

## II Prova in Itinere 6 Giugno 2012

Università di Salerno

Nome e Cognome:

Matricola:

1	2	3	4	5	6	7	tot	bonus
/10	/15	/15	/10	/20	/15	/15	/100	/10

Spazio riservato alla correzione

## 1. 10 punti

(a) Scrivere uno script che prendendo da linea di comando **il nome di una directory**, che supponiamo sia presente nella cwd, ed **una stringa** realizzi in ordine quanto segue:

- 1) faccia vedere sullo standard output i nomi dei file contenuti nella directory specificata;
- 2) inserisca in un file di nome **elenco** i nomi dei file, presenti nella directory specificata, che contengono la **stringa**;
- 3) ordini il contenuto di **elenco** in ordine lessicografico inverso (alla fine **elenco** dovrà contenere i nomi dei file in ordine inverso).

(b) Scrivere un'unica linea di comando che ottiene quanto richiesto nei punti 2) e 3) di (a) (cioé, il file **elenco** dovrà contenere i nomi dei file contenenti **stringa** in ordine inverso).

2. 15 punti

(a) (8 punti) Supponendo di avere nella propria cwd un file di nome **elenco**, scrivere un programma C che in successione:

- 1) visualizzi su standard-output il messaggio "Ora ti mostro **elenco**";
- 2) mostri sullo standard output il contenuto del file **elenco**, utilizzando una delle funzioni **exec**.

(b) (7 punti) Dire che cosa cambiereste nel programma di cui al punto (a) per far sì che dopo il punto 2) venga visualizzato su standard-output il messaggio "Ora conosci il contenuto di **elenco**".

## 3. 15 punti

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

(d) Supponendo di considerare ora il seguente codice

```
(1)  void handler(int);
(2)  void exit1(void);
(3)  void exit2(void);

(4)  int main(void)
(5)  { atexit(exit1);
(6)    printf("Ciao");
(7)    write(1,"Hello\n",6);
(8)    if (fork(>0) atexit(exit2);
(9)    else    printf("Hola\n");}

(10) void exit1(void)
(11) { printf("Exit Handler 1\n"); }
(12) void exit2(void)
(13) { printf("Exit Handler 2\n"); }
```

dire che cosa succede mandandolo in esecuzione. 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();
    fork();
    fork();

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

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

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

## 5. 20 punti

Un sistema con 8 Mbyte (ricorda che 1Mbyte= $2^{20}$  byte) di memoria fisica e dotato di memoria virtuale con paginazione ha le seguenti caratteristiche: indirizzo virtuale di 24 bit e pagine di 512 byte.

Rispondere alle seguenti domande giustificando le risposte:

a) Quante pagine di memoria virtuale sono disponibili nel sistema?

b) Definire la struttura dell'indirizzo fisico indicando la lunghezza dei campi che lo costituiscono.

c) Si consideri la PT sottostante (attenzione: nella tabella i numeri sono tutti in base decimale)

numero pagina	numero frame	valido/invalido
0	520	v
1	1001	v
2	9	v
3	x	i
4	x	i
5	75	v
6	1200	v
7	551	v

Descrivere in maniera sintetica il comportamento del sistema quando viene richiesto di accedere all'indirizzo 000000000000001011111111

## 6. 15 punti

Volendo adottare l'algoritmo LRU per la sostituzione delle pagine, dire:

(a) Quale dovrebbe essere il numero di frame che assicura il *minimo* numero di page fault. Giustificare la risposta.

(b) Quale dovrebbe essere il numero di frame che assicura il *massimo* numero di page fault. Giustificare la risposta.

Si consideri la seguente stringa di riferimenti a pagine

1 2 3 4 2 1 5 6 2 1 3 2 7 6

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

(c) LRU con 4 frame

(d) LRU con 2 frame

## 7. 15 punti

Sei processi arrivano al tempo indicato e consumano la quantità di CPU indicata nella tabella sottostante:

Processo	T. di Arrivo	Burst
$P_1$	0	10
$P_2$	6	6
$P_3$	7	2
$P_4$	12	5
$P_5$	15	1
$P_6$	25	1

Calcolare il turnaround medio ed il waiting time medio per i processi nel caso sia usato l'algoritmo di *scheduling Shortest Job First con prelazione*. Riportare il diagramma di Gantt usato per il calcolo.



**8. 10 punti bonus**

Supponiamo che il sistema di paginazione usato dal SO assegni 3 frame da 512B a ciascun processo e che l'algoritmo di sostituzione delle pagine sia FIFO. Si consideri il seguente programma

```
...  
#define N 256  
int A[N], B[N], C[N];  
int i,j;  
...  
for(i=0; j<N; i++)  
A[i] = B[i] + C[i];
```

Se la dimensione di un intero è 4B, qual è il numero di page faults ?

FOGLIO DA UTILIZZARE PER LA BRUTTA

FOGLIO DA UTILIZZARE PER LA BRUTTA

FOGLIO DA UTILIZZARE PER LA BRUTTA

FOGLIO DA UTILIZZARE PER LA BRUTTA