

02

Bevezetés II.

Parancsok szöveges fájlokra, linkelés,
standard csatornák, átirányítások

Követelmények

- Végleges szöveg a honlapomon + neptunban + coospacen
- zh időpontok:
 - március 19-cel kezdődő héten a gyakorlat idejében
 - május 7-tel kezdődő héten a gyakorlat idejében
- mindenki a felvett gyakorlatra járjon be, máskülönben elesik a zh írás és az óra végi extra feladatok megírásának a lehetőségétől

Parancsok

- PARANCS -egybetűskapcsoló (vagy --hosszúnevűkapcsoló) PARAMÉTER1, PARAMÉTER2,
- Például:
 - ls -l /home/
 - cd /var/
 - man --help
 - exit

Szövegszerkesztők

- CLI:
 - vi, **vim**
 - **nano**, pico
 - joe
 - emacs
- GUI:
 - xedit, **gedit**
 - geany, atom (de ezek már inkább fejlesztő felület, mint egyszerű editorok)
 - (xemacs)

új fájl létrehozása: <szövegszerkesztő> fájlnev (&), majd mentés
(vagy touch)

Kiíratás – emlékeztető

- cat [kapcsolók] [fájl(ok)]
 - Kiírja a fájl(ok) tartalmát a **standard kimenetre**.
 - Több fájl esetén a megadás sorrendje alapján **összefűzi** őket (egymás után íratja ki).
 - Ha a fájlnev helyett a „-” karakter szerepel, vagy nem adunk meg semmit, akkor a **standard bemenetről olvas**.
- more, **less**
 - képernyőre optimalizált fájlkiíratás (ez az, ami tényleg az olvasásra használandó)
 - less a more bővített változata (lapozás, visszatekerés, ...)

Linkelés

- egy állományt több helyen/néven is elérhetővé tegyünk a fájlrendszerben
- Hard link:
 - ln TARGET LINKNAME
 - újra létrehozza a fájlt, ugyanarra az inodera mutat, bármelyik változik akkor egyszerre változik az összes
 - **biztonsági másolat**
 - eredeti törlése esetén megmarad a hard link
- Soft (v. symbolic) link:
 - ln -s TARGET LINKNAME
 - hagyományos értelemben egyszerű link (tükör) jön létre, ami az eredeti állományra mutat
 - **hagyományos link**
 - eredeti törlése esetén megmarad a soft link, de sérül és nem működik többet

Csatornák (streamek)

- 3 standard be- és kimeneti csatorna létezik:
 - Stdin (0) : alapértelmezetten a billentyűzet
 - Stdout (1) : alapértelmezetten a képernyő (terminál)
 - Stderr (2) : alapértelmezetten a képernyő (terminál)
- Ezek átirányíthatóak más eszközökre (device) vagy fájlba

Streamek átirányítás

- < ÁLLOMÁNY: stdin átirányítása (a megadott fájlból **olvas**)
- > ÁLLOMÁNY: stdout átirányítása (a megadott fájlba ír, a létező állomány **felülírásával**)
- >> ÁLLOMÁNY: stdout átirányítása (a megadott fájlba ír, a létező állomány végéhez való **hozzáfűzéssel**)
- 2> ÁLLOMÁNY: stderr átirányítása (a megadott fájlba írja a **hibaüzeneteket**)
- &> ÁLLOMÁNY: stdout és stderr átirányítása ugyanabba a fájlba
- 2>&1: a stderr-t ugyanoda irányítja, ahová a stdout irányítva lett
- 1>&2: a stdout-ot ugyanoda irányítja, ahová a stderr irányítva lett

Feladat

- `mkdir workspace` parancs hibakimenetének átirányítása egy `err.log` fájlba
- `feladat02.dat` fájlt irányítsuk át a `4.txt` -ba, majd még egyszer úgy hogy kétszer szerepeljen a `4.txt`-ben a `feladat02.dat` fájl tartalma
- hozzunk létre egy üres fájlt és a fájl nevét, mint stringet irányítsuk át az adott fájlba, mint szöveges tartalom (hint: `basename`)
- az előző kisebb méretű fájlok tartalmát irányítsuk át egy közös fájlba
- `Stdin`-ről olvasott tartalmat (`cat`, `Ctrl+d`) irányítsuk át egy fájlba, majd a fájl tartalmát irányítsuk át egy másik terminálba (`tty`)