

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 | 6 | 7 | tot |
| /8 | /43 | /10 | /10 | /10 | /10 | /10 | /100 |

NOTA: L'esercizio numero 3 NON deve essere eseguito dagli studenti che devono sostenere SO da 9 cfu

1. 7 punti

Si supponga di mandare in esecuzione il seguente programma:

```
int main(void)
{
    pid_t p;

    fork();

    p = fork();

    fork();

    fork();

    if (p==0) {printf("Ho p=0" \n);}

    exit(0);
}
```

Usando un albero che mostra l'evoluzione dei processi, dire quante volte viene scritta su standard output la frase "Ho p=0".

Si segnino sull'albero i processi che effettuano la scritta.

2. 43 punti

Sia `File` un file contenente i caratteri `abcdef`, e sia `File1` un file contenente i caratteri `pqrs`.

```
-rw----- 1 rescigno 6 Jun 4 09:45 File
-rw--rw--- 1 rescigno 6 Jun 4 09:45 File1
-rw----- 1 tizio 6 Jun 4 09:45 File2
```

(a) (10 punti)

Si assuma che l'utente `tizio` abbia nella sua `cwd` i file `File`, `File2` 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);

    if ((fd2=open("File2",O_RDWR))<0) {printf("errore\n"); exit(0);}
    read(fd2,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 si ottiene settando a 1 il `set-user-id` di `prog.out`?

(b) (8 punti)

Si assuma da qui in avanti che l'utente `rescigno` abbia `File` e `File1`.

Si scriva un programma programma `C` che **in successione**

- mostri su standard output il contenuto di `File`, utilizzando il comando `cat` e una funzione `exec`,

- ridirezioni il contenuto di `File1` in un nuovo file di nome `N-File1`, utilizzando il comando `cat` e una funzione `exec`. (NON va usato l'operatore di ridirezione).

(c) (7 punti)

Si scriva un programma C che

- crei un hard link `H-File` a `File` ed un symbolic link `S-File1` a `File1`;
- setti i permessi di accesso di `H-File` uguali a quelli di `File1`;
- mostri su standard output il numero di link di `File`, `File1`, `H-File` a `S-File1`.

(d) (10 punti)

Giustificando la risposta, si indichino per ciascuno dei file `File`, `File1`, `H-File` a `S-File1`

- la size,
- i permessi di accesso,
- il numero di link,
- il contenuto

(e) (8 punti)

Si assuma di aver creato H-File, S-File1 (mandando in esecuzione il programma scritto in (c)). Sia altro-prog.out l'eseguibile relativo al programma seguente.

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

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

(5)    unlink("H-File");
(6)    unlink("File");
(6)    unlink("File1");

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

(9)    if (fd2=open("S-File1", 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. 10 punti

La formula dell'*anidride carbonica* é CO_2 .

Un sistema é costituito da due processi concorrenti: un processo costruttore di atomi di carbonio, chiamato `proc_carbonio`, e un processo costruttore di atomi di ossigeno, chiamato `proc_ossigeno`; ogni processo ciclicamente produce un atomo e poi stampa il simbolo dell'elemento chimico prodotto.

Si vuole che l'attivit  dei due processi sia sincronizzata in modo tale che la stampa prodotta segua l'ordine COOCOOCOO...

Descrivere con uno pseudocodice i due processi `proc_carbonio` e `proc_ossigeno` utilizzando, per la sincronizzazione, i semafori.

N.B. Nel codice si scriva "*produce ossigeno*" per indicare il momento in cui il processo produce ossigeno, "*produce carbonio*" per indicare il momento in cui il processo produce idrogeno e "*stampa O*" o "*stampa C*" per indicare il momento in cui si stampa.

4. 10 punti

Quattro processi arrivano al tempo indicato, consumano la quantità di CPU indicata e hanno le priorità (numero piccolo corrisponde ad alta priorità) indicate nella tabella sottostante:

| Processo | T. di Arrivo | Burst | Priorità |
|----------|--------------|-------|----------|
| P_1 | 0 | 10 | 3 |
| P_2 | 5 | 1 | 1 |
| P_3 | 3 | 2 | 3 |
| P_4 | 10 | 1 | 2 |
| P_5 | 11 | 5 | 4 |

Calcolare il turnaround ed il waiting time di ogni processo nel caso sia usato l'algoritmo di *Scheduling SJF con prelazione*.

Riportare il diagramma di GANTT usato per il calcolo.

5. 10 punti

Si consideri un disco con blocchi di 1 Kbyte, gestito attraverso una FAT.

3 byte sono necessari per indicare un blocco del disco.

Rispondere, giustificando le risposte, alle seguenti domande:

1. Qual è la massima capacità del disco, espressa in blocchi e in byte?
2. Quanti byte occupa la FAT?
3. Supponendo che il file **pippo** occupi i blocchi fisici 15, 30, 16, 64 e 40 (nell'ordine), quali sono gli elementi della FAT che descrivono il file e quale è il loro contenuto?

6. 12 punti

Data la seguente stringa di riferimenti a pagine,

10 6 2 4 6 8 3 1 4 5 11 8 7 6 10 9 7 8 11 2

Assumendo di avere una memoria fisica costituita da 5 frame blocchi., si riporti la sequenza di occupazioni della memoria riservata al processo nel caso in cui l'algoritmo di sostituzione delle pagine sia

(a) OPT

(b) LRU

(c) Considerando che ogni accesso alla pagina 6 é in scrittura, si determini in ciascun dei casi precedenti il numero di page fault ed il numero di accessi a disco.

7. 10 punti

Si consideri un sistema a memoria virtuale con indirizzi logici a 30 bit, indirizzi fisici a 20 bit, pagine da 512 byte.

- a) Si descriva la struttura dell'indirizzo logico e dell'indirizzo fisico;
- b) Si determini il numero di frame di cui é costituito lo spazio di indirizzamento fisico e il massimo numero di pagine di cui é costituito lo spazio di indirizzamento logico;
- c) esiste in tale sistema la frammentazione? Se si, di che tipo é?

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