SZÉCHENYI ISTVÁN EGYETEM
GYŐR
TÁVKÖZLÉSI TANSZÉK

Mérési Utasítás Linux/Unix jogosultságok és fájlok kezelése

Linux fájlrendszerek és jogosultságok

Linux alatt, az egyes fájlokhoz való hozzáférések szabályozása 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 (1. karakter) és jogosultságok (további 9 karakter)
2. oszlop a fájlra való hivatkozások száma (ezzel most nem foglalkozunk)
3. oszlop fájl tulajdonosa
4. oszlop fájl csoportja
5. oszlop fájl mérete

6-8. oszlop utolsó módosítás ideje (év, hónap, nap)

9. oszlop fájl neve

A fájlokról ezeket az adatokat a fájlrendszer tárolja.

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 jelentése fájlok esetén:

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

Jogok jelentése könyvtárak esetén:

- r: a könyvtár tartalmát lehet listázni
- w: a könyvtár tartalma módosítható (bejegyzés létrehozása, törlés)
- x: könyvtár-hozzáférés (elérhetők a benne levő fájlok, könyvtárak, stb.)

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

chmod 600 /tmp/saját_fájlom
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 egy olyan könyvtárat, melyben alkönyvtár van, akkor az alkönyvtár jogait is módosítja! A tulajdonost, és csoportot is van lehetőségünk módosítani. Tulajdonost a chown paranccsal, ezt csak a root tud, míg csoportot, - a chgrp paranccsal - a felhasználó is, 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. A továbbiakban az első terminálon, root felhasználóként dolgozzon. Hozza létre a /tmp/proba1 fájlt!

```
(touch /tmp/proba1)
```

4. Állítsa be a /tmp/proba fájl tulajdonosának a diak felhasználót!

```
(chown diak /tmp/proba1)
```

5. Állítsa be a /tmp/proba fájl csoporttulajdonosának a diak csoportot!

```
(chgrp diak /tmp/proba1)
```

6. Hozza létre a /tmp/proba2 fájlt!

```
(touch /tmp/proba2)
```

7. Állítsa be /tmp/proba2 fájl tulajdonosának a diak felhasználót és csoporttulajdonosának a diak csoportot egyetlen paranccsal!

```
(chown diak.diak /tmp/proba2)
```
vagy:
```
(chown diak:diak /tmp/proba2)
```

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

```
(touch /tmp/proba3)
```

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

```
(chmod 664 /tmp/proba3)
```

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

```
(mkdir -p /tmp/proba/elso)
```

11. A /tmp/proba könyvtár jogait állítsa be úgy, hogy a tulajdonos tudja módosítani, listázni és a tartalmát elérni, a csoport tudja listázni és a tartalmát elérni, a többi felhasználó pedig csak a tartalmát elérni tudja!

```
(chmod 751 /tmp/proba)
```

12. Váltson át a második terminálra – ahol diak felhasználóként jelentkezett be –, majd lépjen be a /tmp/proba könyvtárba! A továbbiakban diak felhasználóként, itt dolgozzon.

```
(<ALT>+<F2>)(cd /tmp/proba)
```

13. Listázza ki a könyvtár tartalmát! Mit tapasztal?

```
(ls)
```

Ha mindent jól csinált, akkor a rendszer „Engedély 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!

(<ALT>+<F1>)(echo 'nem ures' > /tmp/proba/file)

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

(<ALT>+<F2>)(cat /tmp/proba/file)

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

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

(<ALT>+<F1>)(rm -rf /tmp/proba*)

A törlés természetesen sikeres.

echo, >, >>

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

A > segítségével egy program kimenetét egy fájlba tehetjük (ha nem létezett a fájl, akkor létrejön, 2x kiadva egymás után a parancsot az első fájl felülíródik). A >> segítségével egy program kimenetét egy fájlhoz hozzáfűzhetjük (ha nem létezett a fájl, akkor 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 a 'Hello world!' szöveget echo segítségével! (echo 'hello world')

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

(echo 'Hello world!' > elso)

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

(echo 'Hello' > masodik)

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

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

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

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

7. Fűzze hozzá a /tmp/harmadik nevű fájlhoz a „ world!” szöveget echo segítségével!

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

8. Írassa ki a képernyőre a /tmp/harmadik nevű fájl tartalmát a cat parancs segítségével!

```
(cat harmadik)
```

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. Amennyiben több sornyi szöveget szeretnénk kiíratni vagy fájlba írni, a „<<” és a cat parancsot kell használnunk, a következő módon:

cat << EOF
asédjféask
adsfjaéfdkj
asdfjaésdklfjaés
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 << EOF
adsfasédjlf
asdfasdéfjha
sdafasjdfk
sdjkfhalsj
EOF

Ez a file nevű fájlba 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 << 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 << EOF
234hlk12h
heghjg123
khjjkhg
23hg4
23hg4
...
dofg98
6l5hkj
qw93e8r7
wksdfkasjh
dofg98
EOF
```

more, less, |, sort

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 (működnek 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).

A sort parancs segítségével fájlok sorait rendezhetjük. Alapértelmezésben a rendezés lexikografikus (ABC szerinti) és a sorrend növekvő. Numerikus rendezést érhetünk el a -n kapcsolóval. Csökkenő sorrend a -r kapcsolóval állítható be .

Feladat

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

```
(cat /tmp/szamok)
```

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

```
(tac /tmp/karakterek)
```

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

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

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

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

5. A sort segítségével jelenítse meg növekvő sorrendben a /tmp/szamok állomány számait!

```
(cat /tmp/szamok | sort -n )
```

5. Mit tapasztal a -n kapcsoló használata nélkül?

```
(cat /tmp/szamok | sort )
```

(A számok sorrendje nem a matematikai számsorrend szerint, hanem pozíción belül lett növekvő.)

3. Jelenítse meg ABC szerint fordított sorrendben a /tmp/karakterek állomány sorait!

```
(cat /tmp/karakterek | sort -r )
```