

**Elaborato 1:** Utilizzo della shell e sua programmazione.

**Consegna:** entro il 15 Maggio 2006 ore 23:59.

**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 → Sistemi Operativi LABORATORIO 2006 → Elaborati → "Nome\_docente\_del\_corso" → Elaborato 1
  - b. A questo punto dovrete trovarvi all'interno di anonymous / Sistemi Operativi LABORATORIO 2006 Membri / Elaborati / Nome\_docente\_del\_corso / Elaborato 1
  - c. Cliccare sulla freccia alla destra della voce Elaborato 1, quindi su Nuovo → Documento
  - d. Compilare i campi del form che appare inserendo il file di cui 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 (Drago per Informatica, Pravadelli per Informatica Multimediale).
- 6) Dopo la scadenza del 15/05 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) Occupazione Disco
- 2) Rimozione file "giochi" da disco
- 3) Ricerca utenti
- 4) Esci

### **In dettaglio**

#### *1) Occupazione disco*

Dato un sottoalbero del file system, realizzare una funzione che mostri l'occupazione del disco di ciascun elemento del sottoalbero.

La funzione deve chiedere in ingresso il nome di una directory di partenza e deve restituire l'elenco di tutte le sottodirectory e i file contenuti in essa (in modo ricorsivo) con indicato per ciascuno lo spazio occupato in memoria. La lista deve essere ordinata in base alla quantità di spazio occupata dai vari elementi (dal valore più piccolo al valore più grande). Infine, la lista deve essere *paginata* (visibile una pagina per volta), per permettere una migliore visualizzazione, ed eventuali errori devono essere rediretti su un file di log.

Esempio:

```
20   scuola.GII.Torino.ps
24   Scienze.txt
```

272 ns2-systemc.tar.gz  
300 os/linux/document.pdf  
1700 tesi.ps  
2840 os.tgz  
2916 office52/lib/bck/

## 2) *Rimozione file “giochi” da disco*

Realizzare una funzione che permetta di rimuovere da un sottoalbero del file system tutti i file il cui nome contenga un determinato pattern.

La funzione deve chiedere in ingresso il nome della directory di partenza e un pattern che possa rappresentare un nome di un gioco, ad esempio “scacchi”. Quindi, a partire dalla directory fornita, la funzione deve:

- cercare tutte le occorrenze del pattern in nomi di file e cartelle contenuti all’interno del sottoalbero;
- mostrare la lista di tali occorrenze;
- chiedere all’utente se si vuole procedere con l’archiviazione di tutti i file e le cartelle contenuti nella lista.

Nel caso in cui l’utente decida di procedere con l’archiviazione, la funzione deve chiedere all’utente:

- il nome da dare all’archivio tar che verrà creato;
- il nome della cartella in cui riporre l’archivio;
- il nome di un file di log su cui salvare lo standard output e lo standard error derivanti dall’archiviazione.

Infine, ottenute tutte le informazioni richieste, la funzione deve procedere con l’archiviazione del contenuto della lista visualizzata.

## 3) *Ricerca utenti*

Realizzare una funzione che visualizzi login, nome e shell di un utente del sistema. La funzione deve chiedere la login dell’utente di cui visualizzare le precedenti informazioni. Se l’utente non esiste deve essere stampato un messaggio di errore.

## 4) *Esci*

Fare in modo che si esca dal programma.

### **Note**

Consegnare solo se il programma funziona. È titolo preferenziale per la valutazione (ma non obbligatorio) controllare il comportamento dei casi limite, come ad esempio:

- la digitazione da parte dell’utente di un numero maggiore di 4 nella scelta del menù iniziale
- l’inserimento di input o file non previsti per le varie funzioni