**Gábor Dénes**
**Főiskola**

# PREZENTÁCIÓ

***A UNIX OPERÁCIÓS RENDSZER***

**120**

**Vezetőtanár:**
**Dr. Jedlovszky Pál**

**2000/2001**
**5. szemeszter**

**32 lap**



## A rendszer működése

Bekapcsolás
- Operációs rendszer betöltés
- Felhasználók be- és kilépése
- Rendszer lekapcsolása
  *(Rendszergazda, üzenet)*

Kikapcsolás

**UNIX-ot futtató gépet kikapcsolni tilos!!!**

## Felhasználói csoportok

Rendszergazda
Privilegizált felhasználók
**Felhasználók** *(Ezek vagyunk mi)*



# 1. BEVEZETÉS

**UNIX:**

Többfelhasználós, többfeladatos rendszer

**LINUX:**

A UNIX új, "szabad" változata
Nagy hálózatok operációs rendszere,
de PC-re is alkalmas

## Felépítés

| *alkalmazások* | |
|---|---|
| **segédprogramok** | **shellek** *(parancsértelmezők)* |
| **kernel** *(mag)* | |
| *hardver* | |



## Bejelentkezés

| | | |
|---|---|---|
| **`login:`** | *Ubul* | *(felhasználói név)* |
| **`password:`** | *qwert123* | *(titkos jelszó, nem látszik)* |
| $ | | *(prompt, parancsot vár)* |

## Folyamatok

**Programok futása**
**Háttér folyamatok**
Démonok

## Parancsok szintaxisa

$ Név opciók paraméterek # Kommentár **return**

## Hibaüzenet

Parancs normál végrehajtása esetén általában
**n i n c s v i s s z a j e l z é s !**

## A parancsértelmező

Belépés: a parancsértelmező (shell) aktivizálódik

$ *(Prompt)*
Parancs begépelés
Parancs értelmezés
Parancs végrehajtás: folyamat indul
Folyamat véget ér *(háttérfolyamatnál nem várjuk ki)*
•
Kiléptető parancs *(pl. logout, exit, ^d)*

Kilépés

A parancsok végrehajtása két ütemben történik:
1. Parancsértelmező
2. Maga a parancs

## Shell script

Parancsok file-ba írhatók
Bonyolult parancs sorozatok állíthatók össze.



# 2. A FILE RENDSZER

*"A UNIX-ban minden file"*

## File típusok:

1. Közönséges file
   - Adat-file — írható, olvasható
   - Végrehajtható file
     - Bináris
     - Shell script
2. Katalógus file
3. Speciális file
   - pl: billentyűzet, képernyő, szimbolikus lánc

## A katalógus rendszer

Egyetlen fa szerkezet
**/**közbenső lépcsők/***Ubul***/**saját katalógus
↑ ↑
*root* *bejelentkezési katalógus*

A katalógus tartalma a file műveletekkel automatikusan változik



## Név konvenció

Elvben 128 karakter használható
Kis/nagy betű különböző
Kerüljük a speciális karaktereket: $ * # ? stb.
**.***uborka* rejtett file, másképp kezelendő

## Speciális megnevezések

| | |
|---|---|
| */* | gyökér |
| • | aktuális katalógus |
| •• | szülő katalógus |
| ~ | bejelentkezési katalógus (nem mindenütt) |

## File-ok megnevezése

Az aktuális katalógusban *ezaz*
hangsúlyozottan *./ezaz*
Eggyel lejjebbi katalgusból *../ezaz*
Kettővel lejjebbi katalgusból *../../ezaz*
Abszolút út, a gyökértől indulva
*/home/tanulo/belso/ezaz*
A bejelentkezési katalógustól
*~/belso/ezaz*



## Behelyettesítés file-névbe

| *Karakter* | *Illeszkedés* |
|---|---|
| **?** | Az adott pozícióban bármely karakterre |
| * | Az adott pozíciótól kezdve akárhány, akármilyen karakterre, kivéve a névkezdő • karaktert. |
| [*ai*] | Az adott pozícióban az *a*-ra és *i* -re |
| [*a-p*] | Az adott pozícióban *a* és *p* közötti kisbetűkre |

## Példák

| ***Minta*** | ***Illeszkedik*** | ***Nem illeszkedik*** |
|---|---|---|
| ***Jan?*** | Jani Jano Jana | Jancsi, Janika |
| ***Jan**** | Jani Jano Jana Jancsi Janika Januar | januar •Janu Ja |
| ***Jan[ai]*** | Jana Jani | Semmi másra |
| ***Jan[a-p]*** | Jana Jano Jani | Jancsi, Januar |
| ***Jan[a-p]**** | Jana Jano Jani Janika | Januar, stb. |
| * | Minden közönséges file-ra | Rejtett file-okra |
| ***[0-9]*** | Számjegyre végződőkre | |
| **a*z** | *a*-val kezdődő és *z*-re végződőkre | |
| ***k**** | *k* betűt tartalmazókra | |

# 2.1 Katalógusok kezelése

## Létrehozás

$ **mkdir** MUNKA

$ **mkdir** PROBA MASIK

opció: **-p** Létrehozza a szülőkatalógusokat is

$ **mkdir -p** JATEK/SAKK

## Törlés

$ **rmdir** MUNKA *# Csak üres, több is*

## Hol vagyunk ?

$ **pwd** *# kiírja az aktuális katalógust*

## Katalógus váltás

$ **cd** JATEK # az aktuális katalógustól indul
$ **cd** */users/Ubul/PROBA* # abszolút út, elején */*
$ **cd ••** # visszalépés a szülő katalógusba
$ **cd** # visszalépés a kiindulási katalógusba



# 2.2 File kezelés

## Létrehozás

**Alapelvek**

1. Ha szükség van egy file-ra, a shell létrehozza
2. Helyettesítő karakterek esetén - ha lehet - minden file-ra végrehajtódnak a manipulációk

**Leggyakrabb esetek:**

Szövegszerkesztővel
Másolással
Output átirányítással

Rövid file esetén **cat** paranccsal

$ **cat** >elso
*Ez az elso file*
*Ez a 2. sora*
**^d**

## File megtekintése

$ **cat** *elso*
*Ez az elso file*
*Ez a 2. sora*
$

Ugyanez a **more** illetve a **less** paranccsal
*( Kicsit okosabbak )*



## File másolása

$ **cp** elso x1
↑ ↑
*mit hova*
$ **cp** elso PROBA
# a *PROBA* katalógus-ba *elso* néven másolja
$ **cp** elso masodik PROBA
# mindkettőt másolja
$ **cp** e* PROBA
# minden *e*-vel kezdődőt átmásol

## File áthelyezése (átnevezése)

$ **mv** x1 PROBA # Áthelyezés *PROBA/x1* - be
$ **mv** x1 PROBA/y # *Áthelyezés PROBA/y - ba*
$ **mv** x1 y # Ugyanaz a katalógus: átnevezés

Több file is másolható vagy áthelyezhető, de ekkor az utolsó paraméter katalógus

***Vigyázat: mindkét parancs felülír!***



## File törlése

$ **rm** *y*
Opciók
**- f** *Írásvédettet is töröl*
**- i** *Visszaigazolást kér*
**- r** *Rekurzívan az egész részfa törlése, alkatalógusokkal együtt*

***VESZÉLY!***
***A kitörölt file nem állítható vissza!***

## Listázás

$ **ls**
*JATEK PROBA elso*
# ABC sorrend, aktuális katalógus, csak nevek

$ **ls** *el??* # Csak a megadottakat
*elso*

**Opciók**

-R Kilistázza a teljes fa-struktúrát
-a Kilistázza a rejtett file-okat is
-d Katalógusokba nem megy bele

Példa a teljes fa-struktúrára

$ **ls -R**

*JATEK PROBA elso*

*JATEK:*

*SAKK*

*JATEK/SAKK:*

*PROBA:*

*y*

## Hosszú lista

$ **ls -l** elso

total *122*

*-rwxrw-r-- 1 Ubul user 496 Aug 23 12:10 elso*

A hosszú lista értelmezése:

Blokkok száma
*Utána minden file-ra:*
**file típus**
**hozzáférési jogok**
**láncolási szám** (Ezeket fogjuk részletezni)
tulajdonos
csoport
méret
létrehozás ideje
név



## File típusok

| | | | |
|---|---|---|---|
| - | közönséges | p | speciális cső (pipe) |
| **d** | katalógus | c | karakteres készülékmeghajtó |
| l | szimbolikus lánc | b | blokkos készülékmeghajtó |

## Hozzáférési jogok

3x3 kategória

| | | | |
|---|---|---|---|
| tulajdonos | \| | **r** | olvasás |
| csoport tagok | \| | **w** | írás |
| külsők | \| | **x** | végrehajtás |

**Példa**

- *rwx rw- r--*

1 2 3 4

1 közönséges file
2 a tulajdonos mindent tehet
3 a csoport tagok írhatnak, olvashatnak
4 a kívülállók csak olvashatnak

Katalógus esetén

| | |
|---|---|
| olvasás | kilistázható |
| írás | bejegyzéseket tehetünk, törölhetünk |
| végrehajtás | hozzáférés a katalógushoz |

A teljes útvonalon kell a katalógusokra a végrehajtás!



## Jogosultságok módosítása

$ **chmod** *<új jogok> <file nevek>*

**Példa**

Alapállapot

$ ls *KONYVTAR* -l # Alapállapot

-rw-rw-rw- ubul user 129 Feb 29 10:16 ez_van

**Módosítási lehetőségek**

1. Oktális

$ **chmod** *764 ez_van*

Új jogosultság - *rwx rw- r--* lesz
*7 6 4*

2. Egyenként

3 karakter: *<kinek> <mit> <melyik jogosultság>*

| | | | |
|---|---|---|---|
| u | *tulajdonos* | **+** | *engedélyezés* |
| g | *csoport tagok* | **-** | *tiltás* |
| o | *külső* | | |
| a | *mindhárom csoport* | | |

$ **chmod** u+x *ez_van*

$ **chmod** o-w *ez_van* # az előzővel azonos hatás

Jogosultságot csak a tulajdonos módosíthat

## Két parancs

**chown** tulajdonost vált

**chgrp** csoportot vált



## Láncolás

**Szoros lánc (hard link)**

Ugyanazt az állományt több néven is el lehet érni, különböző katalógusokból is.

$ **ln** elso first

$ **ls** -l elso first

rw-rw-rw- 2 ubul user 496 Aug 23 12:10 elso

rw-rw-rw- 2 ubul user 496 Aug 23 12:10 first

Törlésnél a láncolási szám csökken, zérus esetén törlődik a file.

Cél: névütközések feloldása, névkonvenciók segítése, elérés távoli katalógusokból

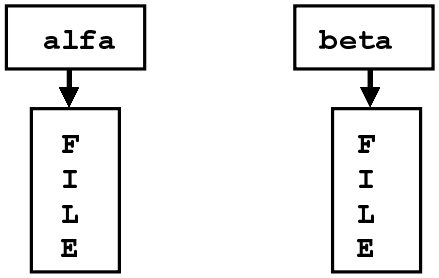
**Szimbolikus lánc**

$ **ln -s** elso szimbola

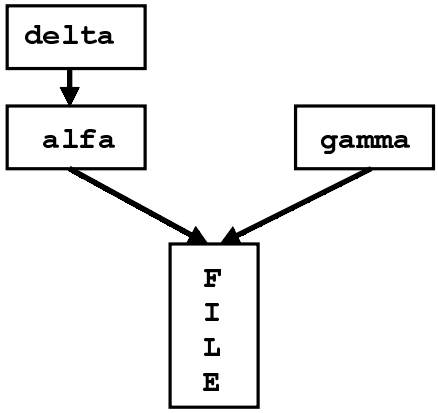
Különböző file-rendszerek (pl.: hard diszk és floppy) között teremthet kapcsolatot.

**Láncolás illusztrációja**

$ **cp** alfa beta




$ **ln** alfa gamma ; **ln –s** alfa delta






$ **grep** <*opciók* > <*minta*> <*file nevek*>
# Stringek keresése file-okban
Opció
**-v** Megfordítja a keresést
Minta
Reguláris kifejezés
^x a sor elején keresi az x-et
x$ a sor végén keresi az x-et
**.** tetszőleges karakter

$ **echo**
Visszaírja a paramétereit
Minden szó külön paraméter

$ **clear**
Letörli a képernyőt
Speciális karakter, átirányítható

$ **sleep** $N$ # $N$ másodpercet vár

$ **mail** <címzés>
Levelek írása és fogadása



# 3. A PARANCS SOR

## Néhány parancs

$ **date** # Kiírja a dátumot és időt
*Fri Sep 22 12:01:16 CET 2000*

$ **who** # Kiírja a bejeletkezett felhasználókat,
*bela vt03 Aug 13 11:26*
*bela vt04 Aug 13 11:28*

$ **sort** # Rendezi a bemenetről kapott sorokat
Opciók
**-r** Visszafelé rendez
**+*N*** $N$ szóval hátrább kezdi a rendezést

$ **wc** # Megszámlálja az inputban a sorokat,
# szavakat, karaktereket
Opciók
**-l -w -c** Csak a sorokat, szavakat, karaktereket



## Perifériák és átirányításuk

**Standard perifériák**

| | | |
|---|---|---|
| **stdin,** | **0** | standard bemenet, billentyűzet |
| **stdout,** | **1** | standard kimenet, képernyő |
| **stderr,** | **2** | standard hibajelzés, képernyő |

A képernyő neve Linuxban */dev/ttyN* *(N=1,2,3,4,5,6)*
Output eltüntetése */dev/null*

Átirányítás file-ba

**<** input átirányítás
**>** output átirányítás felülírással
**>>** output átirányítás hozzáírással

**Példák:**

1. $ **ls** >lista
   $ **cat** lista # Kilistáztuk, és a *listá*-ban
   # megőriztük a katalógust
   $ **wc -w** lista # Megszámoltuk a katalógus-
   # ban lévő file-okat ( + lista!)
   $ **cd** ALFA # Átléptünk az ALFA katalógusba
   $ **ls** >>../lista # Hozzáírtuk az új katalógust
2. $ **wc -l** * 2>/dev/null # Hibajelek nem kellenek

## A csővezeték (pipe)

Az 1. parancs kimenete lesz a 2. bemenete. Jele: **|**

$ **who | sort -r**

# Visszafele rendezve kapjuk a felhasználókat

$ **who | grep Nagy | sort -r**

## Elágaztatás

**tee** <paraméter>

Bemenet a standard inputról,
kimenet a standard outputra és a paraméter fileba

$ . . . **|** . . . **|** **tee** *megoriz* **|** . . .
↑
Itt megőriztük a részeredményt

$ **who | tee** */dev/tty1* **| sort -r**
↑

Közbenső kiíratás eredeti sorrendben
A hibajelzés továbbra is a képernyőre kerül

## Több parancs egy sorban

Elválasztás **;** -vel
Végrehajtás csak **return** után

## Parancsok csoportosítása

**( )** zárójelezéssel



## Parancs behelyettesítés

**` `** „Vissza-aposztrófok” közé rakjuk
A parancs eredménye kerül a megfelelő helyre

**Példa**

$ echo Most **`*ls | wc -l*`** fajl van a katalogusban

## Metakarakterek semlegesítése

| | |
|---|---|
| **** | után 1 karakter |
| **' '** | zárójelpár |
| **" "** | zárójelpár *(kicsit más hatás)* |

**Példa** három csillagot akarunk kiírni

$ **echo '*' * "*"**

## Folyamatok időzítése

Folyamat: egy éppen futó program

1. Normál

$ <Parancs> # Addig kell várni, míg végrehajtódik
$

2. Háttér folyamat

$ *<Parancs>* **&**
1926 # Kiír egy folyamatazonosító számot (PID)
# Elindítja a folyamatot, és rögtön
$



3. Folyamatok listázása

$ **ps**

| PID | TTY | TIME | COMMAND |
|---|---|---|---|
| *1926* | *vt02* | *12:34* | *Parancs* |

4. Folyamat lelövése

$ **kill** *1926*

5. Futás kilépés után

$ **nohup** <Parancs> **&** # Kiépés után fut tovább

Eredmény a **nohup.out** file-ba, automatikusan

6. Időzített végrehajtás

$ **at** *0030* # Éjjel 1/2 1-kor indul
<Parancs sorozat>
^d
$

Az eredményt levélben kapjuk (mail)
Bonyolultabb, többször használt parancsok file-ból olvashatók
Az időpont megadás többféle lehet

**Sajnos!**
Csak kijelölt felhasználók időzíthetnek



# 4. A PARANCSÉRTELMEZŐ

Legfontosabb parancsértelmezők
- Bourne shell
- C shell
- Korn shell
- Bourne again shell (bash) *Linuxban*

Parancs megadása:
1. Parancssor begépelés, utána végrehajtás
2. File-ban összeállított parancs: shell-script

### Bourne again shell

Speciális karakterek: a parancssor begépelését segítik

| | |
|---|---|
| **return** (új sor) | Befejezi a parancssort és elindítja a végrehajtást |
| **backspace** | Törli a kurzor előtti karaktert |
| ← → | Mozgatja a kurzort |
| ↑ ↓ | Az előző/következő parancsot hozza be |
| **#** | Utána a sor végéig kommentár írható |
| **^d** | File vége |

Régebbi shellek: minden megvan, nehézkesebb
**Sajnos!**
Minden rendszerben ki kell próbálni

# Parancssor feldolgozás

Szavakra bontás
Elválasztó (fehér) karakterek: **space, tab, return**
Elválasztás mindig kell!

Parancs szerkezete
**Parancs-név Opciók Paraméterek**
Opciók:
a parancs viselkedését változtatják,
**-** jellel kezdődnek,
egy karakteresek,
egybeírhatók,
leírási sorrendben paraméterezendők
Paraméterek
file nevek, egyéb információ
opciók paraméterei
Példa
$ **ls -l -a**
$ **ls -al** # Egyenértékűek

# Shell változók

1. Közönséges változók
   Olvashatók és írhatók
2. Pozicionális paraméterek
   Csak olvashatók
   Végrehajtáskor a shell átadja a shell scriptnek



Változók lekérdezése
$ **set**
Felsorolja a változókat ABC sorrendben
Konvenció: rendszerváltozók nagy betűsek
saját változók kis betűsek

Hivatkozás: *$név*

Változók definiálása, beállítása, használata
$ kata=/usr/user/Ubul/szabad # Nincs szóköz
$ **cd** $kata
$ **pwd** # Mert kíváncsiskodunk
*/usr/user/Ubul/szabad*

Több shell futtatása
A shell több példányban (rekurzíven) futtatható
$ **sh** # Itt egy belső shell új folyamatként fut
$ **exit** # visszatérés

A shell script mindíg belső shellben fut

Csak az exportált változók adódnak át

Példa
$ szia=Hello ; **export** *szia*
$ **sh**
$ **echo** $szia
Hello
$ **exit**



# Shell scriptek

Összetett parancsok file-ba írva
Végrehajthatók

Létrehozás
Bármilyen szövegszerkesztővel létrehozzuk a file-t
Engedélyezzük a végrehajtást

Példa:
Számoljuk meg a file-okat a katalógusunkban!
Közvetlen megoldás:
$ **ls l wc -w**

Készítsünk ebből shell scriptet!
$ **echo ′ls l wc -w′** >szamlalo
$ **more** szamlalo # Kíváncsiságból kiíratjuk
ls l wc -w

**Fontos!**
Hosszabb shell sciptet szövegszerkesztővel írunk!

Ezután engedélyezzük a végrehajtást
$ **chmod a+x** szamlalo
$ **ls -l** szamlalo # Megint kíváncsiságból
-rwxrwxrwx 1 *Ubul user 11 Oct 12 9:42 szamlalo*

Használata egyszerűen
$ **szamlalo**
29



Használati alternatívák
$ ./szamlalo # Ha nem keres a saját katalógusban
$ **sh** szamlalo # a shell paramétere
$ **sh** <szamlalo # input átirányítás

Ha csak ilyen primitivet lehetne írni, nem sok értelme volna!

# Pozicionális paraméterek

Jele: $$N$$, ahol $$N$$ 1 és 9 közötti szám

Speciális (automatikus) paraméterek
$0 a hívott script file neve
$* az összes paraméter
$# a paraméterek száma

Példa:
Számoljuk meg egy tetszőleges katalógus file-jait!
$ **echo 'ls $1 l wc -w'** >*okos*; **chmod a+x** *okos*
↑
Itt definiáljuk. Az első paraméter kerül ide

Használata
$ **okos** # aktuális katalógus
*29*
$ **okos ..** # A szülő katalógus
$ **okos** $*kata* # A kata által definiált katalógus

# 5. A *vi* EDITOR

Karaktersorozat szerkesztő. Primitív.

*Nem kell szeretni!*

Háromféle üzemmód:
- beírás
- parancs
- ex (utolsó sor)

Átmenetek:

| | | | |
|---|---|---|---|
| beírás | → parancs | **esc** | |
| parancs | → beírás | **i a A i I o O c C s S R** | |
| | | ↑ | |
| | | ezt ajánljuk | |
| | | ↓ | |
| parancs | → ex | **:** | **/ ? !** |
| ex | → parancs | **esc** | **return** |

Belépés
$ **vi** *szerkesztendo* # Ha nincs file-név: új file
Parancs módba kerülünk

Normál kilépés, beírásból
- **esc**
- **:**
- **wq return** a képernyő alján



## Parancs mód

Formátum
*argumentum parancs szövegobjektum*

Argumentum: sorszám vagy ismétlési szám

Parancs: egy betű

Szövegobjektum: amire a parancs vonatkozik

Szövegobjektum kijelölése: a kurzortól

| | |
|---|---|
| **w W** | szó előre, ill. hátra |
| **( )** | mondat előre, ill. hátra |
| **{ }** | bekezdés előre, ill. hátra |
| duplázás | az aktuális sor |

Főbb parancsok

Kurzor mozgatás, 27 féle

Pozicionálás minta kereséssel
- ***/*** *minta* a kurzortól előre
- **?** *minta* a kurzortól vissza

Törlés
- **x** törli a kurzor alatti karaktert
- **d***objektum* törli a megfelelő objektumot
- **dw** a kurzortól a szó végéig töröl
- **dd** törli a kurrens sort
- **D** a kurzortól a sor végéig töröl

A törölt rész pufferbe kerül.



Másolás
- **y** nem törölve helyez a pufferbe (**d** helyett)

A kurzorral új helyre állva:
- **p** a puffer tartalmát a kurzor után beszúrja

Képernyő frissítés
- **^l** Fontos, mert sokszor nem azt érzi az editor, amit mi látunk!!

Váltás ex módba
- **:** a **:** és a kurzor az utolsó sorra kerül

**Parancsok ex módban**

| | |
|---|---|
| **:w** *nev* | elmenti a file-t *nev* néven |
| **:q** | kilép a *vi*-ból. Ha nem mentettünk javítás után, figyelmeztet és nem hajtja végre |
| **:q!** | mentés nélkül is kilép |
| **:wq** | ment + kilép |

## Ajánlott használat Linuxban

Egyszerű szerkesztéshez beírás módban: *Home, End, PgUp, PgDn* és a kurzormozgató billentyűk.
Törlés: *Del*
Parancs módban:
Sorok törlése: *dd* (ismétléssel)
Mozgatás: *yy* és *p* parancsok



# ÚTMUTATÓ

***a UNIX zárthelyi, illetve házi feladat írásához***

1. Minden önálló sorszámú feladatot úgy kell elképzelni, hogy közvetlenül a bejelentkezés után vagyunk.

   A katalógusban csak a feladathoz szükséges file-ok vannak (vagy üres).

2. A feladaton belül a parancsok egymás után kerülnek végrehajtásra, az előző parancs helyes lefutása utáni állapotban.

3. Ha valamelyik kérdésre nem tudjuk a választ, húzzunk vízszintes vonalkát, de a továbbiakban tekintsük úgy az aktuális állapotot, mintha a parancsot helyesen kiadtuk volna.

4. Írjuk minden parancs elé a $ promptot is.

5. Egyes feladatokhoz több egymás utáni parancs kiadása szükséges.

6. Egyes feladatok többféleképpen is megoldhatók. Minden helyes megoldást elfogadunk.

7. A pontszámok egészekre oszthatók.

***Osztályzatok***

| **zárthelyi** | | **házi feladat** | |
|---|---|---|---|
| 48 – 59 | 2 | 90 – 104 | 2 |
| 60 – 71 | 3 | 105 – 119 | 3 |
| 72 – 83 | 4 | 120 – 134 | 4 |
| 84 – 100 | 5 | 135 – 150 | 5 |

**A zárthelyi írásához semmilyen segédeszköz nem használható!**