

SISTEMI OPERATIVI E LABORATORIO
(Esonero e Scritto - Indirizzo Sistemi e Reti)
11 luglio 2008

Cognome: _____ **Nome:** _____
Matricola: _____

Ricordate che non potete usare calcolatrici o materiale didattico. Siate sintetici nelle vostre risposte, anche quando è richiesto di motivarle, sono sufficienti poche righe per rispondere correttamente. (Si ricorda che gli studenti degli anni precedenti devono sostenere l'intero scritto).

ESERCIZI RELATIVI ALLA PARTE DI TEORIA DEL CORSO

(il punteggio conseguito farà media con quello ottenuto nella parte di laboratorio. E' comunque necessario prendere almeno 18 punti per considerare passata la parte di teoria o la parte di laboratorio.)

ESERCIZIO 1 (9 punti)

Si consideri un sistema in cui la tabella delle pagine di un processo può avere al massimo 256 entry. Un indirizzo fisico generato dal sistema è scritto su 16 bit, e la RAM è suddivisa in 128 frame.

(a) Quanto è grande lo spazio di indirizzamento logico del sistema? (esplicitate i calcoli che fate)

Un frame è grande $2^{16} / 2^7$ byte = 2^9 byte. Lo spazio di indirizzamento logico è quindi grande $2^9 * 2^8 = 2^{17}$ byte = 128K byte

(b) Quale dei seguenti indirizzi esadecimali NON è certamente un indirizzo fisico (colonna di sinistra) o logico (colonna di destra) corretto per il sistema sopra descritto? (marcate l'indirizzo scelto con una croce nello spazio apposito)

	Indirizzi fisici	indirizzi logici
1.	<input type="checkbox"/> FFFF	<input checked="" type="checkbox"/> 2000
2.	<input type="checkbox"/> 1FFF	<input type="checkbox"/> 1000F
3.	<input checked="" type="checkbox"/> 1000F	<input type="checkbox"/> FFFF
4.	<input type="checkbox"/> 00000	<input type="checkbox"/> 2FFF

(c) Nelle tabelle delle pagine di questo sistema deve essere presente il bit di validità? (motivate la vostra risposta)

Si, perché il sistema deve implementare la memoria virtuale (in quanto lo spazio di indirizzamento logico è più grande di quello fisico)

(d) Quanto spazio occupa, al massimo la tabella delle pagine di un processo di questo sistema, se il sistema usa l'algoritmo della *seconda chance migliorato*? (è sufficiente riportare la formula aritmetica che esprime la dimensione in byte o bit)

Sono necessari 7 bit per memorizzare il numero di un frame. Inoltre è necessario memorizzare anche il bit di validità, il reference bit e i dirty bit. Ogni entry della PT occupa quindi 10 bit, e la PT

è grande $10 \text{ bit} * 256 \text{ entry}$ (oppure, se si assume di usare due byte per ogni entry, $2 * 256 = 512 \text{ byte}$)

Elencate almeno due semplici tecniche che possono diminuire la probabilità che un sistema vada in thrashing

Vedere lucidi della sezione 9.6.1

ESERCIZIO 2 (9 punti)

- a) Si consideri il problema dei lettori e scrittori, dove i codici del generico scrittore e del generico lettore sono riportati qui di seguito.

Inserite le operazioni di wait e signal mancanti necessarie per il corretto funzionamento del sistema, indicando anche il semaforo mancante ed il suo valore di inizializzazione.

semafori e variabili condivise necessarie con relativo valore di inizializzazione:

semaphore mutex = 1;

semaphore scrivi = 1;

int numlettori = 0;

“scrittore”

{

wait(scrivi);

Esegui la scrittura del file

signal(scrivi)

}

“lettore”

{

wait(mutex);

numlettori++;

if numlettori == 1 **wait(scrivi);**

signal(mutex);

... leggi il file ...

wait(mutex);

numlettori--;

if numlettori == 0 **signal(scrivi);**

signal(mutex);

- b) Indicate una semplice situazione in cui un processo può abbandonare *volontariamente* la CPU mentre è all'interno di una propria sezione critica.

Ad esempio se all'interno della sua sezione critica il processo esegue una system call wait su un semaforo con valore minore o uguale a zero.

- c) Quali sono le proprietà che deve garantire una corretta soluzione al problema della sezione critica?

Progresso, attesa limitata e mutua esclusione.

- d) quali problemi si possono verificare se le tre proprietà non sono rispettate?

Deadlock; starvation; due o più processi contemporaneamente in sezione critica.

- e) Una soluzione al problema della sezione critica basata sul busy waiting è adatta per un sistema che usa un algoritmo di scheduling round robin (motivate la vostra risposta)?

No, un processo che cerca di entrare in una sezione critica occupata spreca inutilmente tutto il suo quanto di tempo senza riuscire ad entrare nella sezione critica (che verrà liberata solo quando il processo che la occupa riottiene l'uso della CPU).

ESERCIZIO 3 (9 punti)

- a) Descrivete con un semplice esempio la variante FAT dell'allocazione concatenata.

Vedere i lucidi della sezione 11.4.2

- b) Che svantaggi ha l'uso della FAT?

Per essere efficiente la FAT va tenuta in RAM, sottraendo così spazio ai processi. Periodicamente deve essere salvata sull'hard disk. Se viene persa la FAT si perdono in un colpo solo tutti i file memorizzati sull'hard disk.

- c) Se un hard disk è grande 16 Giga byte con blocchi da 1 K byte, quanto spazio occupa la FAT di quel disco? (esplicitate i calcoli che fate)

Il disco è grande 2^{34} byte, per cui la FAT deve avere $2^{34} / 2^{10} = 2^{24}$ entry. Per memorizzare il numero di un blocco in una entry sono quindi necessari 3 byte, e la FAT occupa $3 * 2^{24} = 48$ megabyte