

II Prova in Itinere 8 Giugno 2011

Università di Salerno

Nome e Cognome:

Matricola:

| | | | | | | | |
|-----|-----|-----|-----|-----|-----|-----|------|
| 1 | 2 | 3 | 4 | 5 | tot | | |
| /12 | /16 | /12 | /10 | /16 | /14 | /20 | /100 |

Spazio riservato alla correzione

1. *12 punti*

Si supponga di avere nella propria home directory alcuni script i cui nomi hanno estensione **.x**

Scrivere uno script che **prendendo da linea di comando il nome di una nuova directory da creare**:

- 1) crei una sottodirectory all'interno della propria home directory il cui nome sia uguale a quello dato da linea di comando;
- 2) sposti tutti i file con estensione **.x** (cioè file eseguibili) presenti nella home directory, nella directory appena creata;
- 3) faccia in modo che gli script contenuti in tale nuova sottodirectory possano essere invocati da qualunque directory con il nome dello script, senza dover specificare l'intero pathname;
- 4) visualizzi su standard output il valore della variabile d'ambiente PATH.

2. 16 punti

Scrivere un programma C che **prendendo da linea di comando il nome di una nuova directory da creare**, ottenga esattamente nell'ordine dato quanto richiesto di seguito:

- 1) scriva su standard output "Ora creo una directory" mediante una funzione di libreria;
- 2) crei, mediante una delle funzioni **exec**, una sottodirectory all' interno della propria home directory il cui nome sia uguale a quello dato da linea di comando;
- 3) scriva su standard output "Ho creato una directory" mediante una system call.

3. 12 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)        atexit(exit2);
(9)  else wait( );

(10)  printf("Ciao\n");
(11)  write(1,array,6);
(12)  exit(0);}

(13) void exit1(void)
(14) { printf("Exit Handler 1\n"); }
(15) void exit2(void)
(16) { printf("Exit Handler 2\n"); }
```

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

4. 10 punti

Si supponga di mandare in esecuzione il seguente programma:

```
int main(void)
{
    int      fd[2];
    pid_t    p;

    p = fork();
    fork();

    if (p>0) { fork();}

    fork();
    fork();

    sleep(30);
    exit(0);
}
```

Dire, giustificando la risposta, quanti processi sono presenti nel sistema durante i 30 secondi dell'istruzione `sleep(30)`.

5. 16 punti

Si assuma che un SO usi:

30 bit per un indirizzo fisico

32 bit per un indirizzo logico

frame di 4KB

Giustificando le risposte, dire

(a) Quanti bit uso per l'offset?

(b) Con quanti bit identifico un frame?

(c) Quanti frame abbiamo in memoria fisica?

(d) Con quanti bit identifico una pagina?

(e) Quante pagine abbiamo nello spazio di indirizzamento logico?

(f) Se un processo usa tutte le pagine quanto grande in byte la page table?

6. 5 punti

Se

p = numero di bit per identificare una pagina nello spazio di indirizzamento logico;

f = numero di bit per identificare un frame nello spazio di indirizzamento fisico.

Si discuta le potenzialit  e/o svantaggi dei tre casi:

(a) $p = f$

(b) $p < f$

(c) $p > f$

7. 14 punti

Si consideri la seguente stringa di riferimenti a pagine

1 2 3 4 1 2 5 2 1 3 2 3

Si contino i page fault e gli accessi a disco in ciascuno dei tre casi (i frame sono inizialmente vuoti):

- (a) FIFO con 4 frame
- (b) FIFO con 2 frame
- (c) FIFO con 5 frame
- (d) LRU con 4 frame

8. 5 punti

Sia dato un sistema di memoria con indirizzi virtuali suddivisi in 4 campi: **a** , **b** , **c** , **d** , i primi 3 dei quali siano utilizzati per indirizzare tre livelli gerarchici di tabelle delle pagine e il quarto campo rappresenti l'offset entro la pagina selezionata. Indicare dall'ampiezza di quali campi dipende il numero di pagine indirizzate nel sistema:

1. da quella di tutti e quattro i campi
2. da quella del campo **d**
3. da quella del campo **a** e **d**
4. da quelle dei campi **a** , **b** , **c** .

9. 9 punti

Un sistema di allocazione della memoria ha le seguenti pagine libere, in questo ordine:

8KB, 15KB, 3KB, 11KB, 5KB, 7KB, 20KB, 25KB.

Si considerino tre richieste di allocazione che arrivano, una di seguito all'altra, nel seguente ordine:

A) 11K;

B) 4K;

C) 13K.

Indicare a quali pagine vengono assegnate le tre richieste sequenziali A, B e C considerando le politiche *First Fit*, *Best Fit* e *Worst Fit*.

Nota: si assuma che, qualora un blocco libero venga assegnato a seguito di una richiesta di dimensione inferiore, il blocco libero sia comunque interamente assegnato.

| | A) | B) | C) |
|-----------|----|----|----|
| First Fit | | | |
| Best Fit | | | |
| Worst Fit | | | |

10. 20 punti

Cinque processi arrivano al tempo indicato, consumano la quantità di CPU indicata e hanno le priorità indicate nella tabella sottostante:

| Processo | T. di Arrivo | Burst | Priorità |
|----------|--------------|-------|----------|
| P_1 | 0 | 3 | 3 |
| P_2 | 1 | 7 | 5 |
| P_3 | 2 | 2 | 2 |
| P_4 | 6 | 3 | 4 |
| P_5 | 7 | 1 | 1 |

a) (10 punti) Calcolare il turnaround medio ed il waiting time medio per i processi nel caso sia usato l'algoritmo di *scheduling Round Robin con quanto di tempo di ampiezza 2*. Riportare il diagramma di Gantt usato per il calcolo.

b) (10 punti) Calcolare il turnaround medio ed il waiting time medio per i processi nel caso sia usato l'algoritmo di *scheduling a priorità con prelazione*. Riportare il diagramma di Gantt usato per il calcolo.

FOGLIO DA UTILIZZARE PER LA BRUTTA

FOGLIO DA UTILIZZARE PER LA BRUTTA

FOGLIO DA UTILIZZARE PER LA BRUTTA