

02

Bevezetés II.

Parancsok szöveges fájlokra, standard csatornák,
átirányítások, csővezeték

Állományrendszer fizikai felépítése

- 1.Boot block (nulladik blokk): az ebben levő rövid program tölti be a rendszert
- 2.Superblock (első blokk): az állományrendszer részleteit és a belső táblák adatait tartalmazza
- 3.Inode tábla: az inode-ok adatait tartalmazza
 - Az inode (index node)
 - egy adott állomány minden fontos adatát tartalmazza: méretet, típust, tulajdonost, a hozzáférési jogokat, a háromféle dátumot (access: elérés, modify: módosítás, change: inode változás), az állományhoz tartozó lemezblokkok sorszámait, valamint a merev láncok számát avagy a láncszámot (ld. később).
 - Szigorúan véve az inode-okat azonosíthatjuk az állományokkal.
`stat, (touch)`
 - Minden inode egyedi sorszámot kap.
 - Minden könyvtárhoz tartozik egy állomány. Ez a speciális állomány tartalmazza a könyvtárban levő állományok nevét és inode-számát.

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 (&
(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, ...)

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 átírányíthatóak 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, mint stringet irányítsuk át az adott fájlba, mint szöveges tartalom (hint: basename), majd a három fájl tartalmát irányítsuk át egy negyedikbe
- Stdinről olvasott tartalmat (cat, Ctrl+d) irányítsuk át egy fájlba, majd a fájl tartalmát irányítsuk át egy másik terminálba (tty)

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 rendezetten írja ki, 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 (-d)
 - vagy azok amik nem (-u)
 - rendezetten működik csak (sort)