

7. gyakorlat. Shell script III.

I/O átirányítás, szűrők, csövek,: wc, grep, sort, cut, tee, sed, reguláris kifejezések).

Ismétlés, scriptek, szerkezetek

- a) Feladat: Írjunk shell script programot, ami kiszámolja egy szám (n) faktoriálisát! (Pl. while ciklus, i 2-től n-ig, majd a ciklusmagban `f=`expr $f`* $i``. A feladatmegoldásoknál ne használjunk `let` utasítást, elég használni, megtanulni az `expr`-t, és használjuk a normál parancsbehelyettesítést! `` parancs ``) Ellenőrizzük, hogy kaptunk-e paramétert!
- b) Feladat: Írjunk script programot, ami szavakat olvas be egészen addig, míg a vége szót nem írjuk be, majd sorba rendezve kiírja a képernyőre azokat! (Használjunk ideiglenes fájlt!)
- c) Feladat: Írjunk programot, amelyik a paraméterül kapott számokat összeadja és az összeget a szabványos kimenetre írja! A program maximum 10 számot tudjon összeadni! A paraméterül megadott számok helyett megadhatjuk a `–help` kapcsolót, aminek hatására írjunk ki rövid ismertetőt a programról!
- d) Feladat: Írjunk programot, ami szétválogat egy paraméterben megadott fájlt páros és páratlan sorokra, majd ezeket új fájlokban elhelyezi! (dióhéjban: `counter=0; while read line; do if [$(expr $counter % 2)]; echo $line > file_even; else echo $line > file_odd; fi; ((counter ++)); done < bemeneti_file`) vagy `(cat file | while ...)`
- e) Feladat: Írjunk shell script programot, ami a háttérben fut és fél percenként ellenőrzi, hogy egy felhasználó bejelentkezett-e a rendszerbe! A felhasználó nevét paraméterként kell megadni!
- f) Feladat: Jelentkezzünk be egyszerre két terminálon. Ha az egyik ablakban elindítjuk az üzenet nevű most megírandó script programot, akkor az egy percen keresztül 5 másodpercenként írjon egy üzenetet a másik terminálablakunkba! (Megoldás kulcsa: `echo üzenet >/dev/pts/2`, ha `pts/2` a másik terminál azonosítója, vagy használjuk a `'write <felhasznalonev>'` parancsot)

Szűrők

- a) Feladat: Telefonregiszter program, amely kapcsolók segítségével a következőket tudja: `-a` kapcsoló megadása esetén felvesz egy nevet és egy telefonszámot a listába (ügyeljünk a dupla elemekre), `-d` kapcsoló megadása esetén törölje a paraméterként megadott nevet és a hozzá tartozó telefonszámot, kapcsoló nélkül pedig keresse meg az illeszkedő neveket és írja ki hozzá a telefonszámokat. A `–o` kapcsolóra adja meg a regiszterben tárolt telefonszámok darabszámát. Az `–o` kapcsolóra rendezve írja ki a neveket – egyszeresen. (A tároláshoz használt szövegfájl formája szabadon meghatározható, segítségképp: a vesszővel vagy kettősponttal elválasztott mezők használata javasolt)
- b) Feladat: Módosítsuk az 1/a (faktoriális számoló programot) úgy, hogy **szűrőként is tudjuk használni!** (Ha nincs paraméter, `n= `cat``)
- c) Feladat: Módosítsuk az 1/c (paraméterek összeadását végző programot) úgy, hogy szűrőként is tudjuk használni! (Ha nincs paraméter, `for i in `cat``)
- d) Feladat: Írjunk programot, ami a paraméterben megadott fájlokat átnevezi csupa kisbetűsre! (segítség: `echo $*|tr '[A-Z]' '[a-z]'`)

- e) Feladat: Bővítsük az előző programot úgy, hogy kapcsolóval megadható legyen, hogy kisbetűsíteni vagy nagybetűsíteni akarunk!
- f) Feladat: Írjunk programot, ami egy paraméterben megadott állományban az összes elkódolt email címből valódit készít, majd azokat listázza a kimeneten. Egyúttal kerüljenek ki egy fájlba is az eredmények. (tee) Elkódolt email címek például (minden esetben gipsz@jakab.hu cím az eredeti): gipsz kukac jakab pont hu, gipsz at jakab dot hu, gipsz_nospam_jakab.hu, stb.)