

## Prova in itinere 22 Aprile 2013

Università di Salerno

Nome e Cognome:

Matricola:

1	2	3	4	5	6	7	tot	bonus 1	bonus 2
/12	/26	/12	/12	/16	/10	/12	/100	/10	/10

Spazio riservato alla correzione

## 1. 12 punti

Sia **File1** un file lungo 6 byte.

Scrivere un programma C che accoda il contenuto di **File1** a **File2**, ricevendo da linea di comando i nomi dei file **File1** e **File2**, dove

- se **File2** esiste allora viene accodato a **File2** il contenuto di **File1**,
- se invece **File2** non esiste allora **File2** viene prima creato con gli stessi permessi di **File1** e dopo avviene l'accodamento.

## 2. 26 punti

Sia `File` un file presente nella directory `/tmp` contenente i caratteri `abcdef`.

```
-rw----- 1 rescigno 6 Jun 4 09:45 File
```

(a) (7 punti) Assumendo di compilare il programma seguente, dire, giustificando la risposta, quale é l'effetto ottenuto dando `a.out`.

```
int main(void)
{
    int  fd1, fd2, fd3;
    char s[3];

(1)    symlink("File","SFile");
(2)    fd1=open("SFile",O_APPEND | O_RDWR);
(3)    read(fd1,s,3);
(4)    write(1,s,3);

(5)    link("File","HFile");
(6)    fd2=open("HFile",O_APPEND | O_RDWR);
(7)    read(fd2,s,3);
(8)    write(1,s,3);

(9)    fd3=open("File",O_APPEND | O_RDWR);
(10)   read(fd3,s,3);
(11)   write(1,s,3);

    exit(0);
}
```

- (b) (*6 punti*) – Quali istruzioni aggiungereste al programma dato al punto (a) per mostrare sullo standard output la size di `File`, `HFile`, `SFile`.
- Quale dovrebbe essere la size di questi tre file?

(c) (7 punti) Assumendo di aver creato **HFile** e **SFile** (mandando in esecuzione l'eseguibile relativo al programma dato nel punto (a)), e di aver compilato il programma seguente, dire, giustificando la risposta, quale é l'effetto ottenuto dando **a.out**.

```
int main(void)
{
    int  fd1, fd2, fd3;
    char s[3];

    (1)    fd1=open("SFile",O_APPEND | O_RDWR);
    (2)    read(fd1,s,3);
    (3)    write(1,s,3);

    (4)    fd2=open("HFile",O_APPEND | O_RDWR);
    (5)    read(fd2,s,3);
    (6)    write(1,s,3);

    (7)    dup2(fd2,1);

    (8)    fd3=open("File",O_APPEND | O_RDWR);
    (9)    read(fd3,s,3);
    (10)   write(1,s,3);

    exit(0);
}
```

(d) (6 punti) Assumendo

- di aver creato **HFile**, **SFile** (mandando in esecuzione l'eseguibile relativo al programma dato nel punto (a)), e successivamente
- di aver rimosso **File**

dopo aver compilato il programma seguente, dire, giustificando la risposta, quale é l'effetto ottenuto dando **a.out**.

```
int main(void)
{
    int  fd1, fd2;
    char s[3];

    (2)    fd1=open("HFile",O_APPEND | O_RDWR);
    (3)    read(fd1,s,3);
    (4)    write(1,s,3);

    (6)    fd2=open("SFile",O_APPEND | O_RDWR);
    (7)    read(fd2,s,3);
    (8)    write(1,s,3);

    exit(0);
}
```

## 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", 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");

    exit(0);
}
```

Se fosse

```
-r-xr-xr-- 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
```

Siano **strudente** e **straniero** tali che **strudente** faccia parte del gruppo di **rescigno** mentre invece **straniero** non ne faccia parte.

1) Dire se **strudente** o **straniero** o entrambi possono mandare in esecuzione **a.out**.

2) Supponendo che si sia loggato **straniero**, dire

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

2.b) 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. 12 punti

In un hard disk sono necessari 4 byte per contenere l'indirizzo di un blocco; inoltre, ciascun blocco ha una capacità di 1Kb.

Si assuma che un file  $F$ , costituito da 10 blocchi sia memorizzato su tale hard disk e che gli attributi del file siano già presenti in memoria principale.

Assumendo accesso diretto, dire, giustificando la risposta, quanti accessi a memoria sono necessari per **modificare i byte da 2030 a 2045 di  $F$**  per ciascuna delle diverse tecniche di memorizzazione di seguito riportati:

- 1) allocazione contigua;
- 2) allocazione concatenata.



## 5. 16 punti

In una organizzazione dell'allocazione dei file simile a quella adottata in UNIX vi sono 8 puntatori nell'i-node (mantenuto in memoria durante l'accesso al file) di cui

- 6 puntatori diretti a blocchi
- 2 puntatori indiretti a blocchi

Se la dimensione di un blocco é  $Y$  Kb, e un puntatore occupa  $2Y$  bytes, dire giustificando le risposte:

- a) Quanti puntatori ci possono essere in un blocco?
- b) Qual é la dimensione massima di un file per il quale non sono necessari accessi aggiuntivi per accedere a qualunque blocco?
- c) Qual é la dimensione massima di un file?

## 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:

130, 37, 12, 95, 170, 57, 20

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. 12 punti

Sia dato un disco da 1 Gb con blocchi da 8 Kb.

- a) Se sul disco sono allocati solo file di lunghezza 7K, ci sarà frammentazione interna? Ci sarà frammentazione esterna? Motivare la risposta.

- d) Dato il seguente frammento di FAT, dire a quali blocchi bisogna accedere per leggere i byte da 20000 a 25000 i del file **pippo** che inizia al blocco 4

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

## 8. 10 punti - (bonus)

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

```
(1)  int main(void)
(2)  { printf("Ciao\n");
(3)    write(1,"Hello\n",6);
(4)    exit(0);}
```

(a) dire che cosa succede dando `a.out`. Motivare la risposta.

(b) dire che cosa contiene `FILE` dando `a.out > FILE`. Motivare la risposta.

(c) modificando la linea (2) con `printf("Ciao");` dire che cosa succede dando `a.out`. Motivare la risposta.

## 9. 10 punti - (bonus)

Sia dato un disco di 32 Gb diviso in blocchi ampi 2 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.

FOGLIO DA UTILIZZARE PER LA BRUTTA

FOGLIO DA UTILIZZARE PER LA BRUTTA



FOGLIO DA UTILIZZARE PER LA BRUTTA

FOGLIO DA UTILIZZARE PER LA BRUTTA

FOGLIO DA UTILIZZARE PER LA BRUTTA

FOGLIO DA UTILIZZARE PER LA BRUTTA