

Előző gyakorlat - Fel - Következő gyakorlat

Tartalomjegyzék

- 1 Linux jogok
 - ◆ 1.1
chmod
 - ◆ 1.2
Bővebben
- 2 Bash
 - ◆ 2.1
Alapvető
parancsok
 - ◆ 2.2 Pipe,
redirect
- 3 Windows
 - ◆ 3.1
Különbségek

Linux jogok

- Jogosultsági körök
 - ◆ **superuser** (su) felhasználók, pl. rendszergazdák
 - ◆ **owner**, egy adott fájl tulajdonosa, aki létrehozta, pl. *kispista*
 - ◆ **group**, felhasználóknak egy részhalmaza, amelybe az *owner* is beletartozik, pl. *student*
 - ◆ **others**, mindenki más, nyilvános, guest vagy egyéb nem megbízható felhasználó
- Jogosultsági típusok
 - ◆ **w**: Write, írhatja, olvashatja a fájlt
 - ◆ **r**: Read, olvashatja a fájl vagy könyvtár tartalmát
 - ◆ **x**: eXecute, futtathatja a fájlt (programok, script-ek), vagy listázhatja a könyvtárat
 - ◆ **-**: semmi

Ezekből a lehetőségekből válogathatunk, vagyis megadhatjuk, hogy egy adott kör, mely jogokkal rendelkezzen. Például

- a tulajdonos írhatja, olvashatja, futtathatja
- a csoportja olvashatja, futtathatja
- mások semmit

A superuser-ek (su) mindig mindent csinálhatnak, azt nem tudjuk korlátozni, su jogot csak su adhat vagy vehet el.

Ezekre van karakteres és numerikus kód, a tömörebb olvashatóság kedvéért:

- **u**: user, owner
- **g**: group
- **o**: others
- **a**: mindenki: a fájl tulajdonosa, annak csoportja és mindenki más is.
- olvasás: **r** vagy **4**
- írás: **w** vagy **2**
- execute: **x** vagy **1**
- semmi: **-** vagy **0**

Informatika1-2015/Gyakorlat2

Az **a** nem ugyan az, mint az **o**, mert elképzelhet?, hogy a group-nak nem adok jogot valamire, de a csoporton kívüli embereknek igen.

Karakteresen a jogokat egy 10 karakterből álló sztring tartalmazza:

```
1   2           3   4   5   6   7   8   9   10
típus
read write      execute read write execute read write execute
$ ls -l ~
drwxr-xr-x 8 borbely student 4096 Aug 30 23:24 Desktop
drwxr-xr-x 2 borbely student 4096 Mar 27 2012 Downloads
drwxr-xr-x 2 borbely student 4096 Oct 20 2009 Drives
drwx----- 2 borbely student 4096 Apr 20 10:42 mail
drwxr-xr-x 7 borbely student 4096 Sep 6 13:01 public_html
$ _
```

A `public_html` mappának a tulajdonosa: `borbely`, csoportja `student`, azonosító: `drwxr-xr-x`, vagyis:

- típusa **d**: könyvtár
- saját jogaim **rw**x, vagyis minden (hál' Istennek)
- diáktársaim jogai **r-x**, vagyis írni nem írhatják, de minden mást szabad
- mindenki másnak **r-x**, vagyis más is olvashatja és futtathatja

Numerikusan egy 3 számból álló érték határozza meg a jogokat. Balról az első szám az owner jogok, aztán group majd others. A számok 4: olvasás, 2: írás, 1: execute. Ezek összeadódnak. Pl
`/home/student/borbely/public_html` jogai: 755 azaz

- owner (`borbely`) $7=4+2+1$: írás, olvasás és futtatás
- group (`student`) $5=4+1$: olvasás és futtatás
- other $5=4+1$: olvasás és futtatás

chmod

A jogokat a `chmod` paranccsal változtathatjuk meg. Néhány példa:

- `chmod 700 ~/info_hazi`: elrejttem az `info_hazi` mappámat rajtam kívül mindenki el? (nem tudnak belemenni, sem listázni a tartalmát), persze a superuser-ek láthatják.
- `chmod 750 ~/important_work`: ebben a mappában egy group-al együtt dolgozok valamin, a group olvashatja és futtathatja, én írhatom is, senki más nem láthatja.
- `chmod 754 -R ~/public_html`: az **-R** kapcsoló rekurzívan alkalmazza a beállításokat a mappa tartalmára.

B?vebben

- `man chmod`
- leírás

Bash

Alapvet? parancsok

- **pwd**: aktuális könyvtár kiírása.
- **cd**: könyvtárváltás. Ha paraméter nélkül hívjuk meg, akkor a saját mappánkba visz. Pl **cd /mnt, cd, cd ..**
- **ls**: könyvtár tartalmának listázása. Pl: **ls, ls -h, ls /dev, ls -h -l ~**
- **mkdir**: könyvtár létrehozása. **mkdir NewFolder**
- **cp**: másolás. **cp mit.txt hova.txt**
- **mv**: mozgató/átnevezés (az eredeti nem marad meg). **mv mit.txt hova.txt**
- **rm**: törlés, Pl: **rm alma.txt**, rekurzívan töröl minent a mappával együtt: **rm -r NewFolder**
- **quota**: A felhasznált/megmaradt helyet írja ki. Ha a kvótánk betelik, akkor nem kapjuk meg a nekünk címzett leveleket és a grafikus terminálra sem tudunk belépni; ilyenkor a parancssoros terminálra lépve tudjuk kitakarítani a home-unkat.
- **df, du**: lemezen szabad / felhasznált helyet írja ki. A **-h** kapcsolóval olvasható *GB, MB, kB* értékeket kapunk, de lehet fix byte-okban is kiírni. Pl: **df -h ~**
- **mc**: midnight commander fájlböngész?.
- segíts magadon
 - ◆ **info <parancs>**
 - ◆ **man <parancs>** az adott parancsról leírást ad. A leírásban navigálni a 'fel', illetve a 'le' billenty?kkel, továbbá a 'PageUp', 'PageDown'-nal lehet. Kereséshez '/' jel után kell beírni a kifejezést. Kilépés 'q'-val. **man ls**
 - ◆ **<parancs> --help**
 - ◆ **help**
- **history**: parancs történet lekérdezése.
- **apropos**: parancs/program nevek keresése. *apropos math*
- **top**: futó alkalmazások.
- **kill, killall, xkill**: folyamatok leállítása. Egy adott felhasználó összes folyamatának leállítása: **killall -u username**
- **cat**: fájl tartalmának kiírása. Pl: **cat .forward**
- **echo**: kiírja az argumentumait, Pl: **echo Hello World**. Figyelem, beugratós: **echo --help**
- **grep**: szöveg mintázatok keresése állományokban. **grep "http://wiki.math.bme.hut?"http://wiki.math.bme.hu szénakazal.txt**
- **less**: szöveges állományok megjelenítése, könnyebb olvashatóság nagy fájlknál, Pl **less valami.txt**
- **head (-n), tail (-n)**: fájl elejének és végének a kiírása. **head -n 100 valami.txt**
- **sort (-g -k)**: szövegállomány rendezése.
- **wc (-l)**: byte/karakterek/sorok számolás adott fájlra.
- **ssh**: távoli gépre való bejelentkezést teszi lehetővé. Linux-ról Windows-ról
- **scp**: távoli gépre/gépr?l való másolás. Linux-ról Windows-ról
- **wget**: file-ok letöltése internetr?l, HTTP, HTTPS és FTP használatával.
- **exit**: kilépés terminálból (vagy ctrl+d).
- **sudo**: programok futtatása superuserként. (Mat intézetben erre nem lesz lehet?ségünk) Pl. **sudo rm /home/someone_I_hate/important_file.txt**
- **halt, reboot**: kikapcsol, újraindít (Mat intézetben erre sem lesz lehet?ségünk)

Pipe, redirect

Minden program írhat a konzolra, vagy az `stderr`-re, ez az, amit a konzolon szoktunk látni. Azonban ezeket a képerny? helyett fájlba is írhatjuk.

```
$ ls -l ~
```

```
drwxr-xr-x 8 borbely student 4096 Aug 30 23:24 Desktop
drwxr-xr-x 2 borbely student 4096 Mar 27 2012 Downloads
drwxr-xr-x 2 borbely student 4096 Oct 20 2009 Drives
```

Informatika1-2015/Gyakorlat2

```
drwx----- 2 borbely student 4096 Apr 20 10:42 mail
drwxr-xr-x 7 borbely student 4096 Sep 6 13:01 public_html
$ ls -l ~ > folder.txt
$ cat folders.txt
drwxr-xr-x 8 borbely student 4096 Aug 30 23:24 Desktop
drwxr-xr-x 2 borbely student 4096 Mar 27 2012 Downloads
drwxr-xr-x 2 borbely student 4096 Oct 20 2009 Drives
drwx----- 2 borbely student 4096 Apr 20 10:42 mail
drwxr-xr-x 7 borbely student 4096 Sep 6 13:01 public_html
$ _
```

A > jellel egy parancs kimenetét egy fájlba irányítjuk át, majd a fájlt kiírathatjuk a cat-el. Ugyanakkor figyeljük meg a következőt:

```
$ ls -l /home/algebra/wettl/ > myfile.txt
ls: cannot open directory /home/algebra/wettl/: Permission denied
$ cat myfile.txt
$ _
```

A **myfile.txt** üresen maradt, de a konzolon hibaüzenet jelent meg (nem volt jogunk a mappa listázására). Ez azért van, mert az utóbbi **ls** parancs nem a standard kimenetre, hanem a hiba-kimenetre, azaz **stderr**-re írt. Az **stderr**-t a **2>** jelel irányíthatjuk át:

```
$ ls -l /home/algebra/wettl/ > myfile.txt 2> error.log
$ cat myfile.txt
$ cat error.log
ls: cannot open directory /home/algebra/wettl/: Permission denied
$ _
```

Így az egyik fájlba a standard kimenet (**stdout**), a másikban az **stderr** került bele.

Az egyes programok olvashatnak a standard bemenetükről: **stdin**. Az ún. nem-interaktív parancsok olyanok, hogy a parancs kiadása után a befejezéséig nem nyúlunk a billentyűzethez. Ezekkel csak az argumentumaikon keresztül kommunikálhatunk. Interaktív parancsok a futásuk közben olvasnak a standard bemenetükről: **stdin**

- nem-interaktív parancs: `grep`
"http://wiki.math.bme.huneeedle"http://wiki.math.bme.hu haystack.txt
- interaktív parancs: `grep`
"http://wiki.math.bme.huneeedle"http://wiki.math.bme.hu

Az előbbi esetben a **grep** egy megadott fájlban keresi a *needle* kifejezést. Az utóbbi esetben a beírt szövegben keresi ugyanazt.

```
$ grep "needle"
I hate this
why can't I find anything
why can't I find a needle
why can't I find a needle
oh, a needle
oh, a needle
again
$ _
```

A **Ctrl+D** billentyű-kombinációval megszakítjuk a futást. A *needle*-t tartalmazó sorok duplán vannak, mert azokat a **grep** kéresemre kiírta.

Próbáljuk ki így:

```
$ grep "needle" > needle.txt
I hate this
why can't I find anything
why can't I find a needle
oh, a needle
again
$ cat needle.txt
why can't I find a needle
oh, a needle
$ _
```

Ezzel kiválasztottuk a bemenetről a *needle*-t tartalmazó sorokat.

Az egyik parancs kimenete lehet a másik bemenete is, ez a **pipe**, jele egy függőleges vonal: |.

Példa: írjuk ki a home mappánk azon elemeit, melyek nevükben az *info* szövegrészletet tartalmazzák.

```
$ ls -l ~
Desktop
Downloads
Drives
info1
info_hf.txt
mail
myfile.txt
myfolder
needle.txt
public_html
regi_info
regi_info_zh.txt
$ ls -l ~ | grep "http://wiki.math.bme.huinfo"http://wiki.math.bme.hu
info1
info_hf.txt
regi_info
regi_info_zh.txt
$ _
```

Példa: sok elem van egy mappában és apránként végig akarjuk nézni:

```
$ ls /home/student/ | wc -l
821
$ ls /home/student/ | less
```

Egy fájlból kivehetjük a különböző sorokat (rendezés és duplikált sorok törlése):

```
$ cat nevek.txt | sort | uniq > nevek.sorted.txt
```

Ha egy fájl tartalmát akarjuk pipe-olni egy parancsba, azt így is megtehetjük (<):

```
$ cat nevek.txt | sort
$ sort < nevek.txt
```

Ez kombinálható > -el.

```
$ sort < nevek.txt | uniq > nevek.sorted.txt
```

Windows

A DOS után a Windows-os parancssori lehetőségek nem nagyon fejlődtek. A parancssor közelében sincs a Linux-nak, sem használhatóságban, sem funkcióiban, de azért van valami.

- Parancssor indítása
 - ◆ Start menü -> Parancssor
 - ◆ Start menü -> Futtatás -> cmd
 - ◆ Keresés -> cmd
- leírás
- használható: help, help <parancs>
- Windows 10-en új életre kelt a cmd.
- A Windows 10 egy éves évfordulója óta beépített a bash (béta verzió).

Használatos a PowerShell, ami nem mindegyik verzióba beépített, de ingyenes és elterjedt.

Különbségek

Pár szembevetés különbség

- a Windows case-insensitive, vagyis kis és nagybetű nem számít, ellenben a Linux case-sensitive.
- A könyvtár-, fájlnevek:
 - ◆ C:\Windows\System32\
 - ◆ /usr/bin/
- parancssori kapcsolók perrel versus kötőjellel:
 - ◆ Windows: dir /b
 - ◆ Unix: ls -l
- script
 - ◆ Windows-on: .bat
 - ◆ Linux-on: .sh
- pipe, redirect hasonló

Gyakran emulálják/helyettesítik a unix-os lehetőségeket windows-on:

- cygwin
- mingw

Előző gyakorlat - Fel - Következő gyakorlat