05

BASH script programozás I.

cut

- sorok kiválasztott részeit írja ki. Kapcsolók:
  - -b megadott bájtok közötti részt írja ki
  - -c megadott karakterszám közötti részt írja ki
  - **-d elválasztó jel (alapértelmezett a TAB)**

    -f elválasztással született megadott mezők közötti részt írja ki
- Feladat:
  - cutoljuk az ls -l tartalmát (pl első karakter)
  - cutoljuk a ls -1 tartalmából a állományneveket, kiterjesztéseket

BASH (man bash)

- ez egy shell (Bourne Again SHell): hozzáférést biztosít a kernel funkcióihoz, kényelmi szolgáltatások, programok indítása: **parancsértelmező**
- chsh : felhasználó shelljéne változtatása (de mi ezt használjuk, ez az alapértelmezett shell)
- parancssor: parancsok értelmezése a standard bemenetről
- alshellek nyithatóak meg & , fa felépítés

BASH script

- script kötegelt végrehajtása:
  - a parancsértelemező nem a parancssorból, hanem a fileból olvassa soronként a parancsokat
- saját parancsot hozhatunk létre (automatizálás)
- felépítése:
  - #! /bin/bash első sor, parancsértelmező fejrész
  - lehetőség van bármilyen parancs (akár program) végrehajtására
  - soronként egy parancs (de lehet többsoros parancs is sorvégi: \)
  - változók, vezérlési szerkezetek (if, for, while, …)
  - !!!nagyon kötött a szintaxis!!!

BASH script alapok

- Parancsértelmező fejrész: #! /bin/bash
- Egysoros kommentelés: #
- \ a sor végén: következő sor is azonos sornak lesz értelmezve (többsoros parancs)
- 'SZÖVEG':
  - nincs jelentése a spec karaktereknek
  - pl nem lehet változóra hivatkozni, nincs parancsbehelyettesítés
- ”SZÖVEG”: jelentése van a spec karaktereknek:
  - $ változónév hivatkozás
  - ` ` közé írt parancs végrehajtódik és az eredmény behelyettesítődik
  - \ kezdve a fenti spec karakter is kiírható, pl: \$

Feladat: nézzük meg a behelyettesites.sh -t

Szöveg kiírása

- echo – kiír egy stringet a képernyőre (kapcsolóval escape szekvenciák)
- printf ”FORMAT” ”PARAMÉTEREK” – C-hez hasonló módon escape szekvenciákkal (pl: \n, \t, %s, stb ..) értelmezi és kiírja a szöveget
  Példa: printf "Hello %s\n" "World"

Változók

- értékadás: VALTOZO_NEV=érték
  (!!nincs szóköz az = után !!!)
- hivatkozás:

  $VALTOZO_NEV

  vagy ${VALTOZO_NEV}
- beolvasás standard bemenetről:

  read VALTOZO_NEV
- névadás: betűket, számokat és _ jelet tartalmazhat, de nem kezdődhet számmal, hagyomány szerint nagybetűsek
- érték: **szöveg**

  Feladat: Kérjünk be egy szöveget, adjuk át egy változóba majd írassuk ki

Parancsmegadás

- szokásos módon parancs megadása (mint ahogy a parancssorban megszoktuk)
  - ekkor végrehajtódik és a STDOUT/ERR-ra íródik a kimenet
- hivatkozás a parancs eredményére:

    ekkor a parancs lefut és az eredménye fog behelyettesítődni és megjelenni az adott sorban

  - `PARANCS`
  - **$(**PARANCS**)**

Feladat: ls kimenetét adjuk át egy változónak és írassuk ki

Matematikai jelölés

- **((KIF))**
  - csak egész számokon értelmezett, eredmény is egész (használhatóak változók)
  - +, -, *, /, %, ++, - –, **(hatványozás), !, ~, &, ^, &&, ||, ?:, =, +=, -=, *=, /=
  - < <= > >= == != (igaz: 1, hamis: 0)
  - lehet hivatkozásként is használni: $((KIF))

Paraméterek

- hivatkozás: $1, $2, …. , $9, ${10}, ${11}, ….
- összes paraméter listában: $*
- a script neve: $0
- paraméterek száma: $#

Feladat: írjunk egy olyan scriptet, ami

- kiírja az első sorba az összes paramétert
- a következő sorban pedig összeadja az második és harmadik paramétert