

# Operációs rendszerek gyak.

## Linux alapok III., Bash

Cirok Dávid    Hirling Dominik

Szegedi Tudományegyetem

*Cirok.David@stud.u-szeged.hu*

*Hirling.Dominik@stud.u-szeged.hu*

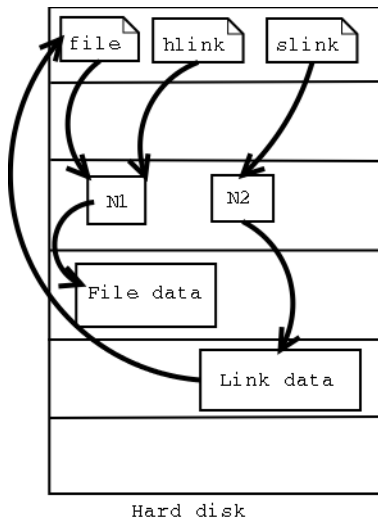
# Linux alapok III., Bash

- 1 Linkelés
- 2 Állománykeresés
- 3 Felhasználók kezelése
- 4 Jogosultságok kezelése
- 5 Shell

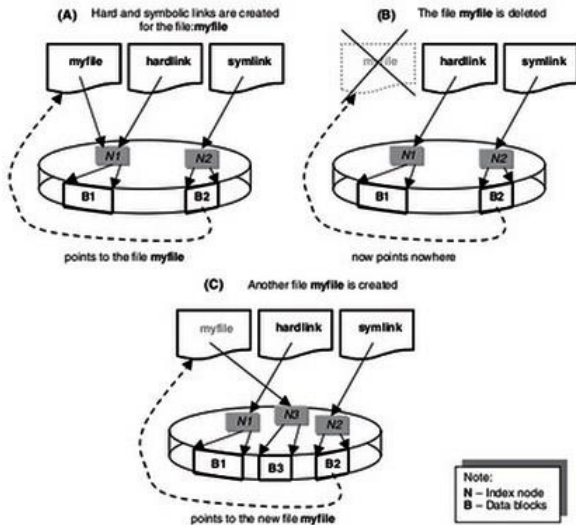
# Linkelés

- Az állományrendszer lehetővé teszi, hogy egy file-t több néven is elérhessünk. Ezt úgynevezett **linkek** segítségével tudjunk elérni. Ezek olyan másolatok, amelyek az eredeti file-ra mutatnak. Két fajtájuk van:
  - **hard** link: ugyanarra az inode-ra mutat, mint az eredeti, megkülönböztethetetlenek egymástól. Ha töröljük az eredetit, ez megmarad.
  - **soft** link: kb. shortcut windows-on: más inode, ami "pointer" az eredeti file-ra. Ha töröljük az eredetit, "nullpointer".
- Használat: `ln FILE LINK`: a FILE-ről készít egy linket LINK néven.
- `ln -s` esetén soft link.

# Linkelés



# Linkelés



# Állománykeresés

- `locate REGEX`: a reg. kif.-re illeszkedő könyvtáraknak, file-oknak megadja az elérési útvonalát
- `find`: ugyanúgy keresésre jó, de sokkal fejlettebb a `locate`-nél, összetett kifejezések, boolean operátorok használhatóak benne és sokkal testreszabhatóbb:
  - `find / -name REGEX`: a root-ból kiindulva a reg. kif.-re illeszkedő nevű állományokat keresi
  - `find . -size +MERET`: az aktuális könyvtárból kiindulva a megadott méretnél nagyobb állományokat keresi
  - `find / -type TIPUS REGEX`: adott típusú állományokat keres, amelyek illeszkednek a reg. kif.-re (f: file, d: dir, l: link)

# Felhasználói információk

- `who`: aktuálisan bejelentkezett felhasználók listázása
- `w`: mint az előző csak részletesebb
- `whoami`: aktuális felhasználó azonosítójának kiírása
- `groups`: aktuális felhasználó csoportjainak kiírása
- `passwd`: aktuális felhasználó jelszavának beállítása (root bárkiét beállíthatja)
- nyilvántartás: `/etc/passwd`, `/etc/group`

# Hozzáférési jogok

- háromféle jog:
  - olvasás (read)
  - írás (write)
  - végrehajtás (execute)
- jogok megadhatók:
  - tulajdonos (owner)
  - csoport (group)
  - mindenki más (all users)

részére

- `chmod <jog> <útvonal>`
  - szimbolikus alak:
    - [FELHASZNÁLÓK] [MŰVELET] [JOGOK]
    - `chmod ug+rx os2017.sh`
  - numerikus alak:
    - háromjegyű szám, tulajdonos | csoport | mindenki
    - számok a jogok összegei: végrehajtás 1, írás 2, olvasás 4
    - `chmod 754 os2017.sh`
- -R kapcsoló



# Shell alapok

- kernel és a felhasználó között közvetít
- parancsértelmező (command interpreter)
- interaktív mód: egyenként megadunk parancsokat
- batch mód: szöveges fájlok (shell scriptek) futtatása
- régen egyedüli mód a géppel való interakcióra, manapság grafikus környezetek (GUI) helyettesíthetik

# Shell típusok

- sh: Bourne shell (legrégebbi)
- csh: C shell
- ksh: Korn shell
- bash: Bourne again shell (ezt használjuk)
- fish
- zsh

# Shell feladatai

- parancssor kezelése
- munkafolyamatok kezelése
- átirányítások elvégzése
- csővezeték kezelése
- aliasok értelmezése
- mintaillesztő karakterek értelmezése
- állománynév-kiegészítés végrehajtása
- shell scriptek futtatása
- személyes beállítások kezelése (környezeti változók, testreszabás)

# Bash alapok

- Bash programok szöveges állományok, amelyek futtatását a bash program végzi
- szöveges fájl elején lévő **parancsértelmező fejrész** határozza meg, hogy a kernel melyik programot használja egy script futtatására
- Bash esetén: `#!/bin/bash`
- a script fájl minden sora külön parancsként lesz végrehajtva
- használhatunk **vezérlési szerkezeteket** is (if, while, for...)
- egysoros kommentekhez: `# komment`
- ha `\` van a sor végén, akkor a következő sor is abba fog számítani (töbsoros parancs)

# Bash scriptek futtatása

- ① `bash SCRIPT PARAMS`
- ② futtatás állományként (ajánlott):
  - parancsértelmező fejléc fontos!
  - végrehajtási jog szükséges: `chmod +x SCRIPT`
  - `./SCRIPT PARAMS` (jelenlegi mappában is kell `./`)