

07

Szűrők, BASH script gyakorlás

zh

- felépítése
 - két bevezető könnyebb feladat (5-5) pontért (hasonló nehézségű, mint az extra feladatok voltak)
 - egy nehezebb, összetettebb feladat 10 pontért
- anyagrész: minden, ami a félév során elhangzott.
 - BASH scriptet kell írni,
 - ismerni kell a vezérlési szerkezeteket,
 - ismerni kell az összes, gyakorlaton megnézett parancsot,
 - kiemelten a szűrőket (regexp, grep, cut, sed)
- amit használhatsz (ne bízd el magad, ezek zh-n való lapozgatása nem lesz elég, ahhoz hogy megold a feladatokat):
 - Rodek Lajos gyakorlati jegyzete (93-as oldalig kell)
 - ? pubban található scriptek (11-es sorszámmal bezárólag)
- amit nem:
 - gyakorlati diasorjaim, honlapomon lévő anyagok
 - Erika jegyzete
 - internet

Szöveg kiírása - kiegészítés

- 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"

if - kiegészítés

- if [kapcsolós feltétel]; then ✓
- if ((matematikai kifejezéses feltétel)); then ✓
- if PARANCS; then

Speciális változók - kiegészítés

- PARANCS

if ((\$? == 0)); then

#Ahol a \$? az előző parancs exit státuszára hivatkozó változó. Általános esetben ez:

- 0 ha minden OK
- 1 ha kisebb warningok vagy hibák történtek
- 2 ha komoly probléma van

- String hossza: `${#VAR_NAME}`

Szűrő

- grep – csak a mintához illeszkedő sorokat írja fájlból (a paramétere tekinthető egyszerűsített reguláris kifejezésnek: karakterek konkatenációja)
- 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**
(az awk-kal majd egészen hasonló dolgokat lehet művelni)

Feladat:

- grep-pel felhasználónevet szűrjük az ls -l tartalmából
- cutoljuk az ls -l tartalmát (pl első karakter)

Fájlok összehasonlítása

- `cmp` – két fájl különbségének a pozícióját írja ki
- `diff` – kiírja azokat a sorokat amikben különbözik a két fájl

„Keresés és csere”

- sed – szöveg cseréje a szövegben:
 - sed s/ezt/erre/g < bemenet .txt > kimenet.txt (minden előfordulásnál cserél)
 - sed s/ezt/erre (ekkor csak az első előfordulásnál)
 - sed /ezt/p (kiírja a mintára illeszkedő sorokat - greppel azonos működés)
 - sed /ezt/d (törli a sort), ha csak a mintára illeszkedő részt szeretnénk törölni, akkor első eset üres második //-el

A reguláris kifejezés megadása a grephez hasonlóan történik, tehát **fordított** az egrephez képest:

- ha pl reguláris kifejezést szeretnénk megadni akkor \- Tehát pl iteráció `\{2,5\}`, halmaz: `\[abc\]`, sor eleje: `^`, sor vége: `\$`

Példák

- `ls -l | sed s/valaki/akarki/g`
- `ls -l /tmp/ | sed /valaki/d`
- `ls -l /tmp/ | grep -v "valaki"`

Gyakorlás a zh-ra

- Írjunk scriptet, ami a paraméterként kapott könyvtárban megadja, a maximális sor számú .txt fájlt. A maximum kiszámításához használjunk függvényhívást.
- MintaZHFeladat.pdf