

Nome e Cognome:

Matricola:

1	2	3	4	5	6	7	tot	bonus
/20	/18	/12	/10	/12	/14	/14	/100	/10

Spazio riservato alla correzione

1. *20 punti*(a) *(11 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 visualizzi sullo standard-output il contenuto di **File** **utilizzando** **HFile**;
- (3) - si crei un hard link **HSFile** ad **SFile**;
- (4) - si visualizzi sullo standard output il numero di link di **File**, **HFile**, **HSFile** e **SFile**.

(b) (5 punti) Supponendo di aggiungere al codice precedente anche le istruzioni seguenti:

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

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

(c) (4 punti) Quali sono il numero di link di `File`, `HFile`, `HSFile` e `SFile` visualizzati sullo standard output?

## 2. 19 punti

Sia `pippo` un file presente nella `cwd` (current working directory).

a) (7 punti) Sia `prog1.out` l'eseguibile corrispondente al codice sottostante:

```
#include<stdio.h>
#include <fcntl.h>
#include <unistd.h>
int main()
{
(1)  fd = open("pippo", O_APPEND|O_WRONLY);
(2)  write(1,"Primo Anno",10);
(3)  dup2(fd,1);
(4)  printf("Sistemi");
(5)  write(1," Operativi ",11);
(6) }
```

dire che cosa succede mandando in esecuzione `prog1.out` e spiegare dettagliatamente il perché;

b) (7 punti) Sia `prog2.out` l'eseguibile corrispondente al codice sottostante:

```
#include<stdio.h>
#include <fcntl.h>
#include <unistd.h>
int main()
{
(1)  fd = open("pippo", O_APPEND|O_WRONLY);
(2)  printf("Primo Anno");
(3)  dup2(fd,1);
(4)  printf("Sistemi");
(5)  write(1," Operativi ",11);
(6) }
```

dire che cosa succede mandando in esecuzione `prog2.out` e spiegare dettagliatamente il perché.

- c) (*4 punti*) disegnare la **Process Table** del processo relativo all'esecuzione di `prog2.out` (riferirsi al punto (5) del codice) e le relative entry nella **File Table** e nella **v-node Table**.

## 3. 12 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", W_OK)<0)
        printf("access error per prova.txt");
    else
        printf("access OK\n");

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

    if (open("tentativo.txt", O_WRONLY)<0)
        printf("open error per tentativo.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**.
- 3) supponendo ora che si sia loggato **studente** ripetere i passi 1) e 2).

*In tutti i casi le risposte vanno giustificate.*







d) Dato il seguente frammento di FAT, dire in quali blocchi fisici sono collocati i seguenti byte:

- byte 131080 del file "pippo" che inizia al blocco 4
- byte 62230 del file "pluto" che inizia al blocco 3

Entry	Contenuto
0	10
1	2
2	0
3	6
4	1
5	8
6	7
7	11
8	12

Nota:  $16K = 2^4 * 2^{10} = 2^{14}$   $1GB = 2^{30}$  Soluzione: La dimensione del disco è di  $2^{30}$  e quindi di  $2^{30}/2^{14} = 2^{16}$  blocchi .

a) Il campo di indirizzamento della FAT (16 bit=2 byte) consente di indirizzare tutti i blocchi.

b) La dimensione della FAT è di  $2^{16}$ . Se risiede su disco occupa  $2^{16}/2^{14} = 4$  blocchi. c)

## 5. 12 punti

Sia "paperino" un file costituito da 10 blocchi memorizzati su disco ed il cui FCB sia gi in memoria principale.

Dire, giustificando la risposta, quanti accessi a disco sono necessari per effettuare le seguenti operazioni effettuate su "paperino", l'una di seguito all'altra:

- a) Nell'ipotesi di **allocazione contigua** 1) cancellare il secondo blocco;  
2) aggiungere un blocco alla fine.

- b) Nell'ipotesi di **allocazione linkata** (ripartendo da "paperino" costituito da 10 blocchi)
  - 1) cancellare il secondo blocco;
  - 2) aggiungere un blocco alla fine.

## 6. 10 punti

Un disco possiede 200 cilindri (numerate da 0 a 199), con richiesta in corso di servizio alla traccia 143, ultima richiesta precedentemente servita alla traccia 125 e con la seguente coda di richieste:

140, 37, 12, 95, 180, 57, 12

Indicare la sequenza di spostamenti della testina per

- (a) una schedulazione SSTF (Shortest Seek Time First),
- (b) algoritmo dell'ascensore (SCAN),
- (c) algoritmo C-LOOK.

## 7. 14 punti

Un filesystem simile a Unix ha gli i-node fatti nel modo seguente:

attributi

ind. blocco 0

ind. blocco 1

ind. blocco 2

ind. blocco 3

ind. blocco 4

ind. blocco 5

ind. blocco 6

ind. blocco indirizzi indirezione singola

ind. blocco indirizzi indirezione doppia

Se i blocchi del disco hanno capacità di 1 Kb e l'indirizzo dei blocchi occupa 32 bit,

a) quale é l'ampiezza massima del file memorizzabile utilizzando questo i-node? (Giustificare la risposta)

b) Se "pippo" un file di dimensione  $1Mb = 2^{20}$ , quanti blocchi servono in totale per la memorizzazione di "pippo" (compresi eventuali blocchi indice)? (Giustificare la risposta)

FOGLIO DA UTILIZZARE PER LA BRUTTA

FOGLIO DA UTILIZZARE PER LA BRUTTA

FOGLIO DA UTILIZZARE PER LA BRUTTA