

8. gyakorlat. Shell script IV.

Reguláris kifejezések, soron belüli mintaillesztés (grep), soron belüli csere (sed) esetleg awk – gyakorlás a korábbi órákon fel nem dolgozott feladatokkal.

Reguláris kifejezések, grep, sed, awk

1. Feladat: Készítsünk scriptet, amelyik a paraméterében megadott fájlból a csak számokat tartalmazó sorokat írja ki! (valós számokra is gondoljunk)
2. Feladat: Készítsünk scriptet, amelyik megszámlolja, hogy hány csak karaktereket tartalmazó sor van a fileban! Megoldások: `s=0; while read ... < file` vagy `grep -c ...`
3. Feladat: Készítsünk scriptet, amelyik egy adott szót (paraméterként megadott) tartalmazó sorokat nem másolja át egy másik fájlba! `grep -v`
4. Feladat: Készítsünk scriptet, amelyik a paraméterében megadott fájlból a csak azokat a sorokat írja ki, amelyekben sem az alma sem a korte szó nem szerepel megadott szó sem szerepel!
`cat file | grep -v „[alma\|korte]”` vagy `cat file | sed „/alma\|korte/d”` vagy `cat file | awk ' $1 !~ /alma\|korte/ {print $1} '`
5. Feladat: Cseréljük ki egy fájlban szereplő minden adott szót (paraméterben) egy másikra! `sed „s/$1/$2/”`
6. Feladat: Egy fájlban kétjegyű számok vannak – írjuk ki fordított sorrendben a számjegyeket! `sed "s/([0-9])\([0-9]\)/\2\1/"`
7. Feladat: Egy fájlban soronként két szó van. Cseréljük meg a szavak sorrendjét a sorokban! `cat file | sed „s/([a-zA-Z]\+)\([\+]\([a-zA-Z]\+\)/\3\2\1/”` vagy `cat file | awk '{print $2,$1}'`
8. Feladat: Egy fájlban számokat találunk soronként egyet-egyet. Adjuk össze ezeket! Megoldás `while read < file` vagy `cat file | awk 'BEGIN {s=0} {s+=$1} END{print s}'`
9. Feladat: Egy fájlban számokat és szövegeket tárolunk vegyesen. A nemcsak számokat tartalmazó sorokat másoljuk át egy új fájlba! Megoldások: `grep -v regkif` vagy `awk ' $1 !~ /regkif/ {print $1} '`
10. Feladat: Írjunk programot, ami beolvasson egy számot a bemenetről, majd kiírja azt fordítva! (dióhéjban: `while [$szam -gt 0]do $aktualis=$(($szam % 10)); $szam=$(($szam / 10)); forditva=$((echo ${forditva}${aktualis}); done`)
Az alábbi megoldás nem használja a let utasítást, helyette `expr`-t használ, és normál parancsbehelyettesítést használ (` parancs `).

```
#!/bin/sh
```

```
#
```

```
echo -n Irjon be egyszamot:
```

```
read szam
```

```
forditva=""
```

```
while
```

```
[ $szam -gt 0 ]
```

```
do
```

```
    akt=`expr $szam \% 10`
```

```
# echo $akt
```

```
    szam=`expr $szam \/ 10`
```

```
# echo $szam
```

```
    forditva=`echo "${forditva}$akt"`
```

```
done
```

```
echo -n A fordított szamsorrend:  
echo $forditva
```

11. Feladat: Írjunk programot, ami beolvas egy számot a bemenetről, majd kiírja a számjegyek összegét! (hasonló az előzőhöz, csak összegezni kell az értékeket)
12. Feladat (sok-sok extra pontért): Írjunk programot, ami megfordítja soronként a paraméterben megadott fájl tartalmát! (segítség: átmeneti tárolásra egy tömböt használjunk)