

Nome e Cognome:

Matricola:

1	2	3	4	5	6	7	8	tot
/10	/10	/15	/15	/10	/16	/12	/12	/100

Spazio riservato alla correzione

1. *10 punti*

Scrivere un comando Bash per:

- (a) Salvare in un file di testo l' output dell' ultimo comando dato.

- (b) Scrivere un comando che fornisce il numero dei comandi contenuti nella history list.

- (c) "Definire" il comando `lh` che lista i nomi dei file presenti nella home directory .

2. 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
-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.

3. 15 punti

Scrivere un programma C che **in successione**

- 1) ridirezioni in un file di nome **elenco** i nomi dei file contenuti nella cwd, utilizzando una delle funzioni **exec** ma senza utilizzare l'operatore di ridirezione;
- 1) mostri sullo standard output la lista dei nomi dei file contenuti nella cwd, utilizzando una delle funzioni **exec**.

4. 15 punti

In un programma C, il codice di **processo padre** usa un exit handler per scrivere sullo standard output `exit padre` mentre il codice di un **processo figlio** usa un exit handler per scrivere sullo standard output `exit figlio`.

(a) Scrivere il programma C in modo da ottenere sullo standard output
`exit figlio`
`exit padre`

(b) Scrivere il programma C in modo da ottenere sullo standard output
`exit padre`
`exit figlio`

5. 20 punti

Considerate un disco con capacità 2^{30} byte e blocchi da 16 KB.

Rispondere alle seguenti domande, giustificando le risposte:

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

6. 15 punti

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

- P1 in esecuzione;
- P2 e P3 pronti (P2 in testa alla coda davanti a P3);
- 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.

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) trascorre un quanto di tempo
- f) termina l'operazione di I/O del processo in attesa
- g) termina l'operazione di I/O del processo in attesa

7. 15 punti

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

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

FOGLIO DA UTILIZZARE PER LA BRUTTA

FOGLIO DA UTILIZZARE PER LA BRUTTA

FOGLIO DA UTILIZZARE PER LA BRUTTA