

EsercizioPerl-A

Realizzare uno script Perl (*users.pl*) che stampa su standard output una statistica sulle shell utilizzate dagli utenti del sistema.

Suggerimenti: il programma fa uso del file */etc/passwd*.

Prerequisiti: variabili (scalari/array/array associativi), input/output (print), strutture di controllo (foreach), files (open, close), funzioni (split, sort, chomp/chop, keys).

Per esempio se da linea di comando scrivo:

```
./users.pl
```

e il file */etc/passwd* è costituito dalle seguenti righe:

```
root:x:0:0:root:/root:/bin/bash
games:x:5:60:games:/usr/games:/bin/sh
man:x:6:12:man:/var/cache/man:/bin/sh
mail:x:8:8:mail:/var/mail:/bin/sh
panetta:x:1000:1000:Claudio Panetta,,,:/home/panetta:/bin/bash
```

lo script stampa in output:

```
/bin/bash=2
/bin/sh=3
```

EsercizioPerl-B

Realizzare uno script Perl (*contabyte.pl*) che prende da linea di comando il nome di una directory e restituisce su standard output lo spazio occupato in byte dalla directory .

Suggerimenti: fare uso in modo opportuno del comando *ls*. Non considerare la presenza di eventuali sottodirectory .

Prerequisiti: variabili (scalari/array), input/output (\$ARGV, print), strutture di controllo (foreach), funzioni (split, qx).

Per esempio se da linea di comando scrivo:

```
./contabyte.pl ~/Desktop/sis_op/prove_varie
```

e il contenuto della directory è il seguente (*ls -l ~/Desktop/sis_op/prove_varie*):

```
drwxr-xr-x 2 panetta panetta          4096 2007-05-02 19:05 dir1
drwxr-xr-x 2 panetta panetta          4096 2007-05-02 19:04 dir2
drwxr-xr-x 2 panetta panetta          4096 2007-05-02 19:04 dir3
-rw-r--r-- 1 panetta panetta      55216 2007-04-24 19:42 esercizio1.zip
-rw-r--r-- 1 panetta panetta      43795 2007-04-24 19:42 esercizio2.zip
-rw-r--r-- 1 panetta panetta 1312204 2006-12-04 17:24 i_promes.txt
-rw-r--r-- 1 panetta panetta 490966 2007-05-02 19:35 i_promes.zip
-rw-r--r-- 1 panetta panetta 156672 2007-05-02 19:04 lezione2.ppt
-rw-r--r-- 1 panetta panetta 6287360 2007-05-02 19:04 lezione3.ppt
```

lo script stampa in output:

Bytes=8358501

EsercizioPerl-C

Realizzare uno script Perl (*stat.pl*) che prende da linea di comando il nome di un file e restituisce su standard output una statistica sulle lettere iniziali delle parole contenute nel file.

Prerequisiti: variabili (scalari/array/array associativi), input/output (\$ARGV, print), strutture di controllo (foreach), files (open, close), funzioni (substr, sort, chomp/chop, keys).

Per esempio se da linea di comando scrivo:

```
./stat.pl ~/Desktop/sis_op/perl/nomicognomi.txt
```

e il file nomicognomi.txt contiene le seguenti righe (cat ~/Desktop/sis_op/perl/nomicognomi.txt):

```
Giovambattista  
Ianni  
Alessandra  
Martello  
Claudio  
Panetta
```

lo script stampa in output:

```
A=1  
C=1  
G=1  
I=1  
M=1  
P=1
```

EsercizioPerl-Cbis

Realizzare uno script Perl (*killproc.pl*) che implementa il comando *killall process_name* il quale uccide tutti i processi con un dato nome. In particolare lo script legge da standard input il nome dei processi da uccidere (cioè quelli aventi nome *process_name*).

Prerequisiti: variabili (scalari/array), input/output (STDIN, print), strutture di controllo (foreach), funzioni (chomp/chop, qx).

Per esempio, si supponga di avere il seguente come output del comando *ps ax*:

```
....  
10599 ?      Rl   0:00 gnome-terminal  
10601 ?      S    0:00 gnome-pty-helper  
10602 pts/0  Ss+  0:00 bash  
10650 ?      S    0:00 gedit file:///home/ale/Scrivania/killproc.pl  
10655 pts/1   Rs   0:00 bash  
....
```

Scrivendo da linea di comando:

```
./killproc.pl
```

lo script chiede di inserire la stringa identificativa dei processi da uccidere:

Inserire il nome del processo: `gedit`

a questo punto lo script ucciderà tutti i processi il cui nome è `gedit`, stampando:

Uccido il processo `gedit` con PID `10650`

In una successiva invocazione del comando `ps ax` il processo non comparirà nella lista di quelli attivi.