

Elaborato 1: Utilizzo della shell e sua programmazione.

Consegna: entro il 19 Maggio 2008 ore 9:00.

Modalità di consegna:

- 1) Rinominare il file contenente l'elaborato con il proprio numero di matricola. Si ricorda che la consegna è individuale, pertanto ogni studente dovrà consegnare una copia dell'elaborato.
- 2) Riportare in calce al file contenente l'elaborato un commento che includa: matricola, nome e cognome, data di consegna, titolo dell'elaborato.
- 3) Fare l'upload del file su <http://amarena.sci.univr.it/>
 - a. Seguire i link: Accesso pubblico → Laboratorio Sistemi Operativi 2008 → "Nome_docente_del_corso"
 - b. A questo punto dovrete trovarvi all'interno di anonymous / Laboratorio Sistemi Operativi 2008 / Nome_docente_del_corso
 - c. Cliccare sulla freccia alla destra della voce Elaborato 1 (sotto la colonna Azione), quindi su Nuovo → Documento
 - d. Compilare i campi del form che appare inserendo il file di cui si vuole fare l'upload in "File locale", il vostro nome, cognome e n° di matricola su "Nome del documento".
 - e. Premere OK
- 4) Si ricorda inoltre che non si potranno né modificare né visualizzare i file di cui è stato fatto l'upload.
- 5) Per qualunque problema durante la sottomissione dell'elaborato contattare il docente del relativo corso (Bombieri per Informatica Multimediale, Pravadelli per Informatica).
- 6) Dopo la scadenza del 13/05 non sarà più possibile effettuare l'upload dell'elaborato. Chi non avrà consegnato perderà definitivamente il diritto di fare l'esame nella modalità orale.

Testo dell'elaborato

Realizzare uno shell-script che visualizzi il seguente menu e implementi le corrispondenti operazioni in funzioni separate come descritto nella sezione "In dettaglio".

MENU

- 1) MioSplit
- 2) MioFind
- 3) Esci

In dettaglio

1) *MioSplit*

L'operazione *MioSplit* deve permettere di spezzare un file in parti di dimensione prefissata in modo simile a quanto effettuato dal comando Linux *split*. In particolare, una volta selezionata la voce di menu 1, l'utente deve poter inserire (in una shell separata rispetto a quella del menu) la riga di comando richiesta per l'esecuzione di *MioSplit* con un sottoinsieme delle opzioni riportate nel *man* del comando Linux *split*.

Ovviamente non è possibile usare il comando Linux *split* per effettuare l'operazione *MioSplit*. Si riporta in fondo a questo documento un riassunto della pagina del *man* relativa al comando Linux *split* con solo le opzioni che devono essere implementate in *MioSplit*.

2) *MioFind*

L'operazione *MioFind* deve permettere di cercare un file all'interno del file system in modo simile a quanto effettuato dal comando Linux *find*. In particolare, una volta selezionata la voce di menu 2, l'utente deve poter inserire (in una shell separata rispetto a quella del menu) la riga

di comando richiesta per l'esecuzione di *MioFind* con un sottoinsieme delle opzioni riportate nel *man* del comando Linux *find*.

Ovviamente non è possibile usare il comando Linux *find* per effettuare l'operazione *MioFind*. Si riporta in fondo a questo documento un riassunto della pagina del *man* relativa al comando Linux *find* con solo le opzioni che devono essere implementate in *MioFind*.

3) *Esci*

Fare in modo che si esca dal programma. Notare che il menu deve essere continuamente visualizzato dopo ogni esecuzione delle operazioni associate alle voci del menu. Il programma deve terminare solo se l'utente seleziona la voce di menu 3.

Suggerimento

Per avviare un nuovo terminale su cui eseguire le operazioni associate alle voci di menu 1 e 2 usare il comando *xterm* con l'opzione *-e* (si veda il *man* per ulteriori dettagli).

FAQ

1. *E' possibile inserire il codice dei punti 1, 2, 3 su script separati, oppure bisogna usare le funzioni?*

E' possibile sviluppare il programma inserendo il codice dei punti 1, 2, 3 su funzioni oppure su file separati.

2. *Come devono essere passate le opzioni per i comandi *MioSplit* e *MioFind*?*

Le opzioni per i comandi *MioSplit* e *MioFind* possono essere fornite come linea di comando oppure in modo interattivo.

3. *E' possibile usare il comando *awk*?*

Si, gli unici comandi che non possono essere utilizzati sono *find* e *split*

4. *Quando viene scelta l'opzione 3, devono chiudersi tutte le finestre aperte dallo script?*

Si, deve rimanere aperta solo la shell di partenza

N.B.: Tutto quanto non esplicitato in questo documento può essere implementato liberamente.

NAME

split - split a file into pieces

SYNOPSIS

split [OPTION] [INPUT [PREFIX]]

DESCRIPTION

Output fixed-size pieces of INPUT to PREFIX1, PREFIX2, ...; default size is 1000 lines, and default PREFIX is 'x'. With no INPUT, or when INPUT is -, read standard input.

-b, --bytes=SIZE
put SIZE bytes per output file

-l, --lines=NUMBER
put NUMBER lines per output file

--help display this help and exit

NAME

find - search for files in a directory hierarchy

SYNOPSIS

```
find [path...] [expression]
```

DESCRIPTION

Find searches the directory tree rooted at each given file name by evaluating the given expression from left to right.

OPTIONS

Command-line arguments following the command name are taken to be names of files or directories to be examined, up to the first argument that begins with '-'. That argument and any following arguments are taken to be the expression describing what is to be searched for. If no paths are given, the current directory is used.

This manual page talks about 'options' within the expression list. These options control the behaviour of find but are specified immediately after the last path name.

EXPRESSIONS

- help, --help
Print a summary of the command-line usage of find and exit.
- name *filename*
filename is the file to be searched into the specified path.
- newer *filename*
File was modified more recently than *filename*.
- perm *mode*
File's permission bits are exactly *mode*.
- type *c*
File is of type *c*:
 - b block (buffered) special
 - c character (unbuffered) special

d directory
f regular file
l symbolic link

-delete

Delete files. If the removal failed, an error message is issued.

EXAMPLES

```
find /tmp -name core -type f -delete
```

Find files named core in or below the directory /tmp and delete them.

```
find $HOME -newer pippo.txt
```

Search for files in your home directory which have been modified after file "pippo.txt".

```
find . -perm 664
```

Search for files which have read and write permission for their owner, and group, but which other users can read but not write to. Files which meet these criteria but have other permissions bits set (for example if someone can execute the file) will not be matched.