

# 02

## Bevezetés II.

Parancsok szöveges fájlokra, standard csatornák,  
átirányítások, csővezeték

# Szövegszerkesztők

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

új fájl létrehozása: <szövegszerkesztő> fájlnev (&)

# 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, ...)

# 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

- Hozzunk kézzel létre három különböző fájlt, számozzuk meg, a fájlnevet irányítsuk át a fájlba (hint: basename, dirname), majd a három fájl tartalmát irányítsuk át egy negyedikbe
- Stdinről olvasott tartalmat irányítsuk át egy fájlba, majd a fájl tartalmát irányítsuk át egy másik terminálba

# Kiíratás

- `wc` – kiírja a fájlban található sorok (1), szavak (2), bájtok (3) számát
- `head` – első tíz sort írja ki (paraméterrel a tíz módosítható másra)
- `tail` – utolsó tíz sort írja ki (itt is hasonlóan)
- `sort` – a sorokat abc sorrendbe rendezi a sorkezdő karakter alapján (de pl számok alapján is rendez -n kapcsolóval)
- `uniq` – egymás után ismétlődő sorokat egyszer írja ki
  - kapcsolóval kiírható csak az ismétlések, vagy azok amik nem
  - rendezetten működik csak (`sort`)

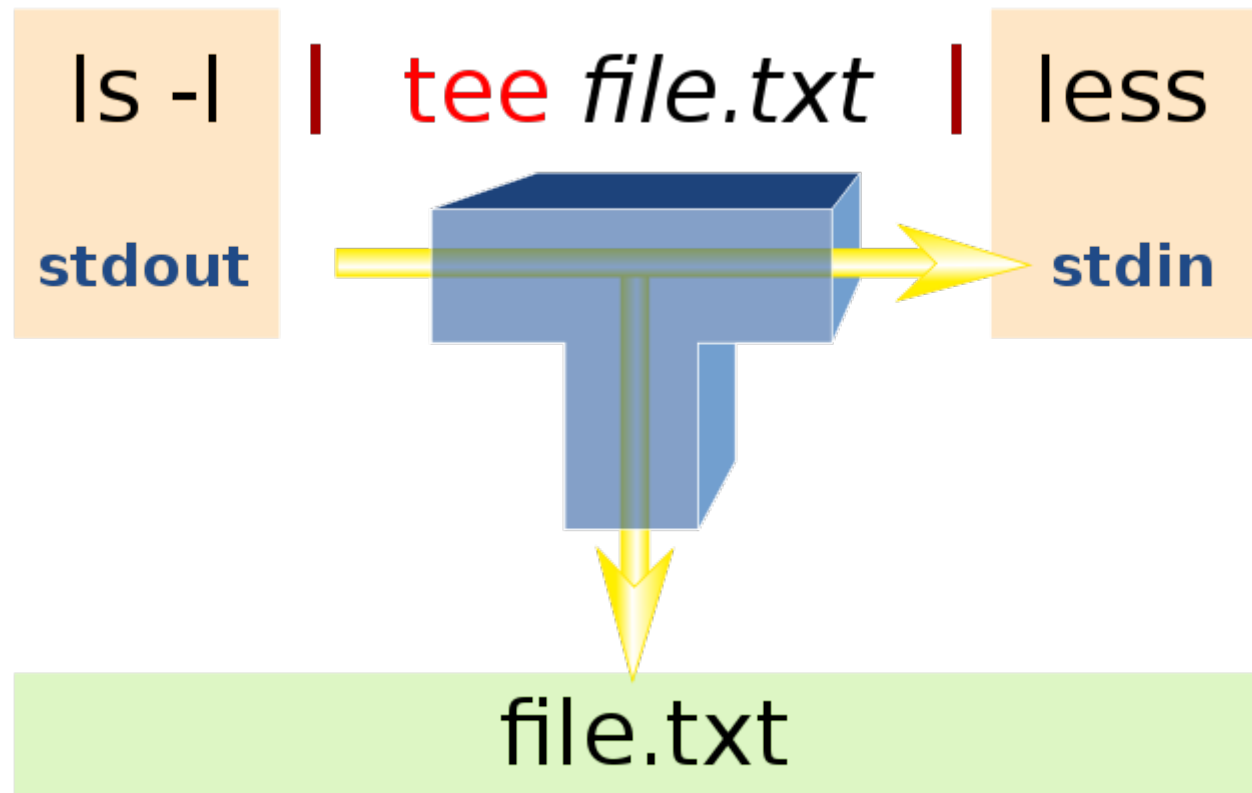
# Csővezetékek (pipeline)

- |
- Egyik parancs kimenetét (stdout) irányítjuk a következő parancs paraméterének (stdin)
- Feladat:
  - Hány állomány van egy könyvtárban?
  - Listázzunk ki lapozhatóan egy olyan könyvtár fájljainak részletes lista nézetét, ami nem fér ki egy képernyőre
  - Írjuk ki a feladat.dat fájl 40.-49. sorát (head és tail felhasználásával)
  - Hány különböző szám szerepel a feladat.dat fájlban?
  - Mi a maximuma a feladat.dat fájlban?



# Szétbontás - tee

- standard kimenetre és a paraméterként adott fájl(ok)ba irányít
- - a: append



# Archiválás, tömörítés

- ZIP (tömörítés):
  - `zip -r archive_name.zip directory_to_compress`
  - `unzip archive_name.zip`
- TAR (archiválás):
  - `tar -cvf archive_name.tar directory_to_compress`
  - `tar -xvf archive_name.tar`
- TAR.GZ (tömörítés):
  - `tar -zcvf archive_name.tar.gz directory_to_compress`
  - `tar -zxvf archive_name.tar.gz`
- TAR.BZ2 (tömörítés, lassabb, alapos) :
  - `tar -jcvf archive_name.tar.bz2 directory_to_compress`
  - `tar -jxvf archive_name.tar.bz2 -C /tmp/extract_here/`

# Mintaillesztés

- \* tetszőleges karakterből álló szó
- ? a helyén valamilyen tetszőleges karakter
- [HALMAZ] a felsorolt karakterek valamelyike
- [ELSŐ-UTOLSÓ] hasonlóan az előzőhöz csak egy tól-iggel megadva
- [^HALMAZ] a fel nem sorolt karakterek lehetnek ott

Feladat: tömörítsük össze az összes olyan fájlt aminek a fájlneve egyetlen tetszőleges, egyjegyű számból áll