

Kötelező feladatok

2011/2012-2

Operációs rendszerek

Beadási mód

- Zip file
 - File neve **hxxxxxxx.zip**
 - Pl. **h123456.zip**
 - A zip egy könyvtárat tartalmaz
 - A könyvtár neve **hxxxxxxx/**
 - Pl. **h123456/**
 - A könyvtárban van elhelyezve a feladat megoldása
 - **h123456.sh**

Beadási mód

- A file tartalmazzon egy fejléct a parancsértelmező fejléc után

```
#!/bin/bash
#####
# Nev: Karacsony Szilveszter
# EHA-kód: VALAMI.SZE
# h-azon.: h123456
# Feladat: 1. feladat
# Gyak.vez.: Kozma Attila
#####
```

Beadási mód

- o pont

- *Ha nem a megadott módon lesz beküldve a feladat!!!*



Határidő

- biro.inf.u-szeged.hu
 - Aki még nem regisztrált
 - REGISZTRÁLJON!!!
 - <http://biro.inf.u-szeged.hu/regiszt>
 - 2011. március 18. 23:59:59
 - Nem fog ellenőrizni!!!
 - Többször is fel lehet tölteni (10)
 - Utolsó feltöltés számít
- Időben kezdjenek hozzá a megoldáshoz!!!

1. kötelező feladat

Leírás: Írjon egy olyan shell script-et, amely egy könyvtár paramétert vár és ebben a könyvtárban lévő file-ok típusairól készít gyakorisági statisztikát, azaz megszámlolja, hogy az egyes fájl típusokból hány darab érhető el az adott könyvtárban, illetve az adott könyvtár alatti alkönyvtárakban (rekurzió!). 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
```

1. kötelező feladat

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
```

1. kötelező feladat

Leírás: 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
```

1. kötelező feladat

Leírás: 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 paraméteréről 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/kotelezol.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
```

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

```
Hiba: IMAGE_DB_DIR nem adott!
```

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

1. kötelező feladat

Leírás: 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
```

1. kötelező feladat

Leírás: 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
```

1. kötelező feladat

Leírás: 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
```

1. kötelező feladat

Leírás: Írjon egy olyan shell script-et, amely fájlkiterjesztések kisbetűssé alakítására alkalmas! A szkript első paramétere egy könyvtár. Ezt kell rekurzívan bejárni, és azon közönséges fájlok (tehát például alkönyvtárakét, mint amilyen a példában a Kvtr.DAT, nem!) nevét csupa kisbetűssé alakítani, amelyeknek a kiterjesztése a többi paraméterben megadottak legalább egyikétől csupán kis-nagy betűben különbözik. (Tehát például txt megadása esetén a TXT, tXt, Txt, stb. kiterjesztésű fájlokat kell átnevezni.) A bejárandó könyvtár után a többi paraméterben tetszőleges számú kiterjesztést megadhatunk. Feltételezhetjük, hogy a kisbetűssé alakítás után keletkező néven még nem létezik fájl az adott könyvtárban.

Példa:

```
$ ls -lR proba
```

```
proba:
```

```
Összesen 4
```

```
-rw-r--r-- 1 pisti pisti      0 2011-02-14 00:53 file2.Txt
-rw-r--r-- 1 pisti pisti      0 2011-02-14 00:51 file.Dat
drwxr-xr-x 2 pisti pisti 4096 2011-02-14 00:58 Kvtr.DAT
-rw-r--r-- 1 pisti pisti      0 2011-02-14 00:57 Bash_ref.PDF
```

```
proba/Kvtr.DAT:
```

```
Összesen 0
```

```
-rw-r--r-- 1 pisti pisti 0 2011-02-14 00:58 File3.TXT
```

```
$ ./lcext.sh proba txt dat
```

```
$ ls -lR proba
```

```
proba:
```

```
Összesen 4
```

```
-rw-r--r-- 1 pisti pisti      0 2011-02-14 00:53 file2.txt
-rw-r--r-- 1 pisti pisti      0 2011-02-14 00:51 file.dat
drwxr-xr-x 2 pisti pisti 4096 2011-02-14 00:58 Kvtr.DAT
-rw-r--r-- 1 pisti pisti      0 2011-02-14 00:57 Bash_ref.PDF
```

```
proba/Kvtr.DAT:
```

```
Összesen 0
```

```
-rw-r--r-- 1 pisti pisti 0 2011-02-14 00:58 file3.txt
```

1. kötelező feladat

Leírás: 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

1. kötelező feladat

Leírás: 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
```

1. kötelező feladat

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

Példa:

```
$ ./user_name.sh
Emma 2
Péter 3
Zsombor 1
$> cat keresztnévek.txt
Emma
Emma
Péter
Péter
Péter
Zsombor
```

1. kötelező feladat

Leírás: 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 közöségi 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
```