

Nome e Cognome:

Matricola:

1	2	3	4	5	6	7	8	tot
/7	/25	/10	/8	/10	/15	/15	/10	/100

Spazio riservato alla correzione

1. 7 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. 25 punti

(a) (15 punti) Scrivere un programma C che in successione:

- 1) visualizzi su standard-output il messaggio "Ridireziona **elenco**" utilizzando una *funzione di libreria*;
- 2) ridirezioni in un nuovo file di nome **elenco_ridiretto** il contenuto del file **elenco** (presente nella cwd), utilizzando una delle funzioni **exec** e senza utilizzare l'operatore di ridirezione.

- (b) (*5 punti*) Disegnare la process table, la file table e la v-node table "finali" relative all'esecuzione del codice di cui al punto (a).

(c) (5 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 puoi guardare `elenco_ridiretto`" utilizzando una *system call*.

(d) (5 punti) Sia `prog.out` l'eseguibile relativo al codice del punto (c). Discutere, giustificando la risposta, che cosa si ottiene nel file `FF` dopo aver eseguito il seguente comando

```
prog.out > FF
```

3. 10 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("tentativo.txt", R_OK)<0)
        printf("access error per tentativo.txt");
    else
        printf("access OK\n");

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

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

    exit(0);
}
```

Se fosse

```
-rwxrwxrwx 1 rescigno 10932 Jun 4 10:45 a.out
-r----- 1 rescigno 1891   Jun 4 09:45 prova.txt
-r----- 1 studente 1891   Jun 4 09:45 tentativo.txt
-r----- 1 rescigno 1891   Jun 4 09:45 risultato.txt
```

a) supponendo che si sia loggato **studente**,

- 1) dire che cosa succede dando **a.out**;
- 2) assumendo 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.*

b) supponendo che si sia loggato **rescigno**,

- 1) dire che cosa succede dando **a.out**;
- 2) assumendo 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. 8 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. 10 punti

Un sistema é dotato di una memoria centrale con TLB che ha le seguenti caratteristiche: Tempo di accesso alla TLB = 10 ns - Tempo di accesso alla memoria = 40 ns

a) Determinare il valore della Hit ratio della TLB necessario ad avere un tempo di accesso medio pari a 15 ns.

b) Si discuta se, aumentando la dimensione della TLB, tale Hit ratio tenderebbe ad aumentare o diminuire.

6. 15 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

7. 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

8. 10 punti

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