

Nome e Cognome:

Matricola:

1	2	3	4	5	6	7	8	tot
/8	/24	/8	/10	/5	/12	/15	/18	/100

Spazio riservato alla correzione

1. *8 punti*

Scrivere uno script che:

- 1) crei un comando che consenta di cancellare dalla directory corrente tutti i file il cui nome cominci con la cifra 2,3,4;
- 2) mandi in esecuzione tale comando;
- 3) conti il numero di comandi creati dall'utente;
- 4) visualizzi su standard output il nome dello script.

2. 24 punti

(a) (16 punti) Sia **File** un file lungo 20 byte.

```
-rw----- 1 rescigno 20  Jun 4 09:45 File
```

Scrivere un programma C in cui

- (1) - si crei un hard link **HFile** ed un symbolic link **SFile** a **File**,
- (2) - si crei un **processo figlio** che visualizzi sullo standard-output il contenuto di **File** in modo tale che i primi 10 byte di **File** siano visualizzati **utilizzando** **HFile** e i secondi 10 byte di **File** siano visualizzati **utilizzando** **SFile**;
- (3) - il **processo padre** crei un hard link **HHFile** ad **HFile**;
- (4) - il **processo padre** visualizzi sullo standard output il numero di link di **File**, **HFile**, **HHFile** e **SFile**.

(b) (4 punti) Supponendo di aggiungere al codice precedente anche le istruzioni seguenti:

```
chmod(HFILE, 0600);  
chmod(HHFILE, 0200);  
chmod(SFILE, 0400);
```

dire quali saranno i permessi dei file `File`, `HFile`, `HHFile` e `SFile` dopo aver mandato in esecuzione l'eseguibile con i cambiamenti sopra riportati e spiegare il perché.

- (b) (*4 punti*) Disegnare la Process Table, la File Table e la v-node Table relative all'esecuzione del programma del punto (a).

3. 8 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
-rw----- 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. 10 punti

Dato il seguente programma C, il cui eseguibile é `a.out`

```
(1)  void handler(int);
(2)  void exit1(void);
(3)  void exit2(void);

(4)  int main(void)
(5)  { char array[6]="Hello ";
(6)    atexit(exit1);
(7)    if (fork()==0)
(8)        {
(9)            while (getppid()!=1);
(10)           atexit(exit2);
(11)        }
(12)    else

(13)    printf("Ciao ");
(14)    write(1,array,6);
(15)    exit(0);}

(16) void exit1(void)
(17) { printf("Exit Handler 1\n"); }
(18) void exit2(void)
(19) { printf("Exit Handler 2\n"); }
```

- (a) dire che cosa succede dando `a.out`. Motivare la risposta.
- (b) modificando la linea (12) con `_exit(0)`; dire che cosa succede dando `a.out`. Motivare la risposta.

5. 12 punti

In una organizzazione dell'allocazione dei file simile a quella adottata in UNIX vi sono 13 puntatori nell'i-node (mantenuto in memoria durante l'accesso al file) di cui

- 10 puntatori diretti a blocchi
- 2 puntatori indiretto a blocchi
- 1 puntatore doppiamente indiretto a blocchi

Se la dimensione di un blocco é 1KB, e un puntatore occupa 4 bytes, dire giustificando le risposte:

- a) Qual la dim. massima di un file per il quale non sono necessari accessi aggiuntivi per accedere a qualunque blocco?
- b) Qual la dim. massima di un file?
- c) Quanti accessi sono necessari per accedere al byte alla posizione 1000K?

6. 5 punti

Sia dato un disco di 16 GB diviso in blocchi ampi 1 KB. Si assuma che la struttura per tener traccia dei blocchi liberi sia del tipo **bitmap**. Si calcoli il numero di blocchi richiesto dalla struttura bitmap.

7. 15 punti

Si consideri un sistema a memoria virtuale con indirizzi a 32 bit, indirizzi fisici a 20 bit, pagine da 512 byt.

- a) Si descriva la struttura dell'indirizzo virtuale e dell'indirizzo fisico;
- b) Si determini la dimensione degli spazi di indirizzamento virtuale e fisico;
- c) Si determini il numero di frame di cui costituito lo spazio di indirizzamento fisico e il numero di pagine di cui costituito lo spazio di indirizzamento virtuale;
- d) quante pagine occupa un programma di 2100 byte;
- e) esiste in tale sistema la frammentazione? Se si, di che tipo ?

8. 18 punti

In un sistema time sharing con priorità sono presenti 4 processi P1, P2, P3 e P4 nel seguente stato:

- P1 in esecuzione;
- P2 e P3 pronti;
- P4 in attesa di una operazione di I/O.

I processi P1 e P4 hanno uguale priorità, e più alta della priorità dei processi P2 e P3; inoltre il processo P3 ha priorità più alta di quella di P2.

Descrivere come cambia lo stato del sistema (cioè lo stato dei processi) se a partire dalla situazione data si verificano nell'ordine tutti e soli i seguenti eventi:

- a) termina l'operazione di I/O del processo in attesa
- b) trascorre un quanto di tempo
- c) il processo in esecuzione chiede una operazione di I/O
- d) il processo in esecuzione chiede una operazione di I/O
- e) termina l'operazione di I/O del processo in attesa
- f) termina l'operazione di I/O del processo in attesa
- g) il processo in esecuzione termina
- h) trascorre un quanto di tempo
- i) trascorre un quanto di tempo

FOGLIO DA UTILIZZARE PER LA BRUTTA

FOGLIO DA UTILIZZARE PER LA BRUTTA

FOGLIO DA UTILIZZARE PER LA BRUTTA