

Cognome e nome: _____ Matricola: _____ Posto: _____

REGOLE. Si inseriscano le proprie risposte su questi fogli.

Non è consentita la consultazione di libri o appunti in forma cartacea o elettronica, né l'uso di palmari e telefoni cellulari.

Per superare l'esame, riportare generalità e matricola negli spazi indicati e acquisire almeno 18 punti su tutti i quesiti.

Per la convalida e registrazione del voto finale il docente si riserva di proporre al singolo candidato una prova orale.

Quesito 1: Rispondere, motivando, ai seguenti quesiti.

[1.A] Si consideri il caso di un *file system* **NTFS**; si discuta brevemente se e perché la struttura di *overhead* per memorizzare un *file* risenta della contiguità con cui i blocchi di tale *file* sono disposti sul disco fisso.

[1.B] Si consideri un sistema con un solo processo attivo, il quale sta eseguendo la seguente porzione di codice:

```
pid_1 = fork();
pid_2 = fork();
pid_3 = fork();
```

Quanti processi risulteranno attivi alla fine dell'esecuzione di queste tre line di codice? (Si motivi la risposta)

Quesito 2:

La seguente soluzione del problema dei lettori-scrittori contiene **cinque** banali errori o mancanze. Lo studente ne modifichi il codice tramite aggiunte, cancellazioni e correzioni. Il risultato dovrà rappresentare una versione corretta, realizzata apportando il minor numero possibile di modifiche all'originale qui di seguito.

(Per coloro che avessero studiato solo sul libro di testo: *P*, corrisponde a *down*, *V* corrisponde a *up*)

Inoltre, lo studente dichiara i valori iniziali delle variabili numeroLettori, mutex e database.

```
Void Lettore (void) {
    while (true) {
        P(mutex);

        numeroLettori++;

        if (numeroLettori==1) V(database);

        V(mutex);

        // leggi il dato

        numeroLettori--;

        if (numeroLettori==0) V(database);

        // usa il dato letto

    }
}
```

```
void Scrittore (void) {
    while (true) {
        // prepara il dato da scrivere

        V(database);

        // scrivi il dato

        P(database);

    }
}
```

VALORI INIZIALI DI TUTTE LE VARIABILI?

Un sistema ha 4 processi e 5 risorse da ripartire. L'attuale allocazione e i bisogni massimi sono i seguenti:

<i>Processo</i>	<i>Allocate</i>	<i>Massimo</i>
<i>A</i>	1 0 1 1 2	1 1 1 4 2
<i>B</i>	2 0 1 1 1	2 2 2 1 3
<i>C</i>	1 1 1 0 0	2 1 1 0 4
<i>D</i>	1 1 1 0 1	1 1 2 1 3

[3.A] Considerando il vettore delle risorse disponibili uguale a $[0 \ 0 \ 1 \ 2 \ x]$, si discuta per quale valore minimo di x questo sia uno stato sicuro e quando invece sia a rischio di deadlock.

[3.B] Per risolvere l'esercizio lo studente ha di fatto ripetutamente utilizzato una parte di un noto algoritmo. Tale algoritmo assegna risorse a processi solo se l'assegnazione fa rimanere il sistema in uno stato sicuro. Come si chiama questo algoritmo?

Si consideri la seguente serie di riferimenti a pagine di memoria: 1, 2, 3, 4, 2, 1, 5, 6, 2, 1, 2, 1, 7, 6, 2, 1, 3.

Si considerino le seguenti politiche di rimpiazzo LRU ed Optimal.

Quanti *page fault* avvengono considerando una RAM con solo 4 *page frame* ed inizialmente vuota?

Si completino inoltre le tabelle mostrando ad ogni istante il contenuto dei 4 page frame di cui è composta la RAM (non è necessario che lo studente mantenga un preciso ordine delle pagine virtuali nelle *page frame*).

Politica di rimpiazzo **LRU**; totale *page fault*?

[illegible]Politica di rimpiazzo **Optimal**; totale *page fault*?[illegible]

Cognome e nome: _____ **Matricola:** _____ **Posto:** _____

Quesito 5:

Sia data una partizione di disco ampia 64 MB organizzata in blocchi dati di ampiezza 1 kB. In caso serva, si consideri l'ipotesi di contiguità nulla di un file (ciascun blocco si trova su disco in posizione non adiacente al blocco precedente e a quello successivo nella composizione del file).

Si consideri che su tale partizione è in uso un file system di tipo ext2 con i-node ampi un blocco e si assuma che l'i-node principale contenga 10 indici di blocco e 1 indice di I e II indizione ciascuno.

Considerando che la dimensione di un file include sia i Byte dei blocchi di dati, sia i Byte occupati dagli i-node necessari a indirizzare i blocchi di dati, si determini l'ampiezza massima di un file consentita dal sistema.

Cognome e nome: _____ Matricola: _____ Posto: _____

Soluzione

Soluzione al Quesito 1

[1.A] Sì ne risente in quanto l'attributo "dati" di NTFS contiene tante coppie valori quante sono le sequenze contigue su disco. Di conseguenza un file enorme contiguamente scritto su disco richiederà unicamente una coppia valori, mentre un file piccolo composto da N blocchi nessuno contiguo a nessun altro richiederà in MFT un numero N di coppie valori.

[1.B] Avremo 8 processi attivi poiché a quello inizialmente attivo se ne aggiunge un altro con la prima fork; ciascuno dei due esegue la seconda fork dunque attivando altri due processi; tutti e quattro eseguono la terza e ultima fork dunque raggiungendo il numero di 8 processi attivi alla fine della porzione di codice.

Soluzione al Quesito 2

```
void Lettore (void) {
    while (true) {
        P(mutex);
        numeroLettori++;
        if (numeroLettori==1) P(database);
        V(mutex);
        // leggi il dato
        P(mutex);
        numeroLettori--;
        if (numeroLettori==0) V(database);
        V(mutex);
        // usa il dato letto
    }
}
```

```
void Scrittore (void) {
    while (true) {
        // prepara il dato da scrivere
        P(database);
        // scrivi il dato
        V(database);
    }
}
```

Valori iniziali delle variabili

numeroLettori: 0; (variabile di tipo intero)

mutex: 1 (variabile di tipo semaforo)

database: 1 (variabile di tipo semaforo)

Soluzione al Quesito 3

[3.A] La matrice delle necessità (massimo numero di risorse richieste dal processo - risorse allocate al processo) è la seguente:

```
0 1 0 3 0
0 2 1 0 2
1 0 0 0 4
0 0 1 1 2
```

Se $x = 0$ oppure $x = 1$, la deadlock è immediata.

Se $x = 2$, il processo D può essere eseguito fino alla fine. Quando ha finito, il vettore delle risorse disponibili è [1 1 2 2 3]. Sfortunatamente ora il sistema si trova in deadlock.

Se $x = 3$, dopo D, il vettore delle risorse disponibili è [1 1 2 2 4] e C può essere eseguito. Dopo il suo completamento, il vettore delle risorse disponibili diventa [2 2 3 2 4]; questo permette a B di essere eseguito e completato. Il vettore delle risorse disponibili diviene dunque [4 2 4 3 5], permettendo il completamento di A.

Quindi il valore più piccolo di x per evitare il verificarsi di deadlock è 3.

[3.B] L'Algoritmo del Banchiere (Banker's Algorithm)

Cognome e nome: _____ Matricola: _____ Posto: _____

Soluzione al Quesito 4Politica di rimpiazzo **LRU**; totale *page fault*? **8** (quelli in grassetto)

r1	r2	r3	r4	r2	r1	r5	r6	r2	r1	r2	r1	r7	r6	r2	r1	r3
1	2	3	4	2	1	5	6	2	1	2	1	7	6	2	1	3
	1	2	3	4	2	1	5	6	2	1	2	1	7	6	2	1
		1	2	3	4	2	1	5	6	6	6	2	1	7	6	2
			1	1	3	4	2	1	5	5	5	6	2	1	7	6

Politica di rimpiazzo **Optimal**; totale *page fault*? **8** (quelli in grassetto)

r1	r2	r3	r4	r2	r1	r5	r6	r2	r1	r2	r1	r7	r6	r2	r1	r3
1	2	3	4	4	4	5	6	6	6	6	6	6	6	6	6	6
	1	2	3	3	3	3	3	3	3	3	3	7	7	7	7	7
		1	2	2	2	2	2	2	2	2	2	2	2	2	2	2
			1	1	1	1	1	1	1	1	1	1	1	1	1	1

L'ultimo rimpiazzo di Optimal può avvenire al posto di qualsiasi pagina (non necessariamente della 1).

Soluzione al Quesito 5

Essendo la memoria secondaria ampia 64 MB e i blocchi dati ampi 1 KB, è immediato calcolare

che sono necessari: $\left\lceil \frac{64MB}{1KB} \right\rceil = 64\text{ K} = 2^6 \times 2^{10} = 2^{16}$ indici, la cui rappresentazione binaria banalmente richiede 16 *bit* (2 B).Sotto queste ipotesi, il *file* di massima dimensione rappresentabile dall'architettura ext2fs fissata dal quesito sarà composto da:

- 10 blocchi, ovvero 10 KB, risultanti dall'utilizzo dei corrispondenti indici diretti presenti nell'i-node principale, al costo di 1 i-node, pari a 1KB (dimensione di un blocco)
- $\left\lfloor \frac{1KB}{2B} \right\rfloor = 512$ blocchi, ovvero 512 KB, risultanti dall'utilizzo dell'intero i-node secondario denotato dall'indice di I indirezione presente nell'i-node principale, al costo di 1 i-node, pari a 1 KB
- $512^2 = 2^{18} = 256\text{ K}$ blocchi, ovvero 256 MB, risultanti dall'utilizzo dell'indice di II indirezione, al costo di $1 + 512 = 513$ i-node, pari a: $513 \times 1\text{ KB} = 513\text{ KB}$

In totale la dimensione di un file così costruito sarebbe di 10 KB + 512 KB + 256 MB di dati più 1 KB + 1 KB + 513 KB di i-node. Tale totale supera la dimensione della partizione (64 MB); la dimensione massima di un file nel sistema considerato è quindi limitato alla dimensione della partizione stessa, ovvero 64 MB.