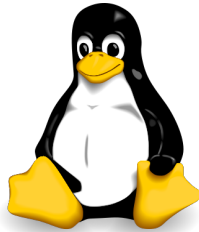


# OPERÁCIÓS RENDSZEREK példatár



Utolsó frissítés: 2014. február 10.



A példatár Szegedi Tudományegyetemen oktatott *Operációs rendszerek* tárgy gyakorlati anyagához illeszkedő feladatok sorát tartalmazza, mely segít a megszerzett tudás elmélyítésében, gyakorlásban, a számonkérésekre való felkészülésben.

## Ajánlott anyagok

---

*Bevezetés a BASH programozásba*

Eredeti: <http://tinyurl.com/BashBeginnerGuide>

Magyar fordítás: <http://tinyurl.com/BashBeginnerGuideHU>

---

*A GAWK felhasználói kézikönyv (letölthető)*

<http://hexahedron.hu/personal/peteri/gawk>

---

*Linux Shell Scripting Tutorial, A Beginner's handbook*

<http://www.freeos.com/guides/lsst>



---

*UNIX / Linux shell scripting tutorial*

<http://tinyurl.com/ShellScriptingTutorial>



---

*Szabilinux, Shell programozás*

<http://www.szabilinux.hu/ufi/10.html>



---

**BÜKI ANDRÁS: *Unix/Linux Héjprogramozás*, 2002**

Kiskapu kiadó, ISBN: 9789639301108



**1. fejezet**

**Parancsok**

## 1.1. Könyvtár- és fájlkezelés

1. Írjon olyan parancsot, amely egy adott állomány egy részletét – a 10. sortól a 15. sorig – írja a szabványos kimenetre!
2. Írjon olyan parancsot, amely egy adott állományból kiszűri az ismétlődő sorokat!
3. Írjon olyan parancsot, amely egy állományból a többször előforduló sorok közül csak egyet ír ki!
4. Írjon olyan parancsot, amely a szöveges állományról másolatot készít, de kihagyja az üres sorokat!
5. Írjon olyan parancsot, amely az állománynak a harmadik és a második oszlopát írja a szabványos kimenetre!
6. Írjon olyan parancsot, amely másolatot készít az állomány azon sorairól, amelyek elején két számjegy található!
7. Írjon olyan parancsot, amely másolatot készít egy állomány 100–200. sorairól, sorszámmal ellátva! A sorok számozása az eredeti állományban elfoglalt helyüket mutassa!
8. Írjon olyan parancsot, amely kiírja a munkakönyvtár öt legnagyobb könyvtárbejegyzésének fontosabb adatait!
9. Írjon olyan parancsot, amely kiírja a szabványos kimenetre a könyvtár azon szabályos állományainak neveit, amelyekre mindenkinek futtatási joga van!
10. Írjon olyan parancsot, amely kiírja a szabványos kimenetre, hány olyan alkönyvtár található a könyvtárban, amely bármely felhasználó által használható munkakönyvtárként!
11. Írjon olyan parancsot, amely a munkakönyvtárban található könyvtárbejegyzések közül a szabályos állományok és könyvtárak legfontosabb adatait kiírja! A listában előbb a könyvtárak, majd a szabályos állományok jelenjenek meg!
12. Írjon olyan parancsot, amely a munkakönyvtárba másolja a könyvtárban található összes kétbetűs nevű szabályos állományt!

13. Írassa ki a szabványos kimenetre a saját HOME könyvtárában található alkönyvtárakat ábécé sorrendben, soronként egyet.
14. Írassa ki a szabványos kimenetre a saját HOME könyvtárában található alkönyvtárakat ábécé sorrendben úgy, hogy csak a 2-5 sorszámú bejegyzések látszódnak, soronként egy-egy.
15. Írassa ki a szabványos kimenetre a saját HOME könyvtárában található állományokat ábécé sorrendben úgy, hogy csak a 3-9 sorszámú bejegyzések látszódnak. A lista tartalmazzon részletes információkat az állományokról.
16. Készítsen statisztikát a HOME könyvtáráról, amelyek a következő adatokat tartalmazzák:
- (a) hány darab fájl és könyvtár található a könyvtárban
  - (b) hány darab nem rejtett könyvtár található a könyvtárban
  - (c) hány darab rejtett állomány található a könyvtárban (a . és .. alkönyvtárakat ne számolja bele)
17. Titkosítsa egy tetszőleges parancs (pl.: `ls -l`) kimenetét 5 eltolású Caesar-kódolással<sup>1</sup>.
18. Titkosítsa egy tetszőleges parancs (pl.: `ls -l`) kimenetét 12 eltolású Caesar-kódolással.
- 19.\* Rendezze a könyvtárak listáját növekvő sorrendbe az alkönyvtárak száma szerint.
20. Jelenítse meg a `.bash_history` fájl tartalmát úgy, hogy az egymás után ismétlődő sorokat csak egyszer tünteti fel.
21. Jelenítse meg a `.bash_history` fájl tartalmát úgy, az ismétlődő sorokat csak egyszer tünteti fel (a nem egymás utáni sorokat is).
22. Jelenítse meg egy könyvtár tartalmát a következő formátumban:

```
<utolsó módosítás dátuma> <fájlnév>
```

---

<sup>1</sup>Caesar-kódolás: <http://hu.wikipedia.org/wiki/Caesar-rejtjel>

23.\*\* Készítsen listát a HOME könyvtárában található duplikált fájlokról. Két fájl duplikált, ha a tartalmuk megegyezik (ennek eldöntésére használja a(z) md5sum programot). A listáról nem kell kiszűrni a merev láncokat.

## 1.2. Csoportok, felhasználók, jogosultságok

24. Készítsen listát a bejelentkezett felhasználók valódi nevééről (nem a felhasználó neveikről)

## 1.3. Folyamatok

25. Írjon olyan parancsot, amely kiírja, hogy hány folyamat található az adott pillanatban a rendszeren!

## 1.4. Reguláris kifejezések

26. Írjon reguláris kifejezést, ami az egész számokra illeszkedik!

27. Írjon reguláris kifejezést, ami a valós számokra illeszkedik!

28.\* Írjon reguláris kifejezést, a szabályos IP(v4 vagy v6)-címekre illeszkedik!

29.\* Írjon reguláris kifejezést, ami a szabályos e-mail címekre illeszkedik!<sup>2</sup>

---

<sup>2</sup> Használjuk a(z) [http://en.wikipedia.org/wiki/Email\\_address#Syntax](http://en.wikipedia.org/wiki/Email_address#Syntax) oldalon leírt szintaxis egy „egyszerűsített” változatát!

**2. fejezet**

**BASH**

## 2.1. Változók, környezeti változók

30. Írjon szkriptet, amely kiírja, hogy a PATH változóban hány könyvtár van.

## 2.2. Paraméterkezelés

31. Készítsen programot, amely megvizsgálja hány paraméterrel indították el! Ha a programot nem egy paraméterrel indították írjon hibaüzenetet majd lépjen ki hibakóddal!

32. Készítsen programot, amely kiírja, hogy hány paramétert kapott a parancssorban, mi volt az első paraméter és mi a program neve!

33. Készítsen programot, amely külön sorban kiírja az összes paraméterét, sorszámmal ellátva!

34. Írjon szkriptet, ami az átadott paramétereket három kategóriába sorolja: létező könyvtár elérési útvonala, létező közönséges fájl elérési útvonala és egyéb érték.

35. Írjon olyan shell-szkriptet, ami csak számokat fogad el paraméterként! Minden nem szám paraméterre írjon ki hibaüzenetet az alapértelmezett hibakimenetre

36. Írjon szkriptet, amely megcseréli két fájlnak az első sorát. A többi sor nem változik. A fájlok elérési útvonalát a szkript paraméterben kapja.

37. Készítsen shell szkriptet, ami paraméterben átadott könyvtárakat hozza létre. Amennyiben az adott néven már létezik könyvtár, akkor törölje ki azt, majd hozza létre újra. A szkript több paramétert is kaphat.

38. Készítsen shell szkriptet, ami útvonalakat kap paraméterként és amennyiben egy útvonal relatív útvonal, átalakítja azt abszolút útvonallá, majd ellenőrzi, hogy létezik-e állomány a megadott útvonalon. A kapott abszolút útvonalak tartalmazhatják a . és .. hivatkozásokat.

39. Leírás: Írjon egy olyan shell script-et, amely az első paraméterben megadott könyvtárban rekurzívan megkeresi az érvénytelen szimbolikus linkeket, majd törli őket. Ha van második paraméter, akkor az itt megadott



fájlba kell írni a törölt linkeket.

Példa:

```
$. ./lcl.sh ~/adatok/ /tmp/torolt_fajlok_listaja.txt
```

40. Írjon egy olyan shell script-et, amely elévült PID fájlok listázására alkalmas! Az első paraméter egy könyvtár, amely többek között .pid kiterjesztésű fájlokat tartalmaz. Ezen fájlok tartalma egy-egy egész szám, amelyek processzus ID-ket jelölnek. Listázza ki ezen .pid fájlok közül azoknak a nevét, amelyek olyan processzus ID-t tartalmaznak, amely nem tartozik egyetlen futó processzushoz sem! Ha a második paraméter egy -d kapcsoló, a szkript törölje is ezeket a fájlokat!

Példa:

```
$ ls /var/run/*.pid
/var/run/acpid.pid           /var/run/gdm.pid
/var/run/atd.pid            /var/run/libvirtd.pid
/var/run/console-kit-daemon.pid /var/run/NetworkManager.pid
/var/run/crond.pid          /var/run/rsyslogd.pid
/var/run/dhclient-wlan0.pid /var/run/upstart-udev-bridge.pid
```

```
$ cat /var/run/gdm.pid
1144
```

```
$ ./stalepid.sh /var/run
gdm.pid
NetworkManager.pid
```

## 2.3. Matematikai kifejezések

41. Írjon szkriptet, mely paraméterül számokat kap. A szkript adja össze a paraméterül kapott számokat, és az eredményt írja az aktuális könyvtárba, egy összeg nevű fájlba.

42. Írjon szkriptet, melynek egy kötelező paramétere van, egy könyvtár. Ezt a könyvtárat rekurzívan megvizsgálja és kiírja, hogy hány darab közön-séges fájl van és ezek össz méretét.

43. Készítsen shell szkriptet, ami egy adott könyvtárban szereplő összes fájl nevét kisbetűssé alakítja.
44. Készítsen shell szkriptet, ami egy adott könyvtárban szereplő összes fájl nevében a szóközöket egy paraméterben átadott karakterre cseréli. A szkript ellenőrizze, hogy az átadott paraméter egyetlen karakterből álljon.
45. Írjon scriptet, amely kiszámolja egy könyvtárban lévő file-ok összegzett méretét (rekurzívan, azaz a könyvtár alkönyvtáraiban lévő file-ok méretével együtt). A script a könyvtárak méretét ne vegye bele az összegzésbe! A script egyetlen beépített linux/unix parancsot sem használhat annak rekurzív kapcsolójával!

Példa:

```
$ ls -lR a
```

a:

```
összesen 8
drwxr-xr-x 2 user user 4096 febr 25 08.27 b
-rw-r--r-- 1 user user 2 febr 24 14.29 valami.txt
```

a/b:

```
összesen 96
-rw-r--r-- 1 user user 63206 febr 24 14.29 en3700.pdf
-rw-r--r-- 1 user user 20532 febr 24 14.29 info_bucsu.odt
-rw-r--r-- 1 user user 2 febr 24 14.29 más.txt
```

```
$ ./sum.sh a
83742
```

46. Írjon egy olyan shell script-et, amely paraméterül egy felhasználónevet vár, majd kiszámolja, hogy ezen felhasználó naponta mennyi időt töltött bejelentkezve az adott gépen. Ha a felhasználó jelenleg be van jelentkezve, akkor az összesítésnél az 'online' szó szerepeljen.

Példa:

```
$ last
gabor pts/1 linux1.inf.u-sze Thu Feb 3 08:12 still logged in lasz
gabor pts/2 linux2.inf.u-sze Thu Feb 2 23:41 - 23:49 (00:08)
balazs pts/4 linux4.inf.u-sze Thu Feb 2 22:40 - 22:43 (00:03)
balazs pts/5 linux2.inf.u-sze Wed Feb 2 20:35 - 20:46 (04:11)
gabor pts/6 linux3.inf.u-sze Wed Feb 2 17:40 - 18:07 (00:27)
csaba pts/8 linux2.inf.u-sze Wed Feb 2 15:14 - 15:24 (00:10)
laszlo pts/11 linux3.inf.u-sze Tue Feb 1 21:43 - 21:44 (00:01)
gabor pts/14 linux1.inf.u-sze Tue Feb 1 21:30 - 22:15 (00:45)
```

```
$ ./login gabor
Thu Feb 3 online
Wed Feb 2 00:35
Tue Feb 1 00:45
```

47. Írjon egy olyan shell script-et, ahol a program első és egyetlen paramétere egy dátumot reprezentál (YYYYMMDD), erről kell megmondani, hogy milyen napra esik. Az érvénytelen dátumokat le kell kezelni. Az ncal nevű programot használni tilos! he, ke, sze, csu, pe, szo, va

Példa:

```
$ ./wd.sh 20090223
he
```

48. Írjon egy olyan shell script-et, amely az adott gépen bejelentkezett összes felhasználó keresztneveiből ábécé sorrend szerinti utónév jegyzéket készít a keresztnevek.txt nevű fájlba. Emellett a stdout-ra ábécé sorrendben írja ki, hogy az egyes keresztnevek hányszor szerepelnek a listában.

Példa:

```
$ ./user_name.sh
Emma 2
Péter 3
Zsombor 1

$> cat keresztnevek.txt
Emma
Emma
```

Péter  
Péter  
Péter  
Zsombor

## 2.4. Vezérlési szerkezetek

49. Készítsen ciklust, amely az 1-10 tartományba eső egész számokat veszi sorra! A ciklus írja ki a számokat és hozzon létre könyvtárakat a számoknak megfelelő névvel!
50. Írjon szkriptet, amely kiírja az aktuális könyvtárban található összes szöveges fájl nevét.
- 51.\* Írjon shell-szkriptet amelyik folyamatosan figyeli a bejelentkezések számát. A szám megváltozásakor írja ki a több vagy a kevesebb szót. A vizsgálatokat elegendő három másodpercenként elvégezni.
52. Írjon szkriptet, ami kिलistázza a közönséges fájlokat egy megadott könyvtárból. A könyvtár nevét paraméterként kapja.
- 53.\*\* Írj szkriptet, ami meghatározza a duplikált fájlokat egy adott könyvtárban. Egy fájl duplikált, ha létezik legalább egy másolata. Ennek eldöntésére használd az md5sum programot. A szkript a vizsgálandó könyvtár nevét parancssori argumentumként kapja, továbbá szűrje ki a merev láncokat is.
- 54.\* Készítsen shell szkriptet, ami útvonalakat kap paraméterként és amennyiben egy útvonal relatív útvonal, átalakítja azt abszolút útvonallá, majd ellenőrzi, hogy létezik-e állomány a megadott útvonalon. A kapott abszolút útvonalak nem tartalmazhatják a . és .. hivatkozásokat.
55. Készítsen programot, amely a saját könyvtárainak legfontosabb állományairól biztonsági mentést készít! A program egy állományból olvassa ki a mentendő könyvtárak neveit (pl a \$ HOME/archivalas.conf) és ezeket mentse a paraméterként megadott könyvtárba!

Példa:

```
$ ./archivalas.sh
```

Használat

archivalas.sh célkönyvtár

```
$cat ~/archivalas.conf
```

```
bin
```

```
Documents
```

```
Pictures
```

```
$. /archivalas.sh Archivum
```

```
Archiválás: bin (bin.tar.gz)
```

```
Archiválás: Documents (Documents.tar.gz)
```

```
Archiválás: Pictures (Pictures.tar.gz)
```

```
#Az Archivum könyvtár tartalma:
```

```
$ls -l
```

```
total 892K
```

```
-rw-r--r-- 1 joe joe 20K 2011-02-05 09:21 bin.tar.gz
```

```
-rw-r--r-- 1 joe joe 672 2011-02-05 09:21 Documents.tar.gz
```

```
-rw-r--r-- 1 joe joe 864K 2011-02-05 09:21 Pictures.tar.gz
```

56.\* Írjon egy olyan shell script-et, amely képek automatikus rendszerezését és átméretezését végzi. A célunk, hogy egy központi könyvtárban (IMAGE\_DB\_DIR környezeti változó alapján) full, medium és thumb mappákban tároljuk az eredeti képeket, azok 50%-os, illetve 10%-os kicsinyítéseit. A scriptnek akárhány képfájl megadhatunk, mint parancssori paramétert, de legalább 1 paraméter szükséges. Ha nincs paraméter, jelezze ki a használat módját! A script a következőt végezze el minden paraméterül kapott képre:

- készítse el 50%-os, és 10%-os kicsinyítését a convert/mogrify parancsok segítségével,
- az eredeti fájlt másolja át az IMAGE\_DB\_DIR/full, az 50%-os-t az IMAGE\_DB\_DIR/medium, a 10%-os kicsinyítést az IMAGE\_DB\_DIR/thumb mappába,
- a fájlok olyan neveket kapjanak, amelyek biztosítják, hogy sohasem írunk felül meglévő képeket az „adatbázisban” (lineáris számokkal történő elnevezés, dátumok használata, stb).
- bármilyen formátumúak is legyenek a bemeneti képek, JPEG formátumú képeket kell készíteni, 'jpg' kiterjesztéssel (nem 'JPEG', 'jpeg', 'JPG').

Ha az IMAGE\_DB\_DIR változó nem írható/olvasható könyvtárra mutat, vagy üres, hibaüzenetet írjon ki a script a hibacsatornára. Ezen felül a script minden esetben a /var/log/kotelezo1.log logfile-t kiegészíti olyan információkkal, hogy mely mappából importáltunk képeket, és azok milyen neveket kaptak.

Példa:

```
$ ls $IMAGE_DB_DIR/full
1.jpg 2.jpg

$ ./kotelezo_1.sh
Használat: kotelezo_1.sh <file> [file ...]

$ pwd
/media/KEPEK_2/Sieles

$ ./kotelezo_1.sh DSCN_102[34].JPG

$ ls $IMAGE_DB_DIR/full
1.jpg 2.jpg 200110202-1.jpg 20110202-2.jpg

$ identify $IMAGE_DB_DIR/thumb/*
1.jpg JPEG 50x50 50x50+0+0 ...
2.jpg JPEG 75x80 75x80+0+0 ...
20110202-1.jpg JPEG 10x20 10x20+0+0 ...
20110202-2.jpg JPEG 100x200 100x200+0+0 ...

$ identify $IMAGE_DB_DIR/full/*
1.jpg JPEG 50x50 500x500+0+0 ...
2.jpg JPEG 75x80 750x800+0+0 ...
20110202-1.jpg JPEG 100x200 10x20+0+0 ...
20110202-2.jpg JPEG 1000x2000 100x200+0+0 ...

$ cat /var/log/kotelezo1.log
2010.02.23: /home/juliska/images/moka1.JPEG -> 1.jpg
2010.02.24: /home/mariska/img/kep3.png -> 2.jpg
2011.02.02: /media/KEPEK_2/Sieles/DSCN_1023.JPG -> 20110202-1.jpg
2011.02.02: /media/KEPEK_2/Sieles/DSCN_1024.JPG -> 20110202-2.jpg

$ unset IMAGE_DB_DIR
```

```
$ ./kotelezo1.sh adatok/kepek/*.jpg > /dev/null  
Hiba: IMAGE_DB_DIR nem adott!
```

```
$ ./kotelezo1.sh adatok/kepek/*.jpg &> /dev/null
```

57. Írjon egy olyan shell script-et, amely kilistázza az azonos nevű processzeket (több példányban futó programokat). A listázás után a felhasználónak lehetősége van az egyező processzek leállítására PID-jük megadásával, amennyiben nem ad meg egyet sem, az összes processz tovább fut.

Példa:

```
$ ps  
  PID TTY          TIME CMD  
 8780 pts/0    00:00:00 bash  
 8793 pts/0    00:00:20 yes  
 8796 pts/0    00:00:18 yes  
 8798 pts/0    00:02:12 gedit  
 8801 pts/0    00:01:20 firefox  
 8813 pts/0    00:00:00 ps
```

```
$ ./kill.sh  
  PID CMD  
 8793 yes  
 8796 yes  
kill cmd? pid:  
8793
```

```
$ ps  
  PID TTY          TIME CMD  
 8780 pts/0    00:00:00 bash  
 8796 pts/0    00:00:28 yes  
 8798 pts/0    00:02:22 gedit  
 8801 pts/0    00:01:30 firefox  
 8813 pts/0    00:00:00 ps
```

58. Írjon egy olyan shell script-et, amely egy könyvtár paramétert vár és ebben a könyvtárban lévő file-ok típusairól készít gyakorisági statisztikát,

azaz megszámolja, hogy az egyes fájl típusokból hány darab érhető el az adott könyvtárban, illetve az adott könyvtár alatti alkönyvtárakban (rekurzív!). A script egyetlen beépített linux/unix parancsot sem használhat annak rekurzív kapcsolójával!

Példa:

```
$ ./counter.sh b
      2 ASCII_text
      1 OpenDocument_Text
      1 PDF_document_version_1.3
$ ls -R
b

b:
a valami.txt

b/a:
en3700.pdf  info_bucsu.odt  más.txt
```

59. Írjon egy olyan shell script-et, amely bejárja a paraméterként megadott directory-t rekurzívan, és írja ki a megadott directory-tól számított relatív útvonallal azokat a file-okat, amelyekre mindenki rendelkezik írási joggal. Azokat a file-okat ne írja ki, amelyek az írási jog ellenére mégse írhatóak (valamelyik ős könyvtárra nincs futtatási joga mindenki másnak)!

Példa:

```
$ls -lR /home/h123456

/home/h123456:
total 8
drwxrwxr-x 2 h123456 users 4096 2009-01-31 00:00 dir1
drwxrwx--- 2 h123456 users 4096 2009-01-31 00:00 dir2
-rw-rw-rw- 1 h123456 users    0 2009-01-31 00:00 1.txt

/home/h123456/dir1:
total 0
-rw-rw-rw- 2 h123456 users    0 2009-01-31 00:00 a.txt
-rw-rw-r-- 2 h123456 users    0 2009-01-31 00:00 b.txt
```



```
/home/h123456/dir2:
total 0
-rw-rw-rw- 2 h123456 users    0 2009-01-31 00:00 c.txt

$ ./listdangerous.sh /home/h123456
1.txt
dir1/a.txt
```

**60.** Írjon egy olyan shell script-et, amely első paraméterében egy számot(n), második paraméterében egy könyvtárat(DIR) adunk meg. Ezek alapján kell az n legtöbbet fogláló közönséges fájlt kilistázni a DIR könyvtárban.

Példa:

```
$/fstat.sh 5 /usr/share/pixmaps/
154545 gdmDebianLogo.xpm
97678  gnome-background-image.png
44487  ubuntu-screensaver.svg
33639  seahorse-applet.svg
11051  gksu-debian.xpm
```

**61.** Írjon egy olyan shell script-et, amely a "basename" parancs használata nélkül a képernyőre írja a paramétereként megadott teljes fájlnevből a könyvtárnevek nélküli fájlnevet.

Példa:

```
$/base_name.sh /home/user/foobar
foobar
```

## 2.5. Függvények

**62.** Írjon függvényt, ami egy paramétert vár! A script döntse el, hogy a paraméterként kapott fájl könyvtár-e illetve hogy üres-e.

**63.** Írjon függvényt számok összeadására! A függvény legyen akárhány paraméterrel meghívható és adja össze minden paraméterét!

**64.** Írjon függvényt számok szorzására! A függvény legyen akárhány paraméterrel meghívható és szorozza össze minden paraméterét!

**65.\*** Írjon függvényt, mely egy paramétert vár és írja ki, hogy mi a fájl típusa (file parancs), ki a fájl tulajdonosa és milyen jogosultsággal rendelkezik a tulajdonos!

**3. fejezet**

**AWK**

### 3.1. Szűrési feladatok

66. Készítsen awk szkriptet, ami kiírja egy állomány leghosszabb szavát.
67. Készítsen awk szkriptet, amely kiírja az aktuális könyvtár összes al-könyvtárának nevét, amely februári keltezésű (bármely év).
68. Készítsen awk szkriptet, amely a bemenet minden sorában felcseréli az első és az utolsó szót, valamint a felcserélt szavakat csupa nagybetűsre cseréli.

### 3.2. Összegzési feladatok

69. Dolgozza fel az `ls -l` parancs kimenetét egy tetszőlegesen választott könyvtár esetén! Számolja meg hány link, hány közöséges fájl és hány könyvtár szerepel a parancs kimenetén!
70. Dolgozza fel az `ls -l` parancs kimenetét egy tetszőlegesen választott könyvtár esetén! Összegezze a fájlok méretét és vesse össze egyezik-e az eredmény azzal, a `du -s` parancs kimenetével!
71. Dolgozza fel az `ls -l` parancs kimenetét egy tetszőlegesen választott könyvtár esetén! Állapítsa meg melyik a legkisebb / legnagyobb fájl (a nevét és a méretét is írjuk ki), és számoljon átlagos fájl méretet!
72. Készítsen awk szkriptet, ami kilistázza a könyvtár tartalmát úgy, hogy a páros sorokban lévő szavak fordított sorrendben jelenjenek meg, míg a páratlan sorok normálisan!
73. Adott  $n$  darab állomány ( $n$  természetes szám). Készítsen awk szkripteket, amelyek a következő műveleteket végzik el.
- (a) Kiírja annak az állománynak a nevét, amelyik a legtöbb mezőt tartalmazza. Adjuk meg a mezők számát is.
  - (b) Kiírja minden állomány nevét, a benne lévő mezők számát valamint azt, hogy átlagosan hány szó van egy állományban.
  - (c) Összeadja az állományok minden sorának első mezőjét és kiszámítja az utolsó mezők átlagát, majd erről minden fájl végén jelentést készít, ami tartalmazza az aktuális fájl nevét és kiszámított adatokat.

74. Készítsen awk szkriptet, ami megadja egy állomány azon sorát, amelyik a legtöbbször szerepel egymás után az állományon belül. Ezen kívül írjuk ki az állomány nevét, valamint azt, hogy hányszor ismétlődött az illető sor.

75. Készítsen awk szkriptet, ami a megadott állományról készít szó gyakoriság stazistikát, tehát megszámolja, hogy az állomány egyes szavai hányszor fordulnak elő, majd a fájl feldolgozása után kiírja azt.

76. Készítsen awk szkriptet, amely paraméterként egy fájlt vár. A fájl a következő formátumú sorokból áll: <ország>;<tájégség>;<kiinduló pont>;<célpont>. A feladat: Kiírni az országokat az őket érintő útvonalak darabszáma szerint. Majd egy üres sor kihagyását követően a tájegységeket kiírni az őket érintő útvonalak összhossza szerint növekvő sorrendben.

Példa Input:

```
Magyarország; Börzsöny; Nagybörzsöny; Márianosztra;18
Románia; Kelemen-havasok; Maroshévíz, Pietros;25
Magyarország; Mátra; Gyöngyös; Mátraszentistván;23
Magyarország; Börzsöny; Diósjenő; Nagy-hideg-hegy;12
```

Példa Output:

```
Országok:
Magyarország 3
Románia 1
\newline
Tájegységek:
Mátra 23
Kelemen-havasok 25
Börzsöny
```

77. Készítsen awk szkriptet, amely paraméterként egy tűzfal log fájlját kapja, ami minden sorában a felhasználók böngészési előzményeit tartalmazza a következő felépítés szerint: <felhasználó neve>;<meglátogatott weboldal neve>;<látogatás időpontja> Feladat: Kiírni a 3 leglátogatottabb weboldalt, az egyes weboldalak mellé azok látogatottsági számával együtt. (szóközzel elválasztva) Holtverseny esetén az összes azonos helyezésű weboldalt írjuk ki.

78. Készítsen awk szkriptet, amelynek paraméter bemenete egy alábbi térfigyelő kamerák aktivitását rögzítő logfile: A fájl az alábbi sorokból épül

fel: <kamera sorszama>:<bekapcsolás órája>:<kikapcsolás órája> Az órák a [0,24] intervallumba esnek. Feladat: Kigyűjteni a fehér foltokat, azaz azon időintervallumokat, amikor egyik kamera sem működött <ora-tol>-<ora-ig> formátumban. A következő sorba pedig a fehér foltok összidejét a következő formátumban: "Teljes ido: X ora", ahol X az említett összidőt jelenti.

Példa Input:

```
1:09:13
2:12:22
3:01:12
2:08:11
3:07:08
1:00:22
```

Példa Output:

```
22-23
23-24
Teljes ido: 2 ora
```

### 3.3. Vegyes feladatok

79. Dolgozza fel az `ls -l` parancs kimenetét egy tetszőlegesen választott könyvtár esetén! Írassa ki csak a könyvtárbejegyzések nevét és méretét! Vegye figyelembe, hogy névben előfordulhat szóköz is, tehát a név nem minden esetben a 8. mező maga! Emellett a linkeket, mint speciális eseteket kezelni kell, hiszen a linkeknél a 8. mezőtől kezdve az utolsóig nem csak a link nevét, hanem magát a célfájl útvonalát is tartalmazza `linknev -> celfajl` formában.

*A következő néhány feladathoz töltsük le az `otos.csv`<sup>1</sup> fájlt, ami a Szerencsejáték Zrt ötös lottó eddigi nyereményeit illetve kihúzott számait tartalmazza. A fájlban van egy fejléc, ami segít az értelmezésben (de ne felejtsek el a feldolgozásból kihagyni!). A nyeremény a fejenként nyert összeget mutatja.*

---

<sup>1</sup><http://www.inf.u-szeged.hu/~grerika/os2011/awk/0502/otos.csv>

80. Listázza ki az összes olyan alkalmat (év + hét) amikor telitalálatos szelvény volt! Ha megvan adva a kihúzás dátuma, azt is jelenítsük meg.
81. Állapítsa meg mikor (év + hét) és mekkora volt a legnagyobb telitaláló nyereménye?
82. Állapítsa meg mikor (év + hét) volt a legtöbb telitalálat és ez hány darab telitalálót jelentett?
83. Állapítsa meg melyik évben volt a legtöbb telitalálatos szelvény és ez hány darab szelvényt jelentett?
84. Állapítsa meg mikor (év + hét) vitték el a legtöbb nyereményt (ötös, négyes, hármas és kettes találatokkal összesen)?
85. Listázza ki melyik számot hány alkalommal húzták ki?
- 86.\* Állapítsa meg melyik 5 számot húzták ki a leggyakrabban/legritkábban?
87. Adott egy állomány, ami az első sort kivéve csak számokat tartalmaz ", " karakterekkel tagolva. Készítsen awk szkriptet ami kiszámítja az egy oszlopba tartozó elemek átlagát és szórását.
- 88.\* Adott egy állomány, ami az első sort kivéve csak számokat tartalmaz ", " karakterekkel tagolva, kivéve az első oszlopot, ami csoport azonosítókat tartalmaz (pl: számok  $1 - n$ -ig). Két rekord egy csoportba tartozik, ha az azonosítójuk megegyezik. Készítsen awk szkriptet ami kiszámítja az egy csoportba tartozó elemek átlagát és szórását.
89. Adott  $n$  darab állomány ( $n$  természetes szám). Készítsen awk szkripteket, amelyek a következő műveleteket végzik el.
- (a) Kírja az állomány azon sorainak a sorszámát, amelyeknek a hossza nagyobb, mint 10 karakter és kírja azokat a sorokat, amelyeknek a hossza kisebb, mint 10 karakter. Egy állomány elemzése után írjuk ki a nevét és a változatlanul kiírt sorok számát.
- (b) Minden sorra, amelynek hossza nagyobb, mint 30 karakter, írjuk ki a sor sorszámát (az állományon belül), az első mezőt és az utolsót. Egy állomány elemzése után írjuk ki a nevét és a mezők számát.
- (c) Kírja minden állományból a mezők első 3 karakterét. Ha a mező hossza

kiseb mint 3, akkor egészítsük ki szóközzel. Egy állomány elemzése után írjuk ki a nevét és a mezők számát.

- (d) Kiírja az állományok azon sorait, amelyek tartalmaznak egy bizonyos karaktersort, legalább k-szor. A karaktersort és a k értéket argumentként adjuk meg. Egy állomány elemzése után írjuk ki a nevét és a kiírt sorok számát.
- (e) Kiszámítja az egy oszlopba tartozó elemek átlagát és szórását az összes fájlra nézve és fájlanként kiírja a részeredményeket.<sup>2</sup>.

**90.** Készítsen awk szkriptet, amely paraméterként egy fájlt vár. A fájl a az ls -lR parancs egy lehetséges kimenetét tartalmazza. A feladat: Kiírni azon fájloknak a nevét, amelyek az aktuális könyvtárhoz képest legalább 3 könyvtár mélységben helyezkednek el. A kiírás után a felhasználótól bekér (standard inputon) az egyik kilistázott fájl nevét. Feladat kiírni ennek a fájlnek a méretét.

**91.** Készítsen awk szkriptet, amely egy CSV formátumú listából HTML táblázatot készít. A CSV állomány minden sora a táblázat egy sorának felel meg. (a mezőket vesszők választják el egymástól) A HTML táblázat 1px vastag kerettel rendelkezzen. (<table border="1"> )

---

<sup>2</sup>Érdeemes a statisztikák számítására alkalmas algoritmusok online változatát használni [http://en.wikipedia.org/wiki/Algorithms\\_for\\_calculating\\_variance#On-line\\_algorithm](http://en.wikipedia.org/wiki/Algorithms_for_calculating_variance#On-line_algorithm)



# Irodalomjegyzék

[1] PERE LÁSZLÓ: *BASH példatár*, 2003<sup>2</sup>

# Tartalomjegyzék

<b>1. Parancsok</b>	<b>2</b>
1.1. Könyvtár- és fájlkezelés . . . . .	3
1.2. Csoportok, felhasználók, jogosultságok . . . . .	5
1.3. Folyamatok . . . . .	5
1.4. Reguláris kifejezések . . . . .	5
<b>2. BASH</b>	<b>6</b>
2.1. Változók, környezeti változók . . . . .	7
2.2. Paraméterkezelés . . . . .	7
2.3. Matematikai kifejezések . . . . .	8
2.4. Vezérlési szerkezetek . . . . .	11
2.5. Függvények . . . . .	16
<b>3. AWK</b>	<b>18</b>
3.1. Szűrési feladatok . . . . .	19
3.2. Összegzési feladatok . . . . .	19
3.3. Vegyes feladatok . . . . .	21