Napredno korištenje operacijskog sustava Linux

Mrežni protokoli

Bojan Novković

Nositelj: doc. dr. sc. Stjepan Groš

Sveučilište u Zagrebu, Fakultet elektrotehnike i računarstva

3. travnja 2020.

1 Uvod

Računalne mreže neophodan su dio današnjeg računarstva i sastavni dio svakog operacijskog sustava. Ovaj tekst prolazi kroz većinu aspekata računalnih mreža u operacijskom sustavu GNU/Linux uz pregršt primjera.

2 Mrežne postavke jezgre

Temeljna podrška za sve operacije vezane uz mrežnu komunikaciju implementirana je u Linux jezgri. Konfiguracija mrežnog dijela jezgre moguća je na više načina prilikom prevođenja jezgre. U nastavku slijedi pregled važnih konfiguracijskih opcija jezgrenog mrežnog podsustava. Primjeri teksta dobiveni su kroz konfiguraciju jezgre pomoću naredbi `make nconfig` ili `make menuconfig`. Svi dani primjeri nalaze se u odjeljku `Networking support -> Network options`

2.1 Podrška za TCP/IP

Temelj današnje mrežne komunikacije nalazi se u `TCP/IP` podsustavu jezgre kojeg je potrebno omogućiti prilikom prevođenja jezgre.

```
[*] TCP/IP networking
```

2.2 Gateway

Ukoliko će računalo vršiti usmjerivanje paketa između lokalne mreže i **ISP-a** tada je potrebno uključiti jezgrin podsustav za usmjerivanje paketa.

```
[*] IP: advanced router
```

Valja napomenuti kako je uključivanje ove opcije moguće i izvršavanjem slijedeće naredbe uz uvjet da je u jezgri omogućen virtualni datotečni sustav `/proc`:
`echo ‘‘1’’ > /proc/sys/net/ipv4/ip_forward`

2.3 Podrška za VPN

Linux jezgra nudi mogućnost tuneliranja paketa prema **IPSec**[1] standardu te nudi podršku za **Authentication Header** i **Encapsulating Security Payload**.

```
[*] IP: AH transformation
[*] IP: ESP transformation
```

2.4 Filtriranje paketa

Ako će računalo obnašati ulogu *firewalla*, tada je potrebno uključiti jezgrin podsustav za filtriranje paketa kako bi se mogla provoditi pravila definirana u *firewallu*. Najkorišteniji podsustav za filtiranje paketa na operacijskom sustavu GNU/Linux jest `netfilter` kojeg se koristi preko naredbe `iptables` o kojoj će biti više riječi kasnije. Treba napomenuti kako `netfilter` nije jedini dostupan sustav za filtiranje paketa već se nudi podrška i za **Berkeley Packet Filter**.

[1] Detaljnije o **IPSecu**: `https://www.routeralley.com/guides/ipsec_overview.pdf`

```
[*] Network packet filtering framework (Netfilter)
[*] BPF based packet filtering framework (BPFILTER)
```

Dodatno, `netfilter` podsustav nudi pregršt opcija za praćenje, snimanje i manipuliranje paketima:

```
< > IPv4 tproxy support
< > Netfilter IPv4 packet duplication to alternate
 destination
< > ARP packet logging
< > IPv4 packet logging
< > IPv4 packet rejection
< > IP tables support (required for filtering/masq/NAT)
< > ARP tables support
```

2.5 IPv6

Linux jezgra nudi podršku i za **IPv6** protokol kojega je potrebno omogućiti prilikom prevođenja jezgre. Unutar izbornika za konfiguraciju **IPv6** protokola nalazi se pregšt opcija u koje nećemo dalje ulaziti.

```
< > The IPv6 protocol
```

3 Vrste mrežnih sučelja

Linux jezgra nudi podršku za velik broj mrežnih uređaja koje predstavlja pomoću mrežnih sučelja. Razlikujemo dvije vrste mrežnih sučelja, **virtualna** i **fizička**. Virtualna mrežna sučelja ne predstavljaju fizički mrežni uređaj iako mogu biti logički povezana s njime dok su fizička sučelja direktna reprezentacija mrežnog uređaja. Nazivi sučelja dodjeljuju se ili prema vrsti, ukoliko se radi o virtualnim sučeljima, ili prema upravljačkom programu potrebnom za ispravan rad, ukoliko je riječ o fizičkom sučelju.

3.1 Vrste virtualnih sučelja

- loopback, `lo`

  Loopback sučelje prisutno je na svakom operacijskom sustavu i koristi se primarno za testiranje mrežnih usluga. IP adresa tog sučelja jest `127.0.0.1` te je dostupno pod simboličkim imenom `localhost`.

- bridge, `br0`

  Bridge sučelja koriste se za povezivanje više fizičkih i/ili virtualnih sučelja u jedno sučelje. Funkcioniraju skoro identično kao L2 *switch*, proslijeđivanje paketa vrši pomoću `MAC` adresa i tablice "učenja" u kojoj se povezuju viđene `MAC` adrese s izvorišnim uređajem.

- tuneli, `tun, tap`

Ova sučelja koriste se primarno za `VPN` komunikaciju, odnosno tuneliranje prometa kroz neki drugi protokol. Međutim, ova vrsta sučelja pogodna je za testiranje i razvijanje mrežnih protokola te za pružanje mrežnih sučelja virtualnim strojevima [2].

- bežična sučelja, `mon0`

Bežična sučelja vežu se na fizički bežični uređaj. Nazivi ovise o načinu rada sučelja[3].

4 Alati za interakciju sa mrežnim podsustavom

U sklopu operacijskog sustava GNU/Linux dostupno je puno alata za interakciju sa jezgrinim mrežnim podsustavom iz korisničkog načina rada. Poznavanje rada s ovim alatima je ključno za uspješno administriranje svih uređaja i mrežnih usluga prisutnih na njima.

4.1 ifconfig, ip

Alati `ifconfig` i `ip` koriste se za konfiguraciju mrežnih sučelja. Oba alata obavljaju istu funkcionalnost, uz razlike u sintaksi, te su na većini distribucija često oba prisutna.

Listing 1: Ispis svih aktivnih mrežnih sučelja pomoću alata `ifconfig`

```
$ ifconfig
enp0s31f6: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
        inet 192.168.1.9 netmask 255.255.255.0 broadcast 192.168.1.255
        inet6 fe80::da05:81ab:dcb4:2385 prefixlen 64 scopeid 0x20<link>
        ether 40:8d:5c:71:6c:39 txqueuelen 1000 (Ethernet)
        RX packets 1466089 bytes 2025690513 (1.8 GiB)
        RX errors 0 dropped 416 overruns 0 frame 0
        TX packets 702906 bytes 62074091 (59.1 MiB)
        TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
        device interrupt 16 memory 0xdff00000−dff20000

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
        inet 127.0.0.1 netmask 255.0.0.0
        inet6 ::1 prefixlen 128 scopeid 0x10<host>
        loop txqueuelen 1000 (Local Loopback)
        RX packets 3952 bytes 14644094 (13.9 MiB)
        RX errors 0 dropped 0 overruns 0 frame 0
        TX packets 3952 bytes 14644094 (13.9 MiB)
        TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

Slijedi usporedni pregled funkcionalnosti oba alata:

[2] Za one koje zanima više: `https://www.kernel.org/doc/Documentation/networking/tuntap.txt`

[3] Opširnije o mogućim načinima rada bežičnih uređaja: `https://wireless.wiki.kernel.org/en/users/documentation/iw`

Operacija	`ifconfig`	`ip`
Pregled sučelja	`ifconfig`	`ip a`
Uključivanje i isključivanje sučelja	`ifconfig <sučelje> up|down`	`ip link set <sučelje> up|down`
Postavljanje statičke IP adrese	`ifconfig <sučelje> <IP>`	`ip address add|del <IP> dev <sučelje>`
Postavljanje podmreže	`ifconfig <sučelje> netmask <vrijednost>`	`ip address add|del <IP>/<netmask> dev <sučelje>`
Omogućavanje DHCP konfiguracije sučelja	`ifconfig <sučelje> dhcp start`	-
Omogućavanje ARP protokola	`ifconfig <sučelje> arp`	`ip link set <sučelje> arp on`

4.2 route

Alat `route` služi za manipulaciju jezgrinom tablicom usmjerivanja.

Listing 2: Prikaz tablice usmjerivanja uz pomoć alata `route`

```
$ route
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
default         _gateway        0.0.0.0         UG    100    0        0 enp0s31f6
192.168.1.0     0.0.0.0         255.255.255.0   U     100    0        0 enp0s31f6
```

Listing 3: Prikaz najkorisnijh naredbi alata `route`

```
# Dodavanje gateway-a
$ route add default gw <IP adresa gateway-a>
# Dodavanje pravila za usmjerivanje
$ route add -net <cidr> gw <IP adresa gateway-a> <sucelje>
# Brisanje pravila za usmjerivanje
$ route del -net <cidr>
```

4.3 arp, arping

Alat `arp` služi za manipulaciju jezgrinom `ARP` tablicom dok alat `arping` služi za slanje `ARP` zahtjeva.

Listing 4: Prikaz `ARP` tablice uz pomoć alata `arp`

```
$ arp -a
Address                  HWtype  HWaddress           Flags Mask            Iface
_gateway                 ether   64:6e:ea:54:81:ec   C                     enp0s31f6
newpad                   ether   c8:5b:76:22:55:eb   C                     enp0s31f6
192.168.1.3              ether   20:c9:d0:9a:f5:79   C                     enp0s31f6
192.168.1.20             ether   00:26:ab:01:da:5a   C                     enp0s31f6
```

Listing 5: Prikaz najkorisnijh naredbi alata `arp`

```
# Dodavanje adrese u ARP tablicu
$ arp -s <IP adresa> <MAC adresa>
# Brisanje unosa za IP adresu na sucelju
$ arp -i <sucelje> -d <IP adresa>
```

Listing 6: Slanje `ARP` zahtjeva uz pomoć alata `arping`

```
$ arping -I <sucelje> 192.168.1.1
ARPING 192.168.1.1 from 192.168.1.9 <sucelje>
Unicast reply from 192.168.1.1 [64:6 E:EA:54:81:EC] 1.127ms
Unicast reply from 192.168.1.1 [64:6 E:EA:54:81:EC] 1.054ms
...
```

4.4 netstat

Alat netstat služi za pregled svih aktivnih konekcija na sustavu. Moguće je filtrirati pregled po raznim vrstama *socketa*[4], npr. `TCP` ili `UDP`. Korisne zastavice za ovaj alat su:

- `-a`, ispisuje sve sockete, neovisno jesu li *"listening"* socketi.
- `-t, -u`, ispisuje sve `TCP` (-t) i `UDP` (-u) sockete.
- `-n`, ispisuje numeričke adrese bez simboličkih imena.
- `-p`, ispisuje *PID* procesa koji je stvorio taj socket.

Listing 7: Prikaz konekcija uz pomoć alata `netstat`

```
$ netstat -antp
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID
tcp        0      0 0.0.0.0:4713            0.0.0.0:*               LISTEN      1320/pulseaudio
tcp        6      0 127.0.0.1:39223         0.0.0.0:*               LISTEN      90222/python
tcp        0      0 0.0.0.0:631             0.0.0.0:*               LISTEN      -
tcp        0      0 127.0.0.1:1337          0.0.0.0:*               LISTEN      190230/ssh
tcp        0      0 192.168.1.9:34496       3.120.198.117:443       ESTABLISHED 1006/firefox
tcp        0      0 127.0.0.1:39223         127.0.0.1:50006         ESTABLISHED 90222/python
tcp        0      0 192.168.1.9:58896       172.217.23.206:443      ESTABLISHED 1006/firefox
```

4.5 dig

Alat `dig` (*Domain Information Groper*) služi za slanje upita `DNS` poslužiteljima.

Listing 8: Prikaz najkorisnijh naredbi alata `dig`

```
# Dohvacanje IP adrese iz simbolickog imena
$ dig www.kset.org +short
# Dohvacanje IP adrese iz simbolickog imena sa specificnog posluzitelja
$ dig @8.8.8.8 www.kset.org +short
# Dohvacanje simbolickog imena iz IP adrese
$ dig -x 8.8.8.8
# Dohvacanje bilo kojeg DNS zapisa
$ dig www.kset.org <tip DNS zapisa>
```

[4]Detaljnije o socketima: `https://www.cs.rpi.edu/~moorthy/Courses/os98/Pgms/socket.html`

4.6 iptables

Alat `iptables` služi za interakciju sa jezgrinim sustavom za filtriranje paketa i najvažniji je alat za konfiguraciju pravila *firewalla*. Pomoću njega definiraju se pravila koja propuštaju ili odbijaju pakete na temelju transportnog protokola, IP adrese, portova, itd. Tipično pravilo sastoji se od željenog uzorka paketa i akcije koju se poduzima prilikom prepoznavanja uzorka. Pravila se prilikom definiranja smještaju u jedan od "lanaca" pravila definiranih u sustavu. Tri predefinirana lanca pravila koji su prisutni na svakom sustavu su:

- `INPUT` - svi paketi namijenjeni *host* računalu,
- `OUTPUT` - svi paketi koje je *host* računalo stvorilo,
- `FORWARD` - svi paketi koje *host* računalo usmjerava.

Akcije se sastoje od "skakanja" na željeni lanac, s time da slijedeći lanci označavaju kraj obrade paketa:

- `ACCEPT` - paket se prihvaća,
- `DROP` - paket se odbacuje bez povratnih informacija,
- `REJECT` - paket se odbacuje uz slanje odgovarajuće `ICMP` poruke ili `TCP` paketa sa `RST` zastavicom.

Pravila se obrađuju slijedno, redom kojim su dodavani u lance. Ukoliko nijedno pravilo nije bilo izvršeno, tada se izvršavaju pretpostavljene akcije definirane za pojedini lanac, poznate pod nazivom `POLICY`, kojom se odbacuje ili prihvaća paket. Preporuča se da se za sve lance `POLICY` postavi na `DROP` kako bi se povećala sigurnost mreže koju *firewall* štiti.

Ovdje valja napomenuti kako će prihvaćanje paketa imati drukčiju akciju ovisno o tome u kojem se lancu pravilo nalazi. Tako će prihvaćanje paketa za pravilo u `INPUT` lancu paket propustiti na daljnju obradu unutar *host* računala, dok će prihvaćanje paketa za pravilo u `FORWARD` lancu paket proslijediti na odredišno računalo.

Definiranje pravila postiže se pokretanjem alata `iptables` sa odgovarajućim argumentima i zastavicama. Primjerice, slijedeća naredba definira pravilo koje propušta sav nadolazeći promet za to računalo na *portu* 80:

```
$ iptables -A INPUT -p tcp --dport 80 -j ACCEPT
```

Slijedi pregled najvažnijih zastavica za alat `iptables`:

- `-A <ime lanca>` - dodaje pravilo u željeni lanac,
- `-p <protokol>` - specificira protokol,
- `--[sport|dport] <port>` - dodaje izvorišni ili odredišni *port* u uzorak,
- `-j <lanac>` - specificira akciju odnosno odredišni lanac na koji se "skače",
- `-i <sučelje>` - dodaje mrežno sučelje u uzorak.
- `-[s|d] <IP adresa>` - dodaje željenu `IP` adresu u uzorak kao izvorište ili odredište

Za iscrpniji popis mogućnosti i detaljnija objašnjenja pojmova konzultirajte odličan *HowTo* na CentOS-ovoj stranici.

Listing 9: Primjer postavljanja `iptables` pravila za računalo sa `DNS` i `SSH` poslužiteljima.

```
# Postavljanje POLICY vrijednosti za sve lance.
# Sve pakete bez odgovarajucih pravila odbacujemo.
$ iptables -P INPUT DROP
$ iptables -P OUTPUT DROP
$ iptables -P FORWARD DROP

# Dopustanje prometa prema DNS posluzitelju
$ iptables -A INPUT -p udp --dport 53 -j ACCEPT
$ iptables -A INPUT -p tcp --dport 53 -j ACCEPT
# Dopustanje prometa prema SSH posluzitelju
$ iptables -A INPUT -p tcp --dport 22 -j ACCEPT

# Dopustanje prometa sa DNS posluzitelja
$ iptables -A OUTPUT -p udp --sport 53 -j ACCEPT
$ iptables -A OUTPUT -p tcp --sport 53 -j ACCEPT
# Dopustanje prometa sa SSH posluzitelja
$ iptables -A OUTPUT -p tcp --sport 22 -j ACCEPT
```

5 Važnije datoteke za konfiguraciju mrežnog sustava

5.1 /etc/network/interfaces

Ova konfiguracijska datoteka sadrži postavke za mrežna sučelja koje se apliciraju prilikom svakog podizanja sustava[5].

Listing 10: Primjer konfiguracijske datoteke `/etc/network/interfaces`

```
# Postavljanje staticke IP adrese za sucelje eth0
iface eth0 inet static
address 192.168.1.5
netmask 255.255.255.0
gateway 192.168.1.254

# Postavljanje DHCP konfiguracije za sucelje eth1
auto eth1 # Automatsko podizanje sucelja prilikom izvrsavanja naredbe ifconfig -a
iface eth1 inet dhcp
```

5.2 /etc/resolv.conf

Ova konfiguracijska datoteka služi za postavljanje `DNS` resolvera. Sadrži niz zapisa u obliku `ključ vrijednost`, a najkorištenija polja su:

- `nameserver` - IP adresa `DNS` poslužitelja
- `search` - domena koju `DNS` resolver dodaje na imena kako bi oformio `FQDN`
- `options` - postavljanje raznih opcija `DNS` resolvera, (npr. `timeout`, `attempts`...)

Listing 11: Primjer konfiguracijske datoteke `/etc/resolv.conf`

```
search kset.org
nameserver 213.191.128.8
nameserver 213.191.128.9
```

[5]Razne distribucije imaju drukčije načine konfiguracije mrežnih sučelja, ovaj način navodimo jer je zastupljen u najkorištenijim distribucijama.

5.3 /etc/hosts

Ova konfiguracijska datoteka služi za postavljanje statičkih `DNS` mapiranja u obliku `IP-adresa simboličko-ime`.

Listing 12: Primjer konfiguracijske datoteke `/etc/resolv.conf`

```
# Static table lookup for hostnames.
# See hosts(5) for details.d
192.168.1.16 git
192.168.1.9 www.app.local
localhost kucni-komp
```

6 Dodatni alati

6.1 ssh

`SSH` je protokol za upravljanje udaljenim poslužiteljima [6]. Nudi mogućnosti za prebacivanje datoteka, izvršavanje naredbi i tuneliranje prometa. Alat `ssh` standardni je alat na većini Linux distribucija.

Listing 13: Prikaz najkorisnijh naredbi alata `ssh`

```
# Prijavljivanje na udaljenog posluzitelja .
# Nakon uspjesne prijave korisniku je dostupna
# komanda linija na udaljenom posluzitelju.
$ ssh <korisnik>@<simbolicko ime>
# Izvrsavanje naredbi na udaljenom posluzitelju
$ ssh -t <korisnik>@<simbolicko ime> <naredba>
# Mapiranje lokalnog porta na port udaljenog posluzitelja
$ ssh -L <lokalni port>:<ciljni posluzitelj >:<port udaljenog posluzitelja>
<korisnik>@<udaljeni posluzitelj>
```

6.2 rsync

Alat `rsync` je brz i svestran alat za pouzdano kopiranje datoteka. Uz lokalno kopiranje nudi mogućnost udaljenog kopiranja pomoću protokola `SSH`. Koristi poseban način prijenosa u kojem šalje samo one datoteke koje se razlikuju u izvorištu i odredištu.

Listing 14: Prikaz najkorisnijh naredbi alata `rsync`

```
# Kopiranje lokalne datoteke na udaljeni posluzitelj .
# Izvoriste i odrediste mogu biti lokalni i udaljeni
$ rsync <lokalna datoteka> <korisnik>@<simbolicko ime>:<odredisni direktorij>
# Kopiranje datoteka u "archive" nacinu koji
# cuva sve metapodatke o datoteci i rekurzivno kopira direktorije
$ rsync -a <lokalna datoteka> <korisnik>@<simbolicko ime>:<odredisni direktorij>
```

7 Web poslužitelj `Nginx`

`Nginx` je program otvorenog koda koji služi za posluživanje na webu, proxyanje prometa, balansiranje i još mnogo toga, a diči se jednostavnošću konfiguracije i brzinom.

Alat je dostupan u obliku paketa na većini distribucija te se njegova instalacija svodi na preuzimanje i instalaciju istoimenog paketa. Konfiguracijske datoteke nalaze se u direktoriju `/etc/nginx`.

[6] U detalje protokola nećemo ulaziti, ali preporučamo pročitati ovu poveznicu: `http://www.avoine.net/cyberedu/2015_07_ssh.pdf`

7.1 Struktura konfiguracijskih datoteka

`Nginx` se sastoji od modula koji su konfigurirani pomoću direktiva koje se nalaze unutar konfiguracijskih datoteka. Direktiva se sastoji od imena konfiguracijskog polja i parametara odvojenih razmacima te završava sa znakom `;`. Postoji i blok direktiva koji ima istu strukturu ali je omotana s vitičastim zagradama. Blok direktive koje moge imati druge direktive unutar vitičastih zagrada zovu se kontekst. Implicitni kontekst za sve direktive smještene van koteksta je `main` kontekst.

Glavna konfiguracijska datoteka jest `/etc/nginx/nginx.conf`. Najčešće se u njoj nalazi `http` blok koji sadrži niz direktiva za podešavanje rukovanja web prometom. Također, na većini distribucija se na kraju te konfiguracijske datoteke nalazi direktiva `include /etc/nginx/conf.d/*.conf;` ili `include /etc/nginx/sites-enabled/*;` kojom se `nginxu` daje do znanja da u tu konfiguracijsku datoteku zaljepi sadržaj svih datoteka u tim direktorijima. Najčešće se radi o konfiguracijskim datotekama za pojedine web stranice, gdje se u pravilu definira jedan ili više `server` konteksta u kojemu se direktivama opisuje virtualni poslužitelj.

7.2 Posluživanje statičnih datoteka

Posluživanje datoteka osnovna je funkcionalnost `nginxa`. U ovom dijelu pretpostavljamo da se datoteke koje želimo posluživati nalaze u direktoriju `/var/www` te da je u glavnoj konfiguracijskoj datoteci prisutna direktiva `include /etc/nginx/sites-enabled/*;`. Prvi korak jest stvaranje konfiguracijske datoteke `/etc/nginx/sites-enabled/static.conf`. U toj datoteci potrebno je definirati `server` kontekst te unutar njega definirati `location` blok. Unutar `location` ili `server` bloka potrebno je dodati `root` direktivu pomoću koje se specificira direktorij iz kojeg će `nginx` posluživati datoteke.

Listing 15: Početno stanje konfiguracijske datoteke `static.conf`.

```
server {
    location / {
        root /var/www;
    }
}
```

Nadalje, unutar `server` bloka potrebno je dodati direktivu `listen 80;` kojom se `nginxu` daje do znanja da taj virtualni poslužitelj treba slušati na portu 80. Time je osnovna konfiguracija posluživanja datoteka gotova te je konfiguraciju potrebno učitati pomoću naredbe `nginx -s reload`, nakon čega je stranica dostupna na adresi `http://localhost`. Međutim, stranica u ovom obliku nije prikladna za korištenje te je konfiguraciju potrebno nadograditi.

Prvi korak nadogradnje sastoji se u dodavanju direktive `server_name <simboličko ime>;` kojom se definira simboličko ime uz koje se taj `server` blok veže. `Nginx` poslužuje zahtjeve za više virtualnih poslužitelja definiranih pomoću više `server` blokova, a odluku o tome koji virtualni poslužitelj treba aktivirati kada primi `HTTP` zahtjev donosi pomoću `server_name` direktive i `Host` zaglavlja u `HTTP` zahtjevu.

Drugi korak nadogradnje sastoji se od dodavanja zapisa o primljenim `HTTP` zahtjevima i greškama koje su generirane tokom rada virtualnog poslužitelja. Tomu služe direktive `access_log` i `error_log`.

Listing 16: Konačno stanje konfiguracijske datoteke `static.conf`.

```
server {
    listen 80;
    server_name www.domain.com;

    access_log /var/log/nginx/domain-access.log;
```

```
    error_log /var/log/nginx/domain-error.log;

    location / {
        root /var/www;
    }
}
```

7.3 Posluživanje web aplikacija

Većina sadržaja na web-u je dinamički generirana i poslužuje se iz zasebnih web aplikacija u kojima je sadržana sva logika za posluživanje i generiranje sadržaja. `Nginx` podržava rad u "proxy" načinu pomoću kojega se može posluživati web aplikacija. Kao i u prethodnoj konfiguraciji, potrebno je staviti `server` i `location` blokove. Međutim, umjesto `root` direktive u `location` bloku potrebno je dodati `proxy_pass` direktivu koja kao argument prima adresu `HTTP` poslužitelja kojem će proslijeđivati sav promet.

Listing 17: Prikaz konfiguracijske datoteke `dynamic.conf`.

```
server {
    listen 80;
    server_name www.domain.com;

    access_log /var/log/nginx/domain-access.log;
    error_log /var/log/nginx/domain-error.log;

    location / {
        proxy_pass http://localhost:8888;
    }
}
```

Prilikom proslijeđivanja prometa `nginx` podržava dodavanje raznih `HTTP` zaglavlja na originalni zatjev kako bi aplikacija mogla biti znati da postoji još jedan poslužitelj između nje i klijenta. Dodatne informacije o svim mogućim direktivama dostupne su na `https://nginx.org/en/docs/`.

Literatura

[1] Više autora, *man stranice*, March 2020.

[2] Više autora, "Dokumentacija mrežnog podsustava linux jezgre." `https://www.kernel.org/doc/html/latest/networking/index.html`, March 2020.

[3] Više autora, "Dokumentacija projekta netfilter." `https://www.netfilter.org/documentation/`, March 2020.

[4] O. Kirch and T. Dawson, *The Linux Network Administrator's Guide, Second Edition*. Berlin, Heidelberg: LDP, 2000.