SZÉCHENYI ISTVÁN EGYETEM
GYŐR
TÁVKÖZLÉSI TANSZÉK

Mérési Utasítás

Linux/UNIX jogosultságok, szövegfájlok létrehozása

# Linux fájlrendszerek.

Linux alatt, az egyes fájlokhoz való hozzáférések korlátozása érdekében, a fájlokhoz tulajdonost, csoportot és a hozzájuk tartozó jogosultságokat rendelünk.

Az ls -l parancs használatakor, a kapott lista a következő információkat írja ki a fájlokról:

1. oszlop fájlok típusa, jogosultságok
2. oszlop ún. inode szám
3. oszlop fájl tulajdonosa
4. oszlop fájl csoportja
5. oszlop fájl mérete
6, 7. oszlop utolsó módosítás ideje
8. oszlop fájl neve

A fájlokról ezeket az adatokat, a fájlrendszer tárolja. Számunkra az 1, 3, 4. oszlopok lesznek az érdekesek.

# chmod, chown, chgrp

A jogosultságok a fájl tulajdonosára, csoportjára, és mindenki másra vonatkoznak:

| fájltípus | Jogok | | |
|---|---|---|---|
| - | Tulajdonos | Csoport | Mindenki más |
| | rwx | r-x | r-x |
| | 111 | 101 | 101 |
| | 7 | 5 | 5 |
| | Olvasás/írás/futtatás | Olvasás/~~írás~~/futtatás | Olvasás/~~írás~~/futtatás |

Fájltípusok például a következőek lehetnek:

- -: reguláris fájl (teljesen egyszerű bináris vagy szöveges állomány)
- d: könyvtár típusú állomány
- c: karakteres típusú eszközfájl (konzol is ilyen például: /dev/tty)
- b: blokk típusú eszközfájl (winchesterek: /dev/hda)
- l: link típusú fájl

Jogok fájlok esetén:

- r: olvasási jog
- w: írási jog
- x: futtatási jog

Jogok könyvtárak esetén:

- r: olvasási
- w: írási
- x: könyvtár-hozzáférési

A fájlok jogait a legegyszerűbb 3 oktális szám segítségével megadni, mint az alábbi példákon is fogjuk látni:

*chmod 600 /tmp/saját_fájlom.txt*
*chmod 640 /tmp/csoportolvashatja*
*chmod 660 /tmp/csoportirhatjais*

Lehetőségünk van könyvtárakban, alkönyvtárakban lévő fájlok jogainak rekurzív módosítására:

*chmod -R 644 /tmp/probakonyvtar*

Ezzel ugyanakkor óvatosan kell bánnunk, hiszen ha rekurzívan változtatunk meg egy olyan könyvtárat melyben alkönyvtár van, akkor az alkönyvtáron beállított jogosultságokat is módosítjuk! A tulajdonost, és csoportot is van lehetőségünk módosítani. Tulajdonost a chown paranccsal, ezt csak a root tudja, míg csoportot, - a chgrp paranccsal - a felhasználó is módosíthat, ha tagja annak a csoportnak melyhez hozza szeretné rendelni a fájlt. Lehetőségünk van egyszerre megváltoztatni a tulajdonost és a csoportot - megfelelő jogosultságok esetén - a chown parancs segítségével.

## *Feladat*

1. Lépjen be root felhasználóként az első terminálon, majd diak felhasználóként a második terminálon.

2. Lépjen be mindkét terminálon a /tmp könyvtárba!

(cd /tmp)

3. Hozza létre root-ként a /tmp/proba.txt fájlt!

(touch /tmp/proba.txt)

4. Rendelje hozzá root felhasználóként, a létrehozott /tmp/proba.txt fájlt a diak felhasználóhoz!

(chown diak /tmp/proba.txt)

5. Rendelje hozzá root felhasználóként, a /tmp/proba.txt fájlt, a diak csoporthoz!

(chgrp diak /tmp/proba.txt)

6. Hozza létre root felhasználóként, a /tmp/proba2.txt fájlt!

(touch /tmp/proba2.txt)

7. Rendelje hozzá root felhasználóként, a létrehozott /tmp/proba2.txt fájlt a diak felhasználóhoz és csoporthoz egy paranccsal!

(chown diak.diak /tmp/proba2.txt)

8. Hozzon létre egy proba3.txt nevű fájlt root felhasználóként, a /tmp könyvtárban!

(touch /tmp/proba3.txt)

9. A /tmp/proba3.txt jogait állítsa be úgy, hogy a tulajdonos és a csoport tudja írni olvasni, a többi felhasználó csak olvasni tudja!

(chmod 664 /tmp/proba3.txt)

10. Hozza létre a /tmp/proba/elso könyvtárstruktúrát root felhasználóként!

(mkdir -p /tmp/proba/elso)

11. A /tmp/proba könyvtár jogait állítsa be úgy, hogy a tulajdonos tudja írni, olvasni, böngészni, a csoport tudja böngészni és olvasni, a többi felhasználó csak böngészni tudja!

(chmod 751 /tmp/proba)

12. Váltson át a második terminálra -ahol diak felhasználóként jelentkezett be -, majd lépjen be a /tmp/proba könyvtárba!

(cd /tmp/proba)

13. Szintén diak felhasználóként, listázza ki a könyvtár tartalmát! Mit tapasztal?

(ls)

Ha mindent jól csinált, akkor a rendszer hozzáférés megtagadva hibaüzenettel válaszol.

14. Váltson vissza az 1. terminálra, majd root felhasználóként hozzon létre egy fájlt, file néven, a /tmp/proba könyvtárban „nem ures” tartalommal!

```
(echo 'nem ures' > /tmp/proba/file)
```

15. Most próbálja meg cat paranccsal megnézni diak felhasználóként a /tmp/proba/file tartalmát! Mit tapasztal?

```
(cat /tmp/proba/file)
```

16. Törölje le root felhasználóval a /tmp könyvtárból, a proba kezdetű fájlokat és könyvtárakat egy paranccsal!

```
(rm -rf /tmp/proba*)
```

A könyvtáron, az olvasási jog hiánya, nem befolyásolja, a könyvtárban lévő fájlokhoz való hozzáférést.

**echo, >, >>**

Az echo parancs kiírja, a mögötte álló kifejezést amennyiben nem értelmezhető kapcsoló vagy idézőjelek „”, aposztrófok „'” között áll, majd sortörést hajt végre. Lehetséges kapcsolója a -n, mellyel nem tesz sortörést a kiírás után. Ha nem adunk meg paraméternek se kapcsolót se szöveget akkor egy sortörést ír ki.

A > segítségével egy program kimenetét egy fájlba tehetjük (ha nem létezett a fájl létrejön, 2x kiadva egymás után a parancsot az első fájl felülíródik). A >> segítségével egy program kimenetét egy fájlhoz hozzáfűzhetjük (ha nem létezett a fájl létrejön, többször kiadva egymás után a parancsot az utoljára kiadott parancs kimenet a fájlhoz fűződik, amennyiben az létezett).

***Feladat***

1. Váltson könyvtárat, lépjen be a /tmp könyvtárba root felhasználóval! `(cd /tmp)`

2. Írassa ki, 'Hello world!' szöveget echo segítségével! `(echo 'hello world')`

3. Írassa elso.txt nevű fájlba, a „Hello world!” szöveget echo segítségével!

```
(echo 'Hello world!' > elso.txt)
```

4. Írassa masodik.txt nevű fájlba, a „Hello” szöveget echo segítségével!

```
(echo 'Hello' > masodik.txt)
```

5. Írassa /tmp/masodik.txt nevű fájlba, a „ world!” szöveget echo segítségével!

```
(echo ' world!' > masodik.txt)
```

6. Írassa harmadik.txt nevű fájlba, a „Hello” szöveget echo segítségével, de ne legyen sortörés a

kiírás után!

```
(echo -n 'Hello' > harmadik.txt)
```

7. Írassa /tmp/harmadik.txt nevű fájlba, a „ world!” szöveget echo segítségével!

```
(echo ' world!' >> harmadik.txt)
```

**cat, tac, <, <<**

A cat parancs a paraméterként megadott állományt, vagy a bemenetére kapott adatokat kiíratja. A tac mint a nevéből is látszik a cat-hez hasonlóan működik, csak fordítva íratja ki a bemenetet. A cat-nak nem csak fájl nevet adhatunk meg, hanem a bemenetére irányíthatunk fájlokat, melyeket kilistáz mintha paraméterként kapta volna. Pl.: cat < proba.txt. Amennyiben több sornyi szöveget szeretnénk kiíratni vagy fájlbaírni, a „<<” és a cat parancsot kell használnunk, a következő módon:

cat << EOF
asédjféask
adsfjaéfdkj
asdfjaésdklfjaés
asdéfkjaésdkj
EOF

Amíg EOF nem fog állni egy külön sorban önállóan, addig írhatunk (törölni vissza sorokat nem lehet!), melyet az cat fog majd megkapni paraméterként. Ha fájlba szeretnénk tárolni a

végeredményt, a következő parancsot kell kiadnunk:

cat > /tmp/file.txt << EOF

adsfasédjlf

asdfasdéfjha

sdafasjdfk

asdfasdkjfhalaa

sdjkfhalsj

EOF

Ez a file.txt be teszi amit begépeltünk.

SZÉCHENYI ISTVÁN EGYETEM
GYŐR
TÁVKÖZLÉSI TANSZÉK

## *Feladat*

1. Készítsen egy több soros (legalább 40 soros, ismétlődésekkel!) állományt, mely számokat tartalmaz több sorban! (pl.:

```
cat > /tmp/szamok.txt << EOF
123
12
43
4
54
876
...
34
564
12
546
34
EOF
```

2. Készítsen egy több soros (legalább 40 soros, ismétlődésekkel!) állományt, mely karakterláncokat tartalmaz több sorban! (pl.:

```
cat > /tmp/karakterek.txt << EOF
234hlk12h
heghjg123
khjjkhg
23hg4
23hg4
...
dofg98
6l5hkj
qw93e8r7
wksdfkasjh
dofg98
sdf8g79
EOF
```

SZÉCHENYI ISTVÁN EGYETEM
GYŐR
TÁVKÖZLÉSI TANSZÉK

# more, less, |

Mivel a terminál 25 karakter magas, a 40 soros állomány túlszalad a képernyőn. A more és less parancsokkal tudjuk 25 sornál hosszabb kimeneteket lapozni. A more csak előre (space segítségével egész oldalakat, enterrel sorokat léptet előre, kilépni q-val), a less előre és hátra is tud lapozni (működik a more-nál megszokott billentyűkombinációk, valamint a pageup, pagedown, fel, le nyilak, kilépni q-val).

A „|” az úgynevezett pipe/cső (<ALTgr>+W). Ennek segítségével egy program kimenetét egy másik programnak adhatjuk át bemenetként (pl.: less, more).

## *Feladat*

1. Írassa ki a képernyőre a /tmp/szamok.txt állományt!

```
(cat /tmp/szamok.txt)
```

2. Írassa ki fordítva a képernyőre a /tmp/karakterek.txt állományt!

```
(tac /tmp/karakterek.txt)
```

3. A more segítségével lapozza a /tmp/karakterek.txt állományt!

```
(cat /tmp/karakterek.txt | more )
```

4. A less segítségével lapozza a /tmp/szamok.txt állományt!

```
(cat /tmp/szamok.txt | less )
```