Kontrola pristupa na nivou fajl sistema

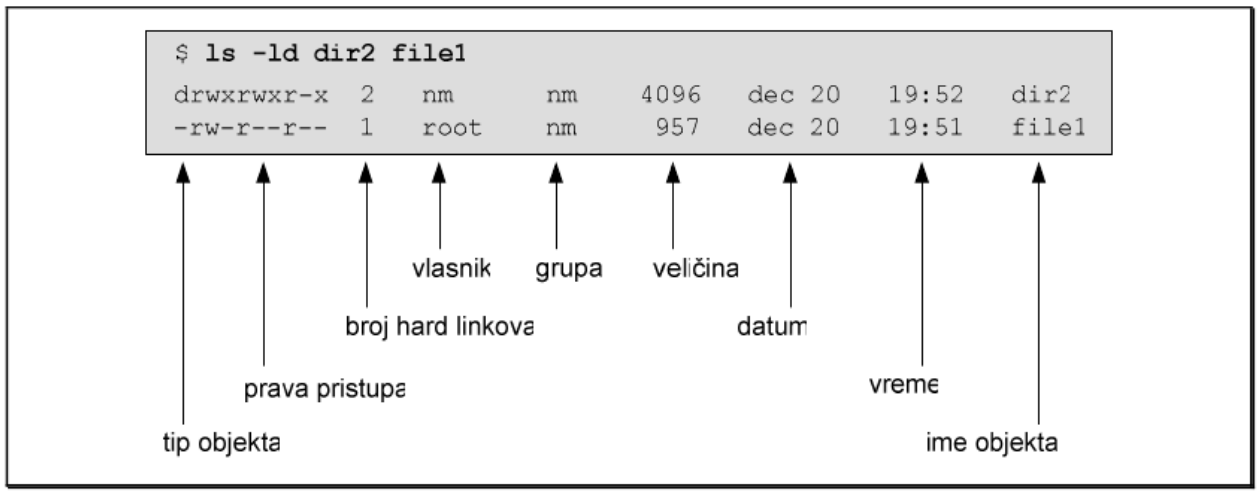
Pristup resursima pod mrežnim operativnim sistemima (kao što je i Linux) je strogo kontrolisan. Fajl sistem je fundamentalni resurs svake radne stanice ili servera, a kontrola pristupa fajlovima i direktorijumima (dodela ovlašćenja za pristup i zaštita od neovlašćenog pristupa) ključna komponenta ozbiljnih zaštitnih polisa u svakom višekorisničkom sistemu.

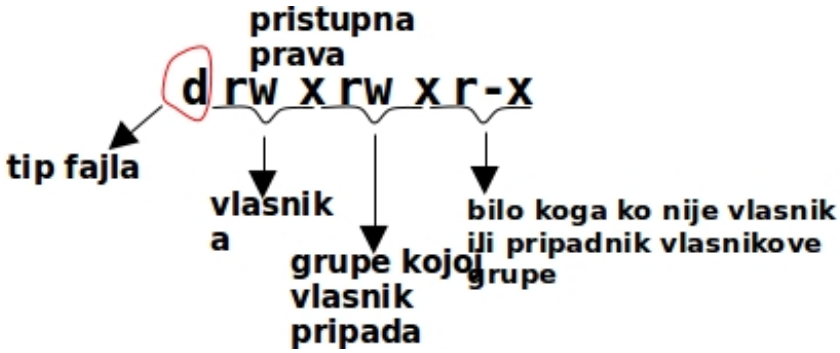
Vlasnički odnosi i prava pristupa

Jedna od najznačajnijih komponenti svake ozbiljne zaštitne politike je kontrola pristupa na nivou fajl sistema. Kontrolom pristupa na nivou fajl sistema određuju se:

- **skup korisnika** koji mogu pristupiti objektima, odnosno fajlovima i direktorijumima,
- nivo pristupa, odnosno **skup akcija** koje autorizovani korisnici mogu izvršiti nad tim objektima.

Kontrola pristupa na nivou fajl sistema zasniva se na vlasničkim odnosima, odnosno vlasništvu nad objektima (pripadnost objekta korisnicima i grupama) i pravima pristupa. Prava pristupa se dodeljuju svakom fajlu i direktorijumu. Prava pristupa za datoteke i direktorijume najlakše se mogu odrediti pomću komande **ls (list)** sa parametrom **-l (long)**, kao što je prikazano u sledećem primeru:





1. Tip fajla - prvi karakter ukazuje na tip datoteke:

- **- (dash)** - je reč o običnoj, regularnoj datoteci
- **d** - reč je o direktorijumu
- **b** - blok uređaj - block special file (npr. /dev/sda)
- **c** - karakter uređaj - character special file (npr. /dev/tty1)
- **l** - simbolički link
- **p** - imenovani pipe
- **s** - socket.

    Prava pristupa - sledećih devet znakova predstavljaju prava pristupa objektu za tri vlasničke kategorije, a to su vlasnik, grupa i ostali. Prva tri karaktera definišu prava pristupa vlasnika, druga tri prava pristupa grupe kojoj fajl pripada i poslednja tri karaktera prava pristupa za ostale:
- **Vlasnik (owner)** najčešće je korisnik koji je kreirao objekat, osim ukoliko superuser (root) ne promeni vlasništvo. U tom slučaju, vlasnik je korisnik kome je vlasništvo dodeljeno. Vlasnik objekta može biti bilo koji korisnik sistema, regularan ili sistemski.
- **Grupa (group)** je korisnička grupa kojoj je fajl formalno priključen. Za razliku od korisnika koji mogu pripadati većem broju grupa, **objekti fajl sistema mogu pripadati samo jednoj grupi**, koja može biti regularna ili sistemska. Najčešće je to primarna grupa korisnika koji je objekat kreirao. Superuser naknadno može promeniti pripadnost objekta grupi.
- **Ostali (others, public)** su svi korisnici koji nisu ni vlasnik objekta, niti pripadaju grupi kojoj objekat pripada. Prava pristupa za svaku vlasničku kategoriju eksplicitno se dodeljuju svakom objektu prilikom kreiranja, a kasnije se mogu promeniti.

Pravo pristupa za svaku grupu se zadaje na isti način, sa istim rasporedom karaktera **rwx**:

- **pravo čitanja (r - read)**,
- **pravo upisa (w - write)**,
- **pravo izvršavanja (x - execute)**.

Ukoliko se na odgovarajućoj poziciji nalazi **crtica -, pravo je ukinuto**.

Primer:

```
$ ls -la ~/a.txt
-rw-rw-r-- 1 marko marko 161 окт  3 23:43 /home/marko/a.txt

$ ls -ld /bin /root
drwxr-xr-x 2 root root 4096 нов 19 03:41 /bin
drwx------ 4 root root 4096 сеп 26 22:51 /root
```

Na primer, sistemski direktorijum **/bin** sadrži najčešće korišćene UNIX komande. Svim korisnicima sistema dato je pravo korišćenja direktorijuma /bin. Svi korisnici sistema mogu da se pozicioniraju na direktorijum, mogu da pročitaju sadržaj i pokrenu komande koje se u njemu nalaze. Pravo upisa dato je jedino superuser-u.

Drugo, sistemski direktorijum /root je home direktorijum superusera, koji nad njim ima sva prava, dok je svim ostalim korisnicima pristup direktorijumu zabranjen.

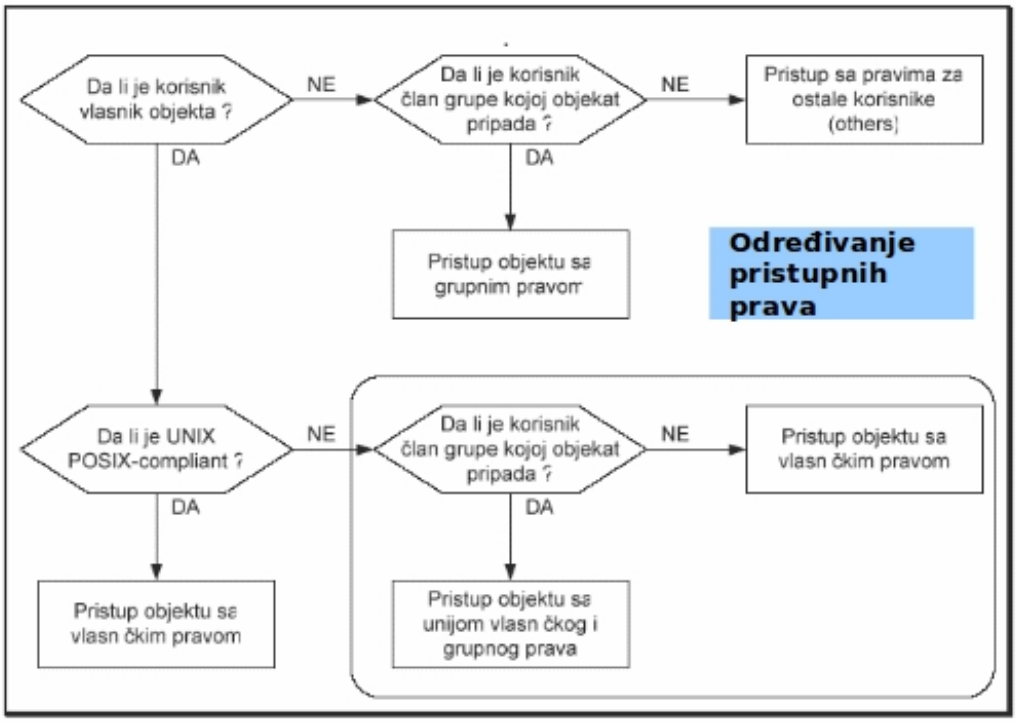
Značenje prava za fajlove i direktorijume bitno se razlikuje, što je prikazano u sledećoj tabeli:

	pristupna prava za fajlove	**pristupna prava za direktorijume**
read (r)	Korisnik može pročitati sadržaj fajla, odnosno može prikazivati fajl na ekranu, štampati ga ili kopirati;	Korisnik može pročitati sadržaj direktorijuma, što znači i da korisnik može da izvrši komandu ls. **Napomena**: za prikazivanje detaljnog listinga direktorijuma (ls -l) neophodno je i x pravo nad direktorijumom
write (w)	Korisnik može modifikovati sadržaj fajla. **Napomena**: može obrisati fajl samo ako mu je dato pravo upisa nad roditeljskim direktorijumom;	Korisnik može modifikovati sadržaj direktorijuma, odnosno dodavati nove fajlove i brisati postojeće, kreirati i brisati poddirektorijume. **Napomena**: može obrisati direktorijum samo ako mu je dato pravo upisa nad roditeljskim direktorijumom;
execute (x)	Korisnik može izvršavati fajl, pod uslovom da se radi o shell programu ili o fajlu u binarnom izvršnom formatu;	Korisnik se može pozicionirati na direktorijum (komandom cd), može prikazivati dugački listing (ls -l) sadržaja i pretraživati direktorijum (find).

Svim fajlovima i direktorijumima dodeljen je korisnički identifikator (UID) i grupni identifikator (GID) vlasnika. Kernel razrešava vlasničke odnose na osnovu ovih identifikatora.

```
$ ls -ln
-rw-rw-r--    1   859   861     20    dec 23    14:04   kyuss
-rw-rw-r--    1   859   861     20    dec 23    15:20   stoner
$ id
uid=859(nm) gid=861(nm) groups=861(nm),0(root)
```

Napomena: Opcija **-n** komande **ls** daje numeričke vrednosti za UID i GID.



Promena pristupnih prava

Prava pristupa mogu promeniti isključivo vlasnici fajlova i direktorijuma, dok root kao superuser može da promeni pristupna prava svakom objektu. Komanda **chmod** može se pokrenuti u **simboličkom (relative)** ili **oktalnom (absolute)** režimu.

Simbolički režim

Korisnik dodeljuje ili oduzima prava u odnosu na postojeća, dok se postojeća prava koja nisu specificirana argumentom komande ne menjaju. Format komande u simboličkom modu je:

```
$ chmod [-R] symbolic_mode[,...] objectname
```

Primer:

```
$ chmod u=rwx myscript
```

symbolic_mode sastoji se od tri komponente:

- **vlasnička kategorija** na koju se komanda odnosi: vlasnik (u), grupa (g), others (o), sve kategorije (a);

- **operator**: dodela prava (+), ukidanje prava (-), dodela tačno određenih prava (=);
- **prava pristupa** koja se dodeljuju ili oduzimaju: r, w i/ili x.

Primeri:

```
$ touch myfile
$ ls -l myfile
-rw-rw-r-- 1 marko marko 0 нов 19 18:28 myfile
```

kreira fajl i lista prava

```
$ chmod go+w myfile
$ ls -l myfile
-rw-rw-rw- 1 marko marko 0 нов 19 18:28 myfile
```

dodata prava upisa kategorijama group i other

```
$ chmod u-w myfile
$ ls -l myfile
-r--rw-rw- 1 marko marko 0 нов 19 18:28 myfile
```

oduzeto pravo upisa vlasniku

```
$ chmod u=rw,go-w myfile
$ ls -l myfile
-rw-r--r-- 1 marko marko 0 нов 19 18:28 myfile
```

dodeljen skup prava rw vlasniku i ukinuto pravo upisa kategorijama group i others

```
$ chmod a= myfile
$ ls -l myfile
---------- 1 marko marko 0 нов 19 18:28 myfile
```

svima ukinuta sva prava.

Parametar -R se koristi za rekurzivnu promenu pristupnih prava direktorijuma i svih objekata (poddirektorijuma i fajlova) koji se u njemu nalaze. Ukoliko se navede parametar -R, argument objectname mora biti direktorijum.

Primer:

```
$ ls -ld parent_dir
drwxr-xr-x      2    nm     nm     4096      Apr 28    09:10     parent_dir

$ ls -l parent_dir
parent_dir:
total 0
-rw-r--r--      1    nm     nm       0    Apr  28    09:09     dir1
-rw-r--r--      1    nm     nm       0    Apr  28    09:10     dir2
-rw-r--r--      1    nm     nm       0    Apr  28    09:09     file1
-rw-r--r--      1    nm     nm       0    Apr  28    09:09     file2
```

```
$ chmod -R o-rx parent_dir
$ ls -ld parent_dir
drwxr-xr-x       2    nm      nm     4096      Apr 28    09:10     parent_dir

$ ls -l parent_dir
parent_dir:
total 0
-rw-r--r-- 1 nm  nm       0     Apr  28     09:09 dir1
-rw-r--r-- 1 nm  nm       0     Apr  28     09:10 dir2
-rw-r--r-- 1 nm  nm       0     Apr  28     09:09 file1
-rw-r--r-- 1 nm  nm       0     Apr  28     09:09 file2
```

Oktalni režim

Komandom chmod u oktalnom režimu dodeljuju se prava pristupa svim vlasničkim kategorijama istovremeno. Prava koja korisnik navede kao argument komande eksplicitno zamenjuju postojeća prava (prethodna prava se ne prolongiraju), tako da se ovaj režim naziva apsolutnim. Komanda zatheva da se u ovom režimu kao argument navedu tri oktalne cifre od kojih svaka predstavlja prava pristupa za jednu vlasničku kategoriju.

Moguće oktalne vrednosti sa odgovarajućim pravima opisane su sledećom tabelom:

Oktalna vrednost	Suma prava po binarnoj vrednosti	Odgovarajuća prava	Definicaj
7	4 + 2 + 1	r w x	čitanje, izmena i izvršavanje
6	4 + 2 + 0	r w -	čitanje i izmena
5	4 + 0 + 1	r - x	čitanje i izvršavanje
4	4 + 0 + 0	r - -	čitanje
3	0 + 2 + 1	- w x	izmena i izvršavanje
2	0 + 2 + 0	- w -	izmena

Oktalna vrednost	Suma prava po binarnoj vrednosti	Odgovarajuća prava	Definicaj
1	0 + 0 + 1	- - x	izvršavanje
0	0 + 0 + 0	- - -	bez prava pristupa

Sintaksa komande chmod u oktalnom režimu je slična sintaksi komande u simboličkom režimu:

```
$ chmod [-R] absolute_mode objectname
```

Apsolutna prava formiraju se pomoću tri oktalne cifre kojima su predstavljena prava pristupa za vlasnika, grupu i ostatak sveta. Parametar -R se, kao i u simboličkom režimu, koristi za rekurzivnu promenu pristupnih prava direktorijuma i svih objekata koji se u njemu nalaze. U tom slučaju argument objectname mora biti direktorijum.

Napomena: Kada se koristi oktalni režim **moraju se navesti sve tri oktalne cifre** u tačnom redosledu (vlasničko pravo, grupno pravo, pravo za ostatak sveta).

```
$ ls -l betatest
-rw-rw-rw-   1   nm    nm 0 dec 23 15:25 betatest

$ chmod 555 betatest
$ ls -l betatest
-r-xr-xr-x   1   nm    nm 0 dec 23 15:25 betatest

$ ls -l denywrites
-rwxrwxrwx   1   nm    nm 0 dec 23 15:25 denywrites

$ chmod 755 denywrites
$ ls -l denywrites
-rwxr-xr-x   1   nm    nm 0 dec 23 15:25 denywrites
```

Promena vlasničkih odnosa

UNIX postavlja inicijalne vlasničke odnose prilikom kreiranja fajla ili direktorijuma. Korisnik koji kreira objekat postaje njegov vlasnik, a objekat se formalno pridružuje primarnoj grupi vlasnika.

Promena vlasnika

Komandom chown (change owner) root kao superuser može da promeni vlasnika objekta, a ukoliko konkretan sistem to dozvoljava, to može učiniti i vlasnik. Regularni korisnici Linux sistema mogu promeniti vlasničke odnose samo ako na sistemu nije aktiviran mehanizam disk kvote (disk quota), kojim se korisnicima ograničava iskorišćenje

prostora na diskovima. Kada se za fajl promeni vlasništvo, prava pristupa starog vlasnika određena su kategorijama group i others. Sledeće komande prikazuju sintaksu za promenu vlasništva:

```
$ chown [-R] new_owner objectname
```

Primer:

```
$ whoami
nm
$ ls -l myfile
-rw-r--r--      1    nm          nm     0    Apr 28       12:07     myfile
$ chown jsmith myfile
chown: changing ownership of `myfile`: Operation not permitted
$ su
Password: ******
# chown jsmith myfile
# exit
exit
$ ls -l myfile
-rw-r--r--      1    jsmith      nm     0    Apr 28       12:07     myfile
```

Postavljanje user ID, postavljanje group ID, sticky bit

Pored osnovnih dozvola, postoje i tri bita informacija definisanih ya fajlove na UNIX sistemima:

- **SUID ili setuid: promena korisničkog ID pri izvršenju**. Može da se postavi samo na fajlove. Ako je SUID postavljen, kada korisnik izvršava fajl, proces će imati ista prava kao vlasnik fajla koji se izvršava. Tekstualna reprezentacija s.
- **SGID ili setgid: promena grupnog ID pri izvršenju**. Za fajlove isto kao i kod SUID, ali se nasleđuju prava grupe kojoj vlasnik fajla pripada. Za direktorijume može da znači da će novo kreirani fajl u tom direktorijumu naslediti grupu kojoj direktorijum pripada, a ne grupu kojoj pripada korisnik koji je kreirao fajl. Tekstualna reprezentacija s.
- **Sticky bit**. Ranije je služio da bi označio da proces ostane (da se “zalepi”) u memoriji nakon što se završi. Sada njegova primena varira od sistema do sistema i najčešće se koristi da bi se sprečilo brisanje fajlova koji pirpadaju drugom korsinku u direktorijumu u kojem korisnik ima prava pisanja. Primer je /temp direktorijum. Tekstualna reprezentacija t.

Oktalna vrednost	Binarna vrednost	Značenje
0	000	setuid, setgid, sticky bit nisu postavljeni
1	001	sticky bit postavljen

Oktalna vrednost	Binarna vrednost	Značenje
2	010	setgid postavljen
3	011	setgid i sticky bit postavljeni
4	100	setuid postavljen
5	101	setuid i sticky bit postavljeni
6	110	setuid i setgid postavljeni
7	111	setuid, setgid, sticky bit i postavljeni

Postavljanje i uklanjanje SUID bita

```
$ ls -l
total 8
-rwxr--r-- 1 root root 104 Aug 19 01:26 hello.sh
$ chmod u+s hello.sh
$ ls -l
total 8
-rwsr--r-- 1 root root 104 Aug 19 01:26 hello.sh

$ ls -l
total 8
-rwxr--r-- 1 root root 104 Aug 19 01:26 hello.sh
$ chmod 4744 hello.sh
$ ls -l
total 8
-rwsr--r-- 1 root root 104 Aug 19 01:26 hello.sh
```

Postavljanje i uklanjanje SGID bita

```
$ ls -ld /javaproject
drwxrwxr-x 2 root javaproject 4096 Aug 19 02:33 /javaproject
$ chmod g+s /javaproject
$ ls -ld /javaproject
drwxrwsr-x 2 root javaproject 4096 Aug 19 02:33 /javaproject

[jones@redhat-server ~]$ touch /javaproject/jones1.txt
[jones@redhat-server ~]$ mkdir /javaproject/jones1dir
[jones@redhat-server ~]$ ls -l /javaproject/
total 12
drwxrwsr-x 2 jones javaproject 4096 Aug 19 02:38 jones1dir
-rw-rw-r-- 1 jones javaproject 0 Aug 19 02:37 jones1.txt
```

```
$ ls -ld /shared/
drwxrwxr-x 2 root adm 4096 Aug 19 02:47 /shared/
$ chmod 2775 /shared/
$ ls -ld /shared/
drwxrwsr-x 2 root adm 4096 Aug 19 02:47 /shared/
```

Sticky bit

```
$ ls -ld /tmp/
drwxrwxrwt 4 root root 4096 Aug 19 02:29 /tmp/

$ chmod o-t dir1
$ ls -l
total 8
drwxr-xr-x 2 root root 4096 Aug 19 03:08 dir1
$ chmod o+t dir1
$ ls -l
total 8
drwxr-xr-t 2 root root 4096 Aug 19 03:08 dir1

$ chmod 1777 dir1/
$ ls -l
total 8
drwxrwxrwt 2 root root 4096 Aug 19 03:08 dir1
```