

Osnove korištenja operacijskog sustava Linux

06. Vlasništvo i dozvole

Dominik Barbarić

Nositelj: doc. dr. sc. Stjepan Groš

Sveučilište u Zagrebu
Fakultet elektrotehnike i računarstva

09.12.2016

Sadržaj

- 1 Dozvole
- 2 Posebne dozvole
- 3 Zadani mode
- 4 Promjena vlasnika

Dozvole (1)

- ▶ Naredba `ls -l` ispisuje informacije o vlasnicima i dozvolama objekta

```
$ ls -l datoteka.txt
```

```
-rw-r--r-- 1 pero users 0 Jan 4 23:19 datoteka.txt
```

- ▶ Objekt je vlasništvo korisnika i grupe
 - Drugo polje označava vlasnika - korisnika (pero)
 - Treće polje označava vlasnika - grupu (users)
- ▶ Prvo polje u prvom bitu sadrži oznaku tipa datoteke, a ostalih 9 bitova se nazivaju **mode** objekta

Dozvole (2)

- ▶ **mode** definira dozvoljene operacije na svakom objektu
- ▶ Devet bitova dijele se u tri grupe od koji svaka čini jedan troznamenasti binarni broj
- ▶ Svaki troznamenasti binarni broj se može prikazati jednom oktalnom znamenkom
- ▶ Svaka oktalna znamenka modea predstavlja skup dozvola koje su dodijeljene sljedećim korisnicima objekta i to:
 - Prva oktalna znamenka definira prava za vlasnika - korisnika
 - Druga oktalna znamenka definira prava za vlasnika - grupu
 - Treća oktalna znamenka definira prava za sve ostale

user
group
others

Dozvole (3)

- ▶ Značenja pojedinih bitova svake znamenke
 - r **read** - Dozvoljeno čitanje
 - w **write** - Dozvoljeno pisanje
 - x **execute** - Dozvoljeno izvršavanje / pretraživanje direktorija
- ▶ Svaki pojedini bit može biti u stanju
 - **uključen** - operacija dozvoljena
 - **isključen** - operacija zabranjena

▶ Primjer 1

$$rwxr-xr-x = 111101101_2 = 755_8$$

	r	w	x
user	+	+	+
group	+	-	+
others	+	-	+

Dozvole (4)

► Primjer 2

`rw-r--r-- = 644`

	r	w	x
user	+	+	-
group	+	-	-
others	+	-	-

► Primjer 3

`r--r--r-- = 444`

	r	w	x
user	+	-	-
group	+	-	-
others	+	-	-

Promjena dozvola (1)

- ▶ Promjena modea obavlja se naredbom `chmod`

`chmod <mode> <objekt>`

- ▶ Mode se može zadati oktalno i simbolički
- ▶ Moguće je rekurzivno mijenjati prava

`chmod -R <mode> <objekt>`

- ▶ Vlasnik datoteke može bez obzira na trenutni mod
 - promijeniti mode
 - obrisati datoteku

Promjena dozvola (2)

► Primjer 4

```
chmod ugo=rwx file1
```

	r	w	x
user	+	+	+
group	+	+	+
others	+	+	+

- Alternativno:

```
chmod a=rwx file1
```

```
chmod 777 file1
```

► Primjer 5

```
chmod u=rwx,go=rx file1 file2
```

```
ili chmod 755 file1 file2
```


Promjena dozvola (3)

► Primjer 6

```
chmod g+w file1 file2 file3
```

	r	w	x
user	*	*	*
group	*	+	*
others	*	*	*

► Primjer 7

```
chmod -x file1
```

```
ili chmod a-x file1
```

	r	w	x
user	*	*	-
group	*	*	-
others	*	*	-

Izvršavanje datoteka

- ▶ Svaka datoteka na UNIX sustavu može biti izvršna (*executable*)
- ▶ Skripta se, tako, može izvršiti korištenjem zadanog interpretera
- ▶ Postavljanjem `x` dozvole svaka se datoteka može izvršiti izravnim pozivanjem

```
/home/linux/skripta.sh
```

```
mode 755
```

```
#!/bin/bash  
echo "Skripta je pokrenuta"
```

```
~$ /home/linux/skripta.sh  
Skripta je pokrenuta  
~$ ./skripta.sh  
Skripta je pokrenuta
```

Promjena dozvola (4)

- Naredba `chmod` može prihvatiti poseban argument prilikom simboličkog zadavanja modea

X (Capital X)

- Direktorijima postavlja `x` dozvolu
- Ostalim datotekama ne mijenja mod
- Omogućuje listanje direktorija bez dodavanja dozvole za izvršavanje datoteka
- Koristan prilikom rekurzivne promjene modea:

```
chmod -R a+X dir1
```

Posebne dozvole (1)

Sticky bit

Sticky bit / Text mode

- ▶ **Kod direktorija**

Dozvoljava brisanje direktorija **samo** vlasniku i root korisniku

- ▶ **Kod datoteka**

Nakon izvršavanja datoteke proces ostaje u memoriji

- ▶ Simbolički se označava s velikim T na mjestu x dozvole za *others* korisnike

```
-rwxr--r-T 1 pero users 0 Jan 4 23:21 datoteka.txt
```

- ▶ Ako *others* ujedno ima i x dozvolu tada se sticky bit označava s malim t

Posebne dozvole (2)

SUID i SGID

- ▶ Za razumijeti preostala dva posebna bita potrebno je shvatiti što se događa s dozvolama korisnika koji pokreće izvršnu datoteku
- ▶ Svaki proces se pokreće s UID i GID primarne grupe korisnika koji ga je pozvao. Pokrenuti proces ima sve ovlasti tog korisnika

Set user ID (SUID) i Set group ID (SGID)

- ▶ Postavljanjem ovih bitova u mode datoteke proces koji pokreće datoteku dobiva dozvole **vlasnika - korisnika** (SUID bit), odnosno **vlasnika - grupe** (SGID) izvršne datoteke

Posebne dozvole (3)

SUID i SGID

- ▶ Simbolički se označava s velikim S na mjestu x dozvole za određenu grupu korisnika

```
-rwSr--r-x 1 pero users 0 Jan 4 23:21 datoteka.txt SUID
```

```
-rw-r-Sr-x 1 pero users 0 Jan 4 23:21 datoteka.txt SGID
```

- ▶ Primijetite da SUID, odnosno SGID ne impliciraju x dozvolu vlasnicima datoteke. U gornjem primjeru samo *others* imaju pravo izvršiti datoteku i u tom trenutku će isti dobiti prava vlasnika.
- ▶ Ako vlasnik, *user* ili *group* ujedno ima i x dozvolu tada se posebni bitovi označavaju s malim s

Posebne dozvole (4)

Promjena dozvola

- ▶ Posebne dozvole se također mijenjaju naredbom `chmod`
- ▶ Ispred uobičajene tri znamenke dodaje se još jedna čiji bitovi odgovaraju posebnim dozvolama
 - **Prvi bit** - SUID
 - **Drugi bit** - SGID
 - **Treći bit** - Sticky bit

▶ Primjer 8

```
$ chmod 5754 file1
$ ls -l file1
-rwsr-xr-T 1 pero users 0 Jan 4 23:23 file1
```

- Alternativno:

```
$ chmod u=rwxs,g=rx,o=rt file1
```

Zadani mode (1)

- ▶ Kreiranjem novog objekta on poprima zadani mode
- ▶ Definira ga trenutni filesystem i procesi koji kreiraju objekt
- ▶ Primjenom **umask** mogu se ograničiti dozvole koje postavljaju nadređeni procesi
- ▶ umask ima isti format kao i mode, no s različitim značenjem bitova
 - **1** - Isključuje dozvolu na poziciji bita
 - **0** - Ne mijenja dozvolu koju je postavio nadležni proces

Zadani mode (2)

- ▶ Naredbom `umask` se mijenja trenutni `umask`
 - **Bez argumenata** - ispisuje trenutnu vrijednost u oktalnom obliku
 - **Argument -S** - ispisuje trenutnu vrijednost u simboličkom obliku
 - **Argument 4 oktalne znamenke** - mijenja vrijednost umaska
Prva oktalna znamenka je za specijalne modove
- ▶ U datoteci s popisom montiranih datotečnih sustava, `/etc/fstab` se mogu navesti tri vrste maski
 - **umask** - Odnosi se na sve vrste datoteka
 - **fmask** - Odnosi se na sve regularne datoteke
 - **dmask** - Odnosi se na sve direktorije
- ▶ Ove vrste maski se mogu navesti i prilikom korištenja naredbe `mount`

Promjena vlasnika

- Promjena vlasnika objekta obavlja se naredbom `chown`

```
chown <korisnik> <objekt>
```

- Promjena grupe objekta obavlja se naredbom `chgrp`

```
chgrp <grupa> <objekt>
```

```
ili chown :<grupa> <objekt>
```

- Moguće je istovremeno promijeniti korisnika i grupu

```
$ chown <korisnik>:<grupa> <objekt>
```

```
$ chown <korisnik>: <objekt>
```

Postavlja korisnika i grupu koja odgovara primarnoj grupi korisnika