem001** (`su - sistem001`). Zabilježite trenutačni direktorij.

    Zašto niste u direktoriju **/home/sistem001**?

12. Kao administrator uredite datoteku **/etc/passwd** tako da je korisniku **sistem001** *home-direktorij* **/tmp**/ i pokušajte se ponovno prebaciti u korisnika **sistem001** da potvrdite da sve radi.

13. Ponovite 12. zadatak, ali postavite **/home/sistem001** za home-direktorij i prebacite se u njega.

    Zašto ovog puta niste u direktoriju /home/sistem001/?

14. Kojim grupama pripada korisnik sistem001?

    ```
    # id sistem001
    ```

15. Izradite nove grupe **grupa001** i **grupa002**. Koji su GID-ovi tih grupa

    ```
    # addgroup grupa001
    # addgroup grupa002
    ```

16. Dodijelite korisniku **sistem001** skupinu **grupa001**.

    ```
    # usermod -a -G grupa001 sistem001
    # id sistem001
    ```

17. Postavite da je korisniku **sistem001** primarna skupina **grupa002**.

    ```
    # usermod -g grupa002 sistem001
    ```

18. Izvršite naredbe za brisanje skupina **grupa001** i **grupa002**. Što se dogodilo?

Zašto?

19. Promijenite GID grupe **grupa002** tako da ga smanjite za 1. Ponovno napravite **grupa001**. Koji su GID-ovi tih dviju grupa?

Zašto?

```
# groupmod -g 1003 grupa002
# addgroup grupa001
```

3.4.1. Dodatna vježba: Napredno upravljanje korisničkim postavkama

1. Onemogućite prijavljivanje na sustav korisnika **ivanhorvat** promjenom njegove ljuske u **/bin/false**. Pokušajte se prijaviti kao korisnik **ivanhorvat**. Što se dogodilo?

2. Promijenite standardnu ljusku (naredba `useradd -D`) za nove korisnike napravljene naredbom useradd u **/bin/zsh**. Izradite korisnika **zshkorisnik** naredbom `useradd` i postavite mu lozinku (**passwd**). Prebacite se s `su - zshkorisnik` u korisnika **zshkorisnik**. Što se dogodilo?

```
# useradd -D -s /bin/zsh
# useradd zshkorisnik
# su - zshkorisnik
```

3. Kao root izvršite `apt-get install -y zsh`.
4. Prebacite se s `su - zshkorisnik` u korisnika **zshkorisnik**. Izvedite naredbu `ps -p $$`. Zadnji zapis govori koja se ljuska rabi. Koja je to ljuska i zašto?

4. Upravljanje grupama i korisnicima

Trajanje poglavlja:
105 min

Po završetku ovoga poglavlja moći ćete:

- ispisati na ekran sve varijable postavljene u okolini **BASH**
- provjeriti i promijeniti vrijednost pojedinačne varijable
- promijeniti konfiguracijske datoteke koje definiraju vrijednosti varijabli pri prijavljivanju u sustav
- napraviti skriptu ljuske
- razumjeti značenje zaglavlja skripti ljuske
- izraditi logičke izraze u skriptama **BASH**
- izraditi složene logičke izraze povezivanjem jednostavnih logičkih izraza
- ostvariti grananje pomoću naredbe **if**
- napisati petlje **for**, **while** i **until**
- primijeniti naredbe za grananje s više mogućnosti **case** i **select**
- prihvatiti korisnički unos naredbom **read** i primijeniti ga u skripti **BASH**
- koristiti se binarnim operatorima za manipulaciju brojevnim varijablama
- koristiti se logičkim operatorima za usporedbu numeričkih tipova ili za usporedbu nizova.

Ova cjelina obrađuje skripte BASH. U cjelini je dan uvid u izradu skripti BASH, grananje, petlje, logičke izraze, numeričke izraze te prihvaćanje i obradu korisničkog unosa.

4.1. Okolina BASH

4.1.1. Varijable okoline

U prethodnom poglavlju prikazane su korisničke postavke i jedna od njih je izbor korisničke ljuske odnosno okoline. Standardna korisnička okolina u *Linux*-u je **BASH**. **BASH** je *Unix*-ova ljuska i skriptni jezik autora Braina Foxa napisan za projekt GNU kao alternativa otvorenom softveru postojeće ljuske **Bourne**. Postoje brojne okoline, a najviše su u uporabi:

Ljuska	Opis
bash	*Bourne again shell* - standardna okolina u većini *Linux*ovih distribucija
zsh	Najbogatija mogućnostima, ali još relativno rijetko korištena
ksh	Standardna ljuska za *Solaris* i *AIX*
tcsh	Standardna ljuska za *BSD distribucije
sh	Originalna ljuska *Bourne*, zastarjela; zamijenio ju *bash*
csh	Originalna ljuska *C*, zastarjela; zamijenili su je *tcsh* i *ksh*

Svaka je okolina malo drugačija i radi malo drugačije u nekim segmentima. Vrijedi pravilo da je složenija i mogućnostima bogatija ljuska (na primjer **zsh**) sporija i zahtjevnija prema resursima.

Tako se na primjer **sh,** odnosno ljuska **bash** i danas rabi kao ljuska za pokretanje jezgre, a napredna svojstva ljuske **bash** (mnoga će biti objašnjena u ovoj lekciji) nisu potrebna u postupku pokretanja jezgre.

Varijable okoline ili globalne varijable dostupne su svakoj okolini odnosno ljusci, a lokalne varijable dostupne samo trenutačnoj. Globalne i lokalne varijable na brojne načine utječu na ponašanje sustava pri pozivima određenih naredbi. Najočitiji je primjer varijabla **PATH** koja određuje u kojim se direktorijima nalaze izvršne datoteke koje će korisnik pozivati. Na primjer, ako promijenimo vrijednost varijable **PATH**:

```
# echo $PATH
/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin
# ls
Desktop    Downloads      linux-3.10.79.tar       Music     Public     test_grupa   test-grupa2
Documents  linux-3.10.79  linux-3.10.79.tar.sign  Pictures  Templates  test-grupa1  Videos
# mv /bin/ls /home/l102/
# ls
bash: /bin/ls: No such file or directory
# /home/l102/ls
Desktop    Downloads      linux-3.10.79.tar       ls Pictures  Templates   test-grupa1  Videos
Documents  linux-3.10.79  linux-3.10.79.tar.sign  Music  Public    test_grupa  test-grupa2
```

Dakle varijabla **PATH** govori okolini gdje da traži izvršne datoteke. Bez nje ili s pogrešno postavljenom varijablom **PATH** morali bismo za sve izvršne datoteke (poput **ls, cd, mkdir, mv** i slično) znati rabiti potpunu putanju.

Do vrijednosti varijable se, kao što je vidljivo u gornjem primjeru, dolazi pomoću posebnog znaka $ ispred imena varijable:

```
# echo PATH
PATH
# echo $PATH
/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin
```

Naredbe za ispis varijabli okoline su **env** ili **printenv**. Naredba za ispis lokalnih i globalnih te za mijenjanje lokalnih varijabli je **set**. Naredba **set** bez parametara prikazuje sve varijable, uključujući i lokalne. Na primjeru ćemo pogledati sve lokalne i globalne varijable:

1 - Globalne varijable

```
# env
SSH_AGENT_PID=3012
DM_CONTROL=/var/run/xdmctl
GPG_AGENT_INFO=/home/l102/.cache/keyring-h1kQH9/gpg:0:1
SHELL=/bin/bash
TERM=xterm
XDG_SESSION_COOKIE=526f3f2a84dd299e75a91eb155116ad3-1433765543.585302-962436292
XDM_MANAGED=method=classic
GJS_DEBUG_OUTPUT=stderr
WINDOWID=52428805
```

```
GNOME_KEYRING_CONTROL=/home/l102/.cache/keyring-h1kQH9
GJS_DEBUG_TOPICS=JS ERROR;JS LOG
USER=root
LS_COLORS=rs=0:di=01;34:ln=01;36:mh=00:pi=40;33:so=01;35:do=01;35:bd=40;33;01:cd
=40;33;01:or=40;31;01:su=37;41:sg=30;43:ca=30;41:tw=30;42:ow=34;42:st=37;44:ex=0
1;32:*.tar=01;31:*.tgz=01;31:*.arj=01;31:*.taz=01;31:*.lzh=01;31:*.lzma=01;31:*.
tlz=01;31:*.txz=01;31:*.zip=01;31:*.z=01;31:*.Z=01;31:*.dz=01;31:*.gz=01;31:*.lz
=01;31:*.xz=01;31:*.bz2=01;31:*.bz=01;31:*.tbz=01;31:*.tbz2=01;31:*.tz=01;31:*.d
eb=01;31:*.rpm=01;31:*.jar=01;31:*.war=01;31:*.ear=01;31:*.sar=01;31:*.rar=01;31
:*.ace=01;31:*.zoo=01;31:*.cpio=01;31:*.7z=01;31:*.rz=01;31:*.jpg=01;35:*.jpeg=0
1;35:*.gif=01;35:*.bmp=01;35:*.pbm=01;35:*.pgm=01;35:*.ppm=01;35:*.tga=01;35:*.x
bm=01;35:*.xpm=01;35:*.tif=01;35:*.tiff=01;35:*.png=01;35:*.svg=01;35:*.svgz=01;
35:*.mng=01;35:*.pcx=01;35:*.mov=01;35:*.mpg=01;35:*.mpeg=01;35:*.m2v=01;35:*.mk
v=01;35:*.webm=01;35:*.ogm=01;35:*.mp4=01;35:*.m4v=01;35:*.mp4v=01;35:*.vob=01;3
5:*.qt=01;35:*.nuv=01;35:*.wmv=01;35:*.asf=01;35:*.rm=01;35:*.rmvb=01;35:*.flc=0
1;35:*.avi=01;35:*.fli=01;35:*.flv=01;35:*.gl=01;35:*.dl=01;35:*.xcf=01;35:*.xwd
=01;35:*.yuv=01;35:*.cgm=01;35:*.emf=01;35:*.axv=01;35:*.anx=01;35:*.ogv=01;35:*
.ogx=01;35:*.aac=00;36:*.au=00;36:*.flac=00;36:*.mid=00;36:*.midi=00;36:*.mka=00
;36:*.mp3=00;36:*.mpc=00;36:*.ogg=00;36:*.ra=00;36:*.wav=00;36:*.axa=00;36:*.oga
=00;36:*.spx=00;36:*.xspf=00;36:
SSH_AUTH_SOCK=/home/l102/.cache/keyring-h1kQH9/ssh
SESSION_MANAGER=local/debian-1:@/tmp/.ICE-unix/2852,unix/debian-1:/tmp/.ICE-
unix/2852
MAIL=/var/mail/root
DESKTOP_SESSION=gnome
PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin
PWD=/home/l102
LANG=en_US.UTF-8
HOME=/root
SHLVL=5
LANGUAGE=en_US:en
GNOME_DESKTOP_SESSION_ID=this-is-deprecated
LS_OPTIONS=--color=auto
LOGNAME=root
XDG_DATA_DIRS=/usr/share/gnome:/usr/local/share/:/usr/share/
WINDOWPATH=7
DISPLAY=:0
COLORTERM=gnome-terminal
XAUTHORITY=/tmp/libgksu-PtVlyX/.Xauthority
```

U sljedećem se primjeru prvo pribave sve varijable i globalne varijable te se pohrane vrijednosti u datoteke **test_env** i **test_set**. Nakon toga se zapisi u datotekama sortiraju i pohrane u datoteke **test_env_sorted** i **test_set_sorted**. Razlika je između tih dviju datoteka upravo u lokalnim varijablama (sve varijable - globalne varijable = lokalne varijable).

```
# env > test_env
# set > test_set
# sort test_env > test_env_sorted
# sort test_set > test_set_sorted
# diff test_set_sorted test_env_sorted | grep "<" | awk '{ print $2 }'

BASH_ALIASES=()
```

```
BASH_ARGC=()
BASH_ARGV=()
BASH=/bin/bash
BASH_CMDS=()
BASH_LINENO=()
BASHOPTS=checkwinsize:cmdhist:expand_aliases:extquote:force_fignore:hostcomplete
:interactive_comments:progcomp:promptvars:sourcepath
BASH_SOURCE=()
BASH_VERSINFO=([0]="4"
BASH_VERSION='4.2.37(1)-release'
COLUMNS=140
DIRSTACK=()
_=env
EUID=0
GJS_DEBUG_TOPICS='JS
GROUPS=()
HISTFILE=/root/.bash_history
HISTFILESIZE=500
HISTSIZE=500
HOSTNAME=debian-1
HOSTTYPE=i486
IFS=$'
LINES=46
LS_COLORS='rs=0:di=01;34:ln=01;36:mh=00:pi=40;33:so=01;35:do=01;35:bd=40;33;01:c
d=40;33;01:or=40;31;01:su=37;41:sg=30;43:ca=30;41:tw=30;42:ow=34;42:st=37;44:ex=
01;32:*.tar=01;31:*.tgz=01;31:*.arj=01;31:*.taz=01;31:*.lzh=01;31:*.lzma=01;31:*
.tlz=01;31:*.txz=01;31:*.zip=01;31:*.z=01;31:*.Z=01;31:*.dz=01;31:*.gz=01;31:*.l
z=01;31:*.xz=01;31:*.bz2=01;31:*.bz=01;31:*.tbz=01;31:*.tbz2=01;31:*.tz=01;31:*.
deb=01;31:*.rpm=01;31:*.jar=01;31:*.war=01;31:*.ear=01;31:*.sar=01;31:*.rar=01;3
1:*.ace=01;31:*.zoo=01;31:*.cpio=01;31:*.7z=01;31:*.rz=01;31:*.jpg=01;35:*.jpeg=
01;35:*.gif=01;35:*.bmp=01;35:*.pbm=01;35:*.pgm=01;35:*.ppm=01;35:*.tga=01;35:*.
xbm=01;35:*.xpm=01;35:*.tif=01;35:*.tiff=01;35:*.png=01;35:*.svg=01;35:*.svgz=01
;35:*.mng=01;35:*.pcx=01;35:*.mov=01;35:*.mpg=01;35:*.mpeg=01;35:*.m2v=01;35:*.m
kv=01;35:*.webm=01;35:*.ogm=01;35:*.mp4=01;35:*.m4v=01;35:*.mp4v=01;35:*.vob=01;
35:*.qt=01;35:*.nuv=01;35:*.wmv=01;35:*.asf=01;35:*.rm=01;35:*.rmvb=01;35:*.flc=
01;35:*.avi=01;35:*.fli=01;35:*.flv=01;35:*.gl=01;35:*.dl=01;35:*.xcf=01;35:*.xw
d=01;35:*.yuv=01;35:*.cgm=01;35:*.emf=01;35:*.axv=01;35:*.anx=01;35:*.ogv=01;35:
*.ogx=01;35:*.aac=00;36:*.au=00;36:*.flac=00;36:*.mid=00;36:*.midi=00;36:*.mka=0
0;36:*.mp3=00;36:*.mpc=00;36:*.ogg=00;36:*.ra=00;36:*.wav=00;36:*.axa=00;36:*.og
a=00;36:*.spx=00;36:*.xspf=00;36:'
MACHTYPE=i486-pc-linux-gnu
MAILCHECK=60
OPTERR=1
OPTIND=1
OSTYPE=linux-gnu
PIPESTATUS=([0]="0")
PPID=8979
PS1='${debian_chroot:+($debian_chroot)}\u@\h:\w\$
PS2='>
PS4='+
SHELLOPTS=braceexpand:emacs:hashall:histexpand:history:interactive-
comments:monitor
UID=0
```

4.1.2. Postavljanje ili izmjena vrijednosti varijable

Dva su načina za postavljanje ili promjenu vrijednosti varijable.

- Prvi je način pomoću operatora "=" s **VARIJABLA=VRIJEDNOST**. Tako se može promijeniti vrijednost svih varijabli.
- Drugi je način pomoću naredbe **set** - `# set VARIJABLA=VRIJEDNOST`. Tako se mijenjaju vrijednosti lokalnih varijabli ili se lokalno mijenja vrijednost globalnih varijabli.

Te promjene (na bilo koji od dva načina) neće biti vidljive u cijelom sustavu (drugim korisnicima, u drugim korisničkim sjednicama istog korisnika itd.). Naredba za objavu vrijednosti varijable je **export.** Nakon što se postavi vrijednost varijable i izvrši `# export VARIJABLA` nova vrijednost varijable bit će vidljiva svim procesima nastalim kao procesi "djeca" (*child process*) procesa u kojem je vrijednost postavljena.

Da bi se varijabla obrisala rabi se naredba **unset**:

```
# testvar="imam_vrijednost"
# set |grep testvar
testvar=imam_vrijednost
# echo $testvar
imam_vrijednost
# unset testvar
# echo $testvar

#
```

4.1.3. Konfiguracijske datoteke

Konfiguracijske datoteke ljuske **BASH** su:

~/.bash_aliases
~/.bash_login
~/.bash_logout
~/.bash_profile
~/.bashrc
/etc/bash.bashrc
/etc/profile
~/.profile

Iz kojih će konfiguracijskih datoteka biti učitane vrijednosti varijabli ovisi o načinu rada okoline BASH.

Postoje tri osnovna načina rada:

1. Interaktivna okolina nastala po uspješnom prijavljivanju u sustav (*login*) naziva se **interaktivna login okolina**. Učitavaju se vrijednosti i izvršavaju naredbe iz datoteke **/etc/profile** i zatim prve pronađene naredbe datoteka **~/.bash_profile**, **~/.bash_login** i

~/.profile . Pri izlasku iz okoline (*logout*) izvršava se datoteka **~/.bash_logout**.

2. Interaktivna okolina koja nije nastala po uspješnom prijavljivanju u sustav (*login*) naziva se **interaktivna non-login okolina**. Učitavaju se vrijednosti i izvršavaju naredbe iz datoteke **~/.bashrc**.

3. Kod **neinteraktivnih okolina** poput na primjer poziva skripti, sustav čita vrijednost varijable **BASH_ENV**. U tu varijablu treba smjestiti ime datoteke koja sadrži željenu konfiguraciju.

Dodatno učitavanje datoteka kao što je **~/.bash_aliases** standardno poziva datoteka **~./bashrc** , a datoteka **/etc/bash.bashrc** je standardno vrijednost varijable **BASH_ENV**.

Datoteke iz **/etc/profile.d** učitaju se samo kada je učitana datoteka **/etc/profile** jer je u njoj definirano pravilo koje čita sadržaj tog direktorija.

4.2. Osnove rada sa skriptama

4.2.1. Pokretanje, unos parametara iz naredbene linije i specijalne varijable

Skripte ljuske su (najčešće mali) programi koji se rabe za automatizaciju složenih akcija. Svaka linija skripte ljuske može zamjenjivati jedan poziv na izvršavanje naredbene linije. Tako već i skripta od dvadesetak linija koja izvršava često korištene akcije može biti velika pomoć administratoru. Svi primjeri u ovom poglavlju mogu se izravno izvršiti u naredbenoj liniji, ali skripte ljuske omogućavaju jednostavno ponavljanje poziva i jednostavno prilagođavanje pri promjenama u sustavu.

Da bi datoteka bila skripta ljuske potrebne su samo dvije stvari:

1. prva linija u datoteci mora biti: **#!/bin/<ljuska>** (gdje je ljuska **bash** ili neka druga, ako nije **bash**)
2. datoteka mora biti izvršna (na primjer **755** dozvole nad datotekom).

Kombinacija specijalnih znakova "**#!**" (referenca je *she-bang*) zadaje interpreter koji se treba rabiti za tumačenje naredbi u datoteci. Tako **#!/bin/zsh** na početku datoteke definira ljusku **zsh** kao interpretera za daljnje linije u datoteci.

Pet je mogućih načina **pokretanja** izvršavanja skripte ljuske. Objasnimo ih na primjeru skripte "**Skripta_1**":

Pokretanje	Objašnjenje
./Skripta_1	Standardno pokretanje.
<ljuska> Skripta_1	Pokreće novu interaktivnu ljusku <ljuska> (bash ili zsh ili sh) koja će izvršiti skriptu i zatim izaći.
source Skripta_1	Pokreće se skripta sa trenutačnom ljuskom kao interpreterom.
. Skripta_1	Pokreće se skripta sa trenutačnom ljuskom kao interpreterom.

exec ./Skripta_1	Sve isto kao kod standardnog pokretanja samo će po izvršavanju trenutačna ljuska izaći.

Nakon bilo kojeg od gore navedenih poziva mogu u nastavku linije biti parametri kojima će se skripta koristiti pri izvršavanju. Na primjer:

```
./Skripta_1 parametar_a parametar_b parametar_c
```

U naredbenoj se liniji zadani parametri mogu pozvati pomoću **specijalnih varijabli** koje se mogu rabiti u svakoj skripti:

Specijalna varijabla	**Objašnjenje**	**Vrijednost u gornjem primjeru**
$*	Popis svih varijabli u naredbenoj liniji.	parametar_a parametar_b parametar_c
$#	Broj varijabli u naredbenoj liniji.	3
$0, $1, $2 $3... $n	Slijedno svi parametri naredbene linije.	#0=./Skripta_1 #1=parametar_a #2=parametar_b itd...
$!	PID posljednjeg pozadinskog procesa.	9792
$$	PID trenutne ljuske.	11211
$?	Izlaz iz zadnje naredbe.	0 (znači uspješno izvršeno)

Za parametre u naredbenoj liniji postoji posebna naredba **shift** koja smanjuje redni broj svih parametara za jedan. Tako $2 postane $1, $3 postane $2 itd. Odnosno #(n)-> #(n-1). Važno je naglasiti da naredba **shift** ne mijenja vrijednost varijable **$0**, nego samo varijabli od **$1** nadalje.

4.3. Logičko grananje

4.3.1. Operatori logičkog grananja

Logička evaluacija u kôdu skripti **BASH** provodi se pomoću operatora **test** ili pomoću uglatih zagrada **[** i **]**.

Sintaksa je:

test <logički_izraz>
ili
[<logički_izraz>**]**

Rezultat se pohranjuje u varijablu **$?**. Vrijednost **0** će biti pohranjena ako je izraz istinit, a neka druga vrijednost ako je izraz lažan. Važno je napomenuti (kako je ranije objašnjeno) da varijabla **$?** sadrži rezultat odnosno izlaz zadnje naredbe. Dakle prva će iduća naredba koja vraća

vrijednost **promijeniti** vrijednost varijable **$?**. Zbog toga se logičko grananje rabi češće izravno u kodu, a ne rabi se vrijednost varijable **$?**.

Na primjer, grananje na osnovi uvjeta postoji li u sustavu datoteka **/bin/bash** standardno će se napraviti ovim izrazom:

```
if [ -f /bin/bash ] ; then
```

a neće biti korištena varijabla **#?**:

```
test -f /bin/bash
if $? ; then
```

Moguće je formirati složene logičke izraze povezivanjem jednostavnih logičkih uvjeta pomoću logičkih operatora ili (**&&)** i **(||)**.

Na primjer, možemo provjeriti postoji li datoteka **/bin/bash** i je li datoteka **/bin/sh** izvršna.

```
test -f /bin/bash && test -x /bin/sh
[ -f /bin/bash ] && [ -x /bin/sh ]
```

Zamijenimo li operator && (i) operatorom || (ili) provjerit ćemo postoji li datoteka **/bin/bash** ili je li datoteka **/bin/sh** izvršna.

Alternativa operatorima **&&** (i) i **||** (ili) su **-a** i **-o.**

Dakle gornje upite možemo provesti i ovako:

```
test -f /bin/bash -a test -x /bin/sh
[ -f /bin/bash ] -a [ -x /bin/sh ]
```

Potrebno je naglasiti važnost točne sintakse u pisanju skripti **BASH**. Sintaksa je naravno važna ovdje kao i u svakom drugom programskom jeziku jer su računala strojevi i zbog toga po definiciji nisu fleksibilna. Interpretori kôda (uključujući **BASH**) razumiju naredbe samo ako su napisane točno u onom obliku u kojem ih očekuju. Stoga je u gornjim primjerima svaki razmak važan. Moguće je na mjestima (katkad) gdje je potreban razmak imati više razmaka, ali ako razmaka nema nastaje pogreška. Na primjer, izvršavanje skripte s linijom u kojoj nedostaje prvi razmak nakon uglate zagrade **[**

```
[-f /bin/bashsd -o -f /bin/sh ]
```

javlja pogrešku:

```
./Skripta_1.sh: line 7: [-f: command not found
```

4.4. Upravljanje tijekom i petlje

4.4.1. Grananje

Grananje se u skriptama **BASH** ostvaruje naredbom **if**.

Sintaksa je:

```
if logički_izraz ; then
               Blok_naredbi_1
            else
               Blok_naredbi_2
 fi
```

Ako je **logički_izraz** istinit, izvršiti će se **Blok_naredbi_1**, ako nije (odnosno ako **logički_izraz** vrati vrijednost različitu od **0**) tada će se izvršiti **Blok_naredbi_2**. Uvijek će se izvršiti samo jedna grana naredbe **if**.

Moguće je naravno u blokovima naredbi imati ponovno grananje, na primjer:

```
if logički_izraz_1 ; then
               Blok_naredbi_1
            else
               if logički_izraz_2 ; then
                           Blok_naredbi_2
                        else
                           Blok_naredbi_3
               fi
fi
```

Dakle ako je **logički_izraz_1** istinit izvršava se **Blok_naredbi_1**; ako nije istinit, tada se, ako je **logički_izraz_2** istinit, izvršava **Blok_naredbi_2**. Ako su oba logička izraza neistinita, izvršava se **Blok_naredbi_3**.

4.4.2. Petlje

U skriptama **BASH** rabe se tri petlje **for**, while i **until**. Te su petlje primarno namijenjene za tri različite namjene:

Petlja	Namjena
for	Ponavljanje unaprijed određeni broj puta.
while	Ponavljanje dok je uvjet istinit.
until	Ponavljanje do trenutka kada uvjet postane istinit.

Mada je intuitivnije koristiti se petljama za njihove prvotne namjene, na primjerima ćemo pokazati da se sve tri petlje mogu rabiti za istu namjenu. Sintakse su:

```
(1)
for Varijabla_uvjeta in set_vrijednosti_varijable;
do
          Blok_naredbi
done

(2)
for ((inicijalizacija;logički_izraz;inkrement)) ; do
        Blok_naredbi
done
while logički_izraz ; do
         Blok_naredbi
done
until logički_izraz ; do
         Blok_naredbi
done
```

(1) Kod petlje **for** pri svakom se izvršavanju varijabli "**Varijabla_uvjeta**" pridružuje vrijednost slijedom jednog elementa iz skupa vrijednosti "**set_vrijednosti_varijable**". Ta se varijabla sa svojom vrijednosti može rabiti u "**Bloku naredbi"**, a izvršavanje petlje završava kada je varijabli dodijeljena zadnja vrijednost iz skupa vrijednosti.

(2) Petlja **for** podržava i drugu sintaksu, identičnu onoj petlje **for** kod programskog jezika C. Prije prvog izvršavanja petlje izvrši se inicijalni korak (inicijalizacija). Nakon toga se provjerava vrijednost logičkog izraza i ako je on istinit, kreće se u izvršavanje bloka naredbi između **do** i **done**. Na kraju svakog izvršavanja bloka naredbi izvrši se naredba inkrement i ponovo se kreće od provjeravanja logičkog izraza.

U petlji **while** blok se naredbi između **do ;** i **done** izvršava dok je"**logički_izraz**" istinit, a u petlji **until** dok nije. Standardno se negdje u bloku naredbi mijenja vrijednost nekih od varijabli korištenih u logičkom izrazu. Ako u bloku naredbi nema naredbe koje može utjecati na evaluaciju istinitosti logičkog izraza, tada govorimo o beskonačnoj petlji.

Pokažimo primjer petlji koje na ekran ispisuju prirodne brojeve od 1 do 10.

```
for Brojac in 1 2 3 4 5 6 7 8 9 10 ; do
        echo $Brojac
done
```

```
for (( Brojac=1; Brojac <= 10; Brojac++))
do
        echo $Brojac
done
```

```
Brojac=1
while [ $Brojac -le 10 ]; do
        echo $Brojac
        let Brojac=Brojac+1
done
```

```
Brojac=1
until [ $Brojac -gt 10 ]; do
        echo $Brojac
        let Brojac=Brojac+1
done
```

Uočimo u ovim primjerima da je vrlo važno zbog preciznosti paziti na "rubne" uvjete - odnosno na to kojom se naredbom koristimo ("veće ili jednako" ili strogo jednako) i kojim brojem (10 ili 11), treba li početi od nule ili od jedinice i slično.

4.5. Prihvaćanje unosa korisnika

4.5.1. Naredbe case i select

Naredba `case` rabi se za ostvarivanje grananja. Grananje za razliku od binarnog grananja **if** može imati više grana, a odluka se provodi na osnovi jednog početnog izraza.

Sintaksa je:

```
case izraz in
        vrijednost_1) lista_naredbi_1;;
        vrijednost_2) lista_naredbi_2;;
        ...
        vrijednost_n) lista_naredbi_n;;
esac
```

Pri izvršavanju se provjerava je li **izraz** jednak pojedinim vrijednostima i ako je tada se izvršava odgovarajući blok naredbi.

Naredba **select** dodaje automatski izbornik za korisnika. Korisnik unosi redni broj elementa (od ponuđenih) i time poziva izvršavanje kôda.

Sintaksa je :

```
select Varijabla_Selecta in Lista_elemenata
do
Blok_naredbi
done
```

Po izvršavanju će program ponovno čekati unos korisnika (izbornik se ne prikazuje ponovo na ekranu) tako da je nužno pomoću naredbi **break** ili **exit** izaći iz okvira naredbe.

U primjeru skripta nudi izbornik s tri mogućnosti za zabavu: 1. Film 2. Kazalište i 3. Sport. Skripta prihvaća unos dok se ne odabere sport i tada javlja poruku o prihvaćanju odabrane mogućnosti za zabavu. Primjer:

```
# cat testna.sh
#!/bin/bash

select Zabava in Film Kazaliste Sport
do
   echo "izabrali ste $Zabava"
   if [ $Zabava = "Sport" ]; then
      break
   fi
done
echo "aaaa Sport pa što ne kaže"
```

Primjer izvršavanja:

```
# ./testna.sh
1) Film
2) Kazaliste
3) Sport
#? 1
izabrali ste Film
#? 2
izabrali ste Kazaliste
#? 1
izabrali ste Film
#? 2
izabrali ste Kazaliste
#? 3
izabrali ste Sport
aaaa Sport pa što ne kaže
```

Napomena

Naredba **break** izlazi iz trenutačnog bloka naredbi – često se rabi kod beskonačnih petlji za izlaz kada je neki uvjet ispunjen. Naredba **exit** izlazi iz cijelog programa odnosno zaustavlja izvršavanje skripte BASH.

4.5.2. Naredba read

Naredba **read** služi za učitavanje korisničkog unosa. Ona zaustavlja izvršavanje programa i iz naredbene linije čita korisnikov unos te ga zapisuje u zadani popis varijabli.

Sintaksa je:

```
read lista_varijabli
```

Naredba **read** bez opcija koristi se varijablom **REPLY** za pohranu unosa. Ta se naredba često kombinira s naredbom **case**.

U skripti u primjeru, pomoću triju mogućnosti korisnik može odabrati akciju. Prva akcija je prikaz trenutačnog vremena i datuma, druga je prikaz trenutačnog radnog direktorija, a treća je mogućnost ispis varijabli okoline. Primjer upotrebe:

```
#!/bin/bash

echo "Unesite što zelite vidjeti \"Datum\", \"Path\" ili \"Okolina\""
read AKCIJA
case $AKCIJA in
  Datum) echo `date` ;;
  Path) echo $PATH ;;
  Okolina) echo `env` ;;
esac
```

Primjer izvršavanja:

```
# ./testna.sh
Unesite što zelite vidjeti "Datum", "Path" ili "Okolinu"
Path
/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin
# ./testna.sh
Unesite što zelite vidjeti "Datum", "Path" ili "Varijable"
Datum
Fri Jun 11 13:52:52 CEST 2015
```

Važno je napomenuti da se izvršavanje prekida nakon što je odabrana neka mogućnost u naredbi **case**. Za ponovo izvršavanje opet treba pozvati izvršavanje skripte.

4.6. Rad s brojevnim tipovima

4.6.1. Binarni operatori (+, -, *, …)

Binarni i unarni operatori su:

Operator	Opis	Primjer	Rezultat primjera
+	Zbrajanje	echo $((20 + 5))	25
-	Oduzimanje	echo $((20 - 5))	15
/	Dijeljenje	echo $((20 / 5))	4
*	Množenje	echo $((20 * 5))	100
%	Modul (ostatak pri dijeljenju)	echo $((20 % 3))	2
++	Post-inkrement (povećava vrijednost varijabli za 1)	x=5 echo $((x++)) echo $((x++))	5 6
--	Post-dekrement (smanjuje vrijednost varijabli za 1)	x=5 echo $((x--))	4
**	Potencija (na - kao x na y-tu)	x=2 y=3 echo $((x ** y))	8

Prioritet binarnih operatora od najvišeg prema najnižem:

Prioritet	Operator i opis
1.	id++ id-- post-inkrement i post-dekrement
2.	++id --id pre-inkrement i pre-dekrement
3.	- + unarni minus i plus
4.	! ~ logička i negacija nad bitovima
5.	** potencija (na - kao x^y)
6.	* / % množenje, dijeljenje i modul
7.	+ - zbrajanje, oduzimanje
8.	<< >> pomak bit ulijevo/udesno
9.	<= >= < > usporedba
10.	usporedba
11.	== != jednakost i nejednakost
12.	& I nad bitovima
13.	^ ekskluzivni ili nad bitovima
14.	| ili nad bitovima
15.	&& logičko i
16.	|| logičko ili
17.	= *= /= %= += -= <<= >>= &= ^= |= operatori pridruživanja

Osnovne binarne numeričke operacije u ljusci BASH provode se pomoću naredbe **expr** ili unutar **$(())**. Važno je napomenuti da ta dva načina ne djeluju jednako. Primjeri razlike u izvršavanju u naredbenoj liniji:

```
# expr 7 + 3
10
# $((7 + 3))
bash: 10: command not found
```

Primjeri razlike u izvršavanju u skripti:

```
# cat binarni.sh
#!/bin/bash
c=$(( 7 + 3 ))
echo "c: $c"
d=expr 7 + 3
echo "d: $d"
e=`expr 7 + 3`
echo "e: $e"
#./binarni.sh
c: 10
./binarni.sh: line 6: 7: command not found
d:
e: 10
```

4.6.2. Operatori za usporedbu

Operatori za usporedbu najčešće se rabe u logičkim izrazima. U petljama (koje se ne izvršavaju nad unaprijed određenim setom vrijednosti) standardno se rabi usporedba varijable kojoj se mijenja vrijednost u petlji i neke konstante kao kriterij završetka izvršavanja petlje.
Operatori za usporedbu u ljusci **BASH** slijede neobičnu logiku. Naime za usporedbu nizova se koriste matematički simboli, a za usporedbu brojevnih tipova se koriste operatori sastavljeni od slova.

Operatori su:

Brojevni operator	Operator nad nizovima	Značenje
-lt	<	strogo manje
-gt	>	strogo veće
-le	Ne postoji operator	manje ili jednako
-ge	Ne postoji operator	veće ili jednako
-eq	==	jednako
-ne	!=	nejednako

Dobro je zapamtiti da su operatori za usporedbu nad brojevnim tipovima skraćenice engleskih izraza:

Brojevni operator	Engleski izraz
-lt	*less than*
-gt	*greater than*
-le	*less or equal*
-ge	*greater or equal*
-eq	*equal*
-ne	*not equal*

4.7. Vježba: Skripte BASH

1. Prijavite se na računalo kao korisnik l102. U GUI-u pokrenite **Terminal** (*Activities* → **Terminal**).
2. Naredbom `su -` postanite **root** korisnik (lozinka: L102).
3. Izradite direktorij **/home/l102/skriptanje** i uđite u njega.

4. Izradite datoteku **prihvat_unosa.sh** i promijenite je u izvršnu.
5. Uđite u datoteku i dodajte zaglavlje tako da datoteka postane skripta ljuske BASH.

   ```
   # vi prihvat_unosa.sh
   ```

   Dodati `#!/bin/bash` na početak datoteke.
6. Dodajte jednu petlju **for** koja se ponavlja 10 puta. Neka u petlji svaki korak poziva ispisuje na ekran trenutačni korak. Dakle, na ekranu neka bude ispisano:

```
Ovo je 1. korak
Ovo je 2. korak
…
Ovo je 10. korak
```

Testirajte izvođenje skripte.

7. Proširite petlju dodatnom naredbom za ispis koja na ekran ispisuje slijedno sve parametre zadane u naredbenoj liniji. Tako da sada ispis na primjer bude:

```
# ./prihvat_unosa.sh druga nije daleko
Ovo je 1. korak
Parametar je: druga
Ovo je 2. korak
Parametar je: nije
Ovo je 3. korak
3.    Parametar je: daleko
…
Ovo je 10. korak
```

Testirajte izvođenje skripte. Što je ispisano na ekran u zadnjim koracima kao vrijednost parametara naredbene linije (pretpostavka je da nije uneseno 10 parametara)?

8. Promijenite skriptu tako da se umjesto petlje **for** izvršava petlja **while** koja rezultira istim ispisom kao u 5. zadatku.

   Moguće rješenje:

9. Naredbu za ispis u petlji zamijenite (preciznije uokvirite) uvjetom if tako da se u slučaju nepostojanja parametra javlja poruka o tome. Dakle, za gornji primjer:

```
# ./prihvat_unosa.sh druga nije daleko
Ovo je 1. korak
1. Parametar je: drugi
Ovo je 2. korak
2. Parametar je: nije
Ovo je 3. korak
3.      Parametar je: daleko
Ovo je 4. korak
4.      Parametar nije zadan
…
Ovo je 10. korak
10. Parametar nije zadan
```

5. Osnovni koncepti računalnih mreža TCP/IP

Trajanje poglavlja:
85 min

Po završetku ovoga poglavlja moći ćete:

- definirati što su to **IP**, **IPv4** i **IPv6**
- definirati koje su valjane adrese u **IPv4**
- definirati mrežnu adresu, mrežnu masku i adresu razašiljanja
- izračunati mrežnu adresu, mrežnu masku i adresu razašiljanja
- definirati mrežne klase i besklasne mreže
- definirati i izračunati rezervirane adrese mreže
- definirati što je TCP/IP
- imenovati četiri sloja na kojima se provodi komunikacija TCP/IP
- imenovati ulogu i raspon privilegiranih portova.

Ova cjelina obrađuje skup protokola za adresiranje TCP/IP i IPv4. Naučiti ćemo adresne klase u IPv4, besklasne mreže, rezervirane adrese i četvorku s točkama.

5.1. Četvorka s točkama

5.1.1. Binarni/decimalni prikaz adresa IPv4

IP (*Internet Protocol*) je mrežni protokol za prijenos podataka kojim se koriste izvorišna i odredišna računala za uspostavu podatkovne komunikacije preko računalne mreže. **IP** je standard na najvećoj računalnoj mreži danas - **Internetu**. Najviše korištena inačica protokola koja je *de facto* standard Interneta je IP inačica 4 (**IPv4**), a slijedeća je inačica IP-a inačica 6 (**IPv6**). Pojedine se inačice razlikuju u načinu adresiranja i brojnim drugim detaljima. Tako se IPv4 koristi 32-bitnom IP-adresom, a IPv6 128-bitnom adresom.

Prepoznatljiv oblik prikaza **IPv4** adrese naziva se četvorka s točkama (*dotted quad*). Kod **IPv4** (*Internet Protocol version 4*) govorimo o 4 bajta (svaki sadrži 8 bitova) odvojenih točkom. Budući da u 8 bitova stane 2^8 = 256 znakova **IPv4** adrese su između **0.0.0.0** i **255.255.255.255**.

IPv4 adresa u osnovi je 32-bitni binarni broj. S obzirom da je u pravilu vrlo teško pamtiti niz od 32 znaka 0 ili 1, češće se rabe druge notacije, a najčešće od svih decimalna. Decimalna se notacija od 32-bitnog binarnog broja dobije tako da se 32-bitni broj odvoji u četiri 8-bitne skupine, svaka se skupina zapiše u dekadskom obliku, a zatim se u zapisu prikažu odvojene točkama.

Pogledajmo to na jednom primjeru:

11000000.10101000.00000001.00000001

Ta ista adresa lakše se pamti u decimalnom obliku:

192.168.1.1

Binarni je zapis važan jer se u izračunu nekih sastavnica mreže i za neke komponente rabe unarne i binarne logičke operacije nad bitovima.

5.2. Rezervirane adrese

5.2.1. Adresa razašiljanja, mrežna adresa i mrežna maska

IP-adresa je numerička oznaka koja je dodijeljena svakom uređaju koji se na računalnoj mreži koristi IP-jem (I*nternet Protocol*) za komunikaciju. Uz **IP adresu** za uspješnu je komunikaciju potrebna i **mrežna maska**. Iz IP-adrese (bilo kojeg uređaja u mreži) moguće je izračunati još dvije važne IP-adrese - **mrežnu adresu** i **adresu razašiljanja** (*broadcast address*).

Mrežna maska je 32-bitni binarni broj koji je uvijek oblika **1*0***. Kao i svaka **IP** adresa mrežna se maska može zapisivati kao četiri binarne osmorke ili kao decimalna četvorka. Mrežna maska se također može zapisati i samo jednim brojem od 1 do 32 koji definira koliko bitova opisuje mrežu.

Primjer je dviju mrežnih maski u spomenuta tri formata zapisa:

Decimalni prikaz	Skraćeni prikaz	Binarni prikaz
255.0.0.0	8	11111111.00000000.00000000.00000000
255.255.128.0	17	11111111.11111111.10000000.00000000

Mrežna maska definira koji su uređaji u istoj mreži - bitovi koji imaju vrijednost 1 pripadaju mreži. Tako ako je na primjer mrežna maska 255.255.128.0 (17) tada su ove adrese u istoj mreži:

- 161.53.224.7 (**10100001.00110101.1**1100000.00000111)
- 161.53.227.224 (**10100001.00110101.1**1100011 .11100000)

Adrese na istoj mreži znače da je komunikacija među uređajima izravna i da ne treba dodatni uređaj za usmjeravanje komunikacije između mreža da bi se ostvarila komunikacija među uređajima.

Adresu uređaja u potpunosti definiraju IP-adresa i mrežna maska. Standardni oblik prikaza je **<IP_adresa>/<Mrežna_maska>**. Dopušten je bilo koji format, ali se zbog jednostavnosti najčešće rabi najkraći prikaz sa četiri decimalne znamenke i decimalnim brojem manjim od 32. Na primjer **161.53.1.173/28**.

Mrežna adresa je podatak potreban pri konfiguraciji usmjernika (*router*). Mrežna adresa je ujedno i prva adresa u danoj mreži. Dakle mrežna se adresa izračunava tako da se u adresi bilo kojeg uređaja svi bitovi koji ne definiraju mrežu zamjene nulama. Na primjer, neka su adresa i mrežna maska 161.35.2.112/24. Tada je mrežna adresa 161.53.2.**0**.

Adresa razašiljanja (*broadcast*) je specijalna adresa u mreži koja je namijenjena za slanje poruka svim čvorovima (članovima) mreže. Adresa razašiljanja je adresa u kojoj su svi ne mrežni bitovi jedinice. U gornjem bi primjeru za 161.35.2.112/24 adresa razašiljanja bila 161.53.2.255. U ovom primjeru sve poruke slane na adresu razašiljanja primit će svi uređaji s adresama između 161.53.2.1 <--> 161.53.2.254.

Pomoću bitovnih logičkih operacija **i**, **ne** i **ili** moguće je iz adrese (bilo kojeg) člana mreže i mrežne maske izračunati adresu mreže i adresu razašiljanja. Postupci su ovakvi:

(1)

```
"MREŽNA ADRESA" = "IP" i "MREŽNA MASKA"
```

Kad je izračunata mrežna adresa, računamo dalje:

(2)

```
"ADRESA RAZAŠILJANJA" = "MREŽNA ADRESA" ili ne ["MREŽNA MASKA"]
```

U gornjem primjeru, dakle:

IP=161.53.2.112 = 10100001.00110101.00000010.01110000

MREŽNA MASKA = 24 = 255.255.255.0=11111111.11111111.11111111.00000000

MREŽA =(1)

```
(1)
10100001.00110101.00000010.01110000 i
11111111.11111111.11111111.00000000
-----------------------------------------------
10100001.00110101.00000010.00000000 = 161.53.2.0
(2)
ne ["MREŽNA MASKA"] =
ne [11111111.11111111.11111111.00000000] =
    00000000.00000000.00000000.11111111

"ADRESA RAZAŠILJANJA" =
10100001.00110101.00000010.00000000    ili
00000000.00000000.00000000.11111111
-------------------------------------------------
10100001.00110101.00000010.11111111 = 161.53.2.255
```

5.3. Mrežne klase i besklasne mreže

5.3.1. Klase A, B i C

Mrežne su klase bile osnovna okosnica mrežne adresne arhitekture od 1981. godine do 1993. godine i uvođenja besklasnog međudomenskog usmjeravanja. U mrežnim klasama **IPv4 adrese** su podijeljene u **pet adresnih klasa**. Svaka je klasa, jedinstveno obilježena pomoću prva četiri bita, a definira određene mrežne raspone kojima se mogu koristiti uređaji (klase A, B i C) ili mreže razašiljanja (klasa D). Peta klasa E rezervirana je za neke buduće istraživačke potrebe. U nastavku su objašnjene prve tri klase jer se klase D i E rabe za posebne namjene koje nisu u okviru ovog programa.

U privatnim mrežama dio se adresa nikada ne rabe na Internetu već su namijenjene za rad lokalne mreže (LAN, *Local Area Network*). Zbog toga svaka mreža gubi najmanje dvije adrese koje se ne mogu rabiti za pristup uređaja mreži, jer su potrebne za funkcioniranje samog LAN-a.

Ime klase	Obavezni početni bitovi	Broj bitova koji definira mrežu	Broj bitova koji definira uređaje u mreži	Broj mogućih mreža
Klasa A	0	8	24	128 (2^7)
Klasa B	10	16	16	16 384(2^{14})
Klasa C	110	24	8	2 097 152 (2^{21})

Ime Klase	Broj mogućih uređaja u mreži	Ukupno adresa u klasi	Početna adresa	Završna adresa
Klasa A	16 777 214 (2^{24} - 2)	2,147,483,648 (2^{31})	0.0.0.0.	127.255.255.255.
Klasa B	65 534 (2^{16} - 2)	1,073,741,824 (2^{30})	128.0.0.0.	191.255.255.255.
Klasa C	254 (2^8 - 2)	536,870,912 (2^{29})	192.0.0.0.	223.255.255.255.

Kako je prikazano u gornjoj tablici, postoje tri osnovne mrežne klase. U tablici su sadržani svi podaci o njima koje treba poznavati. Treba također uočiti kako ih jednostavnije zapamtiti (logika imenovanja) i kako se računaju vrijednosti u tablici.

- Svaka klasa ima na n-tom mjestu bit vrijednosti 0, a sve prethodne bitove vrijednosti 1. počevši od klase A koje ima 0 na prvom mjestu.
- Svaka klasa ima 8 bitova više za definiciju mreže nego prethodna počevši od 8 bitova u klasi A.
- Broj mogućih mreža nije 2 na broj bitova koji definiraju mrežu jer su početni bitovi definirani.

U počecima je Interneta taj klasni sustav bio jednostavan i čist model za usmjeravanje. Ubrzo se taj model pokazao utopijskim. Brojnim organizacijama nije bila dovoljna klasa C – pa im je dodijeljena klasa B, koja je bila prevelika i od koje su iskoristili tek mali postotak adresa. Takvo je stanje dovodilo do brzog iscrpljivanja adresa uz nisku iskoristivost postojećih. Isto je ispravljano 1993. godine uvođenjem **besklasnog međudomenskog usmjeravanja**.

5.3.2. Besklasne mreže

Zapis internetskih adresa u formatu IP/Maska je **CIDR notacija** (*CIDR, Classless Inter-Domain Routing*, besklasno međudomensko usmjeravanje). U notaciji CIDR uvedenoj 1993. godine moguće je rabiti bilo koji broj bitova za definiciju mreže, a ne samo osam i njegove višekratnike.

Klasna je shema zbog (u prethodnom dijelu objašnjenih) nedostataka zamijenjena shemom u kojoj je u svakoj klasi moguće napraviti 2 do 128 podmreža. Zbog ove podijele standardnih klasa besklasne se mreže nazivaju i **besklasne podmreže** (*Classless subnets*).

Određeni se broj bitova koristi za identifikaciju mreže, a ostataka za identifikaciju članova unutar mreže pa je broj adresa dodijeljenih nekoj mreži uvijek potencija broja dva. Postavlja se pitanje ima li smisla za svaku namjenu dodijeliti točno neku potenciju broja dva adresa. Odnosno može li se bolje iskoristiti adresni prostor na neki način da se mreži može dodijeliti bilo koji broj adresa, a ne samo potencija broja dva i ima li to smisla?

Pogledajmo dva primjera:

1. **ISP** (*Internet service provider*) treba **2320 adresa** za potrebe rada sa korisnicima.
2. **Obrazovna ustanova** treba **11 adresa** za svoje uređaje koji trebaju biti dostupni na javnim IP adresama.

U prvom će se slučaju **ISP** ili zatražiti dvije mreže, jednu od 2048 adresa i jednu od 512, ili će zatražiti 4096 adresa za buduće potrebe (ako je planirano dugoročno širenje. Ni jedno rješenje ne predstavljaju veliki gubitak (u postotku neiskorištenih) IP-adresa.

U drugom slučaju čini se mogućim dodijeliti dva segmenta C klase – jedan od osam, a drugi od četiri adrese. No svaka (pod)mreža ima dvije rezervirane adrese te je zapravo dodijeljeno (8-2)+(4-2)=10 iskoristivih adresa. Dakle jedina mogućnost je dodijeliti 16 adresa (14 iskoristivih). Iako se gubitak od ~ 31% čini velikim dodjela adresnog prostora je postupak koji se treba rijetko provoditi i moguće je da će ubrzo i suvišne adrese biti potrebne **obrazovnoj ustanovi**.

5.3.3. Primjer i uporaba naredbe ipcalc

Pogledajmo jedan primjer mreže sa 16 adresa. Recimo da znamo da je jedna od adresa uređaja na mreži 10.43.8.67/28

Opis	**Izračun**	**Binarni prikaz**	**Decimalni prikaz**
IPv4-adresa	Zadano.	00001010.00101011. 00001000.01000011	10.43.8.67
Mrežna maska	Zadano.	11111111.11111111. 11111111.11110000	255.255.255.240
Adresa ***razašiljanja***	Negacija logičkog ili između IPv4 adrese i mrežne maske	00001010.00101011. 00001000.01001111	10.43.8.79
Network address (Network ID)	Logički i IPv4 adrese i mrežne maske	00001010.00101011. 00001000.01000000	10.43.8.64
Slijedno mjesto uređaja u mreži	Logički i IPv4-adrese i negacije mrežne maske	00000000.00000000. 00000000.00000011	3
Raspon adresa mreže			10.43.8.64 do 10.43.8.79

IPv4 adrese koje se mogu dodijeliti uređajima	10.43.8.65 do 10.43.8.78

Izračuni iz gornje tablice mogu se u sustavu napraviti naredbom ipcalc. Naredba ipcalc na osnovi dvaju zadanih parametra računa druge rezervirane adrese u mreži. Gornji primjer pomoću naredbe ipcalc glasi:

```
# ipcalc 10.43.8.67/28
Address:   10.43.8.67           00001010.00101011.00001000.01000011
Netmask:   255.255.255.240 = 28 11111111.11111111.11111111.1111 0000
Wildcard:  0.0.0.15             00000000.00000000.00000000.00001111
=>
Network:   10.43.8.64/28        00001010.00101011.00001000.01000000
HostMin:   10.43.8.65           00001010.00101011.00001000.01000001
HostMax:   10.43.8.78           00001010.00101011.00001000.01001110
Broadcast: 10.43.8.79           00001010.00101011.00001000.01001111
Hosts/Net: 14                    Class A, Private Internet
```

5.3.4. Dodatni sadržaji

- https://en.wikipedia.org/wiki/Classful_network
- https://en.wikipedia.org/wiki/Classless_Inter-Domain_Routing
- http://linux.die.net/man/1/ipcalc

5.4. TCP/IP

5.4.1. Protokoli (IP – UDP, TCP, ICMP, PPP)

TCP/IP (*Internet protocol suite*) čini skup komunikacijskih protokola i model računalnih mreža koji se rabe na Internetu i u sličnim računalnim mrežama. Budući da su osnovni protokoli TCP (*Transmission Control Protocol*) i IP (*Internet Protocol*), taj se skup protokola skraćeno naziva **TCP/IP**.

TCP/IP u potpunosti omogućava povezivanje dvaju mrežnih uređaja definirajući kako podatke treba segmentirati u pakete, kako adresirati, kako usmjeravati kroz mrežu i konačno kamo ih treba dostaviti te kako i tko ih tamo treba obraditi. Taj veliki posao obavlja se u četiri apstraktna sloja, gdje se podjela vrši prema rasponu mrežnih protokola koji su uključeni.

Od najnižeg su prema najvišem slojevi:

Logika je dakle da se komunikacija ostvaruje tako da se na svakom sloju određeni protokol brine o akcijama za je mjerodavan i zatim, ovisno o potrebi, prosljeđuje zahtjeve protokolu u nižem/višem sloju. Tipični predstavnici protokola u svakom sloju dani su u tablici:

Sloj	**Tipični protokoli**
sloj veze	ARP, PPP, OSPF, DSL, IDSN, NDP…
internet sloj	IP (IPv4,IPv6), ICMP, ICMPv6, ECN, IGMP, IPsec…
transportni sloj	TCP, UDP, DCCP, SCTP, RSVP…
aplikacijski sloj	BGP, DHCP, DNS, FTP, HTTP, IMAP, LDAP, MGCP, NNTP, NTP, POP, ONC/RPC, RTP, RTSP, RIP, SIP, SMTP, SNMP, SSH, Telnet, TLS/SSL, XMPP…

Pojasniti ćemo samo ključne protokole:

- **IP – *The Internet Protocol*** je transportni protokol za **TCP**, **UDP** i **ICMP** podatke. **IP-veze** ne brinu se o pouzdanosti prijenosa i ne provode uspostavu veze. To je prepušteno protokolima viših slojeva poput **TCP**-a. **IP-**protokol se brine o adresiranju i usmjeravanju komunikacije među mrežama. To je servis za dostavu datagrama.
- **TCP – *Transmission Control Protocol*** pruža pouzdanu vezu servisima koji je trebaju. Aplikacije koje komuniciraju preko **TCP** ne moraju same paziti na pouzdanost veze. **TCP** pazi na slijed slanja i primanja paketa u komunikaciji, uspostavu veze ili sjednice.
- **UDP – *The user datagram protocol*** pruža u istom sloju kao i **TCP** nepouzdanu vezu i ostvaruje znatno manje dodatno opterećenje. Brži je od **TCP-**protokola, ali ako je potrebno ostvariti kontrolu nad tijekom paketa tada to mora biti ostvareno u aplikaciji koja komunicira preko **UDP**-a.
- **ICMP – *The Internet Control Message Protocol*** rabe mrežni uređaji i uređaji na mreži za komunikaciju o statusu mreže. ICMP se koristi IP-protokolom za komunikaciju i poput **UDP**-a ne uspostavlja veze ni sjednice.

- **PPP –** ***The Point to Point Protocol*** se koristi za **TCP/IP**-komunikaciju preko telefonskih linija. Također se rabi kriptiranim vezama kao što je **PPtP** (*Point-to-Point Tunneling Protocol*).

5.4.2. Popis portova

U računalnim je mrežama **port** logički konstrukt namijenjen identifikaciji servisa ili procesa. **Port** je, zajedno s IP adresom i protokolom, jedinstvena definicija izvora ili odredišta pri komunikaciji. **Port** je 16-bitni binarni broj, pa su u decimalnom zapisu vrijednosti portova između 0 i 65535.

Popis poznatih servisa i portova na kojima se oni pokreću nalazi se u datoteci **/etc/services**. Službeni i potpuni popis održava organizacija **IANA** (*Internet Assigned Numbers Authority*). Zadaća je te organizacije briga o svim globalno jedinstvenim identifikatorima na Internetu.

Broj porta je broj između 0 i 65535, s obzirom da se port opisuje 16-znamenkastim binarnim brojem. Portovi koji počinju s šest nula (1 do 1023 u dekatskom zapisu) **privilegirani** su portovi rezervirani za servise koje pokreće korisnik *root*. Većina standardnih protokola sluša na jednom od **privilegiranih** portova. Skeneri često provjeravaju samo **privilegirane** portove, ako im nije eksplicitno navedeno drugačije.

Na primjer:

```
# nmap -sV localhost

Starting Nmap 6.00 ( http://nmap.org ) at 2015-07-01 10:08 CEST
Nmap scan report for localhost (127.0.0.1)
Host is up (0.0000090s latency).
Not shown: 998 closed ports
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 6.0p1 Debian 4+deb7u2 (protocol 2.0)
25/tcp open  smtp    Postfix smtpd
Service Info: Host:  debian-1.test.lan; OS: Linux; CPE: cpe:/o:linux:kernel
Service detection performed. Please report any incorrect results at
http://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 0.15 seconds
```

Ovo je popis portova kojima se koristi neka aplikacija. Prikazani su samo portovi ispod 1023.

Ako na istom računalu skeniramo sve portove dobijemo znatno veći broj aktivnih portova.

```
# nmap -sV  -p 1-65535 localhost

Starting Nmap 6.00 ( http://nmap.org ) at 2015-07-01 10:06 CEST
Nmap scan report for localhost (127.0.0.1)
Host is up (0.0000090s latency).
Not shown: 65509 closed ports
PORT      STATE SERVICE    VERSION
22/tcp    open  ssh        OpenSSH 6.0p1 Debian 4+deb7u2 (protocol 2.0)
25/tcp    open  smtp       Postfix smtpd
1555/tcp  open  unknown
```

```
1558/tcp  open  xingmpeg?
7010/tcp  open  http       Apache Tomcat/Coyote JSP engine 1.1
7011/tcp  open  unknown
7015/tcp  open  unknown
7110/tcp  open  http       Apache Tomcat/Coyote JSP engine 1.1
7111/tcp  open  unknown
7115/tcp  open  unknown
7410/tcp  open  http       Apache Tomcat/Coyote JSP engine 1.1
7411/tcp  open  unknown
7415/tcp  open  unknown
7510/tcp  open  http       Apache Tomcat/Coyote JSP engine 1.1
7511/tcp  open  unknown
7515/tcp  open  unknown
10050/tcp open  tcpwrapped
35937/tcp open  unknown
37343/tcp open  unknown
37649/tcp open  unknown
40268/tcp open  unknown
42460/tcp open  unknown
50027/tcp open  unknown
52479/tcp open  unknown
54432/tcp open  unknown
59174/tcp open  unknown
Service Info: Host:  debian-1.test.lan; OS: Linux; CPE: cpe:/o:linux:kernel

Service detection performed. Please report any incorrect results at
http://nmap.org/submit/.
Nmap done: 1 IP address (1 host up) scanned in 82.96 seconds
```

Dakle sada se vide svi servisi koji slušaju na portovima računala. Uočimo da veliki broj servisa nije prepoznat (*unknown*). Razlog tome je što se pokreću na nestandardnim portovima i što ne daju podatke o imenu kao odgovor pri zahtjevu za identifikacijom.

5.4.3. Datoteka etc/services

U datoteci **/etc/services** nalazi se popis poznatih servisa i portova/protokola kojima se koriste ti servisi. Tom se datotekom koriste neki sistemski alati za prikupljanje broja porta na osnovi imena servisa. Također se alati za prikaz mrežne aktivnosti koriste datotekom za obrnutu operaciju - za skenirane identificirane portove pribavljaju ime servisa da bi ispis na ekran učinili lakšim za čitanje i tumačenje.

Primjer datoteke **/etc/services**:

```
tcpmux          1/tcp          # TCP port service multiplexer
echo            7/tcp
echo            7/udp
discard         9/tcp           sink null
discard         9/udp          sink null
systat          11/tcp          users
daytime         13/tcp
daytime         13/udp
```

```
netstat         15/tcp
qotd            17/tcp          quote
msp             18/tcp           # message send protocol
msp             18/udp
chargen         19/tcp          ttytst source
chargen         19/udp          ttytst source
ftp-data        20/tcp
ftp             21/tcp
fsp             21/udp          fspd
ssh             22/tcp          # SSH Remote Login Protocol
ssh             22/udp
telnet          23/tcp
smtp            25/tcp         mail
time            37/tcp          timserver
time            37/udp        timserver
rlp             39/udp          resource        # resource location
nameserver      42/tcp    name            # IEN 116
whois           43/tcp          nicname
tacacs          49/tcp         # Login Host Protocol (TACACS)
tacacs          49/udp
re-mail-ck      50/tcp        # Remote Mail Checking Protocol
re-mail-ck      50/udp
domain          53/tcp        # Domain Name Server
domain          53/udp
mtp             57/tcp           # deprecated
tacacs-ds       65/tcp        # TACACS-Database Service
tacacs-ds       65/udp
bootps          67/tcp          # BOOTP server
bootps          67/udp
bootpc          68/tcp          # BOOTP client
bootpc          68/udp
tftp            69/udp
gopher          70/tcp          # Internet Gopher
gopher          70/udp
rje             77/tcp          netrjs
finger          79/tcp
http            80/tcp          www             # WorldWideWeb HTTP
http            80/udp                          # HyperText Transfer Protocol
link            87/tcp          ttylink
kerberos        88/tcp          kerberos5 krb5 kerberos-sec     # Kerberos v5
kerberos        88/udp          kerberos5 krb5 kerberos-sec     # Kerberos v5
supdup          95/tcp
hostnames       101/tcp         hostname        # usually from sri-nic
iso-tsap        102/tcp         tsap            # part of ISODE
acr-nema        104/tcp         dicom           # Digital Imag. & Comm. 300
acr-nema        104/udp         dicom
csnet-ns        105/tcp         cso-ns          # also used by CSO name server
csnet-ns        105/udp         cso-ns
rtelnet         107/tcp                         # Remote Telnet
rtelnet         107/udp
pop2            109/tcp         postoffice pop-2 # POP version 2
pop2            109/udp         pop-2
pop3            110/tcp         pop-3           # POP version 3
pop3            110/udp         pop-3
```

```
sunrpc          111/tcp         portmapper      # RPC 4.0 portmapper
sunrpc          111/udp         portmapper
auth            113/tcp         authentication tap ident
sftp            115/tcp
uucp-path       117/tcp
nntp            119/tcp         readnews untp   # USENET News Transfer Protocol
ntp             123/tcp
ntp             123/udp                         # Network Time Protocol
```

5.4.4. Dodatni sadržaji

- https://en.wikipedia.org/wiki/Internet_protocol_suite
- https://hr.wikipedia.org/wiki/OSI_modelhttps://en.wikipedia.org/wiki/OSI_model
- https://en.wikipedia.org/wiki/Nmap

5.5. Vježba: Identifikacija parametara mreže

1. Prijavite se na računalo kao korisnik l102. U GUI-u pokrenite **Terminal** (*Activities* → **Terminal**).
2. Naredbom `su -` postanite **root** korisnik (lozinka: L102).
3. Zadana je adresa jednog uređaja na mreži s **161.53.17.99/27**. Ručno izračunajte sve ključne adrese mreže pa odgovorite na pitanja.

a. Pripada li adresa **161.53.17.129** toj mreži?

b. Pripada li adresa **161.53.17.12** toj mreži?

c. Koja je adresa razašiljanja (**broadcast**) u toj mreži?

d. Je li mreža besklasna i ako ne koju klasu spada?

e. Koliko uređaja s distinktnim IP adresama stane u ovu mrežu?

f. Koja je adresa mreže?

4. Izvršite naredbu `ipcalc` nad proizvoljnom adresom u mreži da provjerite dobivene rezultate.

6. Konfiguracija mreže

Trajanje poglavlja:
125 min

Po završetku ovoga poglavlja moći ćete:

- pronaći sve mrežne uređaje na računalu
- prikupiti podatke o instaliranim mrežnim uređajima
- imenovati mrežne konfiguracijske datoteke
- urediti postavke mreže u odgovarajućoj datoteci prema potrebi
- pokrenuti, zaustaviti i konfigurirati mrežno sučelje iz naredbene linije
- prepustiti automatsku konfiguraciju **DHCP**-poslužitelju
- provjeriti stanje mreže i **DHCP**-najma
- pogledati i protumačiti tablicu usmjeravanje
- dodati pravila u tablicu usmjeravanja uključujući standardni *gateway*
- nabrojati namjene pojedinih standardnih mrežnih alata
- provjeriti ponašanje i stanje mreže i mrežnih uređaja.

Ova cjelina obrađuje osnovne naredbe za pregled konfiguracije mrežnih sučelja, upravljanje mrežnim sučeljima i prilagođavanje postavki mrežnog sučelja. Naučit ćemo izraditi, pokrenuti, zaustaviti i provjeravati mrežna sučelja i općenito provjeravati dostupnost i valjanost mreže.

6.1. Mrežno sučelje

6.1.1. Mrežna kartica i podrška jezgre

Mrežna kartica (*Network card, NIC-network interface card, network adapter*) dio je računala koji se brine za komunikaciju računala preko računalne mreže, odnosno za priključivanje računala na lokalnu mrežu. Da bi mrežna kartica radila mora postojati potpora u jezgri operacijskog sustava *Linux*.

Mrežno sučelje je sučelje sustava između dviju komponenti računalne opreme na računalnoj mreži.

Fizička adresa (*media access control address, MAC address* ili kraće *MAC*) jedinstveni je identifikator mrežnog sučelja. **MAC-adresu** dodjeljuje proizvođač mrežnih uređaja i pohranjene su u hardveru uređaja u nekom dijelu nepromjenjive memorije (*read only memory, ROM*), a pomoću te se adrese može identificirati uređaj.

Do podataka o mrežnoj kartici može se doći ovako:

- pregledom datoteka **/proc/interrupts** i **/etc/modules**
- izvršavanjem naredbi **dmesg, lsmod** ili **lspci**
- korištenjem dodatnih alata (moraju biti posebno instalirani) poput **lshw** ili **ethtool**.

Najbrže je podatke prikupiti tako da se pomoću naredbe **lspci** saznaju **pci**-brojevi uređaja:

```
# lspci | egrep -i 'network|ethernet'
00:03.0 Ethernet controller: Intel Corporation 82540EM Gigabit Ethernet Controller (rev 02)
00:08.0 Ethernet controller: Intel Corporation 82540EM Gigabit Ethernet Controller (rev 02)
```

Tom se naredbom za sve mrežne uređaje saznaje broj koji ih u potpunosti opisuje – njihov **pci**-broj (*Peripheral Component Interconnect*).

Format **pci**-broja je:

`<domena>:<sabirnica>:<položaj/utor/slot>.<funkcija>`

U primjeru računalo ima samo jednu domenu pa taj parametar nedostaje, a drugi nam govore:

- sabirnica = 0 (kod obje kartice)
- slot = 3 (prva), slot=8 (druga)
- funkcija = 0.

Sada kad znamo **pci**-broj, pomoću njega možemo pribaviti detaljne podatke o mrežnim karticama.

Za mrežnu karticu s PCI brojem 00:03.0

```
# lspci -v -s 00:03.0
00:03.0 Ethernet controller: Intel Corporation 82540EM Gigabit Ethernet
Controller (rev 02)
    Subsystem: Intel Corporation PRO/1000 MT Desktop Adapter
    Flags: bus master, 66MHz, medium devsel, latency 64, IRQ 10
    Memory at f0000000 (32-bit, non-prefetchable) [size=128K]
    I/O ports at d010 [size=8]
    Capabilities: [dc] Power Management version 2
    Capabilities: [e4] PCI-X non-bridge device
    Kernel driver in use: e1000
```

Za mrežnu karticu s PCI brojem 00:08.0

```
# lspci -v -s 00:08.0
00:08.0 Ethernet controller: Intel Corporation 82540EM Gigabit Ethernet
Controller (rev 02)
    Subsystem: Intel Corporation PRO/1000 MT Desktop Adapter
    Flags: bus master, 66MHz, medium devsel, latency 64, IRQ 9
    Memory at f0820000 (32-bit, non-prefetchable) [size=128K]
    I/O ports at d240 [size=8]
    Capabilities: [dc] Power Management version 2
    Capabilities: [e4] PCI-X non-bridge device
    Kernel driver in use: e1000
```

Uz fizičke mrežne uređaje i njima dodijeljena mrežna sučelja na računalu je **uvijek** aktivno i virtualno mrežno sučelje *loopback*. Sučelje *loopback* rabi se za provjeru i za pristup lokalnim servisima. Kad na računalu ne postoje fizički mrežni uređaji, bez tog sučelja klijenti na računalu ne mogu pristupiti resursima koje dijele servisi na tom računalu. Također pri provjeri rada mrežnih servisa jednostavno je provjeriti rad preko mrežnog sučelja *loopback* i tako iz provjere izbaciti moguće kvarove na fizičkim uređajima odnosno pogreške u mrežnim postavkama.

6.1.2. Prikupljanje dodatnih podataka

Broj pribavljen prvom naredbom **lspci** može se iskoristiti za pronalaženje dodatnih podataka pomoću naredbe **dmesg**:

```
# dmesg |grep 00:03.0
[ 0.332635] pci 0000:00:03.0: [8086:100e] type 0 class 0x000200
[ 0.334468] pci 0000:00:03.0: reg 10: [mem 0xf0000000-0xf001ffff]
[ 0.337872] pci 0000:00:03.0: reg 18: [io 0xd010-0xd017]
[ 2.399258] e1000 0000:00:03.0: eth0: (PCI:33MHz:32-bit) 08:00:27:fa:1a:11
[ 2.400233] e1000 0000:00:03.0: eth0: Intel(R) PRO/1000 Network Connection
# dmesg |grep 00:08.0
[ 0.363516] pci 0000:00:08.0: [8086:100e] type 0 class 0x000200
[ 0.364282] pci 0000:00:08.0: reg 10: [mem 0xf0820000-0xf083ffff]
[ 0.366888] pci 0000:00:08.0: reg 18: [io 0xd240-0xd247]
[ 2.910163] e1000 0000:00:08.0: eth1: (PCI:33MHz:32-bit) 08:00:27:a1:3b:b4
[ 2.910455] e1000 0000:00:08.0: eth1: Intel(R) PRO/1000 Network Connection
```

Na ovaj smo način prikupili ove podatke:

Podatak	Kartica 1	Kartica 2
Sučelje	Eth0	Eth1
IRQ	10	9
Inačica upravljačkog programa	e1000	e1000
MAC	08:00:27:fa:1a:11	08:00:27:a1:3b:b4

Prikupljeni se podaci mogu koristiti za ručno učitavanje modula potrebnih za rad mrežnih kartica naredbama **modprobe** ili **insmod**.

6.1.3. Dodatni sadržaji

- https://wiki.debian.org/NetworkConfiguration

6.2. Podaci o adresi poslužitelja

6.2.1. Mrežne konfiguracijske datoteke

Mrežni podaci na računalu pohranjeni su u datotekama **/etc/resolv.conf** i **/etc/hosts** te u datoteci **/etc/hostname** i direktoriju **/etc/network/** na *Debian*u i direktoriju ***/etc/sysconfig/network-scripts/*** na distribucijama *Red Hat*.

U datoteci **/etc/resolv.conf** nalazi se popis **DNS** poslužitelja (*Domain Name Service*).

```
nameserver 161.53.252.36
nameserver 161.53.252.37
```

Napomena

Pri ručnom uređivanju datoteke **/etc/resolv.conf** treba biti oprezan, jer će brojni programi „pregaziti“ zapise u toj datoteci:

- program **Resolvconf**
- pozadinski proces **Network-online**
- DHCP klijenti

Datoteka **/etc/hostname** sadrži ime računala:

```
# cat /etc/hostname
debian-1
```

Datoteka **/etc/hosts** namijenjena je rezoluciji imena u IP-adrese lokalno za računalo, ali i za neka posebna poznata računala. Dodatna se računala upisuju kako se u nekim posebnim slučajevima (kratkotrajna provjera, isprobavanje utjecaja promjene imena i druge testne situacije) ne bi svakog puta trebao raditi novi DNS-zapis i čekati na njegovu propagaciju. Poželjno je imati zapise o dodatnim računalima u datoteci kad je računalo dio klastera pa je poželjno da je rad klastera neovisan o radu **DNS**-servisa i poslužitelja.

Primjer je datoteke **/etc/hosts** sa zapisom o dva računala na kraju:

```
# cat /etc/hosts
127.0.0.1 localhost
127.0.1.1 debian-1.test.lan debian-1
# The following lines are desirable for IPv6 capable hosts
::1 localhost ip6-localhost ip6-loopback
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
# other hosts
161.53.7.213 temp-test1.test.lan temp-test1
161.53.7.215 temp-versioning.test.lan temp-versioning
```

Datoteka **/etc/network/interfaces** središnja je datoteka za konfiguraciju mrežnih postavki u *Debianu*. U njoj su definirana postojeća sučelja, koja se od njih pokreću automatski i postavke tih sučelja. Postavke uključuju pravila usmjeravanja sučelja (*routes*), **IP**-adresu, omogućavanje/gašenje **DHCP**-podrške itd.

```
# cat /etc/network/interfaces
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5)
# The loopback network interface
auto lo enp0s0 enp0s1 enp0s0:0
iface lo inet loopback
# The primary network interface
#allow-hotplug enp0s0
iface enp0s0 inet static
    address 161.53.3.205
    netmask 255.255.255.0
    network 161.53.3.0
    broadcast 161.53.3.255
    gateway 161.53.3.1
 # dns-* options are implemented by the resolvconf package, if installed
 dns-nameservers 161.53.2.69 161.53.2.70
 dns-search test.srce.hr
iface enp0s0:0 inet static
    address 161.53.3.206
    netmask 255.255.255.0
    network 161.53.3.0
    broadcast 161.53.3.255
    gateway 161.53.3.1
    # dns-* options are implemented
by the resolvconf package, if installed
    dns-nameservers 161.53.2.69 161.53.2.70
    dns-search test.srce.hr
iface enp0s1 inet static
    address 161.53.3.217
    netmask 255.255.255.0
    network 161.53.3.0
    broadcast 161.53.3.255
    gateway 161.53.3.1
    # dns-* options are implemented by the resolvconf package, if
installed
    dns-nameservers 161.53.2.69 161.53.2.70
    dns-search test.srce.hr
```

U datoteci u gornjem primjeru podešene su dvije kartice. Prva kartica ima dva sučelja **enp0s0** i **enp0s0:0**, dakle ukupno su podešena tri mrežna sučelja. Konfiguracija pojedinog sučelja počinje s **iface**. Svako sučelje je statički konfigurirano i zadani su adresa, mrežna maska, mreža, adresa razašiljanja i glavni usmjernik (*gateway*). Parametri koji započinju s **dns** namjenjeni su **DNS** rezoluciji i definiraju **DNS** poslužitelje i **DNS** domenu kojoj računalo pripada.

6.2.2. Dodatni sadržaji

- https://wiki.debian.org/NetworkConfiguration#Configuring_the_interface_manually

6.3. Pokretanje i zaustavljanje mreže

6.3.1. Naredbe ifconfig i ip

Naredbe **ifconfig** i **ip** mogu se koristiti za prikaz ili konfiguraciju mrežnih sučelja. Obje su prepune mogućnostima. Naredba **ipconfig** je zastarjela i od verzije 9.0 naredba nije dio standardne instalacije. Za korištenje te naredbe potrebno je dodatno instalirati paket **net-tools**.

Sintaksa je naredbe **ifconfig**:

```
ifconfig sučelje [tip_adrese] mogucnosti | adresa
```

Mogućnosti su:

Mogućnost	Opis
up	Podizanje sučelja.
down	Spuštanje sučelja.
[-]arp	Omogućavanje ili onemogućavanje protokola APR.
[-]promisc	Paljenje i gašenje promiskuitetnog načina rada.
[-]allmulti	Paljenje i gašenje načina rada u kojem se presreću svi *multicast* paketi.
netmask addr	Postavljanje mrežne maske.
add|del addr/prefixlen	Dodavanje i uklanjanje IPv6-adrese.
[-]broadcast [adresa]	Dodavanje i uklanjanje adrese razašiljanja.
multicast	Postavlja zastavice za rad *multicast*a.
address	Definiranje adrese za sučelje.

Naredba **ip** nova je naredba zamišljena kao zamjena za cijeli niz naredbi. Od verzije Debiana 9.0 naredba **ip** je središnji alat za korištenje pri upravljanju mrežnim uređajima i postavkama. Sintaksa naredbe **ip** vrlo je složena kako bi se tom naredbom mogao obavljati veliki obim operacija.

Sintaksa naredbe **ip** je:

```
ip [ opcije] objekt { naredba| help }
```

Opcije	Opis
-V, -Version	Ispis inačice.
-s, -stats, -statistics	Dodatne statistike ili vremenski detalji.
-f, -family	Definira skup protokola, mogući skupovi su inet, inet6 i link.
-o, -oneline	Svaki zapis u vlastitoj liniji (kako bi se olakšalo brojanje zapisa pomoću naredbe **wc**).
-r, -resolve	Svi numerički zapisi adresa koji se mogu razriješiti u imena bit će prikazani pomoću imena, a ne brojeva.

Mogući su objekti:

Objekt	Opis
link	Mrežni uređaj.
address	IPv4 ili IPv6-adresa uređaja.
addrlabel	Dodavanje alternativne oznake.
neighbour	Unos u priručnu memoriju ARP ili NDISC.
route	Unos pravila usmjeravanja u tablicu usmjeravanja.
rule	Unos pravila u tablicu politika usmjeravanja.
maddress	Definiranje ***multicast*** adrese.
mroute	Unos u ***multicast*** tablicu usmjeravanja.
tunnel	Definiranje tunela IP.
xfrm	Okvir za protokol IPsec.

Standardne su naredbe:

Naredba	Opis
add	Dodavanje objekta.
delete	Uklanjanje objekta.
show|list	Prikaz određenog svojstva odabranog objekta.
help	Prikaz opcija za dani objekt.

Važno je napomenuti da ne podržavaju svi objekti svih pet opcija i da brojni objekti podržavaju dodatne naredbe. Za potpun pregled opcija treba pogledati man-stranicu. Obje naredbe mogu mijenjati postavke aktivnih sučelja. Pri prikazu mrežnih sučelja naredbe prikazuju različite detalje o sučeljima. Na primjer:

```
# ifconfig
enp0s0 Link encap:Ethernet HWaddr 08:00:27:fa:1a:11
 inet addr:10.0.2.15 Bcast:10.0.2.255 Mask:255.255.255.0
 inet6 addr: fe80::a00:27ff:fefa:1a11/64 Scope:Link
 UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
 RX packets:514 errors:0 dropped:0 overruns:0 frame:0
 TX packets:355 errors:0 dropped:0 overruns:0 carrier:0
 collisions:0 txqueuelen:1000
 RX bytes:560222 (547.0 KiB) TX bytes:40802 (39.8 KiB)
enp0s1 Link encap:Ethernet HWaddr 08:00:27:a1:3b:b4
 inet addr:10.0.0.1 Bcast:10.0.0.255 Mask:255.255.255.0
 inet6 addr: fe80::a00:27ff:fea1:3bb4/64 Scope:Link
 UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
 RX packets:120359 errors:0 dropped:0 overruns:0 frame:0
 TX packets:215 errors:0 dropped:0 overruns:0 carrier:0
 collisions:0 txqueuelen:1000
 RX bytes:12639656 (12.0 MiB) TX bytes:24415 (23.8 KiB)
```

```
lo Link encap:Local Loopback
 inet addr:127.0.0.1 Mask:255.0.0.0
 inet6 addr: ::1/128 Scope:Host
 UP LOOPBACK RUNNING MTU:16436 Metric:1
 RX packets:220 errors:0 dropped:0 overruns:0 frame:0
 TX packets:220 errors:0 dropped:0 overruns:0 carrier:0
 collisions:0 txqueuelen:0
 RX bytes:13276 (12.9 KiB) TX bytes:13276 (12.9 KiB)
```

```
# ip addr show
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 16436 qdisc noqueue state UNKNOWN
 link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
 inet 127.0.0.1/8 scope host lo
 inet6 ::1/128 scope host
 valid_lft forever preferred_lft forever
2: enp0s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP
qlen 1000
 link/ether 08:00:27:9d:ce:31 brd ff:ff:ff:ff:ff:ff
 inet 10.0.2.15/24 brd 10.0.2.255 scope global enp0s0
 inet6 fe80::a00:27ff:fe9d:ce31/64 scope link
 valid_lft forever preferred_lft forever
3: enp0s1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP
qlen 1000
 link/ether 08:00:27:e4:d0:bd brd ff:ff:ff:ff:ff:ff
 inet6 fe80::a00:27ff:fee4:d0bd/64 scope link
 valid_lft forever preferred_lft forever
# ip link show
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 16436 qdisc noqueue state UNKNOWN mode DEFAULT
 link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
2: enp0s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP
mode DEFAULT qlen 1000
 link/ether 08:00:27:9d:ce:31 brd ff:ff:ff:ff:ff:ff
3: enp0s1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP
mode DEFAULT qlen 1000
 link/ether 08:00:27:e4:d0:bd brd ff:ff:ff:ff:ff:ff
```

6.3.2. Podizanje i spuštanje mrežnih sučelja

Za aktivaciju i deaktivaciju mrežnih sučelja koriste se izrazi podizanje i spuštanje. Sučelja je moguće podizati i spuštati naredbama **ifconfig** i **ip**. U slučaju mrežne konfiguracije iz prethodnog primjera naredbe:

```
# ifconfig enp0s0 down
# ip link set enp0s1 down
```

Rezultirati će spuštanjem obaju sučelja:

```
# ifconfig
lo Link encap:Local Loopback
 inet addr:127.0.0.1 Mask:255.0.0.0
 inet6 addr: ::1/128 Scope:Host
 UP LOOPBACK RUNNING MTU:16436 Metric:1
 RX packets:224 errors:0 dropped:0 overruns:0 frame:0
 TX packets:224 errors:0 dropped:0 overruns:0 carrier:0
 collisions:0 txqueuelen:0
 RX bytes:13516 (13.1 KiB) TX bytes:13516 (13.1 KiB)
```

Iste naredbe sa završetkom **up** rezultirat će podizanjem sučelja.

6.3.3. Dodijeljivanje i uklanjanje adresa mrežnih sučelja

Dodjeljivanje adrese sučelju vrši se naredbama:

```
# ifconfig enp0s0 192.168.0.77 netmask 255.255.255.0 broadcast 192.168.0.255
# ip addr add 192.168.0.77/24 broadcast 192.168.0.255 dev enp0s0
```

Uklanjanje adrese može se provesti samo naredbom **ip**:

```
# ip addr del 192.168.0.77/24 dev enp0s0
```

6.3.4. Dodavanje dodatnog sučelja postojećem

```
# ifconfig enp0s0:1 10.0.0.1/8
# ip addr add 11.0.0.1/8 dev enp0s1 label enp0s1:1
```

Prva naredba dodaje **IPv4**-adresu **10.0.0.1/8** sučelju **enp0s0** pod logičkim imenom **enp0s0:1**.

Druga naredba dodaje **IPv4**-adresu **11.0.0.1/8** sučelju **enp0s1** pod logičkim imenom **enp0s1:1**.

Izvođenje tih naredbi rezultira ovom mrežnom konfiguracijom:

```
# ifconfig
enp0s0 Link encap:Ethernet HWaddr 08:00:27:9d:ce:31
 inet addr:10.0.2.15 Bcast:10.0.2.255 Mask:255.255.255.0
 inet6 addr: fe80::a00:27ff:fe9d:ce31/64 Scope:Link
 UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
 RX packets:3741 errors:0 dropped:0 overruns:0 frame:0
 TX packets:453 errors:0 dropped:0 overruns:0 carrier:0
 collisions:0 txqueuelen:1000
 RX bytes:5215392 (4.9 MiB) TX bytes:40043 (39.1 KiB)
enp0s0:1 Link encap:Ethernet HWaddr 08:00:27:9d:ce:31
 inet addr:10.0.0.1 Bcast:10.255.255.255 Mask:255.0.0.0
 UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
```

```
enp0s1 Link encap:Ethernet HWaddr 08:00:27:e4:d0:bd
 inet6 addr: fe80::a00:27ff:fee4:d0bd/64 Scope:Link
 UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
 RX packets:0 errors:0 dropped:0 overruns:0 frame:0
 TX packets:3873 errors:0 dropped:0 overruns:0 carrier:0
 collisions:0 txqueuelen:1000
 RX bytes:0 (0.0 B) TX bytes:832242 (812.7 KiB)
```

```
enp0s1:1 Link encap:Ethernet HWaddr 08:00:27:e4:d0:bd
 inet addr:11.0.0.1 Bcast:0.0.0.0 Mask:255.0.0.0
 UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
```

```
lo Link encap:Local Loopback
 inet addr:127.0.0.1 Mask:255.0.0.0
 inet6 addr: ::1/128 Scope:Host
 UP LOOPBACK RUNNING MTU:16436 Metric:1
 RX packets:232 errors:0 dropped:0 overruns:0 frame:0
 TX packets:232 errors:0 dropped:0 overruns:0 carrier:0
 collisions:0 txqueuelen:0
 RX bytes:13920 (13.5 KiB) TX bytes:13920 (13.5 KiB)
```

Naredbe možda obavljaju istu operaciju u gore navedenim primjerima, ali ne obavljaju je na isti način pa je poželjno spuštena sučelja podizati istom naredbom kojom su spuštena.

Naredba **ip** rabi se i za prikaz i postavljanje pravila usmjeravanja, ali o tome malo kasnije.

6.3.5. Naredbe ifup ifdown ifquery

Naredbe **ifup** i **ifdown** pokreću i zaustavljaju mrežna sučelja. Važno je reći da su te naredbe namijenjene samo tome i ne rabe se u druge svrhe.

Standardno se naredbe rabe na dva načina:

1. `# ifup/ifdown <sučelje>` se koristi za pokretanje odnosno zaustavljanje pojedinačnog sučelja
2. `# ifup/ifdown -a` naredba **ifup** s mogućnosti **–a** pokreće sva sučelja koja su u datoteci **/etc/network/interfaces** i imaju oznaku *auto* (automatsko pokretanje), a naredba **ifdown –a** spustit će sva trenutačno aktivna mrežna sučelja.

Naredba **ifup** dopušta pokretanje sa zadanim pseudonimom. Taj se pseudonim (*alias*) kasnije prikazuje u ispisu aktivnih mreža i može se koristiti pri pozivu za spuštanje sučelja.

Na primjer:

```
# ifup enp0s0:1
# ifdown enp0s0:1
```

Naredba **ifquery** rabi se za pregled postavljenih postavki za postojeća sučelja. Naredba pregledava postojeće konfiguracije i ne provjerava je li sučelje aktivno:

```
# ifquery -l --allow=hotplug
enp0s1
# ip link set enp0s1 down
# ifquery -l --allow=hotplug
enp0s1
```

U gornjem primjeru naredba **ifquery** pregledava postavljene postavke za postojeća sučelja, a s obzirom na to da ne provjerava je li sučelje aktivno, spuštanje sučelja neće utjecati na rezultat izvršavanja naredbe.

6.3.6. Protokol DHCP i posebne naredbe

DHCP (*Dynamic Host Configuration Protocol*) je standardizirani mrežni protokol za dinamičku dodjelu mrežnih postavki klijentima. Preko DHCP-a se udaljeno računalo može ovlastiti da dodijeli IP-adresu i postavi podatke za mrežnu konfiguraciju. Lokalno se pokreće DHCP klijent i u konfiguraciji sučelja koje se konfigurira preko DHCP-a postavlja se odgovarajuća postavka u datoteci **/etc/network/interfaces**.

Primjer za sučelje enp0s0:

```
iface enp0s0 inet dhcp
```

Dodatna konfiguracija nije potrebna, jer klijent prepoznaje **DHCP**-poslužitelj preko *broadcast* poruka.

Dodijeljena mrežna konfiguracija naziva se i najam (*lease*). Najčešće se pojam odnosi samo na **IP**-adresu, jer isti **DHCP**-poslužitelj istom uređaju dodjeljuje ostale postavke neovisno o najmu.

Ako nastanu poteškoće u radu **DHCP**-a, na primjer dvostruka dodjela adrese ili slično, pomoću naredbi **pump** i **dhclient** može se pokušati popraviti stanje. Popraviti stanje znači ukloniti problem da dva ili više uređaja imaju dodijeljenu istu **IP**-adresu.

Format je naredbi:

```
pump [-krRst?] [-i mrežno_sučelje] [-l broj_sati] [--usage]
dhclient [ -4 | -6 ] [ -S ] [ -N [ -N... ] ] [ -T [ -T... ] ] [ -P [ -P... ] ]
[ -p port ] [ -d ] [ -e VAR=value ] [ -q ] [ -1 ] [ -r | -x ] [ -lf lease-file ]
[ -pf pid-file ] [ -cf config-file ] [ -sf script-file ] [ -s server ] [ -g relay ]
[ -n ] [ -nc ] [ -nw ] [ -w ] [ -B ] [ -I dhcp-client-identifier ] [ -H host-name ]
[ -F fqdn.fqdn ] [ -V vendor-class-identifier ] [ -R request-option-list ]
[ -timeout timeout ] [ -v ] [ --version ]
```

Važnije su mogućnosti za pump**:**

Mogućnost	Opis
-i	Imenuje mrežno sučelje nad kojim se provodi naredba
-l	Definira trajanje najma u satima
-r/-release	Otpušta najam
-R/renew	Obnavlja najam

Važnije su mogućnosti za **dhclient**:

Mogućnost	Opis
-4	Naputak da se koristi DHCPv4
-6	Naputak da se koristi DHCPv6
-d	Pokreće se dhclient kao pozadinski proces
-q	Smanjuje *output* naredbi
-s	Imenovanje DHCP poslužitelja za komunikaciju
-r	Oslobađa najam

Otpuštanje najma najčešće je vezano uz neke pogreške u radu. Kada nije u pitanju pokušaj ispravljanja pogreške/krive konfiguracije, tada su najčešće u pitanju situacije kad se pokušava sakriti identitet uređaja. Primjer izvođenja naredbi za „otpuštanje najma“ (*lease release*) IP-adrese:

```
# dhclient –r
# pump -i enp0s0 –release
```

Naredba za zahtjev novog najma je:

```
# dhclient
# pump -i enp0s0
```

Naredba **pump** može ispisati i detalje o aktivnim najmovima:

```
# pump -i enp0s0 --status
Device enp0s0
 IP: 10.0.2.15
 Netmask: 255.255.255.0
 Broadcast: 10.0.2.255
```

```
Network: 10.0.2.0
Boot server 10.0.2.2
Next server 10.0.2.4
Gateways: 10.0.2.2
Boot file: Debian.pxe
Nameservers: 161.53.252.36 161.53.252.37
Renewal time: Thu Jul 2 13:33:55 2015
Expiration time: Thu Jul 2 16:33:55 2015
```

Podaci o „najmovima“ nalaze se u datoteci **/var/lib/dhcp/dhclient.lease**.

6.3.7. Dodatni sadržaji

- http://www.tecmint.com/ip-command-examples/
- http://www.computerhope.com/unix/ifup.htm

6.4. Usmjeravanje

6.4.1. Promjena pravila usmjeravanja

Usmjeravanje je prema definiciji proces izbora najboljeg puta u mreži. Usmjeravanje paketa je osnova rada Interneta. U prošlosti se izraz rabio i za opis prosljeđivanja paketa između mreža, a danas je standardniji izraz prosljeđivanje.

U mrežama s prospajanjem paketa (u koje spada Internet) usmjeravanje upravlja prosljeđivanjem paketa na čvorovima između izvorišta i odredišta. Svako računalo može obavljati funkciju usmjeravanja, ali ne jednako učinkovito i s jednakim performansama kao za to specijalizirani hardver.

Usmjerivač ili **usmjernik** (*router*) je uređaj koji usmjerava podatkovne pakete na njihovom putu kroz računalnu mrežu pri čemu se taj proces odvija na mrežnom sloju **OSI modela**. Svako računalo spojeno na računalnu mrežu mora znati obavljati osnovnu funkcionalnost usmjeravanja te minimalno mora znati primiti paket s odredišnom adresom tog računala te znati proslijediti pakete koje to računalo šalje natrag prema mreži, najčešće prema najbližem usmjerivaču.

Usmjeravanje se standardno provodi pomoću **tablica usmjeravanja**.

U tablicama usmjeravanja nalazi se niz parova

„ciljano odredište“ :: „lokalno odredište“.

Ciljano odredište može biti mreža, a lokalno je odredište imenovanje usmjerivača kojem se prosljeđuju svi paketi namijenjeni toj mreži. Pogledajmo konkretan primjer usmjerivanja. Naredbom **ip route list** ispisujemo lokalnu tablicu usmjeravanja:

```
# ip route list
default via 10.0.2.2 dev enp0s3
default via 10.0.3.2 dev enp0s8 proto static metric 100
10.0.2.0/24 dev enp0s3 proto kernel scope link src 10.0.2.15
10.0.3.0/24 dev enp0s8 proto kernel scope link src 10.0.3.15 metric 100
169.254.0.0/16 dev enp0s3 scope link metric 1000
```

Tablica usmjeravanja može se dobiti i naredbom **ip** ili **route**. Primjer prikazan na prethodnoj stranici može se objasniti ovom tablicom:

Redni broj	Ciljano odredište	Paket će se prosljediti	Na kojem sučelju
1	ostalo	10.0.2.2	enp0s3
2	ostalo	10.0.3.2	enp0s8
3	10.0.2.0/24	Izravno na cilj	enp0s3
4	10.0.3.0/24	Izravno na cilj	enp0s8
5	169.254.0.0/16	nije definirano	enp0s3

Uočimo dvije svari:

- Postoje dva pravila (treće i četvrto) koja računalu govore da pakete koji su slani na mrežu na kojem je računalo izravno spojeno ne usmjeravaju nikamo. Ti se paketi dostavljaju izravno na odredište.
- Pravilo pod brojem pet je beskorisno jer šalje pakete za određenu mrežu na glavni usmjernik, a svi paketi za koje ne postoji pravilo ionako se šalju preko njega.

Pravila se usmjeravanja paketa (u nastavku teksta kraće referencirana kao rute) mogu postaviti naredbom **route** ili **ip**. Dodat ćemo rute čije je odredište mreža **192.168.55.0/24** preko usmjernika **192.168.1.254**. Primjeri su:

```
# ip route add 192.168.55.0/24 via 192.168.1.254 dev enp0s1
# route add -net 192.168.55.0 netmask 255.255.255.0 gw 192.168.1.254 dev
enp0s1
```

Rute dodane naredbama trajati će do prvog ponovnog pokretanja sustava. Kad su rute stalno potrebne za rad tada se postavljaju u datoteci **/etc/network/interfaces**.

Za gornji primjer treba dodati dvije linije u datoteku **/etc/network/interfaces**:

```
up route add -net 192.168.55.0 netmask 255.255.255.0 gw 192.168.1.254
down route del -net 192.168.55.0 netmask 255.255.255.0 gw 192.168.1.254
```

Kao što je vidljivo u primjeru, potrebno je unijeti pravilo za dodavanje rute pri pokretanju sučelja i pravilo za uklanjanje pri gašenju sučelja. Bez pravila za uklanjanje rute svakim ponovnim pokretanjem mrežnog sučelja nastala bi pogreška zbog pokušaja unosa nove (postojeće) rute.

6.4.2. Mijenjanje i konfiguracija glavnog usmjernika

U svakoj tablici usmjeravanja mora biti imenovan glavni usmjernik (*default gateway*) kojem će se prosljeđivati svi paketi koji nisu zahvaćeni ni jednim drugim pravilom u tablici usmjeravanja. Naredbe su za dodavanje:

```
# ip route add default via 192.168.1.254
# route add default gw 192.168.1.254 enp0s0
```

A isto je pravilo za postavljanje glavnog usmjernika pri svakom pokretanju sučelja koje se dodaje u datoteku **/etc/network/interfaces**:

```
gateway 192.168.1.254
```

Budući da je teško paziti na pogreške pri unosu svih brojeva, u gornjim je naredbama omogućeno imenovanje mreža. Za imenovanje aliasa mreža rabi se datoteka **/etc/networks** koja sadrži zapise formata:

```
„Ime mreže“    „mrežna adresa“ „lista aliasa“
```

Neka na primjer postoji zapis:

```
Sigurna_lokalna_mreza 192.168.55.0
```

u **/etc/networks**. Tada pravilo usmjeravanja iz prethodnog poglavlja može biti:

```
# route add -net Sigurna_lokalna_mreza netmask 255.255.255.0 gw
192.168.1.254 dev enp0s1
```

6.4.3. Dodatni sadržaji

https://en.wikipedia.org/wiki/Routing

6.5. Osnovni mrežni alati

6.5.1. Naredba ping

Postoje brojne naredbe za prikupljanje podataka u svrhu pronalaženja pogrešaka u mrežnoj konfiguraciji.

Naredba **ping** šalje datagram **ICMP ECHO_REQUEST** udaljenom računalu i kao odgovor očekuje **ICMP ECHO_RESPONSE**. U žargonu se poziv **ping** naredbe za udaljeno računalo naziva pinganjem.

Sintaksa je naredbe:

```
ping [opcije] odredište
```

Bez dodatnih opcija naredba **ping** šalje niz paketa do signala za prekid (**CTRL+C**).

Najvažnije su opcije:

Opcija	Opis
-b	„Pinganje" adrese razašiljanja.
-c N	Slanje upita N.
-q	Minimalni izlaz na ekran.
-i N	Zadaje se interval n između slanja paketa.
-n	Samo brojčani prikaz – ne pokušava se provesti razrješenje imena.

6.5.2. Naredba tcpdump

tcpdump je analizator paketa koji se pokreće iz naredbene linije. Pomoću te se naredbe prikazuju **TCP/IP** i drugi paketi koji prelaze (u i iz računala) preko nekog mrežnog sučelja. Alat se ne instalira kao dio osnovne instalacije i treba ga posebno instalirati.

U primjeru u nastavku rezultat naredbe **tcpdump** s opcijom **–v** prikazuje promet paketa za sučelje koje naredba prepozna kao primarno. Pokretanjem naredbe postaje jasno koliko podataka prolazi preko jednog sučelja:

```
# tcpdump -v
tcpdump: listening on enp0s0, link-type EN10MB (Ethernet), capture size 65535 bytes
10:32:06.301001 IP (tos 0x0, ttl 64, id 36168, offset 0, flags [DF], proto UDP (17),
length 63)
 debian-1.local.46587 > stribor21.intranet2.srce.hr.domain: 11869+ A? ftp.hr.debian.org.
(35)
10:32:06.302965 IP (tos 0x0, ttl 64, id 27488, offset 0, flags [none], proto UDP (17),
length 79)
 stribor21.intranet2.srce.hr.domain > debian-1.local.46587: 11869 1/0/0
ftp.hr.debian.org. A 161.53.160.11 (51)
10:32:06.304714 IP (tos 0x0, ttl 64, id 36169, offset 0, flags [DF], proto UDP (17),
length 72)
 debian-1.local.42972 > stribor21.intranet2.srce.hr.domain: 12589+ PTR? 36.252.53.161.in-
addr.arpa. (44)
10:32:06.305388 IP (tos 0x0, ttl 64, id 36170, offset 0, flags [DF], proto UDP (17),
length 65)
 debian-1.local.44934 > stribor21.intranet2.srce.hr.domain: 62225+ A?
security.debian.org. (37)
10:32:06.305432 IP (tos 0x0, ttl 64, id 36171, offset 0, flags [DF], proto UDP (17),
length 65)
 debian-1.local.44934 > stribor21.intranet2.srce.hr.domain: 57757+ AAAA?
security.debian.org. (37)
10:32:06.305626 IP (tos 0x0, ttl 64, id 27489, offset 0, flags [none], proto UDP (17),
length 113)
 stribor21.intranet2.srce.hr.domain > debian-1.local.42972: 12589* 1/0/0
36.252.53.161.in-addr.arpa. PTR stribor21.intranet2.srce.hr. (85)
10:32:06.306077 IP (tos 0x0, ttl 64, id 36172, offset 0, flags [DF], proto UDP (17),
length 68)
 debian-1.local.42079 > stribor21.intranet2.srce.hr.domain: 55804+ PTR? 15.2.0.10.in-
```

```
addr.arpa. (40)
10:32:06.306308 IP (tos 0x0, ttl 64, id 36173, offset 0, flags [DF], proto UDP (17),
length 63)
 debian-1.local.46587 > stribor21.intranet2.srce.hr.domain: 36112+ AAAA?
ftp.hr.debian.org. (35)
10:32:06.308594 IP (tos 0x0, ttl 64, id 27490, offset 0, flags [none], proto UDP (17),
length 113)
.
.
.
(CTRL+c)
^C
81 packets captured
85 packets received by filter
4 packets dropped by kernel
```

U prethodnom je primjeru naredba izvršena u tek nekoliko sekundi. Bez parametara naredba će pokušati prepoznati glavno mrežno sučelje i slušati promet na tom sučelju.

Druge su važnije opcije:

Opcija	Opis
-i <sučelje>	Zadaje na kojem sučelju treba hvatati pakete.
-c N	Izlazi nakon paketa N.
-v, -vv, -vvv	Tri razine dodatno detaljnog ispisa, slijedom prema najdetaljnijem.
-q	Manje detaljni ispis. Izbjegava se prikaz previše detalja o korištenim protokolima.

Na kraju naredbe može se zadati predložak za prepoznavanje (neobavezno). Naredba tada ispisuje samo pakete koji zadovoljavaju zadani predložak. Moguće je filtrirati prema odredištu, izvoru, vrsti paketa, protokolu i slično. Svi oblici predložaka ispisani su u man-stranici.

Primjer je filtriranja samo paketa sa izvorištem **10.0.2.15.**

```
# tcpdump -n src host 10.0.2.15
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on enp0s0, link-type EN10MB (Ethernet), capture size 65535 bytes
10:54:46.396303 IP 10.0.2.15.44086 > 161.53.252.36.53: 32462+ A?
ftp.hr.debian.org. (35)
10:54:46.408134 IP 10.0.2.15.46493 > 161.53.252.36.53: 29030+ A?
security.debian.org. (37)
10:54:46.410782 IP 10.0.2.15.46493 > 161.53.252.36.53: 29824+ AAAA?
security.debian.org. (37)
10:54:46.413289 IP 10.0.2.15.60515 > 212.211.132.250.80: Flags [S], seq
1892551070, win 14600, options [mss 1460,sackOK,TS val 21840325 ecr 0,nop,wscale
4], length 0
10:54:46.414211 IP 10.0.2.15.44086 > 161.53.252.36.53: 57854+ AAAA?
ftp.hr.debian.org. (35)
```

6.5.3. Naredba netstat

netstat (*network statistics*) je naredbeno-linijski alat za prikaz mrežnih veza, tablica usmjeravanja i statistika mrežnih sučelja.

Najvažnije su opcije:

Opcija	Opis
-r	Ispis tablice usmjeravanja.
-i	Ispis aktivnih sučelja i minimalne statistike o njihovoj aktivnosti.
-n	Isključuje rezolucije imena u IP-u.
-p	Prikazuje PID i vlasnika procesa.
-v	Prikazuje detaljni ispis.
-c	Neprestano izvođenje naredbe.
-g	Prikaz informacije o pripadnosti skupinama multicast.
-l	Ispis portova koji trenutačno slušaju.
-t	Prikaz samo komunikacije preko protokola TCP.
-u	Prikaz samo komunikacije preko protokola UDP.

Primjer:

```
# netstat --inet -n
Active Internet connections (w/o servers)
Proto Recv-Q Send-Q Local Address Foreign Address State
tcp 0 0 10.0.2.15:35632 161.53.160.11:80 ESTABLISHED
tcp 0 0 10.0.2.15:60572 212.211.132.250:80 ESTABLISHED
```

Važno je napomenuti da je alat **netstat** zastario i da je poput **ifconfig** alata dio opcionalnog **net-tools** paketa.

U tablici su ispisane nove naredbe.

Zastarjele naredbe	Nove naredbe
netstat	ss, ip -s
netstat –r	ip route
netstat –i	ip –s link
netstat -g	ip maddr

6.5.4. Naredba arp

Alat **arp** prikazuje zapise o rezoluciji adresa koje se nalaze u privremenoj memoriji jezgre (*cache*), dakle privremenu memoriju ARP-a (*Address Resolution Protocol*). Alat je zastario i zamijenjen je naredbom **ip neigh**.

Primjeri izvođenja naredbi za prikaz ARP *cache*a:

```
# ip neigh
10.0.2.2 dev enp0s0 lladdr 52:54:00:12:35:02 STALE
```

```
# arp
Address HWtype HWaddress Flags Mask Iface
10.0.2.2 ether 52:54:00:12:35:02 C enp0s0
```

6.5.5. Naredba lsof

Ime naredbe **lsof** akronim je od ***l**ist **o**pen **f**iles* i naredba ima znatno širu primjenu od nadzora aktivnosti mreže.

Primjeri poziva naredbe za pregled stanja mreže:

Naredba	Opis
lsof –i	Ispis svih otvorenih mrežnih veza
lsof –i –a –p N	Ispis svih otvorenih mrežnih veza koje pripadaju procesu N
lsof -i :N	Ispis svih procesa koji slušaju na portu N
lsof –i tcp, lsof –i udp	Sve mrežne veze protokola TCP/UDP.

Primjer izvođenja naredbe **lsof** za prikaz portova koji komuniciraju preko protokola **TCP**:

```
# lsof -i TCP
COMMAND PID USER FD TYPE DEVICE SIZE/OFF NODE NAME
rpcbind 1763 root 8u IPv4 4932 0t0 TCP *:sunrpc (LISTEN)
rpcbind 1763 root 11u IPv6 4939 0t0 TCP *:sunrpc (LISTEN)
rpc.statd 1794 statd 8u IPv4 4996 0t0 TCP *:54192 (LISTEN)
rpc.statd 1794 statd 10u IPv6 5004 0t0 TCP *:59253 (LISTEN)
master 2653 root 12u IPv4 6054 0t0 TCP localhost:smtp (LISTEN)
master 2653 root 13u IPv6 6056 0t0 TCP localhost:smtp (LISTEN)
sshd 5333 root 3u IPv4 15777 0t0 TCP *:ssh (LISTEN)
sshd 5333 root 4u IPv6 15779 0t0 TCP *:ssh (LISTEN)
pump 5608 root 0u IPv4 17215 0t0 TCP *:bootpc (LISTEN)
nc 7377 root 3u IPv4 46371 0t0 TCP *:36476 (LISTEN)
```

6.5.6. Traceroute

Alat **traceroute** prikazuje put od lokalnog do udaljenog računala. Alat radi tako da postavi nerealno mali **ttl** (*time to live*) na pakete tjerajući usmjerivače na putu paketa da pošalju poruku o pogrešci (ICMP TIME_EXCEEDED). Ttl se postupno povećava dok se ne stigne do odredišta.

Primjer izvođenja naredbe **traceroute** prema poslužitelju **google.com**:

```
# traceroute google.com
traceroute to google.com (208.117.229.183), 30 hops max, 60 byte packets
1 161.53.254.1 (161.53.254.1) 89.174 ms 89.094 ms 89.054 ms
2 193.198.229.181 (193.198.229.181) 109.037 ms 108.994 ms 108.963 ms
3 193.198.228.193 (193.198.228.193) 108.937 ms 121.322 ms 121.300 ms
4 * 193.198.228.201 (193.198.228.201) 113.754 ms 113.692 ms
5 cache.google.com (208.117.229.183) 123.585 ms 123.558 ms 123.527 ms
```

6.5.7. Naredba netcat

netcat je svestran alat kojem je prva namjena omogućiti jednostavno uspostavljanje **TCP** ili **UDP** veza između računala. Skraćeni poziv naredbe je **nc.** Najjednostavnije se rabi tako da se na jednom računalu stavi u način rada slušanja na portu (*listen*):

```
Racunalo1 # nc -l 11233
```

Zatim se pomoću iste naredbe s udaljenog računala na taj port poveže naredbom:

```
Racunalo2 # nc racunalo1 11233
```

Nakon uspostavljanja veze sve što se ispiše na računalu 2 pojaviti će se i na računalu 1. U primjeru je pozdrav upisan samo na računalu 2:

```
Racunalo2 # nc racunalo1 11233
Pozdrav svijetu
Racunalo1 # nc -l 11233
Pozdrav svijetu
```

Alat se može i rabiti za provjeru dostupnosti portova na udaljenom poslužitelju. Jednostavno se postavi upit i definira se **IP** ili ime udaljenog računala (**google.hr**) i port(**80**):

```
# nc -vz google.hr 80
google.hr [208.117.229.181] 80 (http) open
```

6.5.8. Dodatni sadržaj

- http://linux.die.net/man/8/ping
- http://www.tcpdump.org/tcpdump_man.html
- https://en.wikipedia.org/wiki/Netstat
- http://linux.die.net/man/8/netstat
- https://en.wikipedia.org/wiki/Iproute2
- https://en.wikipedia.org/wiki/Lsof

6.6. Vježba: Ručno postavljanje mrežnih parametara

1. Prijavite se na računalo kao korisnik l102. U GUI-u pokrenite **Terminal** (*Activities* → **Terminal**).
2. Naredbom **su -** postanite **root** korisnik (lozinka: L102).
3. Provjerite mrežnu konfiguraciju naredbama **ip** ili **ifconfig**. Koliko je uređaja aktivno i s kojim primarnim adresama?

4. Provjerite poklapaju li se zapisi u središnjoj konfiguracijskoj datoteci mrežnih sučelja s podacima prikupljenim naredbama u prethodnom zadatku. Kako je to moguće?

5. U datoteci **/etc/network/interfaces** dodajte postavke za virtualno sučelje koje je aktivno sa nastavkom **:0** tako da se na tom mrežnom sučelju rabi adresa za 100 viša od adrese koja se nalazi na aktivnom sučelju. Dakle, ako je ime sučelja **enp0s3** i adresa **10.0.2.15/24**, tada na sučelju podesite **enp0s3:0 10.0.2.115/24**.

6. Naredbom **ping** testirajte je li aktivno sučelje **:0**.

```
# ping 10.0.2.115
```

7. Ponovno pokrenite servis **network-online**. Provjerite ponovno mrežnu konfiguraciju. Je li se išta promijenilo? Zašto?

```
# systemctl restart network-online
```

8. Ponovno pokrenite servis **networking**. Provjerite ponovno mrežnu konfiguraciju. Je li se išta promijenilo? Zašto?

9. Naredbom ping ponovno testirajte je li sučelje **:0** aktivno.

```
# ping 10.0.2.115
```

10. Kliknite u gornjem desnom kutu ekrana na ikonu za upravljanje mrežnim postavkama i proučite mogućnosti izbornika **Network Settings**. Uključite mrežnu karticu onog sučelja koje nije uključeno klikom na profil koji nije ukljucen na aktivnoj kartici i provjerite kako je konfigurirana (**Wired Settings** → **settings (*zupčanik na profilu*)** → **IPv4 Tab**).

11. Ručno podesite postavke u GUI-u tako da se sučelju dodijeli adresa za **200** veća od adrese prvog sučelja. Osim adrese sve druge postavke postavite tako da su identične sučelju koje je inicijalno bilo upaljeno.
12. Testirajte postavljeno sučelje pomoću naredbe ping. Radi li?

13. Ponovno pokrenite servis **network-online**. Provjerite ponovno mrežnu konfiguraciju. Je li se išta promijenilo? Zašto?

14. Testirajte je li dostupna vanjska mreža. Je li dostupna?

```
# ping www.google.com
```

15. Ispišite na ekran tablicu usmjeravanja paketa.

16. Iz naredbene linije spustite sučelje koje je inicijalno bilo aktivno. Provjerite koja su sad sučelja aktivna.

17. Provjerite sad u GUI-u stanje mreže. U kojem je stanju mrežna kartica na kojoj je konfigurirano sučelje **enp0s3**? Provjerite radi li sada mreža pokušajem pristupa udaljenom računalu?

```
# ping www.google.com
```

18. Pokrenite ponovno sučelje koje je inicijalno bilo aktivno. Radi li sad mreža?

Napomena

Neke naredbe čitaju konfiguracijske datoteke u **/etc/networks** za detalje pri spuštanju sučelja. Njima nije moguće spustiti sučelja konfigurirana u GUI-u.

7. Osnove administracije poslužitelja

Trajanje poglavlja:
110 min

Po završetku ovoga poglavlja moći ćete:

- pronaći zapise o događajima na sustavu u direktoriju **/var/log/**
- proučiti i podesiti ponašanje sustava za upravljanje sistemskim zapisima
- simulirati sistemske događaje i tako provjeriti konfiguraciju pozadinskog procesa **rsyslog**
- podesiti i izraditi nova pravila za upravljanje sistemskim zapisima
- napraviti raspored za kasnije izvršavanje redovnih zadataka pomoću **cron**a
- zadati odgođeno pokretanje zadatka naredbom **at**
- izraditi arhivu i povratiti datoteke iz arhive
- koristiti **rsync** za učinkovitu lokalnu ili udaljenu sigurnosnu pohranu.

Ova cjelina obrađuje upravljanje, analizu i manipulaciju sistemskim zapisima. Naučit ćemo i izraditi sigurnosnu kopiju podataka, komprimirati je i pohraniti ručno ili automatski.

7.1. Sistemski zapisi i konfiguracijske datoteke poslužitelja

7.1.1. Konfiguracija i smještaj sistemskih zapisa

Osnovne su zadaće administratora računalnih poslužitelja:

- planiranje
- priprema
- instalacija hardvera
- održavanje (proaktivno)
- nadzor
- instalacija nadogradnji i uklanjanje softvera
- upravljanje sigurnosnim kopijama i arhivama
- konfiguracija
- rješavanje problema/uklanjanje nastalih pogrešaka
- održavanje dokumentacije
- podrška korisnicima
- uspostavljanje osnovnih odrednica.

Mnogi se od navedenih zadataka obavljaju uz uređivanje datoteka u direktoriju **/etc/** ili čitanje i tumačenje sistemskih zapisa iz direktorija **/var/log/**.

U direktoriju **/var/log/** nalaze se sistemski zapisi jezgre i svih programa koji su zamišljeni da se pokreću sa ovlastima administratora poslužitelja. Standardne su datoteke u tom direktoriju:

- **messages** – glavna datoteka sistemskih zapisa . Uključuje zapise o akcijama pri pokretanju/zaustavljanju računala. Zapisi **mail**, **cron**, **daemon**, **kern** i **auth** također se mogu nalaziti u ovoj datoteci iako imaju i svoje datoteke. U osnovnim postavkama na Debianu 9 se **mail**, **cron**, **daemon** i **auth** zapisi ne pohranjuju u datoteku **messages** .
- **dmesg** – datoteka sa zadnjim aktivnostima jezgre. Zapisi funkcioniraju na principu FIFO (*First In First Out*) pa je korisna samo ako nema gomile nepotrebnih unosa (aktualna do Debian verzije 9.0).
- **auth.log** – autorizacijske akcije u sustavu.
- **boot.log** – zapisi iz postupka pokretanja sustava (ne mora postojati od Debian verzije 9.0).
- **daemon.log** – zapisi raznih pozadinskih procesa.
- **dpkg.log** – zapisi o instalaciji i uklanjanju paketa naredbom **dpkg**.
- **kern.log** – zapisi o aktivnosti jezgre.
- **maillog ili mail.log** – aktivnosti servisa za elektroničku poštu.
- **xorg.x.log** – aktivnosti X servera (*GUI server*).
- **cron** – sve aktivnosti raspoređivača poslova **cron** (ne mora postojati od Debian verzije 9.0).
- **secure** – svi zapisi o aktivnostima vezanim uz autentikaciju i autorizaciju, uključujući i neuspješne pokušaje.

Većinom datoteka sistemskih zapisa upravlja pozadinski proces **rsyslog**. Kad se pokrene, on čita konfiguracijsku datoteku **/etc/rsyslog.conf**. Zapisi u toj datoteci su ovog oblika:

```
tipX.razinaX; tipY.razinaY  /log/datoteka/za/pohranu.log
```

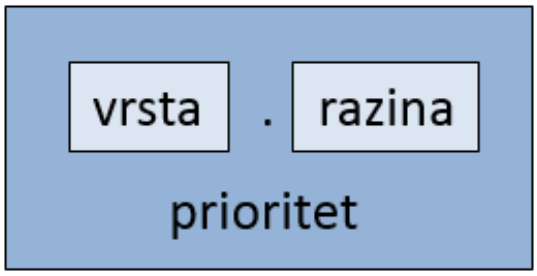
Tip može imati jednu od ovih vrijednosti:

Tip	Opis
auth	autentikacija na sustavu
authpriv	privatna autentikacija
cron	poruke cron pozadinskog procesa
daemon	poruke ostalih pozadinskih procesa
kern	poruke jezgre
lpr	podsustav za linijske pisače
mail	poruke servisa elektoničke pošte
mark	interne poruke (samo za testiranje)
news	podsustav network news
security	zastarjelo – treba rabiti auth
syslog	poruke o aktivnostima središnjeg upravitelja datotekama sistemskih zapisa sustava
user	aktivnosti korisničkih procesa
uucp	podsustav za kopiranja među Unix sustavima
local0 do local7	

Razine mogu biti (od najniže):

Razina	Opis
debug	Najniža razina, sve aktivnosti se bilježe u svrhu pronalaženja pogrešaka u konfiguraciji.
info	Informativni zapisi o aktivnosti - ne indiciraju promjene stanja ili pogreške.
notice	Informativni zapisi o manje standardnim aktivnostima poput pokretanja/zaustavljanja servisa.
warning (ili **warn**)	Upozorenja o mogućim problemima u sustavu.
err (ili **error**)	Pogreške (ne kritične pogreške koje neće rezultirati zaustavljanjem servisa ili zaustavljanjem rada sustava).
crit	Kritične pogreške koje uzrokuju zaustavljanje rada (servisa ili računala). Pogreške od kojih se servis ne može oporaviti.
alert	Ozbiljne pogreške koje mogu rezultirati zaustavljanjem sustava.
emerg (ili **panic**)	Kritične pogreške u radu jezgre ili hardverskih komponenti.

Uređeni par tip i razina naziva se **prioritet**.



Kod razina je važno znati poredak jer se u datoteku sistemskih zapisa prema nekom pravilu bilježe sve poruke u pravilu imenovane razine i svih viših razina.

Tako, na primjer, linija

```
auth.err            /var/log/auth_min_err.log
```

uzrokuje da se zapisi tipa **auth** i razina **err**, **error**, **crit**, **alert**, **emerg** i **panic** zapisuju u datoteku **/var/log/aut_min_err.log**.

Kada se žele zapisati samo zapisi točno određene razine, bez viših razina, potrebno je dodati znak = (jednako) prije razine:

```
auth.=err          /var/log/auth_min_err.log
```

Konfiguracija prihvaća i poseban znak * kao zamjenu za sve tipove ili razine:

```
auth.*             /var/log/auth_sve.log # Svi događaji tipa auth
*.=err              /var/log/err_sve.log # Svi događaji razine err
*.*                /var/log/sve.log      # Svi događaji
```

Sve promjene nad datotekom **rsyslog.conf** postaju aktivne tek pri ponovnom pokretanju servisa.

Načini pokretanja servisa

- Servisi se pokreću ovako:
  - systemctl start ime_servisa
  - na neki nestandardni način.

7.1.2. Primjer datoteke rsyslog.conf

U nastavku donosimo primjer datoteke **rsyslog.conf** iz koje se može iščitati kako standardno izgleda jedna datoteka **rsyslog.conf.**

```
# cat /etc/rsyslog.conf
#  /etc/rsyslog.conf    Configuration file for rsyslog.
#
#            For more information see
#            /usr/share/doc/rsyslog-doc/html/rsyslog_conf.html


#################
#### MODULES ####
#################

$ModLoad imuxsock # provides support for local system logging
$ModLoad imklog   # provides kernel logging support
#$ModLoad immark  # provides --MARK-- message capability

# provides UDP syslog reception
#$ModLoad imudp
#$UDPServerRun 514

# provides TCP syslog reception
#$ModLoad imtcp
#$InputTCPServerRun 514


###########################
#### GLOBAL DIRECTIVES ####
###########################
```

```
#
# Use traditional timestamp format.
# To enable high precision timestamps, comment out the following line.
#
$ActionFileDefaultTemplate RSYSLOG_TraditionalFileFormat

#
# Set the default permissions for all log files.
#
$FileOwner root
$FileGroup adm
$FileCreateMode 0640
$DirCreateMode 0755
$Umask 0022

#
# Where to place spool and state files
#
$WorkDirectory /var/spool/rsyslog

#
# Include all config files in /etc/rsyslog.d/
#
$IncludeConfig /etc/rsyslog.d/*.conf


################
#### RULES ####
################
```

```
#
# First some standard log files.  Log by facility.
#
auth,authpriv.*            /var/log/auth.log
*.*;auth,authpriv.none        -/var/log/syslog
#cron.*                /var/log/cron.log
daemon.*            -/var/log/daemon.log
kern.*                -/var/log/kern.log
lpr.*                -/var/log/lpr.log
mail.*                -/var/log/mail.log
user.*                -/var/log/user.log

#
# Logging for the mail system.  Split it up so that
# it is easy to write scripts to parse these files.
#
mail.info            -/var/log/mail.info
mail.warn            -/var/log/mail.warn
mail.err            /var/log/mail.err
```

```
#
# Logging for INN news system.
#
news.crit            /var/log/news/news.crit
news.err            /var/log/news/news.err
news.notice            -/var/log/news/news.notice

#
# Some "catch-all" log files.
#
*.=debug;\
    auth,authpriv.none;\
    news.none;mail.none    -/var/log/debug
*.=info;*.=notice;*.=warn;\
    auth,authpriv.none;\
    cron,daemon.none;\
    mail,news.none        -/var/log/messages

#
# Emergencies are sent to everybody logged in.
#
*.emerg                :omusrmsg:*

#
# I like to have messages displayed on the console, but only on a virtual
# console I usually leave idle.
#
#daemon,mail.*;\
#    news.=crit;news.=err;news.=notice;\
#    *.=debug;*.=info;\
#    *.=notice;*.=warn    /dev/tty8

# The named pipe /dev/xconsole is for the `xconsole' utility.  To use it,
# you must invoke `xconsole' with the `-file' option:
#
#    $ xconsole -file /dev/xconsole [...]
#
# NOTE: adjust the list below, or you'll go crazy if you have a reasonably
#      busy site..
#
daemon.*;mail.*;\
    news.err;\
    *.=debug;*.=info;\
    *.=notice;*.=warn    |/dev/xconsole
```

7.1.3. Dodatni sadržaji

- http://man7.org/linux/man-pages/man5/rsyslog.conf.5.html

7.2. Alati za rad sa sistemskim zapisima

7.2.1. Naredba logger

Naredba **logger** rabi se za testiranje zapisivanja u središnje datoteke sistemskih zapisa. Bez dodatnih opcija naredba **logger** generira događaje koji rezultiraju upisom zapisa u datoteku **/var/log/messages**. Opcijom **–p** zadaje se prioritet. Tako je moguće ciljano simulirati određene sistemske događaje i provjeriti rad servisa **rsyslog**.

Primjer koji prvo prikazuje generiranje događaja bez opcija, a zatim pomoću opcije **-p** opcijom zadajemo druge opcije:

```
# logger –p local0.notice     "ovo je testna poruka"
# tail -1 /var/log/messages
Jul  3 08:50:01 debian-1 root: –p local0.notice "ovo je testna poruka"

# logger –p auth.crit    "Testiranje lažnog predstavljanja"
# tail -1 /var/log/messages
Jul  3 08:53:30 debian-1 root: –p auth.crit "Testiranje lažnog
predstavljanja"
```

7.2.2. Naredba logrotate

Naredba **logrotate** važna je svim aplikacijama. Iako se može izvršavati iz naredbene linije, standardno se naredba **logrotate** poziva unutar skripti *bash* koje poziva neki od (u idućem poglavlju opisanih) alata za automatizaciju.

Sintaksa naredbe **logrotate** je:

```
logrotate [-dv] [-f|--force] [-s|--state datoteka_stanja]
konfiguracijska_datoteka
```

Opcije naredbe **logrotate** su:

Opcija	Opis
-d, --debug	Pokretanje moda za provjeru rada (simulaciju rada). Promjene neće biti provedene nad datotekama sistemskih zapisa.
-f, --force	Prisilna rotacija sistemskih zapisa (čak i kad prema konfiguraciji rotacija nije potrebna).
-m, --mail <naredba> naslov primatelj	Definira se koja će se naredba rabiti za slanje izvještaja o izvršavanju naredbe preko elektroničke pošte, s kojim naslovom i kome (primatelj).
-s, --state <datoteka_stanja>	Definira se nestandardna datoteka za bilježenje stanja rada servisa **logrotate** . Korisna kad više korisnika (ili pod više korisnika) vrši rotaciju sistemskih zapisa.
--usage	Pribavljanje kratkog pregleda poziva naredbe.
--?, --help	Opsežni pregled korištenja naredbe.
-v, --verbose	Uključivanje opsežnog ispisa aktivnosti naredbe.

Naredba izvršava rotaciju (i kompresiju) sistemskih zapisa. Datoteke sistemskih zapisa (tekstualne datoteke općenito) mogu se kompresijom smanjiti i značajno više od reda veličine.

Središnja konfiguracijska datoteka je **/etc/logrotate.conf**.

Format konfiguracije je:

```
<ime_datoteke> {
parametri
}
```

Pravila unutar vitičastih zagrada primjenjuju se na datoteku (ili datoteke) imenovane prije zagrada. U datoteci **/etc/logrotate.conf** moguće je definirati i opća pravila. Pravila definirana specifično za određenu datoteku imaju veći prioritet te će se na primjer globalno pravilo o intervalu rotacije sistemskih zapisa primijeniti samo na datoteke za koje nije definiran interval u lokalnim parametrima (u zagradama).

7.2.3. Primjer /etc/logrotate.conf datoteke

U primjeru donosimo izgled jedne standardne datoteke **/etc/logrotate.conf**.

Primjer datoteke je:

```
# cat /etc/logrotate.conf
# see "man logrotate" for details
# rotate log files weekly
weekly

# keep 4 weeks worth of backlogs
rotate 4

# create new (empty) log files after rotating old ones
create

# uncomment this if you want your log files compressed
#compress

# packages drop log rotation information into this directory
include /etc/logrotate.d

# no packages own wtmp, or btmp -- we'll rotate them here
/var/log/wtmp {
    missingok
    monthly
    create 0664 root utmp
    rotate 1
}

/var/log/btmp {
```

```
    missingok
    monthly
    create 0660 root utmp
    rotate 1
}
```

U primjeru je vidljivo da je konfiguracija podijeljena na središnji dio, koji opisuje općenita pravila za rad servisa, i na segmente ranije opisanog oblika.

7.2.4. Primjer direktorija /etc/logrotate.d/

Direktorij **/etc/logrotate.d/** rabi se za smještanje pravila rotiranja sistemskih zapisa pojedinih aplikacija. Praksa je jedna aplikacija - jedna konfiguracijska datoteka:

```
# ls /etc/logrotate.d/
apt
dpkg
kdm
rsyslog
aptitude
exim4-base
pm-utils
speech-dispatcher
consolekit
exim4-paniclog
ppp
unattended-upgrades
```

U primjeru je prikazan sadržaj direktorija **/etc/logrotate.d/** iz kojeg je vidljivo da postoje konfiguracijske datoteke za aplikacije apt, dpkg, kdm i sl.

U konfiguraciji je moguće zadati ovlasti i vlasništvo nad datotekama sistemskih zapisa koje se izrađuju pri rotaciji. Također je moguće napisati skripte za pokretanje/zaustavljanje servisa prije/nakon rotacije.

U primjeru koji slijedi su pravila za izradu nove datoteke sistemskih zapisa s pravima **0644** u vlasništvu **puppet:puppet**. Konfiguracija nalaže „ubijanje" procesa i ponovno pokretanje procesa **puppet**:

```
# less puppet
/var/log/puppet/*log {
  missingok
  sharedscripts
  create 0644 puppet puppet
  compress
  rotate 4

  postrotate
    pkill -USR2 -u puppet -f 'puppet master' || true
```

```
    [ -e /etc/init.d/puppet ] && /etc/init.d/puppet reload > /dev/null
2>&1 || true
  endscript
}
```

Kad su postavljene sve skripte, rotacija svih sistemskih zapisa može se izvesti naredbom:

```
# /usr/sbin/logrotate /etc/logrotate.conf
```

7.2.5. Dodatni sadržaji

- http://man7.org/linux/man-pages/man1/logger.1.html
- https://www.digitalocean.com/community/tutorials/how-to-manage-log-files-with-logrotate-on-ubuntu-12-10

7.3. Automatizacija

7.3.1. Servis cron

Cron je softverski alat za vremensko raspoređivanje zadataka. Alat se uglavnom rabi za vremensko planiranje izvršavanja naredbi i skripti ljuske. **Cron** radi tako da svake minute čita specijalizirane datoteke u kojima se nalazi raspored zadataka koji moraju biti izvršeni. Takva datoteka zove se **crontab**.

Korisničke datoteke **crontab** nalaze se u direktoriju **/var/spool/cron/**, ali njihovo uređivanje ne povodi ručno već pomoću naredbe **crontab**. Format zapisa u datotekama je:

```
minute(0-59) sati(0-23) dan_u_mjesecu(1-31) mjesec(1-12) dan_u_tjednu(0-6) naredba
```

Na primjer:

```
# crontab -l
17 *    * * *   cd / && run-parts --report /etc/cron.hourly
25 6    * * *   test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.daily )
47 6    * * 0   test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.weekly )
52 6    1 * *   test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.monthly )
```

Ovo je prikaz datoteke **crontab** korisnika *root*. Gornje linije konfiguracije pokreću naredbe koje će se izvršiti:

- prva - svaki sat u 17. minuti
- druga - svaki dan u 6:25
- treća - svaki tjedan u nedjelju u 6:47
- četvrta - svakog prvog dana u mjesecu u 6:52.

Ta pravila postoje na svakom sustavu za korisnika *root* i omogućavaju da se u odgovarajući direktorij između **/etc/cron.hourly**, **/etc/cron.daily**, **/etc/cron.weekly**, **/etc/cron.monthly** jednostavno smjeste izvršne datoteke koje se trebaju izvršavati svaki sat, dan, tjedan ili mjesec.

Opcije naredbe **crontab** za uređivanje korisničke datoteke **crontab** su:

Opcija	Opis
-e	uređivanje
-l	ispis sadržaja datoteke
-r	brisanje datoteke

U direktorij **/etc/cron.d/** uobičajeno se smještaju male imenovane datoteke **crontab** za rad određenih servisa. Tim se direktorijem koriste paketi za smještanje datoteka **crontab** potrebnih za rad servisa instaliranih tim paketom.

Za kontrolu toga tko može vremenski raspoređivati poslove pomoću servisa **cron** rabe se datoteke **/etc/cron.allow** i **/etc/cron.deny**.

7.3.2. Naredba at

Naredba **at** rabi se za poziv izvršavanja zadatka (jednom) u nekom trenutku u budućnosti. Nekada je bila dio standardne instalacije, a od Debian verzije 8.0 je dio **at** paketa.

Sintaksa naredbe je jednostavna:

```
at          <vrijeme>
```

Vrijeme može biti niz vremenskih opisa poput **now** za trenutačno izvršavanje ili **3am+2days** za izvršavanje zadatka u 3 sata ujutro za dva dana.

Specifikacija svih podržanih oblika zapisa vremena nalazi se u datoteci **/usr/share/doc/at/timespec**.

Brigu o izvršavanju zadataka vodi pozadinski proces **atd**, a popis zadataka u čekanju nalazi se u direktoriju **/var/spool/at/**.
Naredbom **atq** može se dobiti pregled zadataka na čekanju, a zatim se na osnovi njihova serijskog broja naredbom **atrm** može obrisati njihov poziv.

Primjer postavljanja izvršenja naredbe u nekom budućem trenutku (**at**) i njezino brisanje:

```
# at now + 2 minutes
warning: commands will be executed using /bin/sh
at> echo "ide vrijeme 323" > /tmp/ide_vrijeme
at> <EOT>
# atq
1    Fri Jul  2 15:02:00 2015 a root
# atrm 1
# atq
#
```

7.3.3. Dodatni sadržaji

- https://en.wikipedia.org/wiki/Cron
- http://www.computerhope.com/unix/uat.htm

7.4. Sigurnosna pohrana i kompresija

7.4.1. Naredba tar

Tri osnovne strategije sigurnosne pohrane su:

- **potpuna** - svaka inačica sigurnosne kopije sadrži sve datoteke
- **inkrementalna** - prva sigurnosna kopija sastavljena je od svih datoteka nastalih ili izmijenjenih od zadnje potpune sigurnosne kopije. Nakon toga kopiraju se sve datoteke izrađene ili izmijenjene od zadnje inkrementalne kopije.
- **diferencijalna** - kopiraju se datoteke dodane ili izmijenjene od zadnje potpune sigurnosne kopije.

tar je standardni alat za dodavanje arhiva u *Linux*ovim distribucijama.

Naredba **tar** ima brojne opcije:

Opcija	Značenje
-c ili --create	Dodavanje arhive
-x ili --extract	Izvoz zapisa iz arhive
-u ili --update	Obnavljanje arhive razlikama
-r ili --append	Dodavanje datoteke na kraj arhive
--file=<ime> ili –f <ime>	Odabir imena arhive
-v ili –verbose	Prikaz detalja o izvršavanju naredbe
--C dir ili –directory=dir	Izvršavanje operacije u direktoriju naziva dir
-Z	Provođenje i (de)kompresije (tar.Z)
-z	(De)kompresiranje pomoću **gzip** (.tgz ili tar.gz)
-j	(De)kompresiranje pomoću **bzip2** (tar.bz2 ili tar.tbz ili tar.tb2)

Ako opcija imenovanja arhive **–f** nije navedena tad **tar** šalje rezultat izvršavanja naredbe na standardni izlaz. Ta dva poziva naredbe **tar** provode istu operaciju:

```
# tar -c /etc/ > etc.tar
# tar -cf etc.tar /etc/
```

Datoteke se izdvajaju iz postojeće arhive pomoću opcije **–x** :

```
tar –x etc.tar –C /tmp
```

Kompresija i dekompresija uvijek se provode **istom metodom**, u protivnom nastaje pogreška.

Metoda kompresije prepoznaje se po vrsti arhive:

```
# tar czf etc.tar.gz /etc/
tar: Removing leading `/' from member names
# ls
etc.tar.gz
# tar xjf etc.tar.gz -C .
bzip2: (stdin) is not a bzip2 file.
tar: Child returned status 2
tar: Error is not recoverable: exiting now
# tar xzf etc.tar.gz -C .
# ls
etc
etc.tar.gz
```

7.4.2. Alati cpio i dd

cpio je alat namijenjen za kopiranje datoteka iz arhive i u arhivu. Alat se može rabiti za kopiranje datoteka ili izradu i pristupanje arhivama **tar**. Za rad **cpio** potreban je popis svih datoteka koje treba arhivirati.

Prvi korak u primjeru je pribavljanje imena svih datoteka koje arhiviramo naredbom **find**:

```
# find /etc | cpio -o > etc.cpio
10066 blocks
# ls
etc
etc.cpio
etc.tar.gz
```

Iako bogat opcijama, **cpio** zahtijeva oprez pri dekompresiji arhiva jer bez dodavanja opcije **–d neće** izraditi direktorije te bez opcije **–u neće** prepisati postojeće datoteke.

```
cpio -idvu < etc.tar.gz
```

Alatom **dd** može se napraviti sigurnosna kopija uređaja u kojoj je sačuvano sve, uključujući i datotečni sustav i sektor *boot*. Kod naredbe **dd** sve se radi jednom naredbom samo se mijenja koji je parametar ulazni (*input*), a koji izlazni (*output*).

Sintaksa je:

```
dd if=<uređaj ili direktorij> of=<uređaj ili direktorij>
```

Na primjer, moguće je napraviti sigurnosnu kopiju cijelog uređaja na drugom uređaju:

```
dd if=/dev/sda of=/dev/sdb
```

7.4.3. Alat rsync

Mogućnostima najbogatiji alat za sigurnosnu pohranu je **rsync**. To je alat za udaljeno i lokalno kopiranje datoteka. **rsync** dolazi u istoimenom paketu.

Sintaksa je jednostavna:

```
rync <opcije> <izvor> <odredište>
```

Međutim, bogatstvo opcija i mogućnost optimizacije čine **rsync** moćnim.

Najkorisnija opcija za izradu sigurnosne kopije je **-a** ili **--archive**. Ta opcija je zamjena za opcije **–rlptgoD**:

Opcija	Značenje
-r	rekurzivno
-l	kopiranje simboličkih poveznica
-p	spremi dozvole nad datotekama
-t	spremi vrijeme promjena
-g	spremi skupinu
-o	spremi vlasnika
-D	spremi uređaje i posebne datoteke
-a	ekvivalent opcije -rlptgoD. Namijenjen za arhiviranje pa se tako sa ovom opcijom odjednom postavlja niz opcija koje čuvaju sadržaj arhive i ubrzavaju prijenos kada je to moguće

Primjer izvođenja naredbe za izradu sigurnosne kopije direktorija **/home/l102/Pictures** na dva načina:

```
#ls /tmp/backup_l102/
# rsync -a /home/l102/Pictures/ /tmp/backup_l102/
# ls
Screenshot from 2015-03-24 14:56:46.png Screenshot from 2015-03-24 14:57:49.png
Screenshot from 2015-05-08 15:12:34.png
Screenshot from 2015-03-24 14:56:47.png Screenshot from 2015-03-24 14:57:51.png
Screenshot from 2015-05-08 15:13:27.png
Screenshot from 2015-03-24 14:56:52.png Screenshot from 2015-05-08 15:03:04.png
Screenshot from 2015-05-08 15:16:53.png
Screenshot from 2015-03-24 14:56:57.png Screenshot from 2015-05-08 15:09:04.png
Screenshot from 2015-07-08 15:44:36.png
# rsync -a /home/l102/Pictures /tmp/backup_l102/
# ls
Pictures Screenshot from 2015-03-24 14:57:49.png Screenshot from 2015-05-08
```

```
15:13:27.png
Screenshot from 2015-03-24 14:56:46.png Screenshot from 2015-03-24 14:57:51.png
Screenshot from 2015-05-08 15:16:53.png
Screenshot from 2015-03-24 14:56:47.png Screenshot from 2015-05-08 15:03:04.png
Screenshot from 2015-07-08 15:44:36.png
Screenshot from 2015-03-24 14:56:52.png Screenshot from 2015-05-08 15:09:04.png
Screenshot from 2015-03-24 14:56:57.png Screenshot from 2015-05-08 15:12:34.png
```

Uočimo da kad želimo izraditi kopiju direktorija ne rabimo, a kad želimo izraditi kopiju sadržaja direktorija rabimo završno "/".

Za udaljeno kopiranje naredba podržava uporabu para ključeva za autentikaciju te je jednostavno ostvariti periodičnu izradu sigurnosnih kopija dodavanjem naredbe u **cron**.

Pri procesu kopiranja **rsync** prije kopiranja stvara privremenu bazu metapodataka na računalu na kojem je pozvana naredba. Zato se naredba treba pozivati na računalu koje je manje opterećeno i brže, bilo da je to izvorište ili odredište.

7.4.4. Korisne poveznice

- http://www.thegeekstuff.com/2010/04/unix-tar-command-examples/
- http://linux.die.net/man/1/cpio
- http://www.howtogeek.com/135533/how-to-use-rsync-to-backup-your-data-on-linux/

7.5. Vježba: Upravljanje log datotekama

1. Prijavite se na računalo kao korisnik l102. U GUI-u pokrenite **Terminal** (*Activities* → **Terminal**).
2. Naredbom `su -` postanite **root** korisnik (lozinka: L102).
3. Otvorite dodatni terminal i u njemu postanite root i izvršite naredbu

   ```
   tail -f /var/log/syslog
   ```

   (U nastavku vježbe „drugi terminal“).
4. Izvršite naredbu za ponovno pokretanje ssh servisa (`systemctl restart sshd`). Što se pojavilo na drugom terminalu? Zašto?

   ---

5. Izvršite naredbe

   ```
   ip link set enp0s3 down
   ```

   i

   ```
   ip link set enp0s3 up
   ```

   Što se pojavilo na drugom terminalu? Zašto?

   ---

6. U konfiguracijsku datoteku servisa **rsyslog** dodajte pravilo koje će zapise razine **info** i tipa **local2** zapisivati u datoteku **/var/log/vjezba.log**. Napravite datoteku **/var/log/vjezba.log**. Postavite ovlasti na **644** i vlasništvo **root:adm**. Ponovno pokrenite servis **rsyslog**.

   ---

7. Naredbom **logger** generirajte događaj (event) razine **info** i tipa **local2**. Provjerite sadržaj datoteke **/var/log/vjezba**.

   ---

8. Tijekom izvođenja gornjih naredbi, koji su se događaji zabilježili u drugom terminalu?

7.5.1. Vježba: Sigurnosna pohrana i automatizacija

U ovoj je vježbi zadatak automatizirati sigurnosno kopiranje direktorija **/etc/** naredbom **rsync**. Zadatak će se podijeliti na korake:

1. Prijavite se na računalo kao korisnik l102. U GUI-u pokrenite **Terminal** (*Activities* → **Terminal**).
2. Naredbom **su -** postanite **root** korisnik (lozinka: L102).
3. Naredbom **rsync** napravite sigurnosnu kopiju direktorija **/etc/** u direktoriju **/home/l102**. (Obavezno je rabiti opcije **–v** i **–a**).

4. Unesite pravilo **cron** za korisnika **root** koje jednom dnevno izvodi naredbu iz drugog zadatka.

5. Napišite skriptu BASH ljuske koja radi **tar** arhivu direktorija **/etc/** i pohranjuje je u direktorij **/tmp**.

```
# vi /home/l102/bakup.sh
#!/bin/bash
tar -cvf /tmp/etc.tar /etc/
```

6. Napišite skriptu *BASH* ljuske i smjestite je u odgovarajući direktorij tako da se (samo zato jer je u odgovarajućem direktoriju) izvodi svakih sat vremena. Skripta treba pomoću naredbe **tar** izraditi arhivu direktorija **/etc/** u direktoriju **/home/** i pomoću naredbe **rsync** napraviti sigurnosnu kopiju u direktoriju **/tmp**.

7. Testirajte skriptu ručnim pozivom.

   Razmislite: U čemu je nedostatak ovakvog pristupa izradi sigurnosnih kopija?

8. Mrežni servisi

Trajanje poglavlja:
220 min

Po završetku ovoga poglavlja moći ćete:

- razumjeti osnove hijerarhije **DNS**-a
- izraditi **SOA**-zapise za domenu i uređivati/obnavljati ih po potrebi
- prepoznati razine **DNS**-poslužitelja i znati kako se vrši dodjeljivanje imena
- prepoznati kad je korisno imati super server
- implementirati super servere za servise
- provjeriti dostupnost porta pomoću naredbe **telnet**
- protumačiti i promijeniti postavke servisa **vsftpd**
- ostvariti automatsku autentikaciju korisnika pomoću ključeva
- razumjeti autentikaciju poslužitelja pomoću SSH-a
- provjeriti je li izvršna datoteka programski prevedena s uključenom bibliotekom **libwrap**
- primijenit **TCP Wrapper** mrežni sustav ACL (*Access Control List*)
- podesiti **NFS**-poslužitelj za dijeljenje direktorija
- podesiti **NFS**-klijenta za pristup udaljenom dijeljenom direktoriju
- razumjeti namjenu i način rada servisa Samba te dijeliti i pristupiti dijeljenim resursima
- pokrenuti i zaustaviti servis Samba
- dodati korisnike u servis Samba
- razumjeti rad **NTP**-servisa i stratum
- podesiti **NTP**-servis i ručno podesiti vrijeme na računalu
- razumjeti namjenu i složenost **postfixa**
- podesiti postavke **postfixa** bez izravnog uređivanja konfiguracijskih datoteka
- pregledati, razumjeti i upravljati redom čekanja **postfix**a
- pronaći module i virtualne hostove koji su aktivni u servisu **apache2**
- zaustaviti, pokrenuti i pogledati status servisa **apache2**.

Ova cjelina obrađuje mrežne servise. Naučiti ćemo koji su servisi standardni u Linuxu, koja im je namjena i posebnost te kako se pokreću, zaustavljaju i uređuju.

8.1. DNS servisi

8.1.1. Hijerarhija DNS-a i krovne domene

DNS se koristi **hijerarhijskom strukturom**. U ovisnosti o položaju u FQDN-u (*fully qualified domain name*), domena može biti krovna domena ili domena drugog odnosno trećeg stupnja.

U hijerarhijskoj strukturi svaka je domena zadužena:

- za imenovanje korisnika te domene
- za upravljanje formiranjem poddomena
- za delegiranje autoriteta nad imenima u toj poddomeni.

Na primjer, ako postoji računalo s imenom **test.prodaja.primjer.com**.

- ime računala je **test**
- domena trećeg stupnja je **prodaja**
- domena drugog stupnja je **primjer**
- krovna domena je **com**.

Domena **.com** odobrila je formiranje domene **primjer**. Domena **primjer** odobrila je izradu domene **prodaja**, a domena **prodaja** odobrila je dodjelu imena **test** računalu u svojoj domeni.

Postoje tri vrste krovnih domena:

- izvorne krovne domene (.com, .org, .net, .int, .edu, .gov i .mil)
- infrastrukturne krovne domene (.arpa)
- državne krovne domene (.hr, .ba, .us, .gb, .ru i brojne druge).

Napomena

Termin poslužitelj opisuje računalo, a termin DNS-poslužitelj opisuje računalo na kojem je pokrenut DNS-servis, a koje funkcionira kao DNS-server brojnim DNS-klijentima. Takva logika imenovanja rabi se u daljnjim poglavljima i za druge servise (SSH, NTP, *Postfix*, *Apache* itd.).

8.1.2. DNS-klijent

Resolver je klijentska strana **DNS**-a. **Resolver** je zadužen za inicijalizaciju i provođenje slijeda akcija koje omogućavaju prevođenje imena traženog resursa u njegovu IP-adresu. Taj se postupak se naziva **rezolucija** i ovisno o cilju rezolucije može, ali i ne mora, uključivati **DNS**-poslužitelje.

Datoteka **/etc/nsswitch.conf** je konfiguracijska datoteka za izvore imena u nizu kategorija. U datoteci je definirano uz koje i kojim se redom trebaju rabiti izvori podataka.

Kategorije su:

Ime	Opis
aliases	aliasi elektroničke pošte
group	grupe korisnika
hosts	imena uređaja i IP adrese
passwd	korisničke lozinke
protocols	mrežni protokoli
rpc	pozivi udaljenih procedura
services	mrežni servisi

U datoteci **/etc/nsswitch.conf** može se nalaziti ovaj primjer za postavku imena uređaja i IP-adrese:

```
hosts: files mdns4_minimal [NOTFOUND=return] dns mdns4
```

Gornje pravilo definira da se prvo za rezoluciju imena gledaju lokalne datoteke (*files*), nakon toga za adrese .local pokušava *multikast* **DNS** (*mdns4_minimal*), zatim **DNS** (*dns*) i na kraju **mDNS** (*mdns4*) ako **DNS** ne radi ispravno.

Kad program na računalu treba saznati **IP**-adresu pomoću **DNS**-poslužitelja, on poziva biblioteku imena **resolver.** Biblioteka pročita zapise u datoteci **/etc/resolv.conf** koja sadrži popis **DNS**-poslužitelja.

Primjer datoteke **/etc/resolv.conf** :

```
# cat /etc/resolv.conf
# Generated by pump for interface eth0
search local
nameserver 161.53.252.36
nameserver 161.53.252.37
```

Prema zapisima u datoteci **resolv.conf**, **resolver** odlučuje koji **DNS** pitati (prvi, ako ih više zadovoljava kriterije pretrage) i šalje upit. Nakon toga **resolver** čeka odgovor od **DNS**-poslužitelja.

Datoteka **/etc/hosts** sadrži zapise za rezoluciju **IP** u **FQDN**-u (*fully qualified domain name*) i obratno. U datoteci se nalaze i zapisi o samom računalu. Moguće je u datoteku dodatno smjestiti još zapisa o udaljenim računalima s kojima računalo komunicira, a za koje ne postoje **DNS**-zapisi.

Format zapisa:

```
# cat /etc/hosts
127.0.0.1    localhost
127.0.1.1    debian-1.test.lan    debian-1

# The following lines are desirable for IPv6 capable hosts
::1     localhost ip6-localhost ip6-loopback
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
```

8.1.3. DNS-zona

DNS zona je bilo koji distinktni, izdvojeni i neprekidni dio domenskog prostora u sustavu DNS-a za koji je administrativna odgovornost dodijeljena jednom upravitelju. Domenski je prostor oblikovan u hijerarhijski model sastavljen od poddomena ispod krovnih domena.

Zone imaju dvije vrste DNS-poslužitelja:

1. **master** – autoritet za domenu, jedinstven, izmjene na njemu se propagiraju na poslužitelje **slave**.
2. **slave** – može ih biti više, servisiraju zahtjeve (upite) i periodički obnavljaju vlastite zapise podacima s poslužitelja **master**.

Jedno računalo može u isto vrijeme biti **master** nekim zonama, a **slave** drugim zonama. DNS-servis je **bind9**, a središnja konfiguracijska datoteka DNS-servisa je **/etc/bind/named.conf.**

Primjer je konfiguracijske datoteke:

```
# cat /etc/bind/named.conf
// This is the primary configuration file for the BIND DNS server named.
// If you are just adding zones, please do that in /etc/bind/named.conf.local

include "/etc/bind/named.conf.options";
include "/etc/bind/named.conf.local";
include "/etc/bind/named.conf.default-zones";
```

Kao što je vidljivo, ta datoteka učitava tri dodatne datoteke:

- **named.conf.local** – definicije lokalnih zona.
- **named.conf.default-zones** – osnovne zone tj. zapisi koji vode do vanjskih korijenskih poslužitelja te zapisi za rezoluciju *localhost* i reverznu rezoluciju
- **named.conf.options** – mogućnosti za rad servisa *bind9*.

U tim se datotekama nalaze samo specifikacije za rad servisa. Podaci koji se rabe za odgovaranje na upite nalaze se u datotekama u direktoriju definiranom direktivom **directory** u datoteci:

```
directory "/var/cache/bind";
```

U direktoriju **/var/cache/bind** nalaze se datoteke zona.

DNS-servis na distribuciji *Debian Linux* je **bind**, a ime paketa i servisa je **bind9**. Važno je napomenuti da je ime izvršne datoteke i dalje **named**, tako da je naredba za pokretanje servisa:

```
# systemctl start bind9
```

ali izvršna datoteka koja se pokreće je imena **named.** Primjer:

```
# ps -ef |grep -v grep |grep named
# service bind9 start
[ ok ] Starting domain name service...: bind9.
# ps -ef |grep -v grep |grep named
bind 23217 1 4 15:15 ? 00:00:00 /usr/sbin/named -u bind
#
```

8.1.4. Primjer DNS-zona

Pogledati ćemo i protumačiti jednostavan primjer. U datoteci **named.conf.local** nalazi se popis svih zona za koje je poslužitelj **master** ili **slave**, a zadano je ime datoteke u kojoj se nalazi definicija zone. Također je navedeno koja je uloga poslužitelja (**master** ili **slave**). Primjer dijela konfiguracije **named.conf.local** koja definira zonu „test.lan“:

```
zone "test.lan" IN {
  // ovaj server je autoritet za
  // test.lan podatke
  type master;
  file "zone.test.lan ";
};;
```

Slijedi primjer sadržaja datoteke koja definira zonu. Komentari u **DNS**-konfiguracijskim datotekama počinju s ";":

```
# cat zone.test.lan
; dns zona za test.lan
;
@     IN SOA   debian-1 dnsmaster.test.lan. (
                        201405191 ; serial
                        8H        ; refresh
                        4H        ; retry
                        4W        ; expire
                        1D )      ; minimum
; debian-1.test.lan obavlja na ovoj domeni ulogu i
; NS (name server) i MX (mail exchange)
                NS      debian-1
                MX      10 debian-1
; dodijelimo nekoliko aliasa sa CNAME
skladiste    CNAME   dodo
www         CNAME   doma
; upiti za localhost.test.lan
localhost    A       127.0.0.1
; zapisi o imenima poslužitelja
doma         A       10.11.12.3
dodo         A       10.11.12.2
cuvar         A       10.11.12.1
strijelac    A       10.11.12.4
zeus         A       10.11.12.5
```

Posebni znak **@** tumači se kao ime domene definirano u datoteci **named.conf.local**. U primjeru su prikazane četiri vrste **DNS**-zapisa od ukupno 5.

To su:

Ime zapisa	Opis
NS	Poslužitelj zadužen za domenu (name server).
PTR	Reverzni zapis (za dobivanje imena iz IP adrese).
MX	Zapis o poslužitelju elektroničke pošte (mail exchange).
A	Standardni zapis za rezoluciju imena u IP-u.
CNAME	Definicija aliasa.

8.1.5. SOA

SOA (*start of authoritiy*) je autoritativni zapis o određenoj DNS-zoni uključujući primarni poslužitelj zadužen za domenu, adresu elektroničke pošte administratora, serijski broj i parametre vezane uz valjanost zapisa i vrijeme osvježavanja. U prethodnom je primjeru dana SOA-datoteka zone, što je vidljivo po „SOA“ zapisu.

Struktura SOA-datoteke zone je stroga, a oblik joj je:

```
<ime domene> IN SOA <poslužitelj za rezoluciju> <administrator> (
                        <Serijski broj>;
                        <Refresh>;
                        <retry>;
                        <expire >;
                        <minimum>; ) ;
<Zapisi za rezoluciju>…
```

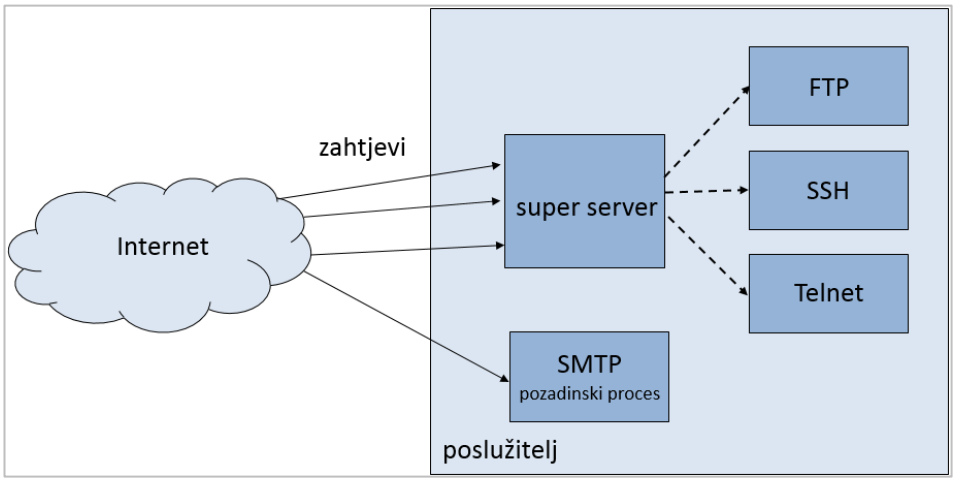
8.1.6. Dodatni sadržaji

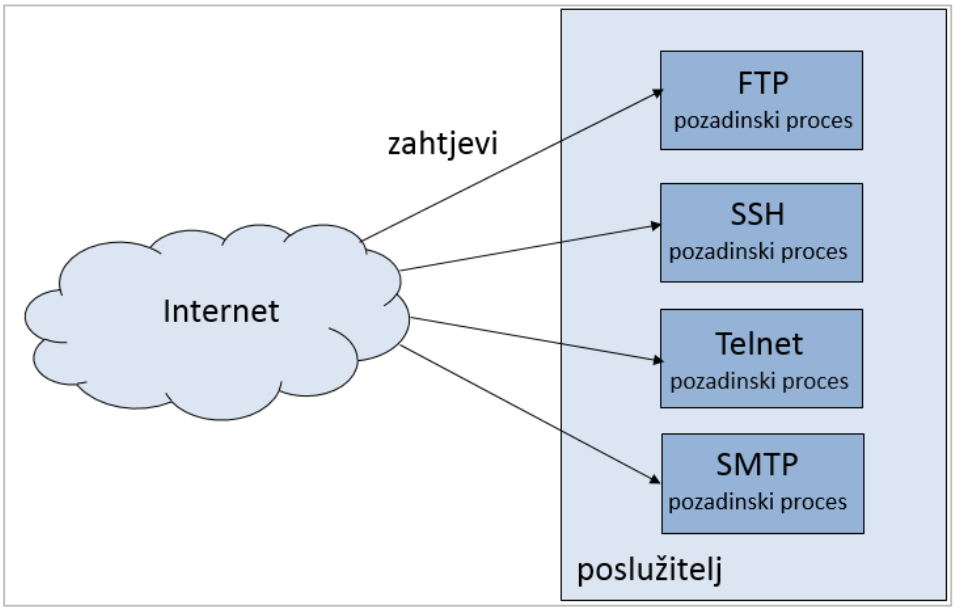
- https://en.wikipedia.org/wiki/List_of_Internet_top-level_domains
- http://www.zytrax.com/books/dns/ch7/

8.2. Super serveri

8.2.1. Super server

Super server je server zadužen za pokretanje servisa na zahtjev. On sluša na portovima za koje je zadužen i pri dolasku zahtjeva u obliku TCP ili UDP-paketa na zadani port pokreće odgovarajući servis koji će poslužiti taj zahtjev.





Prva slika prikazuje kako funkcionira komunikacija prema poslužitelju kad se ne rabi *super server*. Na drugoj slici vidimo da super server prihvaća zahtjeve prema servisima konfiguriranim preko njega (**telnet**, **SSH** i **FTP**), a **SMTP** je i dalje pokrenut kao samostalni servis. Dakle, moguće je kombiniranje: ako je super server aktivan, kroz njega se ne mora odvijati sva komunikacija. Super server samo sluša na portovima servisa za koje je konfiguriran.

Napomena

Potreba za funkcijom koju super serveri obavljaju s vremenom se smanjuje. Razlog tome je sve češće pojavljivanje poslužitelja koji obavljaju samo jednu funkciju. Ako je na poslužitelju standardno pokrenut samo servis **apache2**, tad ne postoji potreba za servisom **xinetd.** Trend **jedan poslužitelj – jedan servis** posebno je uzeo maha u računalnom oblaku gdje je jednostavno napraviti, ukloniti i oblikovati virtualne poslužitelje.

8.2.2. Inetd i xinetd

inetd (*Internet service deamon*) je izvorni *super server* na *Unix* sustavima koji pružaju Internet servise. S vremenom je zbog sigurnosnih propusta u izvornom dizajnu **inetd** napušten i zamijenjen s **xinetd**, **rlinetd** i **ucspi-tcp**.

xinetd (*extended Internet daemon*) je *super server* servis otvorenog kôda za upravljanje IP vezama. **xinetd** je sigurniji od **inetd** i većina modernih Linuxovih distribucija ga koristi.

8.2.3. Konfiguracija xinetd

Konfiguracija se provodi ili preko središnje datoteke **/etc/xinetd.conf** ili korištenjem datoteka u direktoriju **/etc/xinetd.d/**, gdje svaka datoteka opisuje jedan servis.

Primjer konfiguracije za **telnet** u direktoriju **/etc/xinetd.d/**:

```
# cat telnet
service telnet
{
        flags           = REUSE
        socket_type     = stream
        wait            = no
        user            = root
        server          = /usr/sbin/in.telnetd
        log_on_failure  += USERID
        disable         = yes
}
```

Ta konfiguracija definira postavke servisa **telnet**:

Service	**Ime servisa**
Flags	Atributi veze, REUSE govori da se za veze **telnet** ponovno rabi isti *socket*
Socket_type	Postavlja vrstu *socket*a
Wait	*No* za višedretvene (*multi-threaded*) servise, *yes* za jednodretvene (*single-threaded*)
User	Korisnik pod kojim se pokreće servis
Server	Izvršna datoteka za pokretanje servisa
Log_on_failure	Koji će se dodatni parametri bilježiti u slučaju pogreške
Disable	Je li server aktivan *(no* – aktivan, *yes* – nije aktivan)

Nizom takvih datoteka moguće je podesiti da se većina korisničkih servisa pokreće od **xinetd**-a.

8.2.4. Dodatni sadržaji

- https://en.wikipedia.org/wiki/Inetd
- https://en.wikipedia.org/wiki/Xinetd

8.3. Udaljeni pristup

8.3.1. telnet

telnet i **ftp** su dobri primjeri servisa koji rabe, ili bi trebali rabiti mehanizme **xinetd** za slušanje ulazne veze.

telnet je protokol na razini korisničke sjednice (sedmoslojni OSI-model) koji pruža dvosmjernu interaktivnu tekstualno orijentiranu komunikaciju koristeći se virtualnim terminalom. Za komunikaciju se rabi protokol TCP.

Kad se govori o **telnetu** najčešće se govori o **telnet**-klijentu koji se standardno instalira na svim *Linux*ovim distribucijama. Klijent se često rabi kao metoda za provjeru dostupnosti određenog TCP-servisa.

Na primjer, provjera dostupnosti servisa **apache** na lokalnom portu **80** odvija se ovako:

```
# telnet localhost 80
Trying ::1...
Connected to localhost.
Escape character is '^]'.
```

Gornji je test uspješan, jer je uspješno uspostavljena komunikacija s servisom na lokalnom portu 80.

Na operacijskom je sustavu *Debian GNU/Linux* paket za **telnet** server **telnetd** i ne dolazi sa skriptama za pokretanje standardnim za pozadinske procese nego samo s izvršnom datotekom **/usr/sbin/in.telnetd**. Razlog je tome da se za rad servisa treba koristiti **xinetd**om. Primjer konfiguracije prikazan je u poglavlju 8.2.3.

8.3.2. FTP

FTP (*File Transfer Protocol*) je protokol na aplikacijskom sloju koji omogućava prijenos datoteka s jednog računala na drugo preko mreže koristeći se protokolom TCP. FTP radi na principu klijent-poslužitelj i rabi zasebne veze između poslužitelja i klijenta za prijenos i kontrolu. Standardni portovi za **FTP** su **20** za prijenos podataka, a **21** za naredbeno sučelje.

Postoji velik broj implementacija FTP servera za distribucije *Linux* pa tako i za *Debian.* Navedimo samo neke (granične primjere):

- lFTP - *ligtweight FTP* je minimalistička implementacija, ali i dalje bogata opcijama
- gFTP - grafička implementacija za *Linux*
- vsftpd - *very secure FTP daemon* je implementacija s naglaskom na sigurnost, standardna u većini *Linux*ovih distribucija.

8.3.3. vsftpd

vsftpd se instalira iz istoimenog paketa. Može se postaviti za pokretanje super serveru **xinetd**, ali preporuča se da se pokreće kao zaseban servis. Pogledat ćemo konfiguracijsku datoteku servisa da vidimo koje se sve opcije postavljaju u njoj. Datoteka se nalazi u direktoriju **/etc/**:

```
# cat /etc/vsftpd.conf |grep -v "#"
listen=YES
anonymous_enable=YES
dirmessage_enable=YES
use_localtime=YES
xferlog_enable=YES
pam_service_name=vsftpd
rsa_cert_file=/etc/ssl/private/vsftpd.pem
```

Naredba `# grep -v "#"` izbacuje sve zapise koji sadrže "#", a time i (brojne i opsežne) komentare iz prikaza. Navedene opcije serveru **vsftpd** govore sljedeće:

Opcija	Opis
listen	*Yes* ako je samostojeći poslužitelj.
anonymous_enable	Je li dopušteno spajanje s *anonymous*?
dirmessage_enable	Je li dopuštena naredba za prikaz sadržaja?
use_localtime	Rabi li se GMT ili lokalno vrijeme?
xferlog_enable	Hoće li se logirati sve akcije prijenosa datoteka?
rsa_cert_file	Lokacija certifikata za SSL-kriptirane veze.

8.3.4. Prijavljivanje neautoriziranih korisnika

Posebnost je FTP-poslužitelja mogućnost prijavljivanja na sustav neautoriziranih korisnika. Neautorizirani korisnici koriste *anonymous* kao korisničko ime i *email-address* kao lozinku. Neautorizirani korisnici mogu pretraživati i pristupati samo sadržaju direktorija **/var/ftp/** dok autorizirani korisnici pristupaju inicijalno svom direktoriju ***home***, a daljnja prava pristupa su im identična kao i kad pristupe poslužitelju preko konzole.

Na primjeru pogledajmo prijavu neautoriziranog korisnika.

```
# ftp 10.0.2.15
Connected to 10.0.2.15.
220 (vsFTPd 2.3.5)
Name (10.0.2.15:root): anonymous
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp>
```

8.3.5. Dodatni sadržaji

- https://en.wikipedia.org/wiki/Telnet
- https://en.wikipedia.org/wiki/List_of_FTP_server_software

8.4. SSH

8.4.1. Autentikacija poslužitelja i servisa

SSH (*Secure Shell*) omogućava korisnicima pristup naredbenom sučelju na udaljenom računalu. SSH omogućava i uspostavljanje sigurnosnog komunikacijskog kanala preko nesigurne mreže (poput Interneta).

SSH je zamjena za **telnet** koji nema sigurnosnu komponentu i zamjena za druge programe za udaljeni pristup (*rlogin, rsh*) ili udaljeno kopiranje datoteka (*rcp*).

Poslužiteljski program se naziva **sshd** i pokreće se na portu **22**. Klijent se spaja na taj port pomoću naredbe **ssh**. U primjeru je prikazano inicijalno spajanje s klijentom **ssh** na udaljeno računalo:

```
# ssh l102@10.0.2.15
The authenticity of host '10.0.2.15 (10.0.2.15)' can't be established.
ECDSA key fingerprint is c9:b4:b6:8a:d7:37:8a:9b:4a:75:8d:90:0b:49:bf:35.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '10.0.2.15' (ECDSA) to the list of known hosts.
l102@10.0.2.15's password:
Linux debian-1 3.2.0-4-486 #1 Debian 3.2.68-1+deb7u2 i686

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Wed Jul  1 10:34:57 2015
$
```

Kao što je to vidljivo u gornjem primjeru, kod SSH-a se vrši autentikacija i računala i korisnika.

Poslužitelj se autenticira pomoću kriptografskih ključeva. Ključevi se na poslužitelju nalaze u direktoriju **/etc/ssh**. Kad korisnik pri prvom spajanju prihvati nastavak uspostavljanja veze, javni se ključ poslužitelja pohranjuje u datoteku **$HOME/.ssh/known_hosts**.

8.4.2. Autentikacija korisnika

Pri autentikaciji lozinkom korisnik se koristi autentikacijskim podacima koji su pohranjeni u datotekama **/etc/passwd** i **/etc/shadow** na računalu na kojem je pokrenut server **sshd**.

Autentikacija se korisnika može uspostaviti i pomoću ključeva.

1. Prvi je korak izraditi privatni/javni par ključeva naredbom **ssh-keygen**:

```
$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/l102/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /l102/.ssh/id_rsa.
Your public key has been saved in /l102/.ssh/id_rsa.pub.
The key fingerprint is:
08:a1:8a:64:3d:f6:f3:ca:79:5c:9c:9c:4e:6b:cd:50 l102@debian-1
```

```
The key's randomart image is:
+--[ RSA 2048]----+
|    .            |
|  .. .           |
| o.+.            |
|+.. o. .   E     |
|o    o. S +      |
|      o  O       |
|      ..+ =      |
|    . oo + o     |
|     +. .        |
+-----------------+
```

Naredba prihvaća dodatne opcije poput drugog kriptoalgoritma ili druge duljine ključeva.

2. Drugi je korak zaštititi privatni ključ:

```
chmod 600 /l102/.ssh/id_rsa
```

3. Treći je korak dodavanje javnog ključa na udaljeni poslužitelj u datoteku **$HOME/.ssh/authorized_keys.** Za taj korak postoji specijalizirana naredba **ssh-copy-id**:

```
# ssh-copy-id l102@10.0.2.15
l102@10.0.2.15's password:
Now try logging into the machine, with "ssh 'l102@10.0.2.15'", and check in:

  ~/.ssh/authorized_keys

to make sure we haven't added extra keys that you
weren't expecting.
```

4. Zadnji je korak provjera povezivanja bez lozinke:

```
$ ssh l102@10.0.2.15
Linux debian-1 3.2.0-4-486 #1 Debian 3.2.68-1+deb7u2 i686
Last login: Tue May 12 15:48:23 2015
$
```

Autorizacija pomoću ključeva omogućava automatizaciju izvođenja zadataka na udaljenom računalu.

8.4.3. Dodatni sadržaji

- https://wiki.debian.org/SSH

8.5. TCP wrappers

8.5.1. Konfiguracijske datoteke TCP wrappera

TCP Wrapper je mrežni sustav ACL (*Access Control List*). Rabi se za upravljanje pristupom u operacijskim sustavima temeljenim na *Unix*u. **TCP Wrapper** omogućava uporabu IP adrese, adrese mreže, imena poslužitelja i aliasa kao osnove za uspostavu kontrole pristupa.

Osnova rada ***TCP Wrapper***a je biblioteka **libwrap**. Filtriranje na gore opisani način moguće je samo u programima koji su prevedeni s uključenom bibliotekom **libwrap**. Super serveri **inetd** i **xinetd** prevedeni su s **libwrap** kao i malo prije opisani **SSH**, **telent** i **vsftpd.**

Naredbom **ldd** dobije se ispis svih dinamičkih datoteka korištenih pri programskom prevođenju neke izvršne datoteke. Tako možemo provjeriti je li **libwrap** korišten pri prevođenju kombiniranjem s naredbom **grep**:

```
# ldd /usr/sbin/xinetd |grep libwrap
libwrap.so.0 => /lib/i386-linux-gnu/libwrap.so.0 (0xb76f5000)
# ldd /usr/sbin/inetd |grep libwrap
libwrap.so.0 => /lib/i386-linux-gnu/libwrap.so.0 (0xb76f5000)
# ldd /usr/sbin/vsftpd |grep libwrap
libwrap.so.0 => /lib/i386-linux-gnu/libwrap.so.0 (0xb76f5000)
# ldd /usr/sbin/sshd |grep libwrap
libwrap.so.0 => /lib/i386-linux-gnu/libwrap.so.0 (0xb76df000)
```

Filtriranje se provodi postavljanjem pravila pristupa u datoteke **/etc/hosts.deny** i /**etc/hosts.allow**. Oblik pravila za obje datoteke:

```
<ime_servisa>:    <ciljevi_pravila> [EXCEPT <izuzeci_pravila>] [: spawn <naredba>]
```

Ako nije drugačije navedeno paketi koji zadovoljavaju pravilo u datoteci **/etc/hosts.allow** bit će prihvaćeni, a paketi koji zadovolje neko pravilo iz datoteke **/etc/hosts.deny** bit će odbačeni.

Primjer je datoteke:

```
# cat /etc/hosts.allow
vsftpd: localhost : allow
vsftpd: 10. : allow
vsftpd: .insecure.net : allow
vsftpd: ALL : deny

sshd : ALL : allow
sshd : spamer.znani : deny
sshd : 88.4.2. : deny

ALL : ALL : deny
```

Kao što je vidljivo u primjeru, moguće je pomoću pravila u datoteci **hosts.allow** onemogućiti pristup eksplicitno navodeći *deny*. Završnice pravila a*llow* su suvišne, jer bi i bez njih postupanje s prepoznatim paketima bilo isto, s obzirom na to da se nalazimo u datoteci **hosts.allow**.

Napomena

Obrat vrijedi i za datoteku **hosts.deny** – pravila *deny* tamo se podrazumijevaju, a moguće je dodati pravila *allow*.

Oprez je nužan sa pravilima oblika “ALL : ALL : deny“, jer se datoteka čita u slijedu pa bi takvo pravilo na početku zapravo odsjeklo sve servise s TCP W*rapperom*.

8.5.2. Korisne poveznice

- https://en.wikipedia.org/wiki/TCP_Wrapper

8.6. Konfiguracija NFS-a

8.6.1. Konfiguracija poslužitelja

NFS (*Network File System*) je protokol za distribuirani datotečni sustav. Pomoću njega klijentsko računalo može pristupiti datotečnom sustavu na udaljenom računalu (na kojem je pokrenut **NFS**-poslužitelj) isto kao i lokalnim datotečnim sustavima.

Na računalu koje dijeli svoje resurse preko mreže pomoću NFS-a treba:

1. Instalirati paket **nfs-kernel-server**.

2. Napraviti direktorij koji će se dijeliti (ako se ne dijeli postojeći direktorij):

```
mkdir -p /share/dir
chown nobody:nogroup /share/dir
chmod 755 /share/dir
```

3. U datoteci **/etc/exports** omogućiti udaljeni pristup do direktorija koji želimo dijeliti:

```
# cat /etc/exports | grep share
/share/dir 10.0.2.16(rw,sync)
```

Osim imenovanja dijeljenog direktorija (**/share/dir**) potrebno je identificirati i klijenta ili klijente koji imaju pravo pristupa (10.0.2.16 u gornjem primjeru). Opcije navedene u zagradama upravljaju načinom kako će se direktorij dijeliti. Opcija **rw** (*read/write*) omogućuje zapisivanja, opcija **ro** (*read only*) je za dijeljenja samo s ovlastima čitanja. Opcija **sync** osigurava konzistentnost i preporuča se za sva dijeljenja **rw**. To su tek minimalne opcije.

4. Ponovno pokrenuti poslužitelj da bi se učitao novi sadržaj datoteke **/etc/exports**:

```
# systemctl restart nfs-kernel-server
[ ok ] Stopping NFS kernel daemon: mountd nfsd.
[ ok ] Unexporting directories for NFS kernel daemon....
[ ok ] Exporting directories for NFS kernel daemon....
[ ok ] Starting NFS kernel daemon: nfsd mountd.
```

Dijeljeni resurs sad mogu rabiti udaljeni klijenti.

8.6.2. Konfiguracija klijenta

Klijentski paket na operacijskom sustavu *Debian GNU/Linux* je **nfs-common**, a na klijentskom računalu treba biti pokrenut kao pozadinski proces:

```
# systemctl start nfs-common
[ ok ] Starting NFS common utilities: statd idmapd.
```

Također treba napraviti direktorij za montiranje udaljenog datotečnog sustava i dodati zapis koji ga opisuje u datoteku **/etc/fstab**:

```
# mkdir /mnt/nfs
# cat /etc/fstab | grep nfs
10.0.2.15:/share/dir /mnt/nfs nfs defaults 0 0
```

Pri idućem pokretanju sustava ili pri automatskom montiranju pojavit će se novi datotečni sustav:

```
# mount -a
# df -h |grep nfs
10.0.2.15:/share/dir 26G 9.3G 15G 39% /mnt/nfs
```

Važno je napomenuti da kod dijeljenja **rw**-a s udaljenim klijentima nije postavljena kvota na prostor. Tako se preko jednog dijeljenog direktorija može napuniti cijeli datotečni sustav na kojem se nalazi dijeljeni direktorij.

8.6.3. Dodatni sadržaji

- https://en.wikipedia.org/wiki/Network_File_System
- http://linux.die.net/man/5/exports

8.7. Servis Samba

8.7.1. Poslužitelj Samba (smbd i nmbd)

SMB (*Server Message Block*) je mrežni protokol na aplikacijskom sloju koji omogućava dijeljeni pristup do datoteka, pisača i serijskih portova.

Samba je paket programa za *Linux* i *Unix* koji omogućava interoperabilnost s operacijskim sustavom *MS Windows.* ***Samba*** pruža siguran, stabilan i brz pristup datotekama i pisačima svim klijentima koji se koriste protokolom SMB.

Konfiguracija servera

Ime paketa za poslužitelj *Samba* je **samba**, a glavna konfiguracijska datoteka je **/etc/samba/smb.conf.** Pogledajmo sadržaj datoteke **/etc/samba/smb.conf** bez komentara:

```
# cat /etc/samba/smb.conf |grep -v "#"
[global]
   workgroup = WORKGROUP
   server string = %h server
   dns proxy = no
   log file = /var/log/samba/log.%m
   max log size = 1000
   syslog = 0
   panic action = /usr/share/samba/panic-action %d
   encrypt passwords = true
   passdb backend = tdbsam
   obey pam restrictions = yes
   unix password sync = yes
   passwd program = /usr/bin/passwd %u
   passwd chat = *Enter\snew\s*\spassword:* %n\n
*Retype\snew\s*\spassword:* %n\n *password\supdated\ssuccessfully* .
   pam password change = yes
   map to guest = bad user
   usershare allow guests = yes

[homes]
   comment = Home Directories
   browseable = no
   read only = yes
   create mask = 0700
   directory mask = 0700
   valid users = %S

[printers]
   comment = All Printers
   browseable = no
   path = /var/spool/samba
```

```
   printable = yes
   guest ok = no
   read only = yes
   create mask = 0700

[print$]
   comment = Printer Drivers
   path = /var/lib/samba/printers
   browseable = yes
   read only = yes
   guest ok = yes
```

U općem dijelu konfiguracije postavljene su opcije za rad servisa. Nakon toga su definirana dijeljenja resursa i njihova svojstva odnosno određeno je tko i na koji način ima pravo pristupa do resursa. Jednom kad su podešena dijeljenja resursa, potrebno je ponovo pokrenuti poslužitelj *Samba*:

```
# systemctl restart samba
[ ok ] Stopping Samba daemons: nmbd smbd.
[ ok ] Starting Samba daemons: nmbd smbd.
```

Kao što se vidi, servis ***Samba*** sastoji se od dva pozadinska procesa: **smbd** i **nbmd**.

- **smbd** je poslužitelj koji servisira zahtjeve za resursima.
- **nmbd** je poslužitelj koji obrađuje zahtjeve „NetBIOS preko IP“, odnosno obavlja funkciju servera za rezoluciju imena (*nameserver*) za *Windows*ove klijente.

Osim dijeljenja resursa, potrebno je dodati ***Samb-***ine korisnike koji se zatim mogu koristiti tim resursima. Korisnici se dodaju naredbom **smbpasswd**:

```
# smbpasswd -a korisnik1
New SMB password:
Retype new SMB password:
Password updated successfully
```

8.7.2. Sambin klijent

Na operacijskom sustavu *Debian GNU/Linux* klijentski paket ***Sambe*** je **smbclient**. Pomoću klijenta *Samba* može se dobiti popis dijeljenih resursa. Naredba je **smbclient**, a opcija za ispis dijeljenih resursa je **–L**:

```
# smbclient -L windows-1.test.lan
Domain=[WORKGROUP] OS=[Windows 7 OEM] Server=[NT LAN Manager 3.51]
Server=[TRAISS] User=[] Workgroup=[WORKGROUP] Domain=[]
Sharename type Comment
--------- ---- -------
ADMIN$    Disk Remote Admin
C$        Disk Default share
IPC$      IPC  Remote IPC
```

Naredba **smbclient** može se rabiti za pristup dijeljenom *Sambi*-nom resursu, ali standard je koristiti se naredbom **mount** s odabranim tipom **cifs**.

Primjer naredbe **mount** sa tipom **cifs**:

```
mount -t cifs -o korisnik=korisnik1,password=lozinka
//server/dijeljeni_dir /mnt/smb_mount
```

Nekad se rabila danas zastarjela naredba **smbmount**.

Samba podržava i niz grafičkih alata za konfiguraciju. Najpopularniji su:

1. system-config-samba
2. *Swat*
3. *Gadmin-Samba*
4. *Webmin Samba Module*.

Napraviti dijeljeni resurs u ***Sambi*** pomoću grafičkih alata jednako je jednostavno i intuitivno na *Linux*u kao i na *Windows*ima.

8.7.3. Dodatni sadržaji

- https://en.wikipedia.org/wiki/Server_Message_Block
- https://www.samba.org/samba/docs/man/manpages/nmbd.8.html
- https://www.samba.org/samba/docs/man/Samba-HOWTO-Collection/SWAT.html

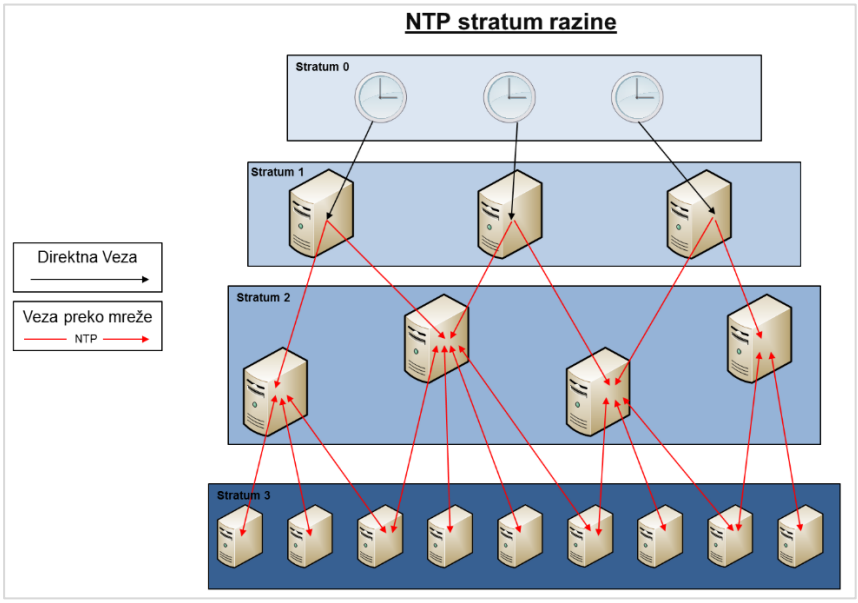
8.8. Konfiguracija NTP-a

8.8.1. Servis NTP

NTP (*Network Time Protocol*) je mrežni protokol za sinkronizaciju vremena između računalnih sustava. **NTP** je zadužen da omogući sinkronizaciju vremena na računalima s atomskim satovima koji imaju mogućnost korekcije vremena prema potrebama (dodavanje sekundi prema potrebama **UTC**-a (*The coordinated Universal Time*).

Da središnji poslužitelji ne bi bili preopterećeni, postavljen je hijerarhijski model gdje samo dio poslužitelja šalje upite središnjem poslužitelju. Tim poslužiteljima upite šalju drugi poslužitelji.

Položaj uređaja u ovoj hijerarhiji opisuje **stratum,** broj čija je vrijednost između 1 i 16. **Stratum 1**, opisuje poslužitelj koji je izravno povezan na atomski sat, a svaki idući čvor u hijerarhiji ima vrijednost **stratuma** uvećanu za 1. Poseban je slučaj kada računalo ne može ostvariti komunikaciju ni sa jednim važećim NTP-serverom te tada **stratum** poprima vrijednost 16. Cilj je svakog uređaja povezati se s NTP-serverom s čim nižim **stratumom** da bi vrijeme na njemu bilo što točnije.



Na gornjoj je slici vidljivo kako funkcionira hijerarhija NTP-a i kako se ostvaruje posluživanje velikog broja poslužitelja s višim stratum brojevima bez rizika preopterećenja poslužitelja s nižim stratum brojem. Važno je napomenuti da je to samo ilustracija; pravo stanje je da poslužitelj stratuma *n* poslužuje više stotina poslužitelja stratuma *n+1*.

8.8.2. Konfiguracija servisa NTP

Središnja konfiguracijska datoteka servisa ntp je **/etc/ntp.conf**. U njoj se podešava popis poslužitelja s kojima se servis pokušava sinkronizirati:

```
# cat /etc/ntp.conf  | grep -v "#"
server 0.debian.pool.ntp.org iburst
server 1.debian.pool.ntp.org iburst
server 2.debian.pool.ntp.org iburst
server 3.debian.pool.ntp.org iburst

restrict -4 default kod notrap nomodify nopeer noquery
restrict -6 default kod notrap nomodify nopeer noquery
restrict 127.0.0.1
restrict ::1
```

Pravila koja počinju sa **server** definiraju poslužitelje s kojima se vrši sinkronizacija. Opcija **iburst** govori da u slučaju dulje nedostupnosti kreće intenziviranje komunikacije da bi se ubrzao oporavak. Kad sistemski sat značajno odstupa od vremena na **NTP** poslužiteljima, tad **NTP**-servis postepeno ispravlja vrijeme kako u sistemskim zapisima ne bi nastale nekonzistentne situacije. Na primjer, zahtjev obrađen prije zapisa o njegovu dolasku i slično.

Za rad NTP-a treba pokrenuti servis **ntpd**. Primjer je pokretanja:

```
# systemctl start ntp
[ ok ] Starting NTP server: ntpd.
#
```

8.8.3. Naredbe ntpdate i ntpq

S paketom **ntp** dolazi servis **ntpd** i naredbeno-linijski alat **ntpq**. Uz njega je moguće instalirati još i alat **ntpdate** iz istoimenog paketa.

ntpq je alat za nadgledanje rada servisa **ntpd**. Pozivom naredbe pokreće se naredbena konzola koja pruža brojne alate za prikupljanje informacija o stanju servisa:

```
# ntpq
ntpq> ntpversion
NTP version being claimed is 2
ntpq> peers
     remote           local      st t when poll reach   delay   offset    disp
==============================================================================
+161.53.131.231  10.0.2.15        2 u   67   64    3    1.346   18.397  63.389
*grampus.irb.hr  10.0.2.15        2 u    5   64    7    1.628   20.288   0.938
+ntp2.mojsite.co 10.0.2.15        2 u   65   64    7    1.411   14.791   0.940
ntpq>
```

Za potpuni popis opcija treba unijeti naredbu **help** u naredbenoj konzoli **ntpq**-a.

Budući da je važno znati točno vrijeme i nije tehnički opravdano čekati postepeni oporavak, pomoću alata **ntpdate** vrijeme se može prisilno podesiti (odjednom) prema **NTP**-poslužitelju.

Pogledajmo na primjeru s opcijom za opširni ispis (**-v**) kako izgleda jedno takvo ubrzano podešavanje vremena prema udaljenom **NTP**-poslužitelju (ntp.srce.hr):

```
# ntpdate -v ntp.srce.hr
 6 Jul 15:15:03 ntpdate[25106]: ntpdate 4.2.6p5@1.2349-o Fri Apr 10 18:48:35 UTC 2015 (1)
 6 Jul 15:15:03 ntpdate[25106]: the NTP socket is in use, exiting
```

Naredba **ntpdate** rabi isti port kao i servis **ntpd** i može biti izvršena samo kad je ugašen **ntpd**. Razlog je takvoj konfiguraciji taj da prisilna promjena može uzrokovati da **ntpd** poništi promjenu sistemskog sata napravljenu naredbom **ntpdate.** Zbog toga je poželjno da je servis **ntpd** zaustavljen pri izvršavanju naredbe **ntpdate**.

8.8.4. Dodatni sadržaji

- https://en.wikipedia.org/wiki/Network_Time_Protocol

8.9. Postfix

8.9.1. Konfiguracijske datoteke u direktoriju /etc/postfix/

Postfix je **MTA** (*Mail Transfer Agent*) odnosno poslužitelj elektroničke pošte, program koji usmjerava i dostavlja poruke elektroničke pošte. **Postfix** je najpopularniji i najstandardniji **MTA** u većini *Linux*ovih distribucija. Konfiguracijske datoteke se nalaze u direktoriju **/etc/postfix/,** a glavne su **main.cf** i **master.cf**. Konfiguracija i broj opcija MTA-a je opsežna i složena. Stoga paket **postfix** dolazi s alatom **postconf** koji služi za konfiguraciju servisa **postfix**. Umjesto pretraživanja velikih datoteka i provjeravanja postoji li neka opcija i na kojem se mjestu nalazi u konfiguraciji, naredbom **postconf** može se jednostavno postaviti vrijednost željenog parametra.

Opcija **–e** u **postconf**-u rabi se za davanje instrukcija za promjenu zapisa u konfiguracijskim datotekama. Primjer postavljanja domene **MTA** i imena poslužitelja u konfiguraciji:

```
# postconf -e "myorigin = test.lan"
# postconf -e "myhostname=debian-1.test.lan"
# service postfix reload
# telnet localhost 25
Trying 127.0.0.1...
Connected to localhost.localdomain.
Escape character is '^]'.
220 debian-1.test.lan ESMTP Postfix (Debian/GNU)
```

Pomoću alata **postconf** moguće je postaviti sve postavke za razne načine rada **postfix MTA**-a.

8.9.2. Naredbe postmap, postalias, newaliases i postqueue

Naredba **postmap** stvara, upravlja ili postavlja upite prema specijaliziranim tablicama **postfix**. **postfix** za rad rabi četiri tablice:

- tablicu aliasa
- tablicu za filtriranje sadržaja
- tablicu usmjeravanja
- tablicu za prevođenje adresa (*address rewriting*).

Naredba **postalias** je naredba za upravljanje baze aliasa **postfixa**. Naredba **newaliases** se često rabi, ali je zapravo riječ o specijaliziranom pozivu naredbe **postalias**. **newaliases** čita konfiguracijsku datoteku **main.cf** i iz nje dohvaća vrijednost baze podataka **alias_database**. Naredba **postalias** zatim u toj bazi postavlja nove *aliase*.

Standardno se *aliasi* nalaze u datoteci **/etc/aliases**. Pomoću *aliasa* se može ostvariti preusmjeravanje elektroničke pošte unutar i izvan domene **MTA**-a.

Ako iz nekog razloga poruke elektroničke pošte ne mogu biti dostavljene, one ostaju u redu čekanja. Nakon što se otkloni razlog nedostavljanja poruka, prisilno se može pokrenuti procesiranje reda čekanja s **postqueue –f** ili **postfix flush**.

postqueue –p je naredba za prikaz reda za isporučivanje elektronske pošte.

Svaki zapis prikazuje:

Polje	Opis
QID	Identifikacijski broj poruke u redu
Size	Veličina sadržaja poruke (zaglavlje nije uključeno)
Priority	Prioritet poruke (glavni kriterij je veličina)
Q-Time	Vrijeme ulaska poruke u red čekanja
Sender/Recipient	Pošiljatelj i primatelj poruke odvojeni nizom “-“ i porukom o razlogu ne dostavljanja poruke.

Primjer izvođenja u trenutku kad su u redu dvije poruke:

```
# postqueue -p
-Queue ID- --Size-- ----Arrival Time---- -Sender/Recipient-------
4A7D18042D 310 Wed Oct 21 15:38:04 root@debian-1.test.lan
 (connect to neumijko.srce.hr[161.53.0.185]:25: Connection refused)
                                    test11@neumijko.srce.hr
2CE6680426 307 Wed Oct 21 15:37:28 root@debian-1.test.lan
 (connect to neumijko.srce.hr[161.53.0.185]:25: Connection refused)
                                    test@neumijko.srce.hr
-- 1 Kbytes in 2 Requests.
```

Brisanje svih ili dijela poruka iz reda izvodi se naredbom **postsuper -d.**

postsuper –d ALL - briše sve poruke
postsuper –d <ID> - briše jednu poruku koja ima zadani ID

Primjer brisanja svih poruka iz reda:

```
# postsuper -d ALL
postsuper: Deleted: 2 messages
# postqueue -p
Mail queue is empty
#
```

8.9.3. Dodatni sadržaji

- http://shearer.org/MTA_Comparison
- https://en.wikipedia.org/wiki/Postfix_(software)
- https://wiki.debian.org/Postfix#Installing_and_Configuring_Postfix_on_Debian
- http://www.postfix.org/postalias.1.html
- http://linux.die.net/man/1/mailq.postfix

8.10. Apache

8.10.1. Konfiguracijske datoteke, pokretanje i upravljanje

Apache (*Apache HTTP Server*) je najpopularniji *web*-poslužitelj.

Središnja konfiguracijska datoteka je **/etc/apache2/apache2.conf**, a ime servisa je **apache2**. Dodatne konfiguracije smještaju se u direktorij **/etc/apache2/conf.d/**.

U direktoriju **/etc/apache2/** postoje još šest poddirektorija:

- **mods-available** – sadrži sve dodatne module.
- **mods-enabled** – sadrži simboličke poveznice na module u direktoriju **mods-available** koji trebaju biti učitani pri pokretanju servera.
- **sites-available** – konfiguracije svih virtualnih hostova
- **sites-enabled** – simboličke poveznice na virtualne hostove koji trebaju biti dostupni.
- **conf-available** – sve dostupne konfiguracijske datoteke koje ne definiraju ponašanje virtualne hostove.
- **conf-enabled** – simboličke poveznice na konfiguracije koje trebaju biti aktivne.

Primjer minimalne konfiguracije **apache2**:

```
    # cat sites-available/default
ServerName grox.net
Listen *:80
ExtendedStatus Off
LogFormat "%h %l %u %t \"%r\" %s %b \"%{Referer}i\" \"%{User-Agent}i\""
combined
```

```
DocumentRoot /var/www
AddHandler cgi-script .cgi
DirectoryIndex index.html
<Directory /var/www>
    Options All
    AllowOverride None
    order deny,allow
    allow from all
</Directory>
```

U jednu se datoteku može upisati proizvoljno virtualnih *host*ova, ali se radi preglednosti preporuča odvojena konfiguracijska datoteka za svaki. Virtualni hostovi su zasebne *web*-stranice za koje je poslužitelj zadužen. Tako je moguće da se na istom fizičkom poslužitelju sa samo jednim mrežnim sučeljem nalazi više *web*-stranica. Virtualni hostovi smještaju se u pojedine datoteke u direktoriju **/etc/apache2/sites-available**, a aktiviraju se izradom simboličke poveznice u direktoriju **/etc/apache2/sites-enabled** i ponovnim pokretanjem servisa **apache2**.

Naredba za upravljanjem servisom **apache2** je **apache2ctl**. Servisom se može upravljati pomoću naredbe **apache2ctl** i pomoću naredbe **service**, ali naredba **apache2ctl** ima nešto više opcija. Na primjer, moguće je dobiti opsežan ispis statistika servisa s opcijom **fullstatus** ili preglednije opcijom **status**.

Primjer je statistika aktivnog servera:

```
# apache2ctl status
 Apache Server Status for localhost
Server Version: Apache/2.2.22 (Debian) mod_ssl/2.2.22 OpenSSL/1.0.1e
Server Built: Feb 1 2014 21:26:17
-------------------------------------------------------------------------
Current Time: Friday, 17-Jul-2015 17:52:31 CEST
Restart Time: Tuesday, 17-Feb-2015 16:09:12 CET
Parent Server Generation: 22
Server uptime: 150 days 43 minutes 18 seconds
Total accesses: 59086653 - Total Traffic: 116.4 GB
CPU Usage: u48.8 s13.33 cu0 cs0 - .000479% CPU load
4.56 requests/sec - 9.4 kB/second - 2115 B/request
1 requests currently being processed, 63 idle workers
.....................................................................
.....................................................................
_____________________________________________W_____________
.....................................................................
.....................................................................
.....................................................................
.....................................................................
.....................................................................
.....................................................................
.....................................................................
.....................................................................
.....................................................................
.....................................................................
```

```
.....................................................................
.....................................................................
.....................................................................
.....................................................................
.....................................................................
.....................................................................
.....................................................................
.....................................................................
.....................................................................
.....................................................................
.....................................................................
.....................................................................
.....................................................................
.....................................................................
Scoreboard Key:
 "_" Waiting for Connection, "S" Starting up, "R" Reading Request,
 "W" Sending Reply, "K" Keepalive (read), "D" DNS Lookup,
 "C" Closing connection, "L" Logging, "G" Gracefully finishing,
 "I" Idle cleanup of worker, "." Open slot with no current process
#
```

Web-serveri mogu sadržavati stranice koje se neprestano posjećuju. Moguće je učitati promjene konfiguracije servisa **apache2** tako da se prekidaju aktivne korisničke sjednice. Opcija za to „graciozno“ ponovno pokretanje servisa je **graceful.** Izravni poziv naredbom je:

```
# apachectl graceful
[ ok ] Reloading web server config: apache2.
```

Taj je način poželjan i zato što je vrijeme nedostupnosti servisa nepostojeće. Budući da je takav način poželjan konfiguracija servisa je postavljena tako da se pozivom naredbe:

systemctl reload apache2

pozove upravo ranije navedena naredba. Pogledamo li u **apache2.service** datoteku vidjet ćemo tu konfiguraciju:

```
# cat /lib/systemd/system/apache2.service |grep ExecReload
ExecReload=/usr/sbin/apachectl graceful
```

8.10.2. Dodatni sadržaji

- https://en.wikipedia.org/wiki/Apache_HTTP_Server

8.11. **Vježba: xinetd**

1. Prijavite se na računalo kao korisnik l102. U GUI-u pokrenite **Terminal** (*Activities* → **Terminal**).
2. Naredbom **su -** postanite **root** korisnik (lozinka: L102).
3. Napišite skriptu imena **pozdrav** koja na standardni izlaz ispisuje “Dobro dosli“. Skriptu spremite u direktorij **/usr/sbin/**.

Promijenite ovlasti tako da datoteka bude izvršna.

4. U direktoriju **/etc/xinetd.d/** napravite datoteku **testna** ovog sadržaja:

```
service testna
{
socket_type     = stream
server   = /usr/sbin/pozdrav
user     = root
wait     = no
disable  = no
}
```

5. U datoteku **/etc/services** dodajte servis imena **testna** koji se koristi portom **55000/tcp**.

```
"testna   55000/tcp  #testna skripta za prikaz pozdrava"
```

6. Ponovno pokrenite servis **xinetd** i pokušajte se lokalno spojiti telnetom na port 55000.

---

7. Promijenite sadržaj datoteke **/usr/sbin/pozdrav** tako da se promjeni poruka. Pokušajte se ponovno spojiti **telnetom**. Što se dogodilo? Zašto?

8.12. **Vježba: DNS**

1. Prijavite se na računalo kao korisnik l102. U GUI-u pokrenite **Terminal** (*Activities* → **Terminal**).
2. Naredbom **su -** postanite **root** korisnik (lozinka: L102).
3. Dodajte ove linije u datoteku **/etc/bind/named.conf.options**:

```
listen-on port 53 { 127.0.0.1;}; ### Master DNS IP ###
allow-query { localhost; }; ### IP Range ###
```

4. Dodajte u datoteku linije **/etc/bind/named.conf.local**:

```
zone "test.lan" IN {
type master;
file "forward.tecaj";
allow-update { none; }; };
zone "0.2.10.in-addr.arpa" IN {
type master;
file "reverse.tecaj";
allow-update { none; }; };
```

5. Proučite sadržaj datoteke **/etc/bind/named.conf.options** i pronađite dio konfiguracije koji definira direktorij u kojem se nalaze datoteke zona (opcija `directory`).

6. U direktoriju iz prethodnog zadatka napravite dvije datoteke:

```
forward.tecaj
$TTL 3H
@      IN SOA @ debian-1.test.lan. (
2015120800    ; serial
1D     ; refresh
1H     ; retry
1W     ; expire
3H )   ; minimum
NS debian-1.test.lan.
debian-1 IN   A      10.2.0.15
debian-2 IN   A      10.2.0.19
```

```
reverse.tecaj
$TTL 3H
@      IN SOA @ debian-1.test.lan.(
2015120800    ; serial
1D     ; refresh
1H     ; retry
1W     ; expire
3H )   ; minimum
NS debian-1.test.lan.
debian-1 IN A10.2.0.15
15            IN PTR debian-1.test.lan.
19            IN PTR debian-2.test.lan.
```

7. Izvršite naredbe za postavljanje odgovarajućih ovlasti nad datotekama (u direktoriju iz 5. zadatka). Nakon toga provjerite konfiguracijske datoteke naredbama `named-checkconf` i `named-checkzone` te ponovno pokrenite servis **bind9**.
8. Dodajte na početak datoteke **/etc/resolv.conf** zapis:

   `nameserver 127.0.0.1`
9. Testirajte rad lokalnog DNS-a naredbama `nslookup` ili `host` za:

```
debian-1.test.lan
debian-2.test.lan
10.2.0.15
10.2.0.19
```

8.13. **Vježba: SSH**

1. Prijavite se na računalo kao korisnik l102. U GUI-u pokrenite **Terminal** (*Activities* → **Terminal**).
2. Provjerite sadržaj direktorija **/home/l102**. Pazite da uključite i prikaz skrivenih datoteka.

3. Naredbom **ssh-keygen** izradite par ključeva. Rabite sve standardne vrijednosti i nemojte unijeti lozinku.

   `$ ssh-keygen` i nakon toga tri puta pritisnuti tipku **[Enter]**

4. Ponovno provjerite sadržaj direktorija **/home/l102**. Pazite da uključite i prikaz skrivenih datoteka. Što se promijenilo?

5. Provjerite sadržaj datoteke **/home/l102/.ssh/id_rsa.pub**.

6. Naredbom **ssh-copy-id** omogućite udaljeni pristup korisniku l102 na lokalni stroj. (U ovoj vježbi obje uloge, i klijenta i servera, obavlja isto računalo). Koji je sada sadržaj datoteke **/home/l102/.ssh/authorized_keys**?

7. Testirajte pristup preko SSH-a autorizacijom pomoću ključeva.

   ```
   $ ssh localhost
   ```

8.14. **Dodatna vježba: Apache2**

1. Prijavite se na računalo kao korisnik l102. U GUI-u pokrenite **Terminal** (*Activities* → **Terminal**).
2. Naredbom „**su -**“ postanite **root** korisnik (lozinka: L102).
3. Instalirajte paket **apache2** `(# apt-get install apache2 -y )`.
4. Napravite direktorij u kojem će se nalaziti web-sadržaji:

   `mkdir -p /var/www/primjer.com/public_html`
5. Napravite datoteku **/var/www/primjer.com/public_html/index.html** ovog sadržaja:

```
<html>
<head>
<title>www.primjer.com</title>
</head>
<body>
<h1>Bravo: Kreirali ste svoj prvi virtualni host</h1>
</body>
</html>
```

6. Izradite datoteku za konfiguraciju naziva **primjer.com** tako da kopirate postojeću osnovnu:

   `# cp /etc/apache2/sites-available/000-default.conf /etc/apache2/sites-available/primjer.com.conf`
7. Uredite novu datoteku tako da postavite ove vrijednosti (ako opcije ne postoje, izradite ih, a ako postoje, izmijenite ih):

```
ServerAdmin webmaster@primjer.com
ServerName primjer.com
ServerAlias www.primjer.com
DocumentRoot /var/www/primjer.com/public_html
```

8. Dodajte svoju konfiguraciju u aktivne konfiguracije:

```
# ln -s /etc/apache2/sites-available/primjer.com.conf
/etc/apache2/sites-enabled/
Ili
#  a2ensite primjer.com.conf
```

9. U datoteku **/etc/hosts** na kraj dodajte zapis:

```
#Virtual Hosts
10.0.2.15      www.primjer.com
```

10. Provjerite dostupnost svoje web-stranice pristupom na stranicu **www.primjer.com** pomoću web-preglednika **firefox** koristeći korisnika l102. Što se dogodilo? Zašto?

11. Ponovno pokrenite servis **apache2**. Pokušajte ponovno pristupiti web-stranicama.

```
systemctl restart apache2
```

9. Osnove sigurnosti

Trajanje poglavlja:
100 min

Po završetku ovoga poglavlja moći ćete:

- zaštiti sustav od korisnika koji ostvare lokalni ili udaljeni neovlašteni pristup
- prikupiti podatke o trenutačnim aktivnostima korisnika i zabraniti im pristup
- izraditi i postaviti pravila za pojedine lance u vatrozidu **iptabales**
- postaviti standardne politike lanca i oblikovati ih prema potrebi
- skenirati poslužitelj ili mrežu za otvorene portove i oblikovati skeniranje prema potrebama
- protumačiti rezultate.

Ova cjelina obrađuje osnovne postavke i sustave koji povećavaju sigurnost. Obradit će se sredstva zaštite od lokalnog napadača, udaljenog napadača i postupci ublažavanja posljedica u slučaju provale u sustav.

9.1. Lokalne postavke sigurnosti Linux-poslužitelja

9.1.1. GRUB i sigurnosne opcije prilikom pokretanja računalnog sustava

Ponekad je lakše osigurati sigurnost udaljenog pristupa do računala nego osigurati sigurnost fizičkog računala. Stoga od neovlaštenog pristupa treba zaštititi i BIOS.

Rescue-mediji mogu se zlorabiti da bi se omogućio neovlašteni pristup do svih datoteka na računalu, ako je ostvaren fizički pristup do računala. Prvi korak zaštite je ograničiti pokretanje sustava samo s lokalnog tvrdog diska, a nakon toga još treba lozinkom zaštititi BIOS.

GRUB pri pokretanju prihvaća naredbeno-linijske opcije. Da bi se uklonila mogućnost zlorabljenja te opcije, potrebno je GRUB zaštiti lozinkom dodavanjem kôda u datoteku **/boot/grub/grub.conf**:

```
set superusers="korisnik1"
password korisnik1 lozinka1
password korisnik2 lozinka2

menuentry "GNU/Linux" {
set root=(hd0,1)
linux /vmlinuz
}

menuentry "Windows" --users korisnik2 {
set root=(hd0,2)
chainloader +1
```

Korisnik1 je administrator i može izvoditi sve akcije kao i kad nije uspostavljena zaštita lozinkom. Svi korisnici mogu pokrenuti *Linux,* ali samo korisnik2 može pokrenuti *Windows.*

Kad je postavljena takva konfiguracija, napadač koji ostvari fizički pristup računalu i dalje neće moći pristupiti podacima.

9.1.2. Ovlasti nad datotekama

Jednom kad (na bilo koji način) napadač uspije pristupiti računalu i dalje je moguće ograničiti potencijalnu štetu slijedeći jednostavne upute:

1. Treba zaštititi ključne datoteke od brisanja (pomoću mogućnosti *immutable*) i datoteke sistemskih zapisa od promjena.

```
# chattr +i /bin/login
# chattr +i /bin/ps
…
# chattr +a /var/log/messages
```

2. Također je korisno imati *loghost*-računalo za udaljenu pohranu datoteka sistemskih zapisa. Tako čak ni korisnik s *root* ovlastima neće moći ukloniti sve zapise o svojim aktivnostima. Za udaljeno bilježenje svih sistemskih zapisa na udaljenom računalu, na primjer **loghost.test.lan**, potrebno je dodati ovu liniju u konfiguraciju **rsyslog**:

```
*.* @loghost.test.lan:514
```

3. Treba postaviti **nosuid** ili **noexec** nad direktorije kojima korisnici imaju pristup:

```
# tail -2 /etc/fstab
/tmp    /tmp    ext4    nosuid        1    2
/home    /home    ext4    noexec    1    2
```

4. Treba paziti na anomalije na datotečnom sustavu poput datoteka za koje se ne može utvrditi vlasništvo (1. primjer) ili datoteka s postavljenim SUID-bitom. Takve provjere treba redovito obavljati (idealno iz **cron**a):

```
# find / -nouser –o –nogroup | mail –s test_permisija
admin@test.lan
# find / -perms +4000 | mail –s test_permisija2 admin@test.lan
```

Također se mogu koristiti specijalizirani alati za otkrivanje upada u sustav kao što su OSSEC, *Samhain* i *OpenDLP*.

9.1.3. Naredbe za pregled aktivnosti korisnika

Naredba **last** ispisuje uspješna prijavljivanja u sustav i ponovna pokretanja servera. Primjer izvođenja naredbe:

```
# last
root     pts/3        :0               Wed Jul 15 15:21   still logged in
root     pts/2        :0               Mon Jul 13 15:48 - 15:28 (1+23:39)
root     pts/1        :0               Fri Jul 10 13:10   still logged in
l102 tty7         :0               Fri Jul 10 10:48   still logged in
(unknown tty7         :0               Fri Jul 10 10:48 - 10:48  (00:00)
reboot   system boot  3.2.0-4-486      Fri Jul 10 10:48 - 15:33 (5+04:45)
root     pts/1        :0               Fri Jul 10 10:40 - down   (00:06)
l102 tty7         :0               Fri Jul 10 10:40 - down   (00:07)
(unknown tty7         :0               Fri Jul 10 10:40 - 10:40  (00:00)
reboot   system boot  3.2.0-4-486      Fri Jul 10 10:39 - 10:47  (00:07)
root     pts/3        :0.0             Fri Jul 10 10:34 - 10:34  (00:00)
root     pts/1        :0.0             Fri Jul 10 10:26 - down   (00:11)
l102 tty8         :0               Fri Jul 10 10:24 - down   (00:14)
(unknown tty8         :0               Fri Jul 10 10:21 - 10:24  (00:03)
root     pts/1        :0               Fri Jul 10 10:13 - 10:13  (00:00)
l102 tty7         :0               Fri Jul 10 10:13 - 10:13  (00:00)
(unknown tty7         :0               Thu Jul  9 16:26 - 10:13  (17:46)
reboot   system boot  3.2.0-4-486      Thu Jul  9 16:25 - 10:38  (18:12)
root     pts/1        :0.0             Fri Jul 10 10:05 - 10:05  (00:00)
l102 tty7         :0               Fri Jul 10 10:04 - 10:05  (00:00)
(unknown tty7         :0               Fri Jul 10 10:04 - 10:04  (00:00)
root     pts/1        :0               Thu Jul  9 10:55 - 10:03  (23:07)
root     pts/1        :0               Thu Jul  9 10:31 - 10:55  (00:23)
reboot   system boot  3.2.0-4-486      Thu Jul  9 10:03 - 10:38 (1+00:35)

wtmp begins Thu Jul  9 10:03:24 2015
```

U gornjem su ispisu u zagradama ispisana trajanja korisničkih sjednica.

Naredbe **w**, **who** i **finger** prikazuju popis svih korisnika spojenih na neku konzolu računala.

```
# w
 15:35:36 up 5 days, 22:16,  4 users,  load average: 0.34, 0.25, 0.28
USER     TTY      FROM             LOGIN@   IDLE   JCPU   PCPU WHAT
l102 :0                          01Jul15 ?xdm?   8:35m  1.88s gnome-session
root     pts/1    :0               01Jul15  0.00s  6.09s  3.58s ssh
root@10.0.2.15
root     pts/2    :0               Thu10   28:16m  1.12s  1.12s bash
root     pts/3    10.0.2.15        Sun23    0.00s  2.83s  0.04s w
# who
l102 :0           2015-07-01 10:34
root     pts/1        2015-07-01 10:35 (:0)
root     pts/2        2015-07-02 10:30 (:0)
root     pts/3        2015-07-05 23:43 (10.0.2.15)
# finger
Login     Name          Tty      Idle  Login Time   Office     Office Phone
```

```
l102  Ime Prezime  *tty7       5d  Jul 10 10:48 (:0)
root      root         *pts/1          Jul 10 13:10 (:0)
root      root          pts/3       8  Jul 15 15:21 (:0)
```

9.1.4. Korisnička ograničenja

U trenutku kad se sumnja na (ozbiljan) sigurnosni incident moguće je zabraniti pristup terminalima svim korisnicima (osim *roota*). Ako postoji datoteka **/etc/nologin** zaustavit će se svi pokušaji prijave na konzolu. Ako se korisnik uspješno autenticira, dobit će kao poruku sadržaj datoteke **/etc/nologin**.

Direktorij **/etc/security/** sadrži niz datoteka koje omogućavaju administratoru ograničavanje korisničke potrošnje resursa. U direktoriju je ukupno osam konfiguracijskih datoteka:

```
# ls /etc/security |grep conf
access.conf
capability.conf
group.conf
limits.conf
namespace.conf
pam_env.conf
sepermit.conf
time.conf
```

Najvažnije su datoteke:

- **access.conf** – onemogućava pristup korisnicima i grupama
- **group.conf** – konfiguracijska datoteka za upravljanje ovlastima članova grupa
- **limits.conf** – najvažnija datoteka koja omogućava ograničavanje brojnih parametara poput veličine datoteka, CPU-vremena, adrese, broja procesa, broj otvorenih datoteka, količina zauzete memorije i slično nad korisnicima i grupama.

9.1.5. Dodatni sadržaji

- http://www.tldp.org/LDP/intro-*linux*/html/sect_03_04.html
- http://man7.org/*linux*/man-pages/man8/nologin.8.html
- http://*linux*.die.net/man/5/limits.conf

9.2. Mrežna sigurnost

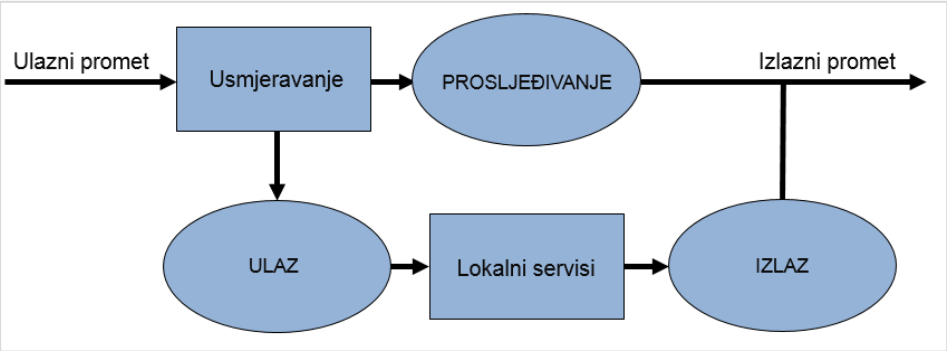
9.2.1. Vatrozid Iptables

Na distribuciji *Debian GNU/Linux* vatrozid **iptables** donosi istoimeni paket i dio je standardne instalacije. **Iptables** imaju tri standardne tablice, a svaku čini nekoliko osnovnih lanaca:

- *Filter* – tablica za filtriranje paketa s lancima: INPUT, OUTPUT i FORWARD

- *Nat* – tablica za translaciju mrežnih adresa: lanci su PREROUTING, POSTROUTING i INPUT
- *Mangle* – sve druge promjene nad paketima (osim NAT): lanci su PREROUTING, INPUT, OUTPUT, FORWARD i POSTROUTING.

Slikom je prikazan tipični promet kroz sustav. Kao što se vidi na slici, svi paketi prolaze kroz jedan od tri lanaca (ulaz, izlaz ili prosljeđivanje) te je vatrozidom moguće filtrirati sve pakete korištenjem jednog od ta tri lanca.



Kombinacija lanca i tablice definira koje su akcije moguće nad paketima.

Tablica *Filter* (svi lanci) podržava: DROP, REJECT, ACCEPT i MIRROR.

Tablica *Nat* podržava:

- REDIRECT i DNAT (u lancima PREROUTING I OUTPUT)
- MASQUERADE (POSTRUTING lanac)
- SNAT (u lancima POSTROUTING i OUTPUT).

U paketu **iptables** nalazi se pet osnovnih alata: **iptables, iptebles-apply, iptables-save, iptables-restore** i **iptables-xml.**

iptables – osnovna naredba za dodavanje i brisanje pravila vatrozida. Osnovna sintaksa je:

```
iptables [-t tablica] naredba lanac akcija
```

Najčešće naredbe su:

Naredba	Opis
-A	Dodavanja pravila na kraj lanca.
-C	Provjeravanje postojanja pravila.
-D	Brisanje pravila iz lanca.
-I	Ubacivanje pravila u lanac (na određeno mjesto u lancu ili na početak).
-L	Ispis pravila.
-F	Brisanje svih pravila (u lancu, ako je on naveden ili ako su sva pravila ista u svim lancima).

Za filtriranje paketa rabe se tablice *filter,* jer svi paketi (kao što se vidi na gornjoj slici) prolaze kroz jedan od lanaca. Najčešće akcije koje se provode nad paketima su:

Akcija	Opis
DROP	Odbijanje paketa.
ACCEPT	Prihvaćanje paketa.
QUEUE	Prosljeđivanje paketa u prostor korisničkih procesa.
RETURN	Povratak paketa u prethodni lanac (kad se rabi gniježđenje lanaca).
REJECT	Slično kao DROP, ali pošiljatelj paketa dobiva povratnu informaciju da je paket odbijen (rijetko korisno).

9.2.2. Opcije -L i -F naredbe iptables i naredbe iptables-apply i iptables-save

Naredba opcijom **–L** rabi se i za prikaz trenutačnih pravila vatrozida. Dodat ćemo dva jednostavna pravila u vatrozid koja puštaju sav *web*-promet (http – port 80 i https – port 443) prema računalu:

```
# iptables -A INPUT -m state --state NEW -p tcp --dport 80 -j ACCEPT
# iptables -A INPUT -m state --state NEW -p tcp --dport 443 -j ACCEPT
# iptables -nL
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
ACCEPT     tcp  --  0.0.0.0/0            0.0.0.0/0           state NEW
tcp dpt:80
ACCEPT     tcp  --  0.0.0.0/0            0.0.0.0/0           state NEW
tcp dpt:443

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
```

Kad se u naredbi **iptables** ne navede tablica, pravilo se unosi za tablicu *filter*.

iptables-apply – naredba za primjenu pravila sa zaštitom od prekidanja vlastitog linka. Naredba primijeni željeni skup pravila i čeka potvrdu korisnika. Ako lošom konfiguracijom korisnik prekine vlastitu vezu, tada naredba neće dobiti korisnikovu potvrdu. U tom će slučaju po isteku vremena za odgovor naredba napraviti *rollback* na prijašnju konfiguraciju vatrozida.

iptables-save – naredba za pribavljanje trenutačnih pravila vatrozida. Unaprijed je definirano da ta naredba šalje *output* na **stdout** pa treba napraviti preusmjerivanje u datoteku:

```
# iptables-save > test
# cat test
# Generated by iptables-save v1.4.14 on Tue Jul  7 16:48:54 2015
*filter
:INPUT ACCEPT [116:10649]
```

```
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [107:9372]
-A INPUT -p tcp -m state --state NEW -m tcp --dport 80 -j ACCEPT
-A INPUT -p tcp -m state --state NEW -m tcp --dport 443 -j ACCEPT
COMMIT
# Completed on Tue Jul  7 16:48:54 2015
```

Naredbom **iptables –F** ili **iptables --flush** brišu se sva aktivna pravila iz vatrozida. Potreban je oprez pri uporabi te naredbe, jer se tada koriste samo politike nad lancima, a ako su te politike DROP, tada računalo postaje nedostupno s mreže.

9.2.3. Naredbe iptables-restore i iptables-xml

iptables-restore – naredba čita standardni input i oblikuje pravila vatrozida prema unosu. Za uporabu datoteke treba se koristiti preusmjeravanjem.

iptables-xml – alat za prebacivanje teško čitljivog oblika vatrozida izrađenog naredbom **iptables-save** u čitljiviji oblik datoteke **xml**.

```
# iptables-xml test
<iptables-rules version="1.0">
<!-- # Generated by iptables*-save v1.4.14 on Tue Jul  7 16:48:54 2015 -->
  <table name="filter" >
    <chain name="INPUT" policy="ACCEPT" packet-count="116" byte-count="10649" >
      <rule >
       <conditions>
INPUT          <p >tcp</p>
        <state >
          <state >NEW</state>
        </state>
        <tcp >
          <dport >80</dport>
        </tcp>
       </conditions>
       <actions>
        <ACCEPT  />
       </actions>

      </rule>

      <rule >
       <conditions>
INPUT          <p >tcp</p>
        <state >
          <state >NEW</state>
        </state>
        <tcp >
          <dport >443</dport>
        </tcp>
       </conditions>
       <actions>
```

```
        <ACCEPT  />
       </actions>

      </rule>

    </chain>
    <chain name="FORWARD" policy="ACCEPT" packet-count="0" byte-count="0" />
    <chain name="OUTPUT" policy="ACCEPT" packet-count="107" byte-count="9372" />
  </table>
<!-- # Completed on Tue Jul  7 16:48:54 2015 -->
</iptables-rules>
```

Napomena

Osim za alat **iptables-xml**, inačice svih naredbi postoje i za IPv6. Naredbe za IPv6 imaju u imenu naredbe **ip6tables** umjesto **iptables**.

9.2.4. Politika lanca

Lanci funkcioniraju tako da se pri dolasku paketa pravila čitaju jedno za drugim i da se primjenjuje prvo pravilo koje paket ispunjava. Kada u lancu nema pravila koje paket ispunjava, tada se primjenjuje politika lanca. Budući da su u gornjem primjeru sve politike postavljene na ACCEPT, svi će paketi biti prihvaćeni.

Da bi vatrozid doista obavljao svoju funkciju, potrebno je promijeniti politike za lance INPUT i FORWARD te postaviti pravilo za lanac INPUT, koje propušta sve pakete koji su rezultat komunikacije koju je započelo računalo (tj. odgovor na sve pakete koji su prošli lanac OUTPUT):

```
# iptables -P FORWARD DROP
# iptables -P INPUT DROP
# iptables -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
# iptables -nL
Chain INPUT (policy DROP)
target     prot opt source               destination
ACCEPT     tcp  --  0.0.0.0/0            0.0.0.0/0           state NEW tcp dpt:80
ACCEPT     tcp  --  0.0.0.0/0            0.0.0.0/0           state NEW tcp dpt:443
ACCEPT     all  --  0.0.0.0/0            0.0.0.0/0           state RELATED,ESTABLISHED

Chain FORWARD (policy DROP)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
```

9.2.5. Dodatni sadržaji

- https://wiki.debian.org/iptables
- http://www.howtogeek.com/177621/the-beginners-guide-to-iptables-the-linux-firewall/
- https://www.frozentux.net/iptables-tutorial/iptables-tutorial.html

9.3. Skeniranje otvorenih portova

9.3.1. Naredba nmap

nmap (*Network Mapper*) je alat za skeniranje mreže i sigurnosnu reviziju. **nmap** može skenirati pojedinačna mrežna sučelja ili velike mreže za što je izvorno dizajniran. **nmap** se koristi IP-paketima za pretraživanje mreže, pretraživanje lociranih uređaja, identificiranje aplikacija na portovima, prikupljanje detalja o operacijskom sustavu, vatrozidu i slično.

nmap je izuzetno bogat opcijama:

```
# man nmap
Usage: nmap [Scan Type(s)] [Options] {target specification}
TARGET SPECIFICATION:
  Can pass hostnames, IP addresses, networks, etc.
  Ex: scanme.nmap.org, microsoft.com/24, 192.168.0.1; 10.0.0-255.1-254
  -iL <inputfilename>: Input from list of hosts/networks
  -iR <num hosts>: Choose random targets
  --exclude <host1[,host2][,host3],...>: Exclude hosts/networks
  --excludefile <exclude_file>: Exclude list from file
HOST DISCOVERY:
  -sL: List Scan - simply list targets to scan
  -sn: Ping Scan - disable port scan
  -Pn: Treat all hosts as online -- skip host discovery
  -PS/PA/PU/PY[portlist]: TCP SYN/ACK, UDP or SCTP discovery to given ports
  -PE/PP/PM: ICMP echo, timestamp, and netmask request discovery probes
  -PO[protocol list]: IP Protocol Ping
  -n/-R: Never do DNS resolution/Always resolve [default: sometimes]
  --dns-servers <serv1[,serv2],...>: Specify custom DNS servers
  --system-dns: Use OS's DNS resolver
  --traceroute: Trace hop path to each host
SCAN TECHNIQUES:
  -sS/sT/sA/sW/sM: TCP SYN/Connect()/ACK/Window/Maimon scans
  -sU: UDP Scan
  -sN/sF/sX: TCP Null, FIN, and Xmas scans
  --scanflags <flags>: Customize TCP scan flags
  -sI <zombie host[:probeport]>: Idle scan
  -sY/sZ: SCTP INIT/COOKIE-ECHO scans
  -sO: IP protocol scan
  -b <FTP relay host>: FTP bounce scan
PORT SPECIFICATION AND SCAN ORDER:
  -p <port ranges>: Only scan specified ports
    Ex: -p22; -p1-65535; -p U:53,111,137,T:21-25,80,139,8080,S:9
  --exclude-ports <port ranges>: Exclude the specified ports from scanning
  -F: Fast mode - Scan fewer ports than the default scan
  -r: Scan ports consecutively - don't randomize
  --top-ports <number>: Scan <number> most common ports
  --port-ratio <ratio>: Scan ports more common than <ratio>
SERVICE/VERSION DETECTION:
  -sV: Probe open ports to determine service/version info
  --version-intensity <level>: Set from 0 (light) to 9 (try all probes)
  --version-light: Limit to most likely probes (intensity 2)
```

```
  --version-all: Try every single probe (intensity 9)
  --version-trace: Show detailed version scan activity (for debugging)
SCRIPT SCAN:
  -sC: equivalent to --script=default
  --script=<Lua scripts>: <Lua scripts> is a comma separated list of
           directories, script-files or script-categories
  --script-args=<n1=v1,[n2=v2,...]>: provide arguments to scripts
  --script-args-file=filename: provide NSE script args in a file
  --script-trace: Show all data sent and received
  --script-updatedb: Update the script database.
  --script-help=<Lua scripts>: Show help about scripts.
           <Lua scripts> is a comma-separated list of script-files or
           script-categories.
OS DETECTION:
  -O: Enable OS detection
  --osscan-limit: Limit OS detection to promising targets
  --osscan-guess: Guess OS more aggressively
TIMING AND PERFORMANCE:
  Options which take <time> are in seconds, or append 'ms' (milliseconds),
  's' (seconds), 'm' (minutes), or 'h' (hours) to the value (e.g. 30m).
  -T<0-5>: Set timing template (higher is faster)
  --min-hostgroup/max-hostgroup <size>: Parallel host scan group sizes
  --min-parallelism/max-parallelism <numprobes>: Probe parallelization
  --min-rtt-timeout/max-rtt-timeout/initial-rtt-timeout <time>: Specifies
      probe round trip time.
  --max-retries <tries>: Caps number of port scan probe retransmissions.
  --host-timeout <time>: Give up on target after this long
  --scan-delay/--max-scan-delay <time>: Adjust delay between probes
  --min-rate <number>: Send packets no slower than <number> per second
  --max-rate <number>: Send packets no faster than <number> per second
FIREWALL/IDS EVASION AND SPOOFING:
  -f; --mtu <val>: fragment packets (optionally w/given MTU)
  -D <decoy1,decoy2[,ME],...>: Cloak a scan with decoys
  -S <IP_Address>: Spoof source address
  -e <iface>: Use specified interface
  -g/--source-port <portnum>: Use given port number
  --proxies <url1,[url2],...>: Relay connections through HTTP/SOCKS4 proxies
  --data <hex string>: Append a custom payload to sent packets
  --data-string <string>: Append a custom ASCII string to sent packets
  --data-length <num>: Append random data to sent packets
  --ip-options <options>: Send packets with specified ip options
  --ttl <val>: Set IP time-to-live field
  --spoof-mac <mac address/prefix/vendor name>: Spoof your MAC address
  --badsum: Send packets with a bogus TCP/UDP/SCTP checksum
OUTPUT:
  -oN/-oX/-oS/-oG <file>: Output scan in normal, XML, s|<rIpt kIddi3,
     and Grepable format, respectively, to the given filename.
  -oA <basename>: Output in the three major formats at once
  -v: Increase verbosity level (use -vv or more for greater effect)
  -d: Increase debugging level (use -dd or more for greater effect)
  --reason: Display the reason a port is in a particular state
  --open: Only show open (or possibly open) ports
  --packet-trace: Show all packets sent and received
  --iflist: Print host interfaces and routes (for debugging)
```

```
  --log-errors: Log errors/warnings to the normal-format output file
  --append-output: Append to rather than clobber specified output files
  --resume <filename>: Resume an aborted scan
  --stylesheet <path/URL>: XSL stylesheet to transform XML output to HTML
  --webxml: Reference stylesheet from Nmap.Org for more portable XML
  --no-stylesheet: Prevent associating of XSL stylesheet w/XML output
MISC:
  -6: Enable IPv6 scanning
  -A: Enable OS detection, version detection, script scanning, and traceroute
  --datadir <dirname>: Specify custom Nmap data file location
  --send-eth/--send-ip: Send using raw ethernet frames or IP packets
  --privileged: Assume that the user is fully privileged
  --unprivileged: Assume the user lacks raw socket privileges
  -V: Print version number
  -h: Print this help summary page.
```

9.3.2. Primjeri izvođenja naredbe nmap

Primjeri izvođenja skeniranja:

- Skeniranje jednog poslužitelja, samo njegovih portova od 1 do 1023.

```
# nmap -v debian-1.test.lan

Starting Nmap 6.00 ( http://nmap.org ) at 2015-10-22 10:46 CEST
Initiating SYN Stealth Scan at 10:46
Scanning debian-1.test.lan (127.0.1.1) [1000 ports]
Discovered open port 22/tcp on 127.0.1.1
Discovered open port 445/tcp on 127.0.1.1
Discovered open port 80/tcp on 127.0.1.1
Discovered open port 111/tcp on 127.0.1.1
Discovered open port 139/tcp on 127.0.1.1
Discovered open port 21/tcp on 127.0.1.1
Discovered open port 2049/tcp on 127.0.1.1
Completed SYN Stealth Scan at 10:46, 0.79s elapsed (1000 total ports)
Nmap scan report for debian-1.test.lan (127.0.1.1)
Host is up (0.00019s latency).
Not shown: 993 closed ports
PORT STATE SERVICE
21/tcp open ftp
22/tcp open ssh
80/tcp open http
111/tcp open rpcbind
139/tcp open netbios-ssn
445/tcp open microsoft-ds
2049/tcp open nfs
Read data files from: /usr/bin/../share/nmap
Nmap done: 1 IP address (1 host up) scanned in 1.03 seconds
 Raw packets sent: 1000 (44.000KB) | Rcvd: 1000 (40.028KB)
#
```

- Skeniranje TCP-portova s dodatnom provjerom za raspoznavanjem otvorenih portova od filtriranih. Provjeravaju se portovi SSH, DNS, POP3 i IMAP-servisi. Provjera se vrši na prvoj polovici 12 podmreža 10.0.0-11.

```
# nmap -sV -p 22,53,110,143,4564 10.0.0-11.1-127
Nmap scan report for 10.0.11.107
Host is up (0.00082s latency).
PORT STATE SERVICE VERSION
22/tcp filtered ssh
53/tcp filtered domain
110/tcp filtered pop3
143/tcp filtered imap
4564/tcp filtered unknown
Nmap scan report for 10.0.11.108
Host is up (0.0027s latency).
PORT STATE SERVICE VERSION
22/tcp filtered ssh
53/tcp filtered domain
110/tcp filtered pop3
143/tcp filtered imap
4564/tcp filtered unknown
Nmap scan report for 10.0.11.109
Host is up (0.00058s latency).
PORT STATE SERVICE VERSION
22/tcp filtered ssh
53/tcp filtered domain
110/tcp filtered pop3
143/tcp filtered imap
4564/tcp filtered unknown
Nmap scan report for 10.0.11.110
Host is up (0.00079s latency).
PORT STATE SERVICE VERSION
22/tcp filtered ssh
53/tcp filtered domain
110/tcp filtered pop3
143/tcp filtered imap
4564/tcp filtered unknown
Nmap scan report for 10.0.11.111
Host is up (0.0028s latency).
PORT STATE SERVICE VERSION
22/tcp filtered ssh
53/tcp filtered domain
110/tcp filtered pop3
143/tcp filtered imap
4564/tcp filtered unknown
Nmap scan report for 10.0.11.112
Host is up (0.0030s latency).
PORT STATE SERVICE VERSION
22/tcp filtered ssh
53/tcp filtered domain
110/tcp filtered pop3
143/tcp filtered imap
4564/tcp filtered unknown
```

```
Nmap scan report for 10.0.11.113
Host is up (0.0033s latency).
PORT STATE SERVICE VERSION
22/tcp filtered ssh
53/tcp filtered domain
110/tcp filtered pop3
143/tcp filtered imap
4564/tcp filtered unknown
Nmap scan report for 10.0.11.114
Host is up (0.0016s latency).
PORT STATE SERVICE VERSION
22/tcp filtered ssh
53/tcp filtered domain
110/tcp filtered pop3
143/tcp filtered imap
4564/tcp filtered unknown
Nmap scan report for 10.0.11.115
Host is up (0.0018s latency).
PORT STATE SERVICE VERSION
22/tcp filtered ssh
53/tcp filtered domain
110/tcp filtered pop3
143/tcp filtered imap
4564/tcp filtered unknown
Nmap scan report for 10.0.11.116
Host is up (0.00089s latency).
PORT STATE SERVICE VERSION
22/tcp filtered ssh
53/tcp filtered domain
110/tcp filtered pop3
143/tcp filtered imap
4564/tcp filtered unknown
Nmap scan report for 10.0.11.117
Host is up (0.0023s latency).
PORT STATE SERVICE VERSION
22/tcp filtered ssh
53/tcp filtered domain
110/tcp filtered pop3
143/tcp filtered imap
4564/tcp filtered unknown
Nmap scan report for 10.0.11.118
Host is up (0.0022s latency).
PORT STATE SERVICE VERSION
22/tcp filtered ssh
53/tcp filtered domain
110/tcp filtered pop3
143/tcp filtered imap
4564/tcp filtered unknown
Nmap scan report for 10.0.11.119
Host is up (0.0021s latency).
PORT STATE SERVICE VERSION
22/tcp filtered ssh
53/tcp filtered domain
110/tcp filtered pop3
```

```
143/tcp filtered imap
4564/tcp filtered unknown
Nmap scan report for 10.0.11.120
Host is up (0.00031s latency).
PORT STATE SERVICE VERSION
22/tcp filtered ssh
53/tcp filtered domain
110/tcp filtered pop3
143/tcp filtered imap
4564/tcp filtered unknown
Nmap scan report for 10.0.11.121
Host is up (0.0011s latency).
PORT STATE SERVICE VERSION
22/tcp filtered ssh
53/tcp filtered domain
110/tcp filtered pop3
143/tcp filtered imap
4564/tcp filtered unknown
Nmap scan report for 10.0.11.122
Host is up (0.0023s latency).
PORT STATE SERVICE VERSION
22/tcp filtered ssh
53/tcp filtered domain
110/tcp filtered pop3
143/tcp filtered imap
4564/tcp filtered unknown
Nmap scan report for 10.0.11.123
Host is up (0.00056s latency).
PORT STATE SERVICE VERSION
22/tcp filtered ssh
53/tcp filtered domain
110/tcp filtered pop3
143/tcp filtered imap
4564/tcp filtered unknown
Nmap scan report for 10.0.11.124
Host is up (0.0014s latency).
PORT STATE SERVICE VERSION
22/tcp filtered ssh
53/tcp filtered domain
110/tcp filtered pop3
143/tcp filtered imap
4564/tcp filtered unknown
Nmap scan report for 10.0.11.125
Host is up (0.0022s latency).
PORT STATE SERVICE VERSION
22/tcp filtered ssh
53/tcp filtered domain
110/tcp filtered pop3
143/tcp filtered imap
4564/tcp filtered unknown
Nmap scan report for 10.0.11.126
Host is up (0.0022s latency).
PORT STATE SERVICE VERSION
22/tcp filtered ssh
```

```
53/tcp filtered domain
110/tcp filtered pop3
143/tcp filtered imap
4564/tcp filtered unknown
Nmap scan report for 10.0.11.127
Host is up (0.0013s latency).
PORT STATE SERVICE VERSION
22/tcp filtered ssh
53/tcp filtered domain
110/tcp filtered pop3
143/tcp filtered imap
4564/tcp filtered unknown
Service detection performed. Please report any incorrect results at
http://nmap.org/submit/ .
Nmap done: 1524 IP addresses (1275 hosts up) scanned in 167.51 seconds
#
```

- Skenira sve portove na zadanom uređaju.

```
# nmap -p- debian-1.test.lan
Starting Nmap 6.00 ( http://nmap.org ) at 2015-10-22 11:32 CEST
Nmap scan report for debian-1.test.lan (127.0.1.1)
Host is up (0.00015s latency).
Not shown: 65522 closed ports
PORT STATE SERVICE
21/tcp open ftp
22/tcp open ssh
80/tcp open http
111/tcp open rpcbind
139/tcp open netbios-ssn
445/tcp open microsoft-ds
2049/tcp open nfs
33326/tcp open unknown
37806/tcp open unknown
39641/tcp open unknown
44294/tcp open unknown
48978/tcp open unknown
60450/tcp open unknown
Nmap done: 1 IP address (1 host up) scanned in 22.64 seconds
#
```

- Skeniranje mreže za lociranje uređaja koji su aktivni. To se skeniranje naziva i lociranje uređaja na mreži ili ping-sken.

```
# nmap -sP 192.168.1.0/24
Starting Nmap 6.00 ( http://nmap.org ) at 2015-10-22 11:37 CEST
Nmap scan report for 192.168.1.0
Host is up (0.0010s latency).
Nmap scan report for 192.168.1.1
Host is up (0.00056s latency).
Nmap scan report for 192.168.1.2
Host is up (0.0015s latency).
```

```
Nmap scan report for 192.168.1.3
Host is up (0.0014s latency).
Nmap scan report for 192.168.1.4
Host is up (0.00091s latency).
Nmap scan report for 192.168.1.5
Host is up (0.00086s latency).
Nmap scan report for 192.168.1.6
Host is up (0.00081s latency).
Nmap scan report for 192.168.1.7
Host is up (0.0014s latency).
Nmap scan report for 192.168.1.8
Host is up (0.0013s latency).
Nmap scan report for 192.168.1.9
Host is up (0.00090s latency).
Nmap scan report for 192.168.1.10
Host is up (0.00070s latency).
Nmap scan report for 192.168.1.11
Host is up (0.00059s latency).
Nmap done: 12 IP addresses (12 hosts up) scanned in 2.20 seconds
#
```

9.3.3. Dodatni sadržaji

- http://nmap.org/book/man-port-scanning-techniques.html
- http://www.cyberciti.biz/networking/nmap-command-examples-tutorials/

9.4. Vježba: Lokalna i udaljena sigurnost

1. Prijavite se na računalo kao korisnik l102. U GUI-u pokrenite **Terminal** (*Activities* → **Terminal**).
2. Naredbom **su -** postanite **root** korisnik (lozinka: L102).
3. Izradite datoteku **/tmp/neunistiva** i zaštitite je od brisanja.
4. Pokušajte izbrisati **/tmp/neunistiva**:

5. Uklonite zaštitu od brisanja nad datotekom **/tmp/neunistiva**. Pokušajte je obrisati kao obični korisnik **l102**.
6. Kako provjeravamo status vatrozida. Zašto?

7. Provjerite status servisa **ssh**. Pokušajte se logirati na **127.0.0.1** kao korisnik **l102**.
8. Otvorite novi prozor kao root korisnik i pomoću **iptables** zabranite sav promet na portu **22**:
9. Provjerite radi li i dalje terminal u kojem imate uspostavljanu **ssh** vezu.

10. Provedite **TCP sken** adrese **127.0.0.1**. Kakvo je stanje porta 22?

11. Uklonite sva pravila iz vatrozida.

12. Provjerite ponašanje drugog terminala i ponovite sken **nmap**. Kakvo je stanje porta 22?

13. Promijenite politiku nad lancem **INPUT** u **DROP**
14. Provjerite ponašanje drugog terminala i ponovite sken **nmap**. Kakvo je stanje porta **22**? Što se još promijenilo?

9.5. Dodatna vježba: iptables standardne postavke

1. Prijavite se na računalo kao korisnik l102. U GUI-u pokrenite **Terminal** (*Activities* → **Terminal**).
2. Naredbom **su -** postanite **root** korisnik (lozinka: L102).
3. Promijenite politike za lance **INPUT** i **FORWARD** u **DROP**.

4. Testirajte ponašanje povezivanja **ssh** na **127.0.0.1** i testirajte stanje portova alatom **nmap**. Također provjerite rad alata **apt** za upravljanje paketima (zaokružite ili upišite stanje).

   ssh radi ne radi

   apt radi ne radi

   nmap poruka: _____

5. Postavite pravilo za lanac **INPUT** koje propušta sve pakete koji su odgovor na komunikaciju koju je započelo računalo (tj. odgovor na sve pakete koji su prošli lanac **OUTPUT**).

```
# iptables -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
```

6. Ponovno testirajte ponašanje povezivanja **ssh** na **127.0.0.1** i testirajte stanje portova alatom **nmap**. Također provjerite rad alata **apt** za upravljanje paketima. Što se promijenilo i zašto?

10. Ispis

Trajanje poglavlja:
70 min

Po završetku ovoga poglavlja moći ćete:

- razumjeti namjenu i ulogu protokola **lpd/lpr** i sustava **CUPS**
- razumjeti mehanizam rada sustava **CUPS**
- instalirati sustav **CUPS**
- razumjeti i namjestiti postavke sustava **CUPS** pomoću *web*-sučelja.

Ova cjelina obrađuje osnove ispisivanja u operacijskom sustavu Linux. U cjelini je obrađen sustav CUPS koji je standardni sustav za ispis na Linuxu.

10.1. Pregled protokola za ispis

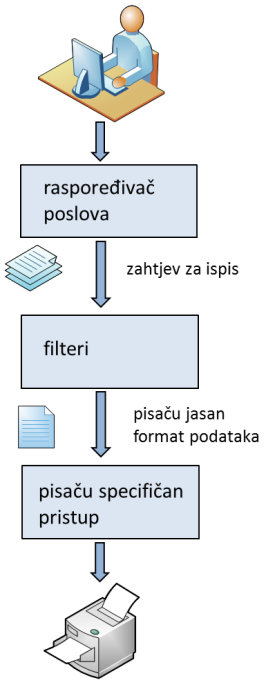
10.1.1. lpd i CUPS

lpd (*Line Printer Daemon protocol*) ili **lpr** (*Line Printer Remote protocol*) je mrežni protokol za slanje zahtjeva za ispisom na udaljeni pisač. **LPD**-pisač definiran je pomoću **IP**-adrese računala koje je poslužitelj zahtjeva i imena reda (*queue*) na tom računalu.

CUPS (*Common Unix Printing System*) je moderan sustav za upravljanje pisačima s operacijskih sustava temeljenih na **Unix**-u. Računalo sa pokrenutim **CUPS**-serverom djeluje kao poslužitelj za zahtjeve za ispisom. Pomoću servisa **CUPS** računalo može prihvatiti klijentske zahtjeve, obraditi ih i proslijediti na odgovarajući pisač.

CUPS pruža mehanizam koji omogućava slanje zahtjeva za ispisom na pisače na uobičajeni način. Podaci za ispis šalju se raspoređivaču poslova koji poslove prosljeđuje sustavu za filtriranje, a on ulazne podatke pretvara u format razumljiv pisaču. Ti se podaci zatim šalju na odgovarajući pisač na poseban izlaz koji je specifičan za taj pisač. To može biti izravni pristup preko paralelnog, serijskog ili **USB**-porta Ili cups-pdf kad se obavlja **PDF**-virtualni ispis.

Na slici je prikazan načina rada sustava **CUPS**. Sustav obavlja sve korake između računala koje podnosi zahtjev i pisača.



Napomena

CUPS podržava protokol **lpd** pa se sva računala koja mogu vršiti ispis preko protokola **lpd** za istu namjenu se mogu koristiti i servisom CUPS.

10.1.2. Dodatni sadržaji

- https://wiki.debian.org/SystemPrinting
- https://www.cups.org/documentation.php?VERSION=2.1&Q

10.2. CUPS

10.2.1. Konfiguracijske datoteke /etc/cups/

CUPS se na distribuciji *Debian* instalira iz istoimenog paketa. Pri instalaciji paketa vidljiv je velik broj paketa o kojima ovisi paket CUPS. Većina tih paketa su imena **printer-driver-*** i sadrže upravljačke programe za pojedine modele pisača.

Servis CUPS po pokretanju sluša na portu 631. Središnja konfiguracijska datoteka je **/etc/cups/cupsd.conf**. Postavit ćemo osnovna pravila pristupa u datoteci /**etc/cups/cupsd.conf** tako da je omogućeno pokretanje ispisa iz mreže 192.168.1.0/24:

```
# 1
Listen localhost:631
Listen 192.168.1.42
Listen /var/run/cups/cups.sock
#2
Browsing On
```

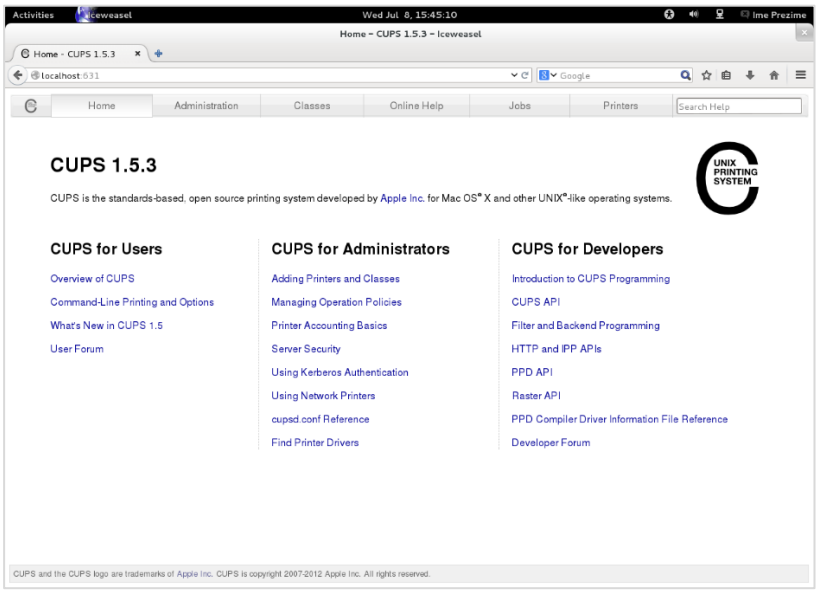
```
BrowseOrder allow,deny
BrowseAllow @LOCAL
#3
<Location/>
Order allow,deny
Allow localhost
Allow 192.168.1.*
</Location>
#4
<Location /admin/conf>
AuthType Basic
Require user @SYSTEM
Order allow,deny
Allow localhost
Allow 192.168.1.150
</Location>
```

- Prvi odjeljak konfiguracije definira da servis sluša na portu 631 *localhost* i 192.168.1.42. Ako računalo ima još adresa/sučelja na njima, **CUPS** neće slušati.
- Drugi odjeljak definira da su dijeljeni pisači vidljivi u lokalnoj mreži.
- Treći odjeljak ograničava pristup do servera na *localhost* i mrežu 192.168.1.0/24.
- Četvrti odjeljak ograničava pristup konfiguracijskim datotekama na *localhost* i na jednu (administratorsku) radnu stanicu 192.168.1.150.

Konfiguracija se dalje provodi preko *web*-sučelja na *localhost*: 631 ili 192.168.1.42:631.

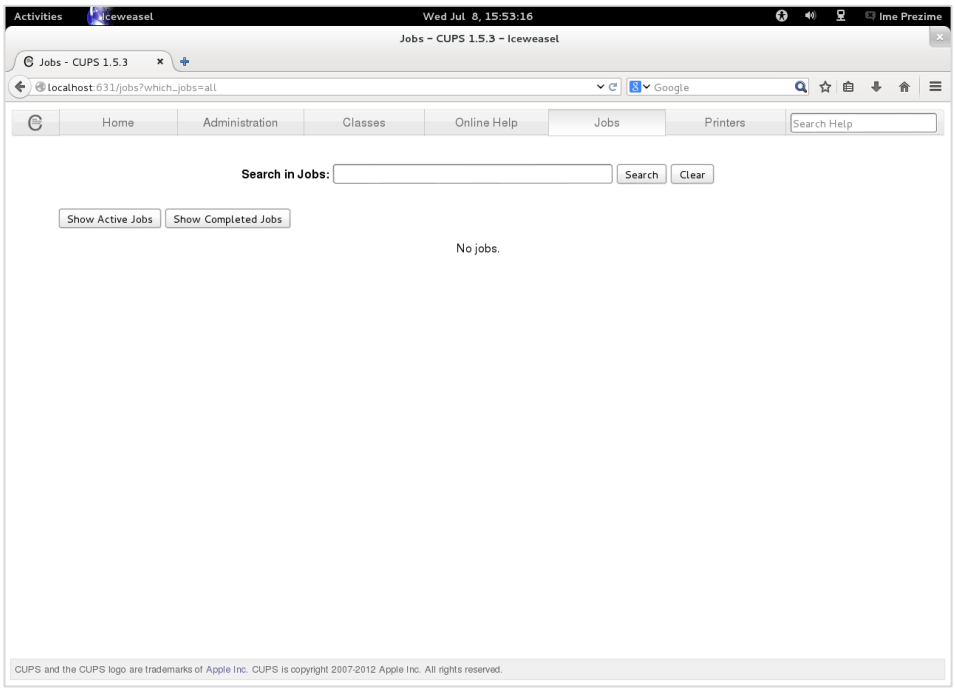
10.2.2. Web-sučelje CUPS-a

Kroz grafičko sučelje na lokaciji http://localhost:631/ treba provesti dodavanje novog printera. Osim administrativnih operacija *web*-sučelje omogućava i pristup do brojnih uputa.

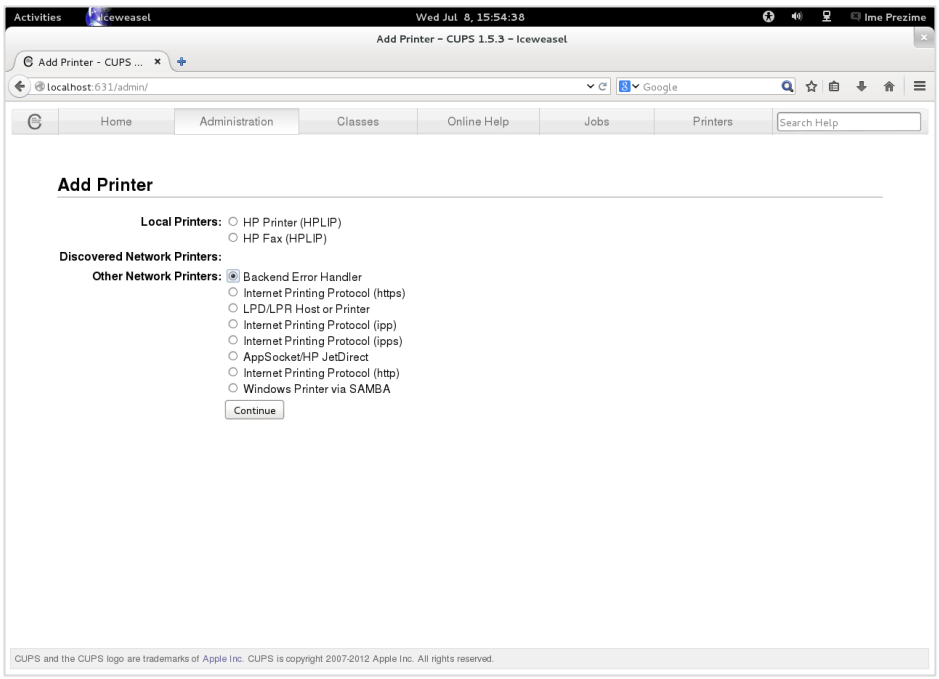


Grafičko je sučelje jednostavno za uporabu, ali ipak zahtjeva osnovno razumijevanje rada CUPS-a.

Izborom opcije ***Jobs*** može se pratiti izvođenje poslova.



Izborom opcije ***Administration*** mogu se dodavati i uklanjati pisači:



Važno je napomenuti da CUPS rade pomoću IPP-a (*Internet Printing Protocol*) za razliku od servisa **LDP**-a (*Line Printer Deamon*). CUPS također podržava komunikaciju preko protokola **http**, **https**, **ipp** i **ipps**.

10.2.3. Dodatni sadržaji

- http://man7.org/linux/man-pages/man1/logger.1.html
- https://www.digitalocean.com/community/tutorials/how-to-manage-log-files-with-logrotate-on-ubuntu-12-10

10.3. **Vježba: Konfiguracija CUPS**

1. Prijavite se na računalo kao korisnik l102. U GUI-u pokrenite **Terminal** (*Activities* → **Terminal**).
2. Naredbom **su -** postanite **root** korisnik (lozinka: L102).
3. Otvorite tekstnim preglednikom datoteku **/etc/cups/cupsd.conf**. Pronađite opciju koja definira na kojem portu sluša servis **cups**. Koji je to port? _____
4. Pomoću web-preglednika pristupite **localhost** portu **631**.
5. Proučite opcije u web-sučelju za konfiguraciju **CUPS**. Posebno provjerite koliko je pisača dostupno na stranici **http://localhost:631/printers/** pomoću tipke *printers*.

---

6. Instalirajte paket **cups-pdf**.

```
# apt install cups-pdf
```

7. Ponovno pristupite stranici **http://localhost:631/printers/** (ako je niste napustili, napravite refresh). Koliko je sad dostupno pisača?

---

8. Promijenite port slušanja servisa **cups** u **11631**. Ponovno pokrenite servis **cups**.
9. Ponovno pristupite stranici **http://localhost:631/**. Pokušajte pristupiti na **http://localhost:11631/**.

10.4. **Dodatna vježba: Ispis u virtualni pdf pisač**

1. Prijavite se na računalo kao korisnik l102. U GUI-u pokrenite **Terminal** (*Activities* → **Terminal**).
2. Naredbom „**su -**“ postanite **root** korisnik (lozinka: L102).
3. Otvorite datoteku **/etc/fstab** pomoću alata **libreoffice**.

```
# libreoffice /etc/resolv.conf
```

4. Pokrenite ispis datoteke (samo je jedan pisač instaliran).
5. U direktoriju **home** korisnika nalazi se PDF poddirektorij u kojem se trebala napraviti datoteka imena **resolv.pdf**. Otvorite datoteku **resolv.pdf** pomoću alata **libreoffice**. Je li datoteka identična **/etc/resolv.conf** ?

---

11. Grafička okolina X

Trajanje poglavlja:
160 min

Po završetku ovoga poglavlja moći ćete:

- imenovati ključne inačice grafičke okoline X
- navesti godinu nastanka prve inačice „moderne" grafičke okoline X
- povezati pojmove **X11R7** i **X.Org**
- imenovati središnju konfiguracijsku datoteku *X.Org servera*
- prepoznati konfiguracijske datoteke *X.Org servera*
- prepoznati i imenovati komponente konfiguracije datoteke **xorg.conf**
- izraditi i provjeravati testnu datoteku **xorg.conf**
- opisati način rada i ulogu *X.Org servera* i *XKlijenata*
- pokrenuti *XKlijent* na udaljenom računalu
- upravljanje pravima pristupa udaljenih *XKlijenata*
- pokrenuti *XKlijente* s opcijama zapisanim u datoteku **Xresources**
- objasniti namjenu naredbe **xrdb**
- pokrenuti *X.Org server* promjenom runlevela
- pokrenuti *X.Org server* iz komandne linije
- definirati razlike u konfiguraciji *X.Org servera* ovisno o načinu pokretanja servera
- definirati što je upravitelj prikazom
- imenovati najčešće upravitelje prikazom
- razumjeti konfigurabilnost upravitelja prikazom
- argumentirano i objektivno birati između različitih upravitelja prikazom
- prepoznati i razlikovati desktop-okoline **KDE Plasma**, **GNOME** i **Xfce**
- koristiti desktop-okolinu **GNOME** za brzo i jednostavno pozivanje aplikacija
- odabrati desktop-okolinu na osnovi svojstava koja su vam potrebna.

Ova cjelina obrađuje grafičku okolinu X i brojne upravitelje prikazom. U lekciji je obrađen X.Org server i Xklijenti kao i grafičke okoline GDM, KDM i XDM koje ujedinjuju X.Org server i XKlijente u smislenu cjelinu za rad.

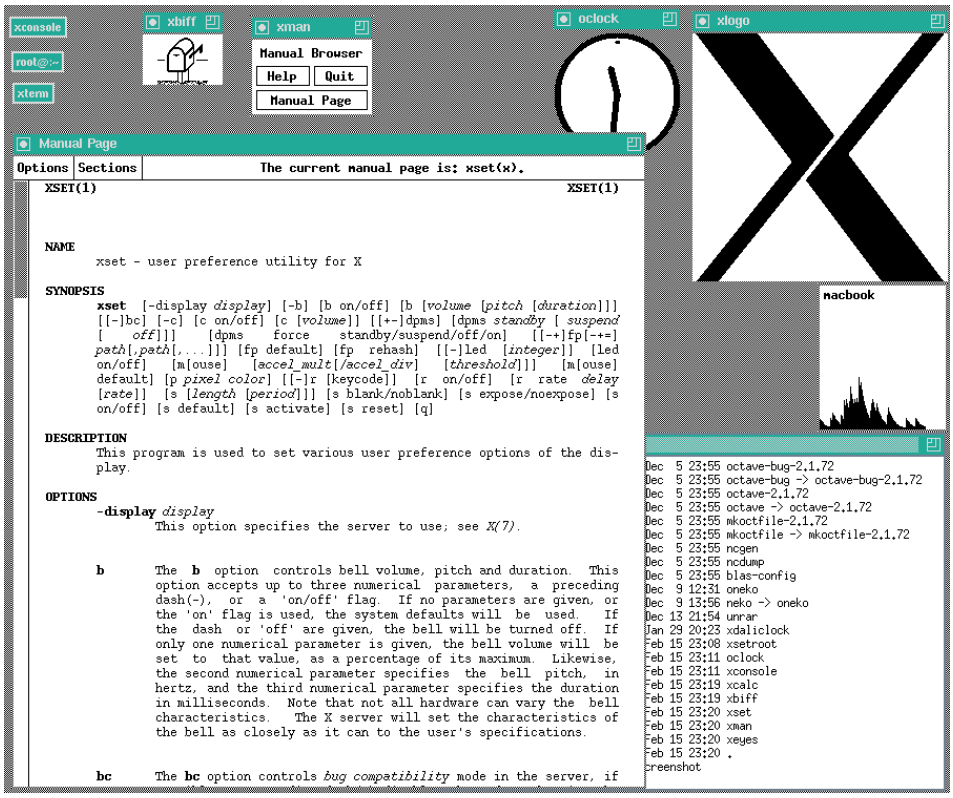
11.1. O grafičkoj okolini X

11.1.1. Povijest

Sustav *X Windows* (poznat još i kao X11, X ili X-Windows) je sustav za upravljanje prozorima za standardne jedinice za prikaz uobičajen na operacijskim sustavima utemeljenim na *Unix*u. X pruža okosnicu za okolinu grafičkog korisničkog sučelja omogućavajući izradu i upravljanje prozorima te korisničku interakciju pomoću miša i tipkovnice.

Slika prikazuje povijesni primjer sustava *X Windows* s upravljačem prozora **twm** (*Tab Window Manager*) i aktivnim klijentskim aplikacijama **xlogo**, **xterm**, **oclock**, **xbiff**, **xman** i **xload**.

Napomena: Za povećanje slike treba mišem kliknuti na nju.



Izvor: Wikipedia

Sustav *X Windows* izvorno je razvijen u sklopu projekta **Athena** na *Massachusetts Institute of Technology* (MIT) 1984. godine. Dugo se vremena sustav *X Windows* temeljio na *X Windows* verziji 11, inačici 6, skraćeno nazvanim X11R6. Moderna inačica grafičkog sučelja nastala je 2004. godine kad su snage ujedinili fondacija *X.Org* i članovi razvojnog tima koji su prije toga radili na projektu *XFree86*. Tom suradnjom nastao je moderni sustav *X Windows X11R7*, po fondaciji *X.Org* jednostavnije nazvan sustav *X.Org Windows* ili *X.Org server*.

Prva izdana inačica poslužitelja *X.Org* bila je **X11R6.7.0**. U sljedećoj inačici (X11R7.0.0 – izdana 21.12.2005. godine) i svim inačicama nakon nje *X.Org* je uveo modularnost nad izvornim kodom. Inačica X11R7.0.0 smatra se inačicom 1.0 ili 1.0.1 sustava *X.Org*. Od te inačice do ožujka 2015. godine izdano je 17 novih inačica, od kojih je svaka donosila manja ili značajnija proširenja funkcionalnosti od prethodne.

Primjeri su proširenja:

- protokol **RandR** koji omogućuje rad na više ekrana te njihovo upravljanje
- **XInput 2.2** koji podržava dodir na više točaka (*multi-touch*)
- podrška za **systemd-logind**.

X.Org je preddefinirani poslužitelj *X Windows* za *Debian* od inačice 4.0.

11.2. Konfiguracija X.Org

11.2.1. Automatska konfiguracija

Konfiguracija *X.Org servera* odvija se automatski i sve se pohranjuju u brojnim datotekama u sustavu. Ipak, ako postoji datoteka **xorg.conf** smještena u direktoriju **xorg.conf.d/,** tada se poslužitelj *X.Org* koristi tom konfiguracijskom datotekom.

X.Org server sadrži mehanizme za automatsko generiranje konfiguracija pri pokretanju poslužitelja, a ako nisu izvršene promjene postavki nakon instalacije, tada nije potrebno postojanje niti datoteke **xorg.conf** niti direktorija **xorg.conf.d/**.

Automatski konfiguracijski mehanizam djeluje kroz dva pristupa:

1. Probama se prikupljaju informacije koje se mogu prikupiti na takav način ili (ako je to moguće) da se vrijednosti postavljaju zaključivanjem na osnovu prikupljenih parametara iz drugih konfiguracijskih datoteka. Dakle, ti se parametri ne moraju izrijekom definirati u konfiguracijskoj datoteci.

2. Za većinu važnih konfiguracijskih parametara postoje **pričuvne „sigurnosne“ vrijednosti** da bi se osiguralo pokretanje *X.Org servera* u nekoj uporabljivoj konfiguraciji.

Ta se dva pristupa kombiniraju da bi se omogućilo autokonfiguriranje i uspješno pokretanje *X.Org servera* bez potrebe za korisničkom interakcijom.

11.2.2. xorg.conf i druge konfiguracijske datoteke

Pri svakom pokretanju *X.Org server* provjerava brojne lokacije na datotečnom sustavu i iz njih učitava pronađene konfiguracijske datoteke. Lokacije su na kojima se nalaze konfiguracijske datoteke:

Naziv datoteke	Namjena datoteke
/etc/X11/xorg.conf	Konfiguracijska datoteka *X.Org servera*.
/etc/X11/xorg.conf-4	Konfiguracijska datoteka *X.Org servera*.
/etc/xorg.conf	Konfiguracijska datoteka *X.Org servera*.
/usr/etc/xorg.conf	Konfiguracijska datoteka *X.Org servera*.
/usr/lib/X11/xorg.conf	Konfiguracijska datoteka *X.Org servera*.
/etc/X11/xorg.conf.d-4	Konfiguracijski direktorij *X.Org servera*.
/etc/xorg.conf.d	Konfiguracijski direktorij *X.Org servera*.
/usr/etc/xorg.conf.d	Konfiguracijski direktorij *X.Org servera*.
/usr/lib/X11/xorg.conf.d	Konfiguracijski direktorij *X.Org servera*.
/var/log/Xorg.n.log	Datoteka sistemskih zapisa *X.Org servera* za displej n.
/usr/bin/∗	Izvršne datoteke klijenata.
/usr/include/∗	*Header*-datoteke.
/usr/lib/∗	Dijeljene biblioteke.
/usr/share/fonts/X11/∗	Fontovi.

/usr/share/X11/XErrorDB	Baza klijentskih pogrešaka.
/usr/lib/X11/app-defaults/∗	Specifikacije resursa klijenata.
/usr/share/man/man?/∗	Man-stranice.
/etc/Xn.hosts	Datoteka za kontrolu pristupa displeja n.
/etc/X11/xorg.conf.d	Konfiguracijski direktorij *X.Org servera*.

Nakon instalacije *X.Org servera*, datoteka ***xorg.conf*** može se izraditi pribavljanjem konfiguracije iz sustava naredbom **X :1 –configure**. Tako izrađena datoteka može se zatim uređivati. Postavljanjem u odgovarajući direktorij ili pozivom iz naredbene linije pribavljena datoteka se može rabiti za oblikovanje *X.Org servera*.

Mogućnost **:1** nije obavezna, ali osigurava uspješno pribavljanje konfiguracije čak i kad je poslužitelj X pokrenut na displej adresi **:0**. Pri pokretanju naredbi, na ekranu se prikazuju informacije o tijeku pribavljanja konfiguracijskih podataka, a sama konfiguracija pohranjuje se u datoteku **~/xorg.conf.new**.

Primjer je sadržaja datoteke xorg.conf.new (klikom na naziv datoteke ona se otvara u novom prozoru).

Konfiguracijska datoteka poslužitelja **X.Org** sastoji se od odjeljaka ovih oblika:

```
Section  "ImeOdjeljka"
           Konfiguracijski_podaci 1
           Konfiguracijski_podaci 2
              ...
EndSection
```

Ime	Opis
Files	Putanja do datoteke.
ServerFlags	Kontrolne zastave servera.
Module	Dinamičko učitavanje modula.
Extensions	Aktiviranje proširenja.
InputDevice	Opis uređaja za unos.
InputClass	Opis klasa za unos.
Device	Opis grafičkih uređaja.
VideoAdaptor	Opis Xv video adaptera.
Monitor	Opis monitora.
Modes	Opis grafičkih načina rada.
Screen	Konfiguracija zaslona.
ServerLayout	Definicija rasporeda elemenata na zaslonu.
DRI	Konfiguracije specifične za DRI (*Direct Rendering Infrastructure*).
Vendor	Konfiguracije specifične za proizvođača.

Slijed odjeljaka je proizvoljan. Budući da postojanje konfiguracijske datoteke nije obavezno, tako je i postojanje svih odjeljaka proizvoljno. Ako parametar nije izrijekom definiran, a potreban je za rad

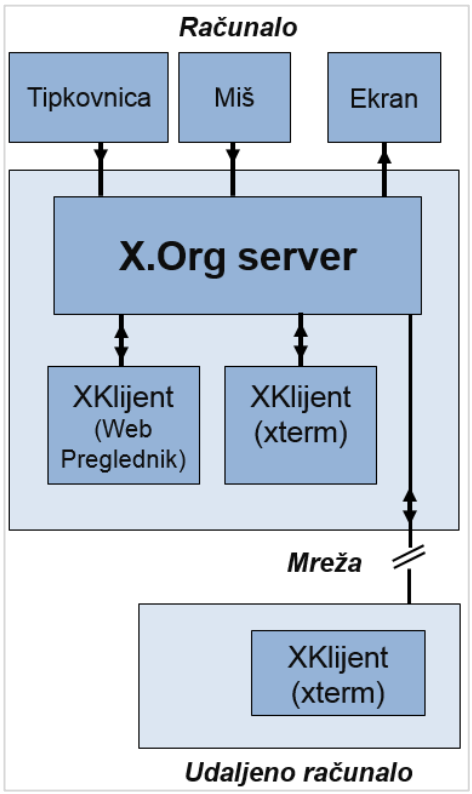
servera, tada sustav rabi osnovne ili probama prikupljene vrijednosti. Tako se ostvaruje fleksibilnost konfiguracije, uslijed koje korisnik koji želi podesiti neki element poslužitelja ne mora podesiti, prepisati niti pribaviti druge elemente.

Naredbom # X -config ~/xorg.conf.new -retro provjerava se valjanost prikupljene konfiguracije. Rezultat naredbe treba biti ekran ispunjen okomitim sivo-bijelim crtama i aktivni pokazivač miša u obliku slova X.

11.3. Upravljanje XKlijentima

11.3.1. XKlijenti

Programi koji se pokreću u okolini X nazivaju se **XKlijenti**. *XKlijent*ima je *X.Org server* potreban zbog toga što predstavlja poveznicu između njih i uređaja za unos i uređaja za prikaz, kako je prikazano na slici.



Moguće se postavke *XKlijenata* razlikuju od klijenta do klijenta, a ovise o namjeni, veličini i složenosti samog *XKlijenta*. Korisnička konfiguracija *XKlijenta* provodi se uređivanjem datoteke ***~/.Xresources***.

Parametri postavljeni u toj datoteci primijeniti će se na *XKlijenta*, ako se na njega i odnose, odnosno, ako se *XKlijent* za prikaz koristi konfiguriranim elementom. Sadržaj datoteke učitava se pri pokretanju *X.Org servera*. To znači da učitavanje promjena konfiguracije za, na primjer, **xclock** zahtijeva ponovno pokretanje *X.Org servera*. Također je moguće naredbom **xrdb**, namijenjenom za upravljanje bazom podataka resursa *X.Org servera*, pozvati učitavanje parametara:

```
$ xrdb -merge ~/.Xresources && xclock &
```

Tom se naredbom, prije pokretanja *XKlijenta* **xclock**, prvo provede spajanje postojećih zapisa u bazi s konfiguracijom u datoteci ***Xresources*** i to tako da vrijednosti u datoteci imaju prioritet.

X.Org server neovisan je o klijentima koji se u njemu pokreću. Datoteka za konfiguraciju *X.Org servera* definira samo komponente servera, dakle, komunikaciju s komponentama za prikaz i ulaznim komponentama (miš, tipkovnica i sl.). Datotekama za konfiguraciju *X.Org servera* ne

konfiguriraju se svojstva *XKijenata* poput fonta, direktorija za fontove, dimenzija i inicijalnog mjesta *XKlijenata* itd.

11.3.2. DISPLAY

XKlijenti su izdvojeni od *X.Org servera* i mogu se izvoditi lokalno na *X.Org serveru* ili na udaljenim *X.Org serverima* koji su mrežom povezani sa računalom koje pokreće *XKlijenta*. Pri pokretanju, *XKlijenti* trebaju informaciju pod kojim se *X.Org serverom* pokreću. Osnovni klijenti (poput **xterm** i **xclock**) pri pokretanju prihvaćaju parametar **–display**. Drugi je način postavljanje vrijednosti varijable DISPLAY. Dakle, ako želimo pokrenuti **xclock** na udaljenom računalu IP-adrese 161.53.2.0 tada možemo pokrenuti:

```
$ xclock –display 161.53.2.0:0
```

ili

```
$ export DISPLAY=161.53.2.0:0
$ xclock
```

X.Org server mora dopustiti izvođenje klijenata s udaljenih računala. Za dodavanje i uklanjanje servera iz kontrolnog pristupnog popisa *X.Org servera* rabi se naredba **xhost**.

```
$ xhost + 161.53.2.0
161.53.2.0 being added to access control list
```

11.4. Pokretanje X.Org servera

11.4.1. startx

X.Org server može se automatski pokrenuti ulaskom u odgovarajuću razinu izvođenja kako je objašnjeno u 2. poglavlju.

Drugi je način pokretanja poslužitelja pomoću naredbe **startx**. U slučaju pokretanja iz naredbene linije, naredbom se **startx**, pomoću dodatnih mogućnosti mogu definirati svojstva poput rezolucije i pokretanja određenog upravitelja prikazom (*display manager*).

Sintaksa naredbe **startx** je:

```
startx [ [ klijent ] opcije... ] [ -- [ server ] [ display ] opcije... ]
```

Naredba startx koristit će se mogućnostima postavljenim u konfiguracijskim datotekama navedenim u poglavlju 11.2.2., ako su tamo definirani.

Tri su značajne razlike u ta dva načina pokretanja poslužitelja:

1. Kad se server pokreće naredbom ne treba provesti postupak prijave. Server će se pokrenuti pod korisnikom koji je pokrenuo izvršavanje naredbe i isti će se korisnik prijaviti u upravitelj prikaza. Zbog toga se, kad se rabi korisničko grafičko sučelje (GUI), ne pokreće

xlogin Xklijent koji upravlja postupkom prijavljivanja u sustav.

2. Učitat će se različite specifične konfiguracije. U slučaju pokretanja iz naredbene linije, pomoću naredbe **startx**, poznato je koji korisnik pokreće server pa se mogu učitati i korisniku specifične postavke definirane u konfiguracijskim datotekama, posebno u datoteci ***~/xinitrc***.

3. Pri pokretanju naredbom **startx** izrijekom se može definirati koja će se konfiguracijska datoteka rabiti. Zbog te se razlike serveri najčešće pokreću iz naredbene linije, a ne promjenom razine izvođenja *(runlevel)* ili ponovnim pokretanjem sustava.

11.5. **Upravitelj prikazom**

11.5.1. Upravitelj prikazom - GDM, KDM i XDM

Upravitelji prikazom (*Display Manager*) poznati su kao i *login*-upravitelji. To su grafička sučelja koja se pokreću na kraju procesa pokretanja operacijskog sustava i upravljaju čitavim prikazom. Postoje brojne implementacije upravitelja prikazom kao što postoje i upravitelji prozorima i desktop-okoline.

Upravitelji prikazom međusobno su različiti, a i u pojedinom upravitelju pristupom je moguće izvršiti prilagodbe i promjene te implementirati vizualno i funkcionalno vrlo različite teme.

Razmotrimo, primjera radi, dva upravitelja prikazom - A i B. Moguće je, na primjer, upravitelja prikazom A toliko promijeniti (nazovimo taj promijenjeni upravitelj prikazom A') da nova konfiguracija upravitelja prikazom (A') ima manje funkcionalnih i vizualnih sličnosti s izvornom konfiguracijom (A) nego s upraviteljem prikaza B.

Značajno zastupljenih upravitelja prikazom postoji desetak, ali su najviše zastupljena ova tri:

- **XDM** – *X Window Display Manager*
- **GDM** – *GNOME display manager*
- **KDM** – *KDE display manager*.

Dizajn **XDM** vođen je standardom XDMCP (*X Display Manager Control Protocol*). XDM je minimalistički upravitelj prikazom i kao takav sposoban je raditi na najvećem spektru različitih konfiguracija softvera i hardvera. Primarna namjena XDM-a je omogućiti pristup potrebnim grafičkim funkcionalnostima s minimalnom instalacijom. Ako je apsolutno nužno na poslužitelju instalirati upravitelj prikazom, najčešće će se rabiti ovaj. Zbog minimalnosti sustava, XDM posjeduje i minimalne funkcionalnosti pa se većina korisničkih desktop okolina koristi novijim i funkcionalnostima bogatijim upraviteljima prikazom.

GDM se distribuira s grafičkom okolinom GNOME. GDM je osnovni upravitelj prikazom u distribuciji *Debian Linux* i instalira se ako nije izrijekom izabran neki drugi upravitelj prikazom ili ako nije odabrana instalacija koje ne uključuje GUI. Izvršna datoteka GDM i ime paketa je **gdm3**, a izvršna datoteka za pokretanje upravljača prikazom je ***/usr/sbin/gdm3***.

KDM je, uz GDM, najkorišteniji upravitelj prikazom. Razvila ga je KDE, međunarodna zajednica slobodnog softvera. Osim uobičajenih razlika, poput lokacije konfiguracijskih datoteka, imena i lokacije središnje izvršne datoteke, postoje i implementacijske i konfiguracijske razlike.

Iako su i **KDM** i **GDM** različiti na mnogo razina, izbor između njih nije lagan jer:

- ni jedan upravljač prikazom nije objektivno „bolji“
- kompatibilnost s vanjskim softverom je identična ili gotovo identična. Na primjer, tipični softver s grafičkom komponentom koji nije razvijen kao dio jednog od upravitelja prikazom biti će ili kompatibilan s oba ili nekompatibilan s oba.

Pri izboru upravitelja prikazom, osim osobne udobnosti i osjećaja, korisnik ima tek jednu stvar koja ga može usmjeriti, a to je **kompatibilnost s operacijskim sustavom**. Poznato je da će upravitelj prikaza koji je standardan za neku distribuciju (kao GNOME za *Debian*) biti podvrgnut temeljitijim i sveobuhvatnijim provjerama kompatibilnosti.

U *Debian*u su na raspolaganju ove mogućnosti:

- izabrati upravitelja prikazom pri instalaciji
- naknadno ga instalirati iz repozitorija paketa
- prevesti ga (*compile*) iz izvornog kôda.

11.6. Izbor desktop okoline

11.6.1. Desktop okoline

Desktop-okolina ujedinjuje razne *XKlijente* s ciljem oblikovanja grafičkog sučelja pomoću elemenata poput ikona, alatnih traka, desktop-pozadina i desktop-*widget*a. Većina desktop-okolina uključuje i skup integriranih aplikacija i usluga. Desktop-okolinu čini još i upravitelj prozorima. Postojeći upravitelj prozorima može biti zamijenjen nekim drugim kompatibilnim upraviteljem prozorima.

Desktop-okolina pruža korisnicima cjelovite i intuitivne alate za prilagodbu grafičke korisničke okoline vlastitim potrebama. Korisnici, ako to žele, mogu kombinirati aplikacije iz različitih desktop-okolina. Na primjer, korisnici **KDE**-a mogu instalirati aplikacije **GNOME** poput *web*-preglednika *Epiphany*. Nedostatak korištenja brojnih aplikacija iz različitih desktop-okolina je oslanjanje aplikacija na očekivane pakete biblioteka koje, kad su instalirane na (njima) nestandardnim desktop-okolinama, nedostaju. Rezultat je situacija u kojoj za integraciju softvera treba zadovoljiti brojne ovisnosti.

Nadalje, aplikacije namijenjene određenoj desktop-okolini bolje se integriraju s tom okolinom. Vizualno i funkcionalno miješanje aplikacija može rezultirati neočekivanim ponašanjem. Na primjer, moguće je da dio aplikacija reagira na jedan klik, a dio na dvoklik ili da se promjena prozora (veličina, oblik i druge postavke) drugačije izvodi na drugim aktivnim površinama i slično.

Dekstop okolina	Kratki opis
Cinnamon	Grana GNOME 3
Enlightenment	Neovisno razvijana, zasnovana na bibliotekama *enlightenment foundation*.
GNOME	Sadrži i *GNOME Classic*
KDE Plasma	Aktualna inačica desktop-okoline KDE
LXDE	„Lightweight X11 Desktop Environment“-cilj je te desktop-okoline smanjenje zahtjeva prema procesoru i RAM-u.
LXQt	Port LXDE, iduća inačica LXDE
MATE	Grana GNOME 2
Xfce	Modularnost i ne zahtjevnost prema računalnim resursima su karakteristike XFCE-a.

11.6.2. GNOME

Desktop-okolina GNOME ima cilj biti jednostavna i usmjerena na potrebe korisnika. Razvijena je kao dio projekta **THE GNOME*,*** koji je dio projekta **GNU** i u potpunosti je sastavljena od slobodnog i otvorenog softvera. Desktop-okolina *Gnome* podržava distribucije izvedene iz **BSD**-a, ali primarno je razvijena za *Linux*ove distribucije.

Aktualna inačica desktop-okoline **GNOME** je **GNOME 3**. Inačica izdana 6. travnja 2011. godine izazvala je raspravu zbog velikih promjena u odnosu na prethodnu. Promjene su bile tolike da je *Debian* s **GNOME 2** prešao na **Xfce**. Tek tijekom 2014. godine je **GNOME** ponovno standardna desktop-okolina distribucije *Debian*.

11.6.3. Ljuska GNOME

Ljuska GNOME službeno je korisničko sučelje desktop-okoline **GNOME**. Izgled okoline može varirati, ali osnovna konfiguracija napravljena je od gornje alatne trake i desktop-površine.

Alatna je traka sastavljena od (s lijeva na desno):

1. tipke *Activities*
2. izbornika aplikacija
3. sata
4. integriranog sistemskog izbornika za status.

Izbornik aplikacija prikazuje ime aplikacije trenutačno u fokusu i omogućava izvršavanje operacija poput gašenja, otvaranja novog prozora i pristupa postavkama aplikacije. Važno je naglasiti da je prikazana samo aplikacija koja je u fokusu, a ne sve aplikacije sa trenutačno otvorenim prozorom.

Integrirani sistemski izbornik nalazi se u gornjem lijevom kutu i sastoji se od raznih sistemskih indikatora, prečaca do sistemskih postavki i prečaca za pristup sjedničkim (*session*) akcijama kao što su prijavljivanje i odjavljivanje korisnika te gašenja računala.

Najvažniji dio alatne trake je tipka *Activities* koja obavlja funkciju sličnu tipki *Start* u *Windows*ima. Pritiskom tipke *Activities* otvara se posebni desktop-„pregled“ (*Overview*). U top „pregledu“ vide se

sve trenutačne aktivnosti i moguće je jednostavno prelaziti između različitih aplikacija i radnih površine te pokretati nove aplikacije. Na lijevoj strani nalazi se izbornik prečaca do omiljenih aplikacija na slici označen brojem 1. Gore, desno od njega, nalazi se izbornik u kojem su, kad je odabrana opcija *Windows,* prikazane aktivne aplikacije i jednostavno je prelaziti između njih. Kad se odabere opcija *Applications,* pojavljuju se ikone svih instaliranih aplikacija i moguće ih je pokrenuti klikom. Također je pomoću prozora za pretraživanje u gornjem desnom kutu moguće pokrenuti željenu aplikaciju.

11.6.4. Konfiguracija GNOME

Teško je govoroiti o konfiguraciji **GNOME**, a ne spomenuti njezinu instalaciju. **GNOME** se može instalirati iz izvornog kôda preuzetog sa stranica projekta https://www.gnome.org/getting-gnome/ ili instalacijom odgovarajućih paketa (središnji paket u *Debianu* je G*nome*). Alternative instalaciji paketa G*nome* su paketi G*nome-desktop-environment* koji uključuje manji skup aplikacija ili paket G*nome-fifth-toe* koji uključuje značajno više aplikacija. Programeri koji žele razvijati dodatne komponente za GNOME mogu instalirati pakete G*nome-core-devel* ili G*nome-devel* od kojih je prvi manji, a drugi potpuni paket.

Jedan od češće korištenih načina za konfiguraciju desktop-okoline **GNOME** je pribavljanje postojeće, gotove konfiguracije **GNOME** koja se korisniku sviđa i (manja) prilagodba željama i potrebama korisnika. Na stranici http://gnome-look.org/ mogu se pribaviti potpune, modificirane ljuske **GNOME** koje se korisniku svide. Posebno je zanimljivo vidjeti do koje se razine može promijeniti izgled desktop-okoline **GNOME**. Na primjeru okoline **GNOME** prilagođene da izgleda kao Windows 7 http://gnome-look.org/content/show.php/Win2-7+Pack?content=113264. Taj je primjer posebno zanimljiv jer su ga vjerojatno izradili entuzijasti s naših prostora, što se može zaključiti po tome što su podržani jezici engleski, danski i hrvatski.

Na stranici https://extensions.gnome.org/ moguće je nabaviti razne dodatke za osnovnu ljusku GNOME. Neki od dodataka su minimalne promjene u ponašanju ljuske, dok su druge značajne izmjene načinu rada i funkcionalnostima ljuske.

Postojeća se konfiguracija može prilagoditi vlastitim potrebama. Naredba **gnome-tweak-tool** otvara sučelje za promjenu postavki ljuske **GNOME**. Isto se sučelje može otvoriti pomoću tipke *Activities* izborom *Advanced Settings*. Opcije su inicijalno podijeljene na konfiguracije:

- prozora
- sučelja (*interface*)
- upravljača datotekama (*File Manager*)
- fontova
- ljuske.

Naglasimo riječ inicijalno, s obzirom na to da proširenja dodana u ljusku šire opcije u pojedinim navedenim izbornicima, a neke i dodaju dodatne izbornike.
Često postoje i kategorije opcija:

- radna površina (*Desktop*)
- proširenja ljuske
- teme.

11.6.5. KDE Plasma

KDE je međunarodna zajednica slobodnog softvera koja izrađuje aplikacije za razne platforme kao što su sustavi *Linux*, FREEBSD, *Microsoft Windows* i *OS X*. Cilj je zajednice razvoj osnovnih funkcionalnosti za desktop i za aplikacije za ispunjavanje svakodnevnih potreba korisnika i programera aplikacija.

KDE Plasma 5 je desktop koji je zajednica **KDE** razvila primarno za *Linux*ove distribucije, a prva je inačica izdana 15. srpnja 2014. godine. Desktop-okolina *Plasma 5* označava kraj migracije desktop-okoline **KDE** na aplikacijski okvir (*application framework*) *QtQuick*.

KDE se može instalirati isto kao i **GNOME**, pomoću izvornog kôda ili instalacijom odgovarajućih paketa. Postoje tri paketa koji se mogu instalirati ovisno o potrebama.

Naziv paketa	Opis
kde-plasma-desktop	Minimalna inačica s minimalnim funkcionalnostima.
kde-standard	Standardna inačica idealna za većinu namjena.
kde-full	Potpuna inačica sa svim funkcionalnostima.

11.6.6. Razlike između GNOME-a i KDE Plasme

Velika promjena između **GNOME**-ovih inačica 2 i 3 bila je u dugmetu *Activities* koje funkcionira slično dugmetu *Start* u operacijskom sustavu *Windows 8*. Ta promjena je vizualna i funkcionalna. **KDE** je, s druge strane, s inačicom 5 uveo velike tehničke promjene sa strane načina rada, ali vizualno i funkcionalno *Plasma 4* i *5* sliče nekom klonu *Windows*a*7*.

Važna razlika je i da prozori aplikacija **GNOME** u osnovnim postavkama imaju samo dugme za gašenje. **KDE** ima tri kontrole kakve su standardno dostupne i u operacijskom sustavu *Windows*:

- *Close*
- *Maximize/Restore* (kontekstno ovisno)
- *Minimize*.

Što se tiče brzine rada i procesorske zahtjevnosti, prednost ima **KDE** koji je velike preinake napravio prelaskom na inačicu 5, 2014. godine. Te preinake nisu bile vezane samo uz izgled nego i na programski okvir na kojem funkcionira **KDE**. **KDE** je tako modernizirao dio kôda koji upravlja načinom korištenja hardverskih grafičkih komponenti. Ukupni je rezultat da **KDE** radi brže i ima manje zahtjeva. **GNOME** bez korištenja ubrzanja 3-D (3-D *acceleration*) ne može raditi. U tom slučaju **GNOME** se vraća na osnovni, klasični izgled ograničenih funkcionalnosti. **KDE** bez poteškoća radi bez ubrzanja 3-D sa svim funkcionalnostima. Potrebno je napomenuti da je ta razlika važna samo kod pokretanja sustava na **vrlo** ograničenim ili zastarjelim hardverskim konfiguracijama i najčešće u nekim testnim okolinama koje uključuju virtualizaciju hardvera.

11.6.7. Xfce

Xfce je desktop-okolina slobodnog softvera za *Unix* i platforme temeljene na *Unix*u kao što su *Linux*, *Solaris* i BSD. Ciljevi okoline su **Xfce** brzina, mala zahtjevnost za resursima, vizualna privlačnost i jednostavnost korištenja. U programskom dizajnu **Xfce** stavlja naglasak na modularnost. **Xfce** u stvari čini niz zasebno pakiranih dijelova koji zajedno pružaju sve

funkcionalnosti desktop-okoline. Takav dizajn omogućava izbor podskupa dijelova ovisno o potrebama korisnika.

Xfce se na *Debian*u može instalirati iz izvornog kôda sa stranica http://archive.xfce.org/xfce/. Ime paketa za instalaciju je *xfce4*, a ako je to potrebno, moguće je proširiti instalaciju dodatnim korisnim aplikacijama instalacijom paketa **xfce4-goodies** ili svim u repozitoriju dostupnim aplikacijama instalacijom svih paketa ***xfce*** odnosno **xfce4-***.

11.6.8. Razlike između GNOME i Xfce

S tehničke strane **Xfce** dijeli više sličnosti s **GNOME**-om, ali starijom inačicom, **GNOME 2**. **Xfce** se temelji na GTK+ 2 – *widget toolkit*u za izgradnju grafičkih korisničkih sučelja isto kao i **GNOME 2.**

Xfce kao i **KDE** bez poteškoća radi bez ubrzanja 3-D i očekivano je najmanje zahtjevan od sve tri desktop okoline. Vizualno je **Xfce** sličniji **GNOME**-u s alatnom trakom smještenom na vrhu. Funkcionalno je **Xfce** kombinacija dviju prije opisanih desktop-okolina. Tako je alatna traka smještena na vrh ekrana kao kod okoline **GNOME**, ali klikom na „aplications menu“ otvara izbornik sličan **KDE**-ovom. Na dnu u sredini nalazi se alatna traka sa najčešće korištenim aplikacijama:

- terminal (korisnički)
- upravljač datotekama
- *web*-preglednik
- *eMail* klijent (ako je grafički email klijent instaliran)
- prečac za pretraživanje (eng. *Application search*)
- prečac do korisnikova direktorija *home*

Konfiguracija **Xfce**-a provodi se isključivo kroz niz grafičkih sučelja. Većina konfiguracije odvija se pomoću *Settings manager*a kojem se pristupa putanjom *Application Menu* → *Settings* → **Settings Manager**.

Kao (nepotrebno) loše naglašavanje razlika ističe se pozicioniranje tipke za upravljanje korisničkom sjednicom i gašenje. Smještena u gornjem lijevom uglu odmah do „glavne“ tipke, *Application Menu* ta važna opcija poziva za poboljšanjem konfiguracije.

11.6.9. Dodatni sadržaji

- *The GNOME Project* - http://en.wikipedia.org/wiki/The_GNOME_Project
- *GNU Project* - http://en.wikipedia.org/wiki/GNU_Project
- *Xfce* - http://en.wikipedia.org/wiki/Xfce
- *KDE Plasma 5* - http://en.wikipedia.org/wiki/KDE_Plasma_5
- *KDE* - http://en.wikipedia.org/wiki/KDE

11.7. **Vježba: Grafička okolina X**

1. Prijavite se na računalo kao korisnik l102. U GUI-u pokrenite **Terminal** (*Activities* → **Terminal**).
2. Naredbom „**su -**“ postanite **root** korisnik (lozinka: L102).
3. Pokrenite naredbu **top** i proučite rezultat. Koji se servisi najčešće pojavljuju na vrhu popisa?

4. Naredbom „**su – l102**“ postanite l102 korisnik. Pokrenite Xklijenta **xclock**, zatim izađite iz Xklijenta. Pokrenite ga ponovno kao pozadinski proces.

```
# xclock      CRTL+C
# xclock &
```

5. Pokrenite još jednom **xclock** kao pozadinski proces. Što se dogodilo?

6. Unesite u datoteku **.Xresources** u direktoriju home trenutačnog korisnika (l102):

```
! xclock ---------------------------
xclock*update:             1
xclock*analog:             false
xclock*Foreground:         white
xclock*background:         black
```

7. Pokrenite **xclock** naredbom

```
xrdb -merge ~/.Xresources && XClock &
```

   Što se dogodilo?

8. Pokrenite naredbu **reboot**. Pri pokretanju sustava zaustavite pokretanje u izborniku ***GRUB2*** i odaberite uređivanje (e). Promijenite liniju koja završava s “**ro quiet**“ tako da završava s „**ro text**“. Pokrenite sustav (**[CTRL]+[X]** ili **[F10]**).
9. Prijavite se u sustav kao korisnik **l102**. Što se dogodilo? Kako je to drugačije od standardnog pokretanja?

I. Rješenja

Vježba 1.3.

3.a. `#lsmod`

`#lsmod | grep usb-core`

`(ohci_hcd,ehci_pci,usbhid,ehci_hcd,ohci_pci)`

4.

```
# uname –r
```

rezultat: `5.10.0-16-amd64`

```
# ls /lib/modules/5.10.0-16-amd64/
# cd /lib/modules/5.10.0-16-amd64/
```

7.

```
# modprobe snd-hda-codec-idt
# modprobe snd-hda-codec
# modprobe snd-hda-codec-hdmi
# lsmod |wc -l
```

8.

```
# modprobe –r snd-hda-codec
# modprobe –r snd-hda-codec-idt
# modprobe –r snd-hda-codec-hdmi
# lsmod |wc –l
```

U nekim koracima nije bilo moguće ukloniti modul.

Vježba 1.4.

4. `$ mkdir –p /tmp/jezgra`

Vježba 2.5

4. `rm -rf /lib/systemd/system/default.target`

`ln -s /lib/systemd/system/multi-user.target /lib/systemd/system/default.target`

6.

NE - CLI

Broj aktivnih procesa je manji od 5. zadatka

7.

Pokrece se gui sucelje.

Broj procesa ja najveći

8. `ln -s /lib/systemd/system/graphical.target /etc/systemd/system/default.target`

9. vrijednosti su se vratile na ranije (5. zadatak)

Vježba 2.6

3.

2 jezgre: `5.10.0-14-amd64 i 5.10.0-16-amd64`

4.

4 datoteke, za svaku jezgru ima i rescue mode

7. Sustav se pokrenuo u single user modu.

9. I dalje ista jezgra 5.10.0-16-amd64 ali sada u single user modu

12. 5.10.0-16-amd64 jezgra jer single user mode samo ogranici pristup ali koristi istu jezgru

Vježba 3.4

10.

```
#passwd sistem001
```

11.

Ne postoji definicija home direktorija za sistem001

Vježba 4.7

3.

```
# mkdir -p /home/l102/skriptanje
# mkdir -p /home/l102/skriptanje
```

4.

```
# touch prihvat_unosa.sh
# chmod +x prihvat_unosa.sh
```

6.

```
#!/bin/bash
for Brojac in 1 2 3 4 5 6 7 8 9 10 ; do
        echo "Ovo je $Brojac. korak "
done
```

7.

```
#!/bin/bash
for Brojac in 1 2 3 4 5 6 7 8 9 10 ; do
        echo "Ovo je $Brojac. korak "
     echo "Parametar je: $1"
     shift
done
```

8.

```
#!/bin/bash brojac=1
while [ $brojac -le 10 ];
do
echo "Ovo je $brojac. korak"
let brojac+=1
done
```

9.

```
#!/bin/bash
brojac=1
broj_parametara=$#

while [ $brojac -le 10 ];
do
echo "Ovo je $brojac. korak"
if [ $brojac -le $broj_parametara ] ; then
echo "$brojac. Parametar je: $1"
     else
echo "$brojac. Parametar nije zadan"
fi
shift
let brojac+=1
done
```

Vježba 5.5.

```
$ ipcalc 161.53.17.99/27

Address:   161.53.17.99         10100001.00110101.00010001.011 00011

Netmask:   255.255.255.224 = 27 11111111.11111111.11111111.111 00000
```

```
Wildcard:  0.0.0.31             00000000.00000000.00000000.000 11111

=>

Network:   161.53.17.96/27      10100001.00110101.00010001.011 00000

HostMin:   161.53.17.97         10100001.00110101.00010001.011 00001

HostMax:   161.53.17.126        10100001.00110101.00010001.011 11110

Broadcast: 161.53.17.127        10100001.00110101.00010001.011 11111

Hosts/Net: 30                    Class B
```

Vježba 6.6.

3.

```
# ifconfig
#ip addr show
1 aktivan 10.0.3.?
```

4.

ne poklapa - kartice su definirane u gui-u

5.

primjer :

```
# vim /etc/network/interfaces
"
auto lo enp0s8:0
iface lo inet loopback
iface enp0s8:0 inet static
      address 10.0.3.115/24
```

```
      gateway 10.0.3.2
"
```

6.

```
# ping 10.0.3.115
```

- nije

7.

nije - jer pomoću network-online konfigurira se mreža kroz GUI

8.

```
# systemctl restart networking
```

9.

sucelje je sada aktivno

12. Ne

13.

I dalje ne jer network-online ne pokrene ponovno mrežna sučelja već samo njihov konfigurator.

14. Da

15.

```
# route
# ip route list
```

16.

```
# ip link set enp0s3 down
```

17. Ne radi

18.

```
# ip link set enp0s3 up
```

Ping i dalje ne prolazi odnosno mreža ne radi dok se ne spusti drugo sucelje i time ne pocisti routing

7.5.

4. NEma zapisa

5. Zapisi samo kada se pali sučelje

6.

na kraj datoteke /etc/rsyslog.conf:

```
"local2.info    /var/log/vjezba.log"
# systemctl restart rsyslog

# touch /var/log/vjezba.log
# chmod 664 /var/log/vjezba.log
# chown root:adm /var/log/vjezba.log
```

7. `# logger -p local2.info "Dogadjaj vrlo vazan"`

Vježba 7.6.

3.

```
# rsync -av /etc /home/l102/
```

4.

```
# crontab -e
```

Dodati na kraj:

```
0 5 * * * rsync -av /etc /home/l102/ > /dev/null 2>&1
```

5.

```
# vi /home/l102/bakup.sh
#!/bin/bash
tar -cvf /tmp/etc.tar /etc/
```

6.

```
# vi /home/l102/bakup2.sh
"
#!/bin/bash
rm -rf /home/etc.tar
tar -cvf /home/etc.tar /etc/
rsync -av /etc /tmp/
"
# chmod +x /home/l102/backup2.sh

# mv /home/l102/backup2.sh /etc/cron.hourly/
```

7.

```
# /etc/cron.hourly/backup2.sh
# ls /tmp/
```

Samo jedna kopija se cuva

Greške starije od 1 sat nije moguće ispraviti

Datoteke nisu zastićene

Prenosi se previše podataka nepotrebno...

Vježba 8.11.

3. `# vim /usr/sbin/pozdrav`

```
#!/bin/bash
echo Dobro dosli!
```

```
# chmod +x /usr/sbin/pozdrav
```

6.

```
# telnet 127.0.0.1 55000
```

7. poruka se promijeni

Vježba 8.12.

5.

```
# cat /etc/bind/named.conf.options |grep dire
      directory "/var/cache/bind";
```

7.

```
# chgrp bind forward.tecaj
# chgrp bind reverse.tecaj
# named-checkconf /etc/bind/named.conf.local
# named-checkconf /etc/bind/named.conf.options
# named-checkzone tecaj.local forward.tecaj
# named-checkzone tecaj.local reverse.tecaj
# systemctl restart bind9
```

Vježba 8.13.

2. `ls -a /home/l102/`

4. Nastao je direktorij .ssh

6. `$ ssh-copy-id l102@127.0.0.1`

Vježba 8.14.

10. Dobili smo default stranicu apache servera na debianu jer je portebno ponovno pokrenuti apache2

Vježba 9.4.

3.

```
# touch /tmp/neunistiva
# chattr +i /tmp/neunistiva
```

4. Nije moguće cak ni sa -f opcijom

5.

```
# chattr -i /tmp/neunistiva # su - l102
$ rm -f /tmp/neunistiva
$ rm: cannot remove `/tmp/neunistiva': Operation not permitted
```

6. Provjerom stanja lanaca jer vatrozid sacinjavaju lanci

```
# iptables -nL
```

7.

```
# service ssh status
# ssh l102@127.0.0.1
```

8.

```
# iptables -A INPUT -p tcp --dport 22 -j DROP
```

9.

Ne radi

10.

```
# nmap -v 127.0.0.1
# Port 22 is filtered
```

11.

```
#iptables -F
```

12. terminal opet radi i port je otvoren prema rezulatu skeniranja

13.

```
# iptables -P INPUT DROP
```

14. Terminal je ponovno neaktivan a skeniranje traje dugo.

All 1000 scanned ports on localhost (127.0.0.1) are filtered

Vježba 9.5.

3.

```
# iptables -P INPUT DROP
# iptables -P FORWARD DROP
```

6. ssh ne tadi jer je server i ishodište i odredište (odbijaju se paketi kada je odredište)

```
apt radi
```

Vježba 10.3.

3.

```
# less /etc/cups/cupsd.conf|grep -v "#"
# less /etc/cups/cupsd.conf|grep Listen
```

Port je 631

4.

```
$ firefox 127.0.0.1:631
```

5. No printers.

7. Vidljiv je pdf printer

8.

```
# vim /etc/cups/cupsd.conf
# systemctl restart cups.service
```

9.

```
$ firefox 127.0.0.1:631
```

nema sucelja - "Unable to connect"

```
$ firefox 127.0.0.1:11631
```

vidljivo

Vježba 10.4.

3. `# libreoffice /etc/resolv.conf`

5. `$ libreoffice ~/PDF/resolv.conf`

Jest, identican je originalnoj datoteci.

Vježba 11.7.

3.

```
gnome-*
top
Vbox
```

5.

Imamo 2 sata

7. Dobili smo digitalni sat

9. Na ekran se ispisuju poruke o koracima pokretanja

Bilješke