



ISMÉTLÉS

A 2. GYAKORLAT ANYAGÁNAK ÁTTEKINTÉSE



A LINUX PARANCSORA

- **Prompt:**
hxxxxxxx@linux2: /home\$ (felhasznalo@gepneve:munkakonyvtar\$)
- **Parancsok szerkezete:**
parancs (-opciók) (paraméterek)
parancs (--opció) (paraméterek)
- Kis- és nagybetű eltérő jelentéssel bír!
- **Segítség:** man parancs
- **Speciális karakterek:**
^C – folyamat megszakítása (végregezen)
^D – szövegbevitel vége
^Z – előtérben futó folyamat megszakítása (később folytatható)

A LINUX PARANCSSORA

- **Több parancs egyidejű kiadása**

Lehetőség van egy sorba több parancsot is beírni. Ezek végrehajtása lehet független egymáshoz, de a közéjük írt speciális karakterekkel kapcsolatot is teremthetünk köztük.

- **Lehetséges kapcsolatok**

Független végrehajtás – `parancs1; parancs2`

Csak sikeres végrehajtás esetén következhet az újabb parancs – `parancs1 && p2`

Csak sikertelen végrehajtás után jöjjön az újabb parancs – `parancs1 || parancs2`

FONTOSABB KÖRNYEZETIVÁLTOZÓK

- **\$HOME**

A bejelentkezett felhasználó home könyvtárának abszolút elérési útja.
Ez a ~ értéke.

Más felhasználók home útvonalának lekérdezése: **finger**

- **\$PATH**

A programok végrehajtásához használt keresési útvonalak listája, melyeket : választ el egymástól.

- **\$PS1, PS2**

Az első és másodlagos promptok szövege.

ILLESZKEDÉSEK

■ Speciális karakterek

- ^ — a sor elejére illeszkedik
- \$ — a sor végére illeszkedik
- * — tetszőlegesen sok, tetszőleges fajta karakter lehet
- ? — pontosan egy tetszőleges karakterre illeszkedik
- [halmaz] — a következő egy karakter a halmazból kerül ki

■ Példák

Pdf végződésű fájlnevek:	*.pdf
'A'-val kezdődő sorok:	^A
Évszám:	[0 2][0-9][0-9][0-9]
Kisbetűvel kezdődő kétbetűs fájlok:	[a-z]?

CSŐVEZETÉK

- **Mire való?**

Tetszőlegesen sok parancs összefűzhető vele.

A következő parancs bemenete az előző parancs kimenete lesz.

Egyirányú adatáramlás.

A háttérben történik, a monitoron nem jelennek meg az adatok.

- **Példák:**

```
ls | wc -l
```

```
find . -name 'm*' | xargs chmod 700
```

ÁTIRÁNYÍTÁS

- **Szabványos ki- és bemenet**

- < filename – szabványos bemenet helyett a fájlból olvas
- > filename – szabványos kimenet helyett a fájlba ír (felülír!)
- >> filename – ua. előző, de az eredeti tartalom mögé fűzi

- **Monitoron történő megjelenítés**

A fájlba írandó tartalom a monitoron is megjelenik.

- ls > lista – ls | tee lista
- ls >> lista – ls | tee -a lista

PROCESSZEK KEZELÉSE

- Lista

A shellben futó összes processzt kilistázza, a hozzájuk tartozó PID-t is.

A **ps** paranccsal indítható. Mivel a program fut, miközben a listát készíti, ezért önmagát is megjeleníti.

- Előtérbe/háttérbe hozás

`fg PID` illetve `bg PID`

- Háttérben indítás

& kerül a parancs végére, például `gedit&`

- Leállítás

`kill PID` vagy `kill %job`

`kill -9 PID` v. `kill -9 %job`

- terminálja a processzt
- azonnali leállításra kényszeríti

ÚJABB LEHETŐSÉG SZÖVEGFÁJL KEZELÉSRE

- **A parancs kinézete**

```
echo szoveg > file  
echo 'szoveg' > file  
echo -n 'szoveg' >> file  
echo -e 'elott\tutan' > file
```

- **Aposztróphasználat**

Akkor van szükség az aposztrófok használatára, ha speciális karaktereket szeretnénk beírni (**\t** a tabulátor, **\n** a sortörés). Az aposztrófok közti felesleges white space karaktereket nem vonja össze.

- **Kapcsolói**

- n nem rak sortörést az utolsóként bevitt sor után (alapértelmezetten tesz)
- e speciális karakterek bevitele is megengedett, ekkor az aposztrófok közti **\t** és **\n** karakterpárokat átalakítja tabulátorra illetve sortöréssé.

SFTP

- **Bejelentkezés**

`sftp hxxxxxxx@linux.inf.u-szeged.hu`

- **Parancsok**

A teljes lista lekérdezhető a **?** karakterrel.

Sok linuxos parancsnak van l-es megfelelője (ls, cd, mkdir...) a lokális fájlrendszer használatához.

- **Fájlok mozgatása**

`get forrásútvonal (célútvonal)` – távoliról lokálisra

`put forrásútvonal (célútvonal)` – lokálisról távolira

Ha elhagyjuk a második paramétert: változatlan névvel kerül az aktuális mappába.

A második paraméter egy mappa útvonala: másolat készül bele az eredeti névvel.

A második paraméter egy még nem létező fájl útvonala: a megadott helyre a megadott néven készül a másolat.

FELADATOK

- Készítsünk egy `3gyak` nevű mappát a `/home/hxxxxxx` könyvtárunkba.
- Állítsuk át a jogosultságát `677`-re.
- Az aktuális könyvtárból egyetlen parancs kiadásával hozzunk létre egy `uj` nevű könyvtárat a `3gyak` nevűn belül. Mit tapasztalunk? Miért?
- Adjuk meg a hiányzó jogot a tulajdonosnak, majd ismételjük meg a parancsot!
- Másoljuk bele a `/pub/progalap/Gyakorlat/gyak03` könyvtár `anyag.txt` állományát `tananyag` néven az `3gyak` könyvtárba.
- Írassuk ki, hány sort, szót és karaktert tartalmaz a fájl!
- Készítsünk hozzá erős linkelést `hardlink` néven.
- Másoljuk át a `3gyak` mappa teljes tartalmát az `uj` nevű mappába.

FELADATOK

- Jelentkezzünk be `sftp` kapcsolattal a kabinetes szerverre!
- Készítsünk a lokális fájlrendszeren egy `celmappa` nevű állományt!
- Másoljuk át bele a `3gyak` mappa teljes tartalmát!
- A távoli számítógépen hozzunk létre egy mappát `viSSza` néven!
- Másoljuk át a `viSSza` mappába a lokális gépen található `tananyag` állományt `anyag` néven!

FELADATOK

- Az `tananyag` azon sorait, melyek `txt`-re végződnek, irányítsuk át a `textfiles` nevű állományba. Ha az állomány már létezett, írjuk felül!
 - A keletkezett állomány leghosszabb sorának karakterszámát fűzzük a `textfiles` állomány végére úgy, hogy az eredmény a képernyőn is megjelenjen!
 - Szűrjük ki a képernyőre `textfiles` állomány azon sorait, melyek kétjegyű számmal kezdődnek!
-

FELADATOK

- Az `tananyag` azon sorait, melyek `txt`-re végződnek, irányítsuk át a `textfiles` nevű állományba. Ha az állomány már létezett, írjuk felül!
- A keletkezett állomány leghosszabb sorának karakterszámát fűzzük a `textfiles` állomány végére úgy, hogy az eredmény a képernyőn is megjelenjen!
- Szűrjük ki a képernyőre `textfiles` állomány azon sorait, melyek kétjegyű számmal kezdődnek!

```
grep txt$ tananyag > textfiles  
wc -L textfiles | tee -a textfiles  
grep ^[0-9][0-9]* textfiles
```

FELADATOK

- Írasd ki a `textfiles` állomány utolsó előtti sorát a képernyőre!
 - Vond meg a tulajdonos és a csoport olvasási jogát az aktuális könyvtár összes `text`-tel kezdődő állományától anélkül, hogy a többi jogosultság megváltozna!
 - A `3gyak` nevű mappában vagyunk. Fűzzük hozzá a `textfiles` állományhoz az új mappabeli `tananyag` tartalmát anélkül, hogy az megjelenne a képernyőn. (Tudjuk, hogy a `3gyak` és az új nevű mappa azonos könyvtárban van.)
-

FELADATOK

- Írasd ki a `textfiles` állomány utolsó előtti sorát a képernyőre!
- Vond meg a tulajdonos és a csoport olvasási jogát az aktuális könyvtár összes `text`-tel kezdődő állományától anélkül, hogy a többi jogosultság megváltozna!
- A `3gyak` nevű mappában vagyunk. Fűzzük hozzá a `textfiles` állományhoz az `uj` mappabeli `tananyag` tartalmát anélkül, hogy az megjelenne a képernyőn. (Tudjuk, hogy a `3gyak` és az `uj` nevű mappa azonos könyvtárban van.)

```
tail -n 2 textfiles | head -n 1  
chmod ug-r text*  
cat ../uj/tananyag >> textfiles
```