



Számítógépes alapismeretek

Dr. Illés Zoltán,
Mail: illes@inf.elte.hu

Visszatekintés



- Számítógépek, jelek, információk, tárolásuk
- Számábrázolás
 - Fixpontos, lebegőpontos számábrázolás
- Kódolás
 - ASCII, UTF-8 stb kódtáblák
- Felépítés, operációs rendszer szerep
- Kliens-Szerver különbségek
- Grafikus, karakteres kapcsolat
- Fájltrendszer, alapvető parancsok

Mi jön ma?



- További parancsok
- Folyamatok előtérben, háttérben
- I/O átirányítás
- Szűrők
- Irány programozni!
- Reguláris kifejezések

További parancsok



- Parancsok előtérben: normál parancskiadás
- Parancs futtatás a háttérben: & karakter a parancs végén
 - & megnevezései: ampersand, and, és jel, at jel
- sleep 15 # 15 másodperc várakozás, közben nincs parancskiadási lehetőség
 - CTRL+Z stop jelet küld a folyamatnak
- sleep 15 & # a várakozó folyamat a háttérben fut, közben újabb parancsot adhatunk ki
 - Eredményül kapjuk: [1] 28321, az első a „jobszám”, a második a pid
- jobs – listázza a futó parancsainkat(„jobjainkat”)
- ps – process status parancs, láthatjuk a háttérben futó sleep-et is

Folyamatok háttérben



- Mi van ha egy háttérfolyamatot előtérbe akarunk tenni: `fg %1`
- Egy előtérben futó folyamatot ha leállítottunk(`ctrl-z`) a háttérben is folytathatjuk: `bg %1`
- Jelzés küldés: `kill -signál folyamat`
 - A folyamatot vagy a job sorszámával (kell a % jel) vagy a processz azonosítóval jelöljük.

```
szamalap.inf.elte.hu - PuTTY
illes@valerie:~$
illes@valerie:~$ sleep 25 &
[1] 28345
illes@valerie:~$ kill -SIGSTOP %1
illes@valerie:~$ ps
  PID TTY          TIME CMD
 27724 pts/4    00:00:00 bash
 28345 pts/4    00:00:00 sleep
 28351 pts/4    00:00:00 ps

[1]+  Megállítva                  sleep 25
illes@valerie:~$ bg %1
[1]+ sleep 25 &
illes@valerie:~$
[1]+  Kész                        sleep 25
illes@valerie:~$
```

Még többet a folyamatokról!



- top parancs: látjuk a futó folyamatok adatait, globális rendszer állapotot!
 - Mindenki egyenrangú?
 - Unix, Linux rendszerek prioritása: -20-tól 19-ig (40 szint)
 - A kisebb szám a nagyobb prioritás!
 - nice parancs, módosítja az indítandó parancs prioritását.
 - Alapból a prioritás 0, top vagy ps -l parancs NI oszlop!
 - nice -n 5 sleep 20& # 5 lesz az új prioritás, 0-hoz 5-öt ad
 - nice -n -10 sleep 20& # növelni root joggal lehet!!!!

Idézőjelek



- Parancssorban legális karakterek: .,_,-,számok, normál karakterek
- Idézőjelek: ',",\ – módosítják a klasszikus karakter értelmezést
- 'Aposztrof karakterek között megszűnik minden speciális \ \$ % stb. hatás'
 - pl: echo 'alma\fa' #alma\fa
- "Macskakörmön belül a \$, a \, a ` és a ' karakterek hatása megmarad"
 - Példa: a=fradi; echo "hajrá \$a !" # hajrá fradi
- \x karakter: módosítja x eredeti jelentését
 - Példa: fa=virág; echo alma\fa #alma\$fa
 - echo alma\$fa #almavirág

Kimenet, Bemenet, átirányítások



- Kimenet, bemeneti eszközök
 - stdin (0) - billentyűzet, alapértelmezett bemenet
 - stdout (1) - monitor, alapértelmezett kimenet
 - stderr (2) – monitor, alapértelmezett hibakimenet
- Átirányítás
 - Kimenet: > jel segítségével
 - Pl: echo alma barack szilva >gyumolcs
 - echo alma >&2 #hogya ne a 2 nevű állomány legyen
 - Bemenet: < jel segítségével
 - Pl: passwd juli <alma; cat alma # almafa, almafa

I/O átirányítások II.



- Kimenet, hozzáfűzés (append)
 - >> fájlnev, Pl: echo dió >>gyumolcs
 - Ha nem létezik a fájl, akkor létre is hozza!
- Hibakimenet (stderr) átirányítás
 - 2>, 2>>
- Szimmetria miatt(☺): <<
 - Bemenet átirányítás a helyben megadott szövegre

```
$ cp 2>hiba  
$ mv ezt 2>>hiba  
$ cat hiba
```

```
cat <<alma  
<input type=text name=X>  
<input type=button>  
alma
```

Szűrők



- Parancs vs. Szűrő
 - Képes egy parancs kimenetét saját bemeneteként fogadni!
- Önmagában egy szűrő nem szűrő!
 - Legalább 2 parancs összekapcsolásáról van szó!
- Műveleti jel: |
- Ilyen nagyon ismert, gyakran használt szűrőparancs például a WC!
 - Word Count!
 - Feladata: sorok, szavak, karakterek megszámlálása!
 - `wc [-lwc] [file]` #Paraméter nélkül várja ctrl-d-ig a bemeneti adatokat!

Szűrők használata



- Használhatók normál parancs alakban, paraméter nélkül, ekkor a bemenetről (billentyűzet) várja az adatokat
 - Input vége: CTRL+D
 - Paraméteres formában módosulhat a bemenet!
 - Pl: `wc alma.fa`
- Szűrő formában, | karakterrel!

```
szamalap.inf.elte.hu - PuTTY
illes@panda:/$ wc
alma a fa alatt
Esik az eső
      2      7      29
illes@panda:/$ ls|wc
      27      27      124
illes@panda:/$ ls
afs      dev  h2      lib      opt      sbin  tmp
bin      etc  home    lib64    proc     srv   usr
boot     h    initrd  media    root     sys   var
cluster  h1   LDAPFETCH mnt      run      T
illes@panda:/$ ls -l|wc
      28      247      1338
illes@panda:/$
```

Fontosabb szűrők



- Fontosabb kész szűrők:

- cut, tee, sort, uniq, wc, grep

- Példa:

```
$ who >nevek  
$ sort nevek #sorok szerinti sorrend  
$ who|sort -r -u # fordított sorrend, egyedi sorok  
$ who|wc -l #bejelentkezett felhasználó szám
```

- Ami biztos: man (ual) lekérdezése

- Példa: man sort, man -k kulcsszó

CUT - kivágás



- Standard inputon vagy fájlból vághatunk ki mező(ke)t, oszlop(ka)t, sorokból!
- `cut -c1-5` 1-5 karakteroszlop kivágás
 - példa: `date | cut -c4-8` `# Oct`
- `cut -f1,3,5-7` 1,3,5,6,7 mezők kivágása
 - Alapértelmezett mezőelválasztó: Tab
 - Új mezőelválasztó: `-d char`
 - Példa: `cat /etc/passwd | cut -f1,7 -d: # név, shell`

Grep – sorok szűrése



- A paraméterül adott mintával rendelkező sorok kiválasztása.
- Fontosabb paraméterek:
 - -v mintát nem tartalmazó sorok
 - -i kis és nagybetűket nem különböztet meg
 - -w Csak önálló szóként találja meg (traPista nem)
 - -r Rekurzívan a paraméterül adott könyvtárra.
 - -l Csak a fájl neveket írja ki. (fájlban keres)
 - -c csak a sorok számát írja ki
 - -n megszámozza a sorokat

Grep használata



- Szűrő példa:
 - `cat nevsor | grep Pista` # Eredményül kapjuk a Pista-t tartalmazó sorokat.
- Parancs forma:
 - `grep -r 'fradi' ./script` # script könyvtárban a fradi-s sorokat (fájlokban) keresi
 - `grep -r -l par *` # Az összes állományban, alkönyvtárakban keresi a par szót, eredményül csak a fájl nevét írja ki.

Mintaillesztés, reguláris kifejezések



- Egy szövegminta általános megadása – Reguláris kifejezések-speciális karakterek
 - ^ Sor elejétől kell egyezni a mintának.
 - Pl: '^alma' : a sor elején alma szó áll
 - \$ Sor végétől kell egyezni a mintának.
 - Pl: 'barack\$' : a sor végén a barack szó áll
 - . Egy tetszőleges karakter
 - * Előző minta ismétlése 0 vagy többször!
 - Pl: '^alma.*fa\$ ' - alma és fa között akárhány(0 is lehet) karakter

További mintaillesztés



- Karakterhalmazok megadása: [...]
 - [a-z] (kisbetű)
 - [^a-c] Nem a, b vagy c betű
 - [A-Za-z0-9] Alfa numerikus (szám vagy betű)
 - \w Alfa numerikus, mint előző
 - \W Nem alfa numerikus
 - \d számjegy, azonos a [0-9]-el
 - \s szóköz, tab, sortörés
- Szavak illesztése
 - \< Szókezdet, Pl: grep "\<Zol"
 - \> Szóvég, Pl: grep "\>nán"

E(F)Grep – mintaillesztés I.



- egrep , fgrep– bővített grep -E, fixed, grep -F
 - `ali|éva` , vagy kapcsolat,
 - Példa: `ls | egrep "par|pelda"`
 - + Előző minta legalább egyszer
 - Példa: `ls | egrep "<p+f"` # szó elején 1 vagy több p, majd f betű.
 - ? Előző minta nulla vagy egyszer ismétlődik
 - Példa: `ls | egrep „param\d?”` #param, param1, param2,...
 - {n} előző karakter pontosan n szer!
 - Példa: `cat almafa | egrep ^[a-b]\w{5}`

E(F)Grep – mintaillesztés II.



- $\{2,4\}$ Előző minta 2,3 vagy 4-szer ismételve
 - $\{1,\}$ Előző minta legalább egyszer
- $()$ Egy csoportba fogunk egy mintát.
 - Ismétlődéshez célszerű, azaz ezt követi egy ismétlésre vonatkozó utasítás $+,?,\{n\}$
 - Példa: `[0-9]{8}(\s[0-9]{8}){1,2} #bankszámla`
 - A `[0-9]` helyett `\d` is jó lenne.
- Speciális karaktert célszerű `\` mögé írni.
 - Példa: `^[+-]?\d+([\.\.]\d+)? #előjeles szám tizedes ponttal \., vagy vesszővel`
- Többi lehetőséghez: man

További minták



- E-mail cím(rövidebb változat):
 - `\w+[\.-_\w]*@\w+(\.\w+)+`
- Óra:perc:másodperc
 - `[01][0-9]|2[0-3]:[0-5][0-9]:[0-5][0-9]`
- Stb.
- Minden programozási nyelv tartalmazza ezt vagy ennek kicsit bővített, módosított lehetőségeit!

Szövegszerkesztők



- Széles skála áll rendelkezésre!
- Jellemzően: vi, pico, mcedit, joe, stb.
- Alapvető használatukban a segítség a képernyőn!
 - Kivéve vi!
- Néhány vi jellemző!
 - Kétféle üzemmód, parancs vagy szerkesztő mód!
 - Parancsmódból szerkesztő módba: i, a, o, stb
 - Szerkesztő módból parancs módba: ESC
 - Parancs módban: mentés: w név, mentés és kilépés: wq (quit)
 - Kilépés mentés nélkül: q!

Köszönöm a figyelmet!

