

Prova in itinere 7 Giugno 2016

Università di Salerno

1. **Codice comportamentale.** Durante questo esame si deve lavorare da soli. Non si può consultare materiale di nessun tipo. Non si può chiedere o dare aiuto ad altri studenti.
2. **Istruzioni.** Rispondere alle domande. Per la brutta usare i fogli posti alla fine del plico (NON si possono usare fogli aggiuntivi); le risposte verranno corrette solo se inserite nello spazio ad esse riservate oppure viene indicata con chiarezza la posizione alternativa.
Per essere accettata per la correzione la risposta deve essere ordinata e di facile lettura.
TUTTE le risposte vanno GIUSTIFICATE. Ciascuna risposta non giustificata vale ZERO.

Nome e Cognome: _____

Matricola: _____

Firma _____

Spazio riservato alla correzione: non scrivere in questa tabella.

1	2	3	4	5	Tot	bonus
/22	/28	/16	/18	/16	/100	/15

1. *22 punti*

L'operatore `>>` utilizzato in Bash, accoda l'output del comando alla sua sinistra al file presente alla sua destra; per esempio il comando

```
ls >> elenco
```

accoda l'elenco dei file presenti nella cwd al contenuto del file `elenco` (creando `elenco` se non esiste).

Sia `vecchio-elenco` un file presente nella current working directory.

(a) *(10 punti)* Scrivere un programma C che esegua quanto segue:

1. utilizzando una delle funzioni `exec` e "senza" l'utilizzo dell'operatore `>>` esegua il comando

```
ls >> elenco
```

2. assegni al file `elenco` gli stessi permessi di `vecchio-elenco`.

(b) (*7 punti*) Che cosa cambiereste nel codice precedente per fare in modo che:

3) solo dopo aver eseguito l'accodamento, venga mostrato su standard output la frase
`guarda il contenuto di elenco.`

N.B. Si consiglia di riscrivere l'intero codice.

- (c) (5 punti) Mostrare mediante un disegno il contenuto della *Process Table*, *File Table* e *V-node Table* relative all'esecuzione del programma scritto al **punto a)**, immediatamente prima della fine.

2. 28 punti

Sia `File` un file contenente i caratteri `abcdef`.

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

(a) (10 punti)

Si assuma che l'utente `tizio` abbia nella sua `cwd` il file `File` e l'eseguibile `prog.out` relativo al codice seguente.

```
-rwx--x--x 1 rescigno 1024  Jun 4 09:45 prog.out
```

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

    if ((fd1=open("File",O_RDWR))<0) {printf("errore\n"); exit(0);}
    read(fd1,s,3);
    write(1,s,3);

    exit(0);
}
```

Rispondere alle domande seguenti, **giustificando ciascuna delle risposte**

- 1) Può `tizio` mandare in esecuzione `prog.out`?
- 2) Se sì, che cosa succede quando `tizio` manda in esecuzione `prog.out`?
- 3) Che cosa potrebbe fare l'utente `rescigno` così che l'esecuzione di `prog.out` da parte di `tizio` possa avere un esito diverso?

(b) (6 punti)

Si assuma da qui in avanti che l'utente **rescigno** abbia **File** nella sua **cwd** e sia **new-prog.out** l'eseguibile relativo al programma seguente; dire, **giustificando la risposta**, quale é l'effetto ottenuto dando **new-prog.out**.

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

    (1)    link("File","HFile");
    (2)    if ((fd1=open("HFile",O_RDWR))<0) {printf("errore\n"); exit(0);}
    (3)    read(fd1,s,3);
    (4)    write(1,s,3);

    (5)    symlink("File","SFile");
    (6)    if ((fd2=open("SFile",O_RDWR))<0) {printf("errore\n"); exit(0);}
    (7)    read(fd2,s,3);
    (8)    write(1,s,3);

    exit(0);
}
```

(c) (6 punti)

Quali istruzioni aggiungereste al programma dato al punto (b) per mostrare sullo standard output la size di `File`, `HFile`, `SFile`.

– Quale dovrebbe essere la size di questi tre file? **Giustificare la risposta**

(d) (6 punti)

Si assuma di aver creato HFile, SFile (mandando in esecuzione new-prog.out). Sia altro-prog.out l'eseguibile relativo al programma seguente.

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

(2)    if ((fd1=open("File",O_RDWR))<0) {printf("errore\n"); exit(0);}
(3)    read(fd1,s,3);
(4)    write(1,s,3);

(5)    unlink("HFile");
(6)    unlink("File");

(7)    read(fd1,s,3);
(8)    write(1,s,3);

(9)    if (fd2=open("SFile", O_RDWR)<0) {printf("errore\n"); exit(0);}
(10)   read(fd2,s,3);
(11)   write(1,s,3);

(12)   exit(0);
}
```

Dire, **giustificando la risposta**, quale é l'effetto ottenuto mandando in esecuzione altro-prog.out?

3. 16 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 **pluto** la cui taglia é 5Kb sia allocato su tale hard disk e che gli attributi del file (cioé il FCB) siano già presenti in memoria principale.

Dire, giustificando la risposta, quanti accessi a disco sono necessari per **cancellare il quarto blocco di pluto** ed **aggiungere un nuovo blocco (presente in memoria principale) alla fine di pluto** per ciascuna delle diverse tecniche di memorizzazione di seguito riportate:

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

4. 18 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 indirizzi indirezione singola

ind. blocco indirizzi indirezione doppia

Se i blocchi del disco hanno capacità di 512 byte e l'indirizzo dei blocchi occupa 16 bit,

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

- (b) Se **paperino** é un file di dimensione 2^{18} byte, quanti blocchi servono in totale per la memorizzazione di **paperino** (compresi eventuali blocchi indice)? (Giustificare la risposta)

- (c) Assumendo che l'i-node del file `pip` sia già in memoria principale, dire, giustificando la risposta, quanti accessi a disco sono necessari per
 - (c1) leggere dal blocco 7
 - (c2) scrivere sul blocco 2^9 .

5. 16 punti

Sia dato un disco da 2 Gb con blocchi da 4 Kb.

- a) Assumendo che i blocchi liberi siano gestiti mediante una *bitmap*, dire quanto é grande in byte tale *bitmap*.

- b1) Assumendo che sia utilizzata una FAT per l'allocazione dei file di tale sistema, dire quanti blocchi sono necessari per memorizzare tale FAT.

- b2) Dato il seguente frammento di FAT, dire a quali blocchi bisogna accedere per leggere il byte 100000 del file **paperoga** che inizia al blocco 3

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

6. (bonus) 15 punti

Sia `File` un file presente nella `cwd`. Si consideri il codice seguente.

```
int main(void)
{
    int  fd;

    fd=open("File",O_RDWR);

    dup2(fd,1);

    write(1,"Ciao",4);

    exit(0);
}
```

Ricordando che `dup2` é una **operazione atomica**, si faccia quanto segue:

- (a) riscrivere il codice precedente cosí da sostituire la chiamata `dup2(fd,1)`; con una sequenza di system call che simulino i passi che `dup2(fd,1)`; effettua atomicamente.

(b) modificare il codice scritto nel punto (a) così che sia possibile avere durante una sua esecuzione una situazione che evidenzi la necessità che `dup2` sia una operazione atomica. Si motivi e si commenti la scelta fatta.

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

FOGLIO DA UTILIZZARE PER LA BRUTTA