

Nome e Cognome:

Matricola:

1	2	2	2	3	4	5	6	tot
/10	/17	/5	/8	/10	/15	/20	/15	/100

Spazio riservato alla correzione

1. 10 punti

Supponendo di essere posizionati nella directory root ("current directory" = /) e che in tale directory sia presente il file **pippo** Scrivere uno script che in sequenza realizzi quanto segue:

- 1) crei all'interno della directory root una sottodirectory **first-dir**;
- 2) crei all'interno della directory **first-dir** una sottodirectory **second-dir**;
- 3) si sposti dentro **second-dir** usando un pathname assoluto;
- 4) si copi il file **pippo** nella directory **second-dir** utilizzando pathname relativi.

2. Sia **FF** un file che contenga le 4 lettere **abdc** tale che

```
-rw--w---- 1 rescigno 4 Jun 4 09:45 FF
```

a) (*17 punti + 5 punti bonus*) Si scriva un programma **C** che in successione (il punto 3b) é opzionale):

1) visualizzi su standard output la frase **Ordiniamo FF**;

2) scriva all'interno di un altro file **SS** il contenuto di **FF** in ordine inverso, utilizzando una delle funzioni **exec** (**senza** l'utilizzo dell'operatore **>**) ed il comando **sort -r**;

3a) visualizzi su standard output la frasi **Fatto!**;

3b) (*bonus 5 punti*) visualizzi in successione su standard output le frasi **Puoi controllare SS**
e

Hai fatto un buon lavoro utilizzando due *exit-handler*.

- b) (*5 punti*) Mostrare mediante un disegno *Process Table*, *File Table* e *V-node Table* relativi all'esecuzione del programma di cui al punto a).

- c) (*8 punti*) Siano **HFF** e **SFF**, rispettivamente, hard link e symbolic link a **FF**.
- 1) Dire, giustificando la risposta, quali sono i permessi di **HFF** e di **SFF**;
 - 2) Assumendo di aver eliminato il file **FF**, dire, giustificando le risposte, che cosa si ottiene leggendo 2 byte da **HF** e 2 byte da **SF**.

3. 10 punti

Si consideri il seguente programma e si supponga di compilarlo.

```
#include<sys/types.h>
#include<fcntl.h>
#include<unistd.h>

int main(){
    if (access("prova.txt", O_WRONLY)<0)
        printf("access error per prova.txt");
    else
        printf("access OK\n");

    if (open("tentativo.txt", O_RDONLY)<0)
        printf("open error per tentativo.txt");
    else
        printf("open OK\n");

    if (open("prova.txt", O_WRONLY)<0)
        printf("open error per prova.txt");
    else
        printf("open OK\n");

    exit(0);
}
```

Se fosse

```
-r-xr-xr-x 1 rescigno 10932 Jun 4 10:45 a.out
-rw----- 1 rescigno 1891  Jun 4 09:45 prova.txt
--w-r--r-- 1 straniero 1891  Jun 4 09:45 tentativo.txt
```

supponendo che si sia loggato **straniero**, dire

1) che cosa succede dando **a.out**.

2) Si assuma ora di settare il set-user-id di **a.out**. Dire se ci sono cambiamenti dando **a.out**.

In tutti i casi le risposte vanno giustificate.

4. 15 punti

Considerate un disco con capacità 2^{30} byte e blocchi da 8 KB.

Rispondere alle seguenti domande, giustificando le risposte:

- 1) Quanti byte servono per gli indirizzi di blocco?
- 2) Calcolare la dimensione (in byte) della FAT.
- 3) Quanti blocchi occupa la FAT se memorizzata su disco?
- 4) Qual' é il contenuto della FAT relativamente ad un file i cui dati sono memorizzati (in sequenza) nei blocchi 3, 6, 0 e 10?

5. 20 punti

Un processo genera la seguente sequenza di riferimenti a pagine virtuali:

1, 2, 3, 1, 4, 1, 2, 5, 4, 3, 6, 8, 3, 8, 6, 2, 4, 1, 2, 6, 5, 8

calcolare il numero e la sequenza di page fault che si verificano usando i seguenti algoritmi di rimpiazzamento (i frame sono inizialmente vuoti): (a) ottimo con 3 frame

(b) FIFO con 7 frame

(c) LRU con 3 frame

6. 15 punti

In un sistema con memoria virtuale le pagine sono composte da 2^{12} byte, la RAM é fatta di 2^7 frame, e lo spazio di indirizzamento logico massimo é di 2^8 pagine.

- (a) Qual é la lunghezza in bit di un indirizzo logico?
Qual é la lunghezza in bit di un indirizzo fisico?

- (b) Si consideri la PT sottostante (attenzione: nella tabella i numeri sono tutti in base decimale)

numero pagina	numero frame	valido/invalido
0	52	v
1	100	v
2	x	i
3	126	v
4	x	i
5	1	v
6	70	v
7	10	v
8	x	i
9	x	i
10	x	i
11	87	v
12	125	v
13	21	v
14	56	v
15	12	v
16	121	v
17	20	v

(b1) Dati i seguenti indirizzi logici (in esadecimale):

$0345C$

$0A2EA$

dire se causano page fault o meno; in questo ultimo caso ricostruire l'indirizzo fisico (in esadecimale).

FOGLIO DA UTILIZZARE PER LA BRUTTA

FOGLIO DA UTILIZZARE PER LA BRUTTA

FOGLIO DA UTILIZZARE PER LA BRUTTA