

# Számítógépes rendszerek

## 1. gyakorlat

Kapcsolódás a szerverhez

Alap UNIX parancsok

### Kapcsolódás a szerverhez

Szerver: [szamrend.inf.elte.hu](http://szamrend.inf.elte.hu)

Kapcsolódás:

- terminál – PuTTY
- adatmozgatás – SCP (WinSCP)

# WinSCP



Letöltés: winscp.net (ingyenes)

Kapcsolódás a szerverhez:

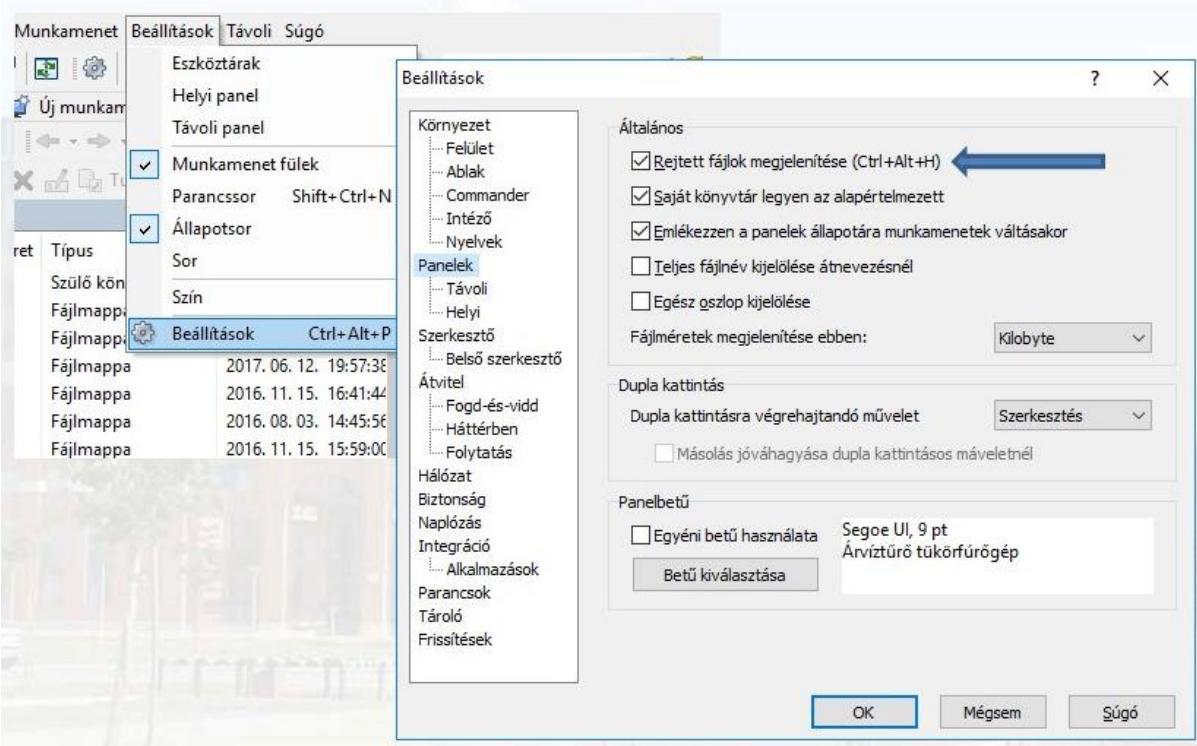
- Kiszolgáló: szamrend.inf.elte.hu
- Felhasználónév: {inf-es azonosító}
- Jelszó: {inf-es jelszó}

Alapértelmezésben a bejelentkezéskor a HOME könyvtárban vagyunk.

# WinSCP



Rejtett fájlok megjelenítése:



## Saját könyvtárak

- HOME: \$HOME vagy ~ – saját fájljaink  
pl. /afs/inf.elte.hu/user/k/kl/klettner
- public: link HOME-ból – weboldalunk  
pl. /h2/public/k/kl/klettner

## public könyvtár



- Elérés böngészőből:  
`people.inf.elte.hu/{inf-es azonosító}`  
Pl. <http://people.inf.elte.hu/klettner>
- üres (nincs benne kezdőlap) → Forbidden
- kezdőlap: index.html → html formátumú fájl

# PuTTY



Letöltés: [www.putty.org](http://www.putty.org) (ingyenes)

Kapcsolódás a szerverhez:

- Host: szamrend.inf.elte.hu
- login as: {inf-es azonosító}
- password: {inf-es jelszó}

Alapértelmezésben a bejelentkezéskor a HOME könyvtárban vagyunk.

## Alap UNIX parancsok

- **man [parancsnév]**  
Manuál olvasása. Kilépés: q  
man -k kulcsszó: a megadott kulcsszót tartalmazó manuálok listázása
- **bash**  
Új shell (parancsértelemező) indítása.
- **logout**  
Kijelentkezés a gépről.
- **exit**  
Kilépés az aktuális shell-ből.

## Alap UNIX parancsok

- **passwd (kpasswd) [felhasználónév]**  
Jelszó megváltoztatása. Ha nem adunk meg felhasználónevet, a saját jelszavunkat változtatjuk meg.
- **who**  
Bejelentkezett felhasználók listázása.
- **whoami**  
Az aktuális felhasználó azonosítóját írja ki.
- **finger**  
Felhasználó adatainak lekérdezése.

## Alap UNIX parancsok

- **date [kapcsolók] [+formátum]**  
Dátum és idő kiíratása és beállítása.  
kapcsolók:
  - u: egyezményes időzóna (UTC)
  - s: beállítás
    - pl. date -s'2010-01-01 12:00:00'
- formátum:
  - + karakterrel kezdődik, részletesen lásd. man
    - pl. date '+%A'

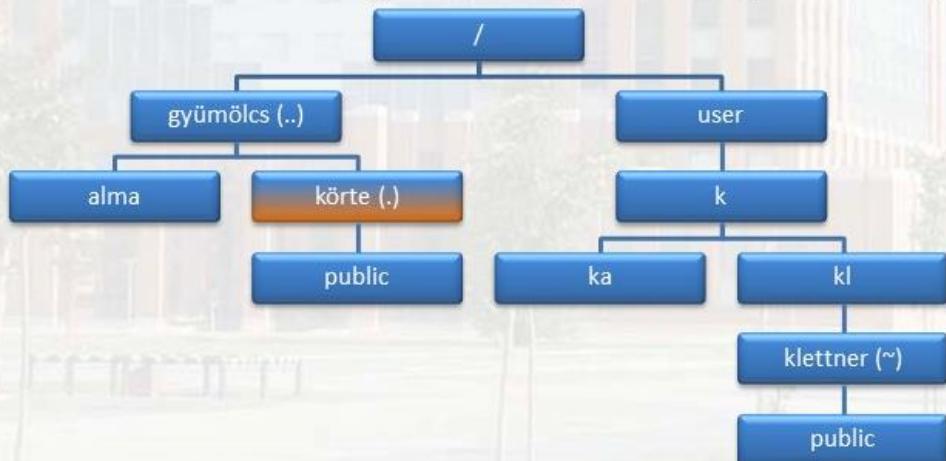
# **Számítógépes rendszerek**

## **2. gyakorlat**

UNIX könyvtárkezelő parancsok

# Könyvtárszerkezet

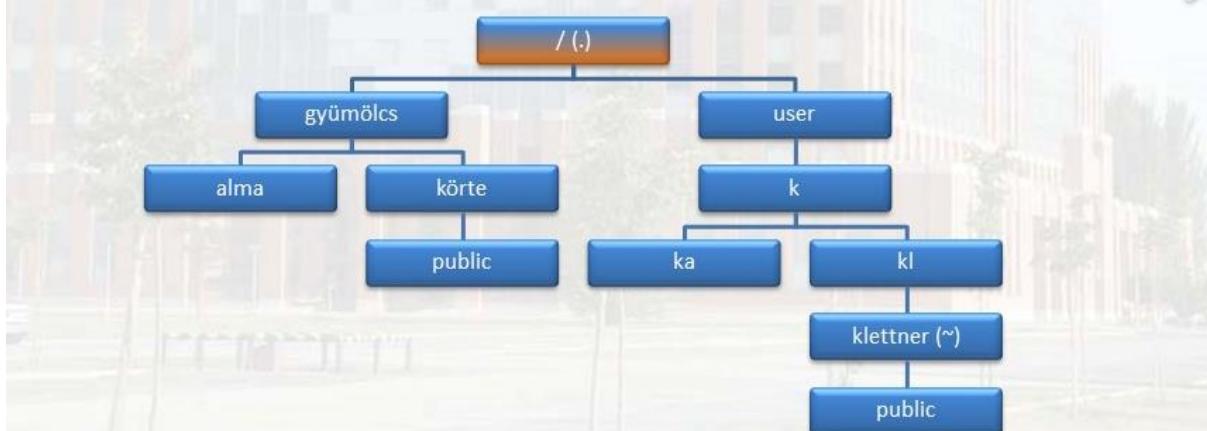
- / root, a könyvtárfa kiindulópontja, legfelső szintje a könyvtárszerkezetnek
- . a munkakönyvtár
- .. a munkakönyvtár szülőkönyvtára
- ~ a felhasználó saját könyvtára (HOME)



# Könyvtárhivatkozások



- A könyvtárhivatkozásokban a / jelet használjuk elválasztójelként.  
Pl. user/k/kl/klettner
- A hivatkozás elemei könyvtárak, kivéve az utolsó elemet, ami lehet fájlhivatkozás is.



# Könyvtárhivatkozások



- Abszolút hivatkozás: A gyökérhez viszonyítva adjuk meg a teljes elérési utat. / jellel kezdődik a hivatkozás.

Pl. /gyümölcs/körte/public

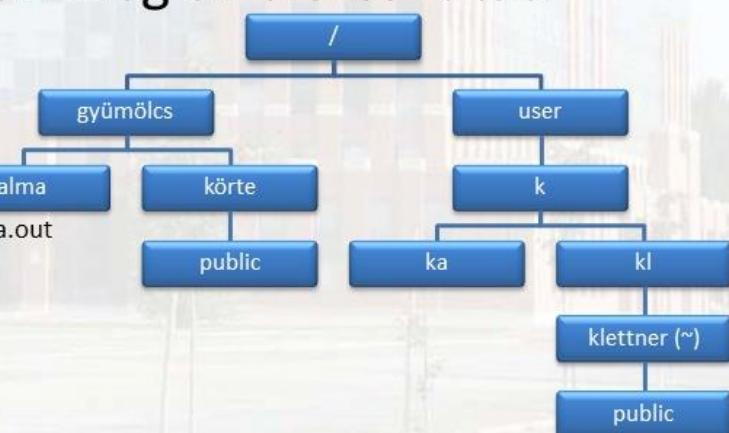
- Relatív hivatkozás: A munkakönyvtárhoz viszonyítva adjuk meg az elérési utat.

Pl. ../../ka

public

~/public

./a.out



## Helyettesítő jelek



- A hivatkozásokban a ? pontosan egy darab tetszőleges karaktert helyettesít.
- A hivatkozásokban a \* tetszőleges számú (akár nulla darab) karaktert helyettesít.
- A hivatkozásokban a [ ] a közé zárt karakterek bármelyikét, de csak azokat helyettesíti.
- A hivatkozásokban a [^] vagy [!] a közé zárt karakterek kivételével, tetszőleges karakterek bármelyikét helyettesíti.

# Könyvtárkezelő parancsok



- **pwd**

A munkakönyvtár teljes elérési útja.

- **ls [kapcsolók] [hivatkozás]**

Könyvtár tartalmának listázása.

Kapcsolók:

-a – minden listáz

-l – részletes lista

-R – rekurzív listázás

-h – "emberi" méretlista

-S – méret szerint csökkenően rendezve

-t – utolsó módosítás időpontja szerint rend.

-r – fordított sorrendben rendezve

# Könyvtárkezelő parancsok



- **cd [elérési út]**

A munkakönyvtár a paraméterként megadott könyvtár lesz.

- **mkdir [kapcsolók] könyvtárnév**

A megadott nevű könyvtár létrehozása.

- **rmdir [kapcsolók] könyvtárnév**

A megadott nevű üres könyvtár törlése.

# Könyvtárkezelő parancsok



- **cp [kapcsolók] forrás cél**

Másolás.

Kapcsolók:

- r – rekurzív (könyvtár)
- i – kérdezzen rá a felülírásokra
- u – csak a régebben módosított fájlokat írja felül
- f – ne álljon le, ha valamelyik fájlt nem lehet megnyitni vagy létrehozni

# Könyvtárkezelő parancsok



- **mv [kapcsolók] forrás cél**

Áthelyezés, átnevezés.

Kapcsolók:

- i – kérdezzen rá a felülírásokra
- u – csak a régebben módosított fájlokat írja felül
- f – ne álljon le, ha valamelyik fájlt nem lehet megnyitni vagy létrehozni

# Könyvtárkezelő parancsok



- **rm [kapcsolók] törlendők**

Törlés.

Kapcsolók:

- r – rekurzív (könyvtár)
- v – visszajelzés a művelet sikerességéről
- i – kérdezzen rá az egyes elemekre
- f – ne álljon le, ha valamelyik fájlt nem lehet törölni

# Könyvtárkezelő parancsok



- **touch fájlnév**

Üres fájl létrehozása, vagy meglévő fájl utolsó módosítási idejének beállítása az aktuális időpontra.

- **In célpont linknév [kapcsolók]**

A célontra mutató link létrehozása.

Kapcsolók:

- s – szimbolikus link létrehozása

# Részletes lista (ls -l) 1. oszlop

- első karakter a bejegyzés típusa:
  - szabályos fájl
  - d könyvtár
  - l hivatkozás
  - ...
- következő 3x3 karakter az elérési jogok:
  - [2-4]: tulajdonos (user; u)
  - [5-7]: tulajdonos csoportja (group; g)
  - [8-10]: mindenki más (others; o)Minden csoport három karakter: rwx
  - r: olvasási jog
  - w: írási jog
  - x: futtatási jog

# Könyvtárkezelő parancsok



- **chmod [kapcsolók] jog hivatkozás**

Elérési jogok megváltoztatása.

Kapcsolók:

- v – visszajelzés a művelet sikerességéről
- R – rekurzív

**jog:** lehet módosítás vagy beállítás

Jogok módosítása:

[felhasználó] + |- jogosultság

felhasználó: u és/vagy g és/vagy o

+ jogosultság megadása, - elvétele

jogosultság: r és/vagy w és/vagy x

# Könyvtárkezelő parancsok



Jogok megadása:

Jogokat szimbolizáló bitek:

Felhasználó	u	u	u	g	g	g	o	o	o
Jog	r	w	x	r	w	x	r	w	x
Oktális helyiérték 8 <sup>x</sup>	2	2	2	1	1	1	0	0	0
Oktális bitérték	4	2	1	4	2	1	4	2	1

A beállítandó jogértéket oktális számként adjuk meg.

## Könyvtárkezelő parancsok



- **tar feladat [kapcsolók] [fájlok]**

Archiválás és archívumból való visszaállítás.  
feladat:

- c – új archívum létrehozása és fájlok  
hozzáadása
- r – fájlok hozzáadása archívumhoz
- t – archívumfájl tartalmának listázása
- x – visszaállítás archívumból

kapcsolók:

- f – archívumfájl megadása: -f archívumfájl
- j – bzip2 tömörítés (.bz2)
- z – gzip tömörítés (.gz)

## Könyvtárkezelő parancsok



- **zip [kapcsolók] archívum [tömörítendők]**

Fájlok tömörítése zip archívumba.

Kapcsolók:

- v – visszajelzés a művelet sikerességéről
- r – rekurzív

pl.: zip csapatok fradi.txt honved.txt  
zip -r ~/minden \*

# Könyvtárkezelő parancsok

- **unzip [kapcsolók] archívum**  
zip archívum kitömörítése vagy listázása.  
Kapcsolók:
  - l – archívum tartalmának listázása
  - v – archívum tartalmának részletes listázásapl.:    **unzip -l minden**  
         **unzip -v minden**  
         **unzip minden**

## Számítógépes rendszerek 3. gyakorlat

UNIX fájlkezelő parancsok

# Fájlkezelő parancsok



- **cat [kapcsolók] fájlok**

A fájlokat a megadás sorrendjében egymás után fűzve kilistázza a képernyőre.

Kapcsolók:

-n – beszámorra a sorokat

-b – beszámorra a nem üres sorokat

pl.: cat -n csapatok.txt fradi.txt honved.txt  
cat -b csapatok.txt fradi.txt honved.txt

# Fájlkezelő parancsok

- **more fájl**

Kilistázza a fájlt a képernyőre lapozással.

Lapozás:

- enter – soronként

- szóköz – laponként (képernyőnként)

- q – kilépés

# Fájlkezelő parancsok



- **less fájl**

Kilistázza a fájlt a képernyőre kétirányú lapozással.

Lapozás:

- le és fel kurzormozgató billentyűk
  - soronként le és fel
- PageDown és PageUp billentyűk
  - laponként (képernyőnként) le és fel
- q
  - kilépés

# Fájlkezelő parancsok



- **head [kapcsolók] fájl**

A fájl elejét listázza ki.

Kapcsolók:

- n{sorok száma}
  - a megadott számú sor listázása

- c{karakterek száma}
  - a megadott számú karakter listázása

Ha a számosságot negatív előjellel adjuk meg, a végéről hagy el annyit, amennyit megadtunk.

pl.: head szavak.txt

head -c5 szavak.txt

head -n5 szavak.txt

head -n-5 szavak.txt

# Fájlkezelő parancsok



- **tail [kapcsolók] fájl**

A fájl végét listázza ki.

Kapcsolók:

- n{sorok száma}

- a megadott számú sor listázása

- c{karakterek száma}

- a megadott számú karakter listázása

Ha a számosságot pozitív előjellel adjuk meg, annyiadik pozíciótól listáz, amennyit megadtunk.

pl.: tail szavak.txt

tail -c6 szavak.txt

tail -n6 szavak.txt

tail -n+18 szavak.txt

# Bemeneti és kimeneti eszközök

Alapértelmezett:

- 0 (stdin): bemenet (billentyűzet)
- 1 (stdout): kimenet (monitor)
- 2 (stderr): hibakimenet (monitor)

Hivatkozás célként: &egységszám

Egyéb:

/dev/null: "kuka"

# Bemeneti és kimeneti eszközök



Sok parancs esetében, ha nem adunk meg forrás fájlnevet, akkor az alapértelmezett input eszkösről (stdin) várja a feldolgozandó adatokat. Az ilyen parancsok esetében (pl. cat, head, tail, ...) ezt a man külön említi.

Amennyiben a parancs jellegéből adódóan nem következik, meddig fogad adatokat, úgy az adatbevitelt a [ctrl]+[d] billentyűkombináció zárja le.

## Input-Output átirányítás



>cél Kimenetet cél-ba irányítja. Ha cél egy fájl, akkor mindenkor létrehozza.

>>cél Kimenetet cél-ba irányítja. Ha cél egy létező fájl, akkor hozzáfűzi, ha egy nem létező fájl, akkor létrehozza.

<forrás forrás tartalmát bemenetre irányítja.

<< „here” dokumentum.

# here

Használat:

```
<<határoló  
sorok  
határoló
```

## Könyvtárkezelő parancsok

- **find [útvonal] [kapcsolók] [kifejezések]**

A megadott kifejezéseknek megfelelő könyvtárbejegyzések rekurzív keresése.

Kapcsolók:

-maxdepth {szint} – keresési mélység

Kifejezések:

-empty – üres fájlok és könyvtárak

-name "{minta}" – a mintának megfelelő bejegyzések. Kis- és nagybetűket megkülönbözteti.

-iname "{minta}" – a mintának megfelelő bejegyzések. Kis- és nagybetűket nem különbözteti meg.

# **find**



Kifejezések:

-exec {parancs} \; – minden találat esetén végrehajtja a parancsot úgy, hogy annak egyik paramétere a találat.

parancs: bármilyen parancs lehet, amit a shell-be írhatunk. Ahová a parancsba a találatot szeretnénk illeszteni, oda {} jelet kell írni.

# **find**

(

Kifejezések:

-size {méret} – a méretnek megfelelő fájlok.

méret: [{előjel}]{mennyiség}[{mértékegység}]

előjel: + (nagyobb mint)

- (kisebb mint)

mértékegység: c (byte)

k (kilobyte)

M (Megabyte)

G (Gigabyte)

-type {típus} – a típusnak megfelelő bejegyzések.

típus: f (szabályos fájl)

d (könyvtár)

l (szimbolikus link)

# find



Kifejezések:

-mtime [{előjel}]{nap} – az utoljára a megadott napja módosult bejegyzések.

nap: aktuális időponttól számított 24 óra.

*előjel:*

- *nincs megadva*: az utolsó módosítás több, mint nap\*24, és kevesebb, mint (nap+1)\*24 órája történt.

- *pozitív (+)*: az utolsó módosítás több, mint (nap+1)\*24 órája történt.

- *negatív (-)*: az utolsó módosítás kevesebb, mint nap\*24 órája történt.

# **find**



Kifejezések:

- not {kifejezés} – Tagadás. A következő kifejezés igazságértékét megfordítja.
- {kifejezés1} -o {kifejezés2} – Vagy kapcsolat. A feltétel igaz, ha a két kifejezés legalább egyike igaz.
- Zárójelezés: Kettőnél több kifejezés esetén a logikai kapcsolatok kiértékelését zárójelezéssel szabályozhatjuk vagy egyértelműsíthetjük. Pl.  
`find \(-size +1k -name "*e" \) -o -name "o*"`  
`find -size +1k \(-name "*e" -o -name "o*" \)`

## **Számítógépes rendszerek**

### **4. gyakorlat**

Szűrők, csővezeték

# Szűrők



Azt a UNIX parancsot (programot), mely, ha nem adjuk meg annak az állománynak a nevét, amit be kell olvasnia és a tartalmát fel kell dolgozna, akkor a standard inputról várja a feldolgozandó adatokat, továbbá, ha nem adunk meg kimeneti fájlnevet, akkor a standard outputra ír, szűrőnek nevezük.

## Csővezeték (Pipe)

- **parancs1 | parancs2**

parancs 1 kimenete parancs 2 bemenetére irányítódik át (vagyis parancs1 kimenete parancs2 bemeneteként jelenik meg).

## Szöveg kiíratása



### echo [kapcsolók] szöveg

Kiírja a standard kímentre a megadott szöveget.

Kapcsolók:

- n – ne írjon sorvége jelet a szöveg után
- e – escape szekvenciák használata

escape szekvenciák:

- \n sorvége jel
- \r kurzor vissza jel
- \t tabulátor jel
- \a csengő (hangadás) jel

## Sorok manipulációja



### • sort [kapcsolók] [fájlok]

Rendezve írja a megadott összes fájl vagy a standard bemenet sorait a standard kímenetre.

Kapcsolók:

- r: fordított sorrendben
- u: többször előforduló sorokat csak egyszer írja ki

# Szöveg statisztika



- **wc [kapcsolók] [fájl]**

Kiírja a standard kimenetre a fájl vagy a standard bemenet sorainak, szavainak és byte-jainak számát végül a nevét.

Kapcsolók:

- l: a sorok számát írja ki,
- c: a byte-ok számát írja ki,
- m: a karakterek számát írja ki,
- w: a szavak számát írja ki,

# Sorok manipulációja



- **tr [kapcsolók] halmaz1 [halmaz2]**

Karaktereket cserél, töröl, vagy tömörít a bemenet soraiban.

Kapcsolók:

- d: a halmaz1-ben szereplő karakterek törlése,
- s: a halmaz1-ben szereplő karakterek tömörítése,  
halmazok megadása: aposztrófok között

# Sorok manipulációja



- **cut [kapcsolók] [fájl]**

Kivágja a megadott fájl vagy a standard bemenet soraiból az adott mezőt, karaktert. A mezőket listában adhatjuk meg.

Kapcsolók:

- c lista: karakterek kivágása,
- f lista: mezők kivágása,
- d 'elválasztó': mezőelválasztó megadása.

# Sorok manipulációja

- **cut [kapcsolók] [fájl]**

A listában állhat

- egy szám pl. "cut -c2" vagy
- egy intervallum pl. "cut -c3-5",  
"cut -c-6",  
"cut -c4-" vagy
- ezek felsorolása, pl. "cut -c3,5,6",  
"cut -c3,5,8-11"

# Sorok manipulációja

(

- **grep [kapcsolók] minta [fájl]**

Megkeresi a mintának megfelelő sorokat a megadott fájl(ok) vagy a standard bemenet soraiban.

A **minta** egy reguláris kifejezés.

Kapcsolók:

-s:	nem ad hibaüzenetet,
-l:	csak a fájlneveket írja ki,
-r:	rekurzív keresés,
-w:	teljes szóra keresés.

# Reguláris kifejezések



- Előadás anyaga:
  - 3. előadás 20-27 dia
- Irodalom, részletes leírások:
  - [https://hu.wikipedia.org/wiki/Reguláris\\_kifejezés](https://hu.wikipedia.org/wiki/Regul%C3%A1ris_kifejez%C3%A9s)
  - <http://people.inf.elte.hu/csa/regkif.htm>
- Webes tesztelő környezet:
  - <https://regexr.com/>

## Számítógépes rendszerek

### 5. gyakorlat

Shell Script I.

Folyamatvezérlés

# Folyamatok

Unix (Linux):

- Többfeladatos (multitasking)
- Időosztásos (time sharing)
- PID (process identifier)

## Párhuzamosan futó folyamatok kezelése

- Lekérdezés
- Szüneteltetés
- Leállítás
- Időzítés
- Priorizálás
- Előtérbe vagy háttérbe küldés

# **Előtérben és háttérben futtatás**

## **parancs**

Parancs indítása előtérben  
beavatkozási lehetőségek:

- [ctrl]+[z] szüneteltet (felfüggeszt)
- [ctrl]+[c] leállít (megszakít)

## **parancs &**

Parancs indítása háttérben

## **PID és JobID**

PID: operációs rendszeren belüli egyedi  
azonosító

JobID: parancs értelmezőn belüli egyedi  
azonosító

# Futó folyamatok lekérdezése

## ps [kapcsolók]

Futó folyamatok listázása

Kapcsolók:

- e – környezet mutatása
- f – fa szerkezetben mutatja
- u – mutatja a futtató felhasználót és az indítás idejét is
- a – a terminálon futó összes folyamat
- x – a felhasználó összes folyamata
- ax – a kiszolgálón futó összes folyamat
- l – mutatja a prioritásokat is

# Futó folyamatok lekérdezése

## jobs [kapcsolók]

Háttérben futó folyamatok listázása.

Kapcsolók:

- l – a PID-eket is listázza
- n – csak a terminált folyamatokat listázza
- s – csak a nem futó folyamatokat listázza

# Futó folyamatok lekérdezése

**top**

Futó folyamatok automatikusan frissülő monitorozása.

Billentyűparancsok:

q – kilépés

u – szűrés felhasználóra

A – alternatív megjelenítések

d – frissítési intervallum beállítása

z – színes

## Előtérbe vagy háttérbe küldés

**fg [JobID]**

Folyamat előtérbe küldése

**bg [JobID]**

Folyamat háttérbe küldése

# No hangup

(2)

## **nohup parancs**

Folyamat kijelentkezéskori megszakadásának megakadályozása.

A parancs kimenete hozzáfűződik a nohup.out fájlhoz.

# Szignálok

(1)

A Unix rendszer a folyamatok vezérlését a folyamatoknak küldött ún. szignálok (signals) segítségével végzi.

# Fontosabb szignálok



- **TERM, 15:** TERM szignál küldése, ez a normál leállítás. A programnak nem kell kötelezően leállnia.
- **KILL, 9:** KILL szignál. A program kilövése, mindenkorban leállítja a programot, hasznos ha esetleg egy lefagyott program nem reagálna a TERM szignálra.
- **INT, 2:** INT szignál ([CTRL]+[C]). A program félbeszakítása, mely rendszerint a program elhalálozását vonja maga után.
- **STOP, 19:** STOP szignál ([CTRL]+[Z]). A program felfüggessztése.

## Szignálok küldése

**kill [kapcsolók] PID**

**kill [kapcsolók] %JobID**

Szignál küldése a megadott azonosítójú folyamatnak.

Kapcsolók:

-{szignál} – a megadott szignált küldi

# Szignálok küldése

## killall [kapcsolók] név

Szignál küldése az összes megadott nevű folyamatnak.

Kapcsolók:

-s {szignál} – a megadott szignált küldi

## Csapdák (eseménykezelés)

### trap 'parancs' szignál

A megadott szignál érkezésekor az aposztrófok közé írt parancsot hajtja végre.

# Folyamat erőforrás-felhasználás



## time parancs

A megadott parancs futtatását követően kiírja, hogy az adott parancs futtatása mennyi erőforrást vett igénybe.

## strace parancs

### strace –pPID

A megadott parancs futtatása közben kiírja, az adott parancs összesrendszerhívását.  
Hibakereséshez használatos.

# Számítógépes rendszerek

## 6. gyakorlat

Shell Script II.

Változók

Vezérlési szerkezetek

## Speciális karakterek

- \ A backslash-t követő karakter elveszíti módosító hatását. pl. \' \" \\ \\$
- \$ Változó neve és értéke közötti megkülönböztetés. (Id. később)
- ' Speciális karakterek vagy szavak speciális jellegét feloldja.
- " "Speciális karakterek vagy szavak speciális jellegét feloldja, kivéve \\$ ''
- ` ` A közé írt jelsort végrehajtja, de a kimenet végére nem újsor, hanem szóköz kerül.

## .profile

- HOME könyvtárba.
- minden belépéskor lefut.

# Scriptek

- **sh:** parancsértelmező
- **Forrás fájl:** egyszerű szöveges fájl futtatási joggal
- **Végrehajtás:** soronként
- **Parancsok elválasztása:** sorvég vagy ;
- **Komment:** # (sor végéig)
- **Első sor:** #!/bin/sh (ki értelmezze)
- **Futtatás:** ./parancsnév

## Változók

- **env**  
Kilistázza a környezeti változókat és tartalmukat, az általunk létrehozott normál változókat nem írja ki.
- **set**  
Kilistázza a környezeti változókat, majd az általunk létrehozott normál változókat.

# Változók

- **azonosító=érték**  
Változó létrehozása.
- **\$azonosító**  
Változó tartalmának elérése.
- **unset azonosító**  
Változó törlése.
- **export azonosító**  
Saját változó környezeti változová tétele.

## Környezeti változók



A rendszerről nyújtanak hasznos információkat:

- **HOME**: a felhasználó saját könyvtárának abszolút elérési címe
- **PWD**: az aktuális könyvtár abszolút elérési címe
- **PATH**: a rendszerparancsok elérési helyeinek listája, kettősponttal elválasztva
- **USER**: a felhasználónévünk
- **RANDOM**: egy véletlenszerű számot ad 0 és 32767 között
- **PS1**: parancssori promptot meghatározó változó

# Script paraméterek

**\$0** parancs neve

**\$1** első paraméter

**\$2** második paraméter

...

**\$9** kilencedik paraméter

**\$#** paraméterek száma

**\$\*** összes paraméter

## Kifejezés kiértékelése

- **expr kifejezés**

A kifejezés értékét adja eredményül.

## Adat beolvasása



- **read változó**

Beolvas a standard bemenetről egy sor adatot (az első ENTER leütésig) és az adott nevű változóban tárolja.

Ha nem adunk meg változónevet, akkor a \$REPLY változóba kerül az eredmény.

## Számsorozat előállítása



- **seq [kapcsolók] [kezdőérték] [lépés] végérték**

Kiír a standard kimenetre egy **kezdőértékkel** kezdődő **végértékig** tartó számsorozatot a **lépés** értékével haladva a **végérték** felé.

Kapcsolók:

-s'elválasztás' – elválasztó karaktersorozat megadása

# Elágazások



- **test** logikai\_kifejezés vagy

[ logikai\_kifejezés ]

A logikai kifejezés eredményét a \$? Változóba helyezi. (1: hamis; 0: igaz)

**logikai\_kifejezés:** kif1 op kif2 ...

- Számok összehasonlítása:

kif1. -lt kif2. kisebb mint (less than)

kif1. -gt kif2. nagyobb mint (greater than)

kif1. -le kif2. kisebb vagy egyenlő (less or equal)

kif1. -ge kif2. nagyobb vagy egyenlő (greater or equal)

kif1. -eq kif2. egyenlő (equal)

kif1. -ne kif2. nem egyenlő (not equal)

# Elágazások

- Szövegek összehasonlítása:

kif1. = kif2.      megegyező

kif1. != kif2.      eltérő

-n kif.      nem nulla hosszúságú

-z kif.      nulla hosszúságú

- Összetett kifejezések:

! kif.      tagadás

kif1. -a kif2.      és (and)

kif1. -o kif2.      vagy (or)

# Elágazások



```
if logikai_érték
then
# ha igaz (0) a logikai_érték, akkor ide lép be
elif logikai_érték_2
then
# ide akkor lép be ha a 2. logikai érték igaz
...
else
# ha egyik logikai_érték sem igaz, akkor ide lép
fi
```

# Elágazások



**case** kifejezés **in**

**érték1)**

# kifejezés=érték1

**;;**

**érték2)**

# kifejezés=érték2

**;;**

**...**

**\***)

# kifejezés semelyik eddigi értékkel nem egyezett

**;;**

**esac**

# Ciklusok



**for** ciklusváltozó **in** lista

**do**

# ciklusmag

**done**

Ciklusváltozó rendre felveszi a listában szereplő értékeket, és minden érték esetén végrehajtja a ciklusmagot.

# Ciklusok

**while** logikai\_érték

**do**

# ciklusmag

**done**

Ismétli ciklusmag végrehajtását, míg logikai\_érték igaz. Tehát logikai\_érték benmaradási feltétel.

## Ciklusok



**until** logikai\_érték

**do**

# ciklusmag

**done**

Ismétli ciklusmag végrehajtását, míg logikai\_érték hamis. Tehát logikai\_érték kilépési feltétel.

# Számítógépes rendszerek

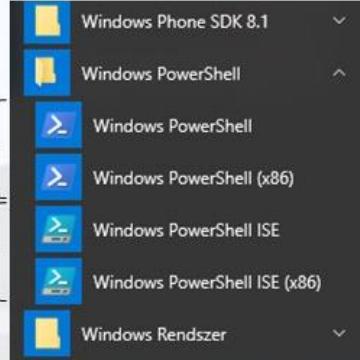
## 7. gyakorlat

PowerShell

Alapok

### A PowerShell indítása



- Parancssor, karakteres
  - Grafikus
  - Első használat (rendszerelőként):
    - Help frissítése: Update-Help
- 
- The screenshot shows the Windows Start Menu with several PowerShell-related entries:
  - Windows Phone SDK 8.1
  - Windows PowerShell
  - Windows PowerShell
  - Windows PowerShell (x86)
  - Windows PowerShell ISE
  - Windows PowerShell ISE (x86)
  - Windows Rendszer

# A PowerShell jellemzői



- Nem kis- nagybetű érzékeny.
- Ékezetes karakterek használata is lehetséges.
- A legtöbb parancsnak van aliasa. Ezek a UNIX- és DOS-beli megfelelők.
- Parancsok szerkezete: Ige-Főnév.
- Utasítások szerkezete:  
`<parancs> <paraméterek> <argumentumok>`
- Csővezeték használatata ugyanúgy történik, ahogy UNIX-ban.
- Csővezeték eredménye objektum.
- Az utasítások objektumokon dolgoznak.

## Segítségkérés

- **Get-Help <parancsnév>**  
pl.
  - Get-Help ExecutionPolicy
  - Get-Help Set-ExecutionPolicy
  - Get-Help ls
  - Get-Help dir
  - Get-Help Get-ChildItem -Examples

# Könyvtártartalom listázása



- **Get-ChildItem**

- Recurse – rekurzív listázás
- Path <elérési út> – listázandó könyvtár megdása
- Force – rejtett fájlokat is listázza
- Directory – csak könyvtárakat listáz
- File – csak fájlokat listáz

## Könyvtárbejegyzés attributumok

**darhs!**

d: directory (mappa)

a: archiválandó

r: read only (csak olvasható)

h: hidden (rejtett)

s: system (rendszer)

l: link (parancsikon)

## Munkakönyvtár helyének módosítása



- **Set-Location <munkakönyvtár>**  
Munkakönyvtár helyének beállítása.
- **Push-Location <munkakönyvtár>**  
Munkakönyvtár helyének beállítása, előző munkakönyvtár helyének verembe mentése.
- **Pop-Location**  
Munkakönyvtár helyének veremből való visszaállítása.

## Könyvtárbejegyzés létrehozása

- **New-Item**
  - Name – a bejegyzés neve
  - Value – kezdeti tartalom megadása
  - ItemType – a bejegyzés típusa:
    - File
    - Directory
    - SymbolicLink
    - HardLink
    - ...

# Másolás

- **Copy-Item**

- Destination – cél megadása
- Recurse – rekurzív másolás

# Átnevezés

- **Rename-Item**

- NewName – új név megadása

# Áthelyezés

- **Move-Item**

- Destinatiton – cél megadása

# Törlés



- **Remove-Item**

- Include – amit töröljön
- Exclude – amit kihagyjon Include-ból
- Force – védett fájlokat (rejtett, rendszer) is

## Könyvtárbejegyzés létezésének ellenőrzése

- **Test-Path**

Értéke igaz, ha a megadott könyvtárbejegyzés létezik.

## Fájl tartalmának megadása



- **Set-Content**

- Value – új tartalom megadása
- NoNewline – nem tesz a végére sora melést

# Fájl tartalmának kiegészítése



- **Add-Content**

- Value – új tartalom megadása
- NoNewline – nem tesz a végére soremelést

# Fájl tartalmának listázása

- **Get-Content**

- TotalCount – listázandó sorok számának megadása a fájl elejétől
- Tail – listázandó sorok számának megadása a fájl végétől

# Kiírás a képernyőre



- **Write-Host**

- Separator – listaelemeket elválasztó karakterszorozat megadása
- Tail – listázandó sorok számának megadása a fájl végétől
- NoNewline – nem tesz a végére soraelést
- ForegroundColor – szövegszín megadása
- BackgroundColor – háttérszín megadása

## Kiírás

- **Write-Output**

# Dátum



- **Get-Date**

-Year	– év
-Month	– hónap
-Day	– nap
-Hour	– óra
-Minute	– perc
-Second	– másodperc
-Millisecond	– ezredmásodperc
-DayOfWeek	– hétfő napja
-DayOfYear	– nap sorszáma éven belül

# Szűrők

- **Format-List**  
Formázott lista előállítása
- **Format-Table**  
Formázott tábla előállítása
- **Sort-Object**  
Rendezés
- **Where-Object**  
Szűrés
- **Select-Object**  
Kiválogatás

# Szűrők

- **Select-String**

Kiválogatás minta alapján

- **Measure-Object**

Statisztika

- **Out-File**

Kimenet fájlba írása

- **Get-Member**

Az osztály adatai (metódusai)

# Változók



Azonosító:           \$<azonosító>

Értékadás:           <változó> = <érték>

Típus megadása: [típus] <változó> = <érték>

Tartalom törlése: Clear-Variable <változó>

Változó törlése: Remove-Variable <változó>

Változó lista:       Get-Variable

Tömbök:              mint C++

Tömb bővítése: <azonosító>=<azonosító>+<elem>

Associatív tömb: @{<index1>=<érték1>;...}

# Típus deklaráció és kényszerítés

[string]

[char]

[byte]

[int]

[long]

[bool]

[single]

[double]

[DateTime]

## Típuskonverziók

- <érték>.To<típus>()

Konvertálja <érték>-et <típus> típusra, amennyiben lehetséges.

Pl.:

    \$szoveg.ToChar.Array()  
    \$szam.ToString()

# Operátorok, függvények



<https://technet.microsoft.com/en-us/library/hh848303.aspx>

<https://ss64.com/ps/syntax-compare.html>

<http://www.madwithpowershell.com/2013/10/math-in-powershell.html>

Pl.

- [math]::pi
- [math]::sin(2)
- [math]::pow(2,3)
- [math]::sqrt(2)

# Szöveg műveletek



- <string>.Split('<mezőhatároló>')[<sorszám>]  
Mező kivágása.
- <string>.Substring(<első>,<darab>)  
Szövegrész kivágása.
- <string> -**replace** <régi>,<új>  
Csere.  
<string> összes <régi> karaktersorozatát  
lecséréli <új> karaktersorozatra, és az így  
kapott új stringet adja át a kimeneten.

# PowerShell Script

- Script futtatási jog megadása:
  - rendszergazdaként indítani
  - set-executionpolicy remotesigned
- Kiterjesztés: .ps1
- Futtatás:
  - PS ISE: RunScript vagy [F5]
  - PS: .\<script>
  - DOS: powershell .\<script>

# Script paraméterek



**\$args:** összes paraméter

**\$args[n]:** n. paraméter

**\$args.Count** vagy

**\$args.Length:** paraméterek száma

**\$MyInvocation.MyCommand.Name:** Script neve

**Kötelező paraméter:**

param

(

[Parameter(Mandatory=\$true)][<típus>]<változó>

)

## Érték beolvasása billentyűzetről

- **Read-Host**

**Beolvas adatot a billentyűzetről.**

Értéke a beolvasott adat lesz.

-Prompt – Kérdés beolvasás előtt, végén :

## Hibaüzenetek megjelenésének (2) tiltása

- **\$ErrorActionPreference="SilentlyContinue"**  
A script további soraira vonatkozik.
- **-ErrorAction SilentlyContinue**  
Az utasításra vonatkozóan, amennyiben támogatja.

# Számítógépes rendszerek 8. gyakorlat

PowerShell

Vezérlési szerkezetek

# **Lista (tömb) bejárása**

```
foreach ($ciklusváltozó in lista)
```

```
{
```

```
    ciklusmag
```

```
}
```

## **if elágazás**

```
if (feltétel)
```

```
{
```

```
    sorok
```

```
}
```

```
elseif (feltétel2)
```

```
{
```

```
    sorok
```

```
}
```

```
else
```

```
{
```

```
    sorok
```

```
}
```

# **switch elágazás**

```
switch (változó)
```

```
{
```

```
    érték1 {tevékenység1}
```

```
    érték2 {tevékenység2}
```

```
...
```

```
    default {tevékenység}
```

```
}
```

# Ciklusok

While (ciklusfeltétel)

{

    ciklusmag

}

Do

{

    ciklusmag

} While (ciklusfeltétel)

For(kezdőértékkadás; ciklusfeltétel; léptetés)

{

    ciklusmag

}

## Csővezeték



csővezeték enumerátor:                   \$input

elemszám:                                       @(\$input).Count

eredeti állapot helyreállítása: \$input.Reset()