

SISTEMI OPERATIVI
13 febbraio 2018 – corso A

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

Ricordate che non potete usare calcolatrici o materiale didattico, e che potete consegnare al massimo tre prove scritte per anno accademico.

ESERCIZI RELATIVI ALLA PARTE DI TEORIA DEL CORSO

ESERCIZIO 1 (7 punti)

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

Inserite le istruzioni mancanti necessarie per il funzionamento del sistema secondo la soluzione vista a lezione, indicando anche il semaforo e i valori di inizializzazione mancanti.

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) La soluzione riportata garantisce l'assenza di starvation? (motivate la vostra risposta)

no: se almeno un lettore è in fase di lettura e continuano ad arrivare altri lettori, nessun processo scrittore riuscirà mai a scrivere nel file.

- c) Sia dato un sistema operativo time sharing che implementa la memoria virtuale. Indicate separatamente le cause per cui un processo in stato *running* potrebbe abbandonare **volontariamente** e **involontariamente** il processore.

volontariamente

termina

operazione di I/O

chiamata di fork

chiamata di wait con semaforo < 1

involontariamente

quanto di tempo scaduto

page fault

- d) Tre processi P1 P2 e P3 arrivano uno dopo l'altro – nell'ordine indicato ma in tempi diversi – in coda di ready. I tre processi sono caratterizzati da un unico burst di CPU e da nessun burst di I/O. In quale caso il waiting time medio del sistema relativo ai tre processi risulta lo stesso sia adottando un algoritmo di scheduling FCFS che adottando un algoritmo di scheduling SJF preemptive?

È sufficiente che: $\text{Burst_P1} \leq \text{Burst_P2} \leq \text{Burst_P3}$

- e) Elencate le tecniche che un SO usa per conservare sempre il controllo della macchina anche quando non è in esecuzione codice del SO stesso.

Uso di istruzioni privilegiate, uso di un timer che restituisce il controllo al SO quando scade, uso di registri speciali per controllare l'accesso ad aree di ram riservate ai vari processi (oppure, paginazione e/o segmentazione)

- f) Riportate lo pseudocodice che descrive l'implementazione dell'operazione di Wait, e dite quale informazione fornisce il valore corrente della variabile semaforica.

Per il codice si vedano i lucidi della sezione 6.5.2. Se la variabile è positiva, quanti processi possono superare il semaforo senza addormentarsi su di esso, se negativa il suo valore assoluto ci dice quanti processi sono addormentati sulla waiting list del semaforo.

ESERCIZIO 2 (7 punti)

Un sistema usa una paginazione a due livelli, la tabella delle pagine esterna del processo più grande occupa esattamente un frame, e all'interno di una tabella delle pagine vengono usati 20 (esadecimale) bit per scrivere il numero di un frame. Si sa che in un indirizzo logico l'offset massimo all'interno di una pagina è 7FF.

- a) Quanto sono grandi lo spazio di indirizzamento logico e fisico del sistema? (esplicitate i calcoli che fate, e lasciate ed esprimete la grandezza dei due spazi come potenze di due).

Una pagina, e quindi un frame, sono grandi $2^{11}=2048$ byte, e poiché vengono usati 4 byte (20 esadecimale bit) per scrivere il numero di un frame, la tabella delle pagine esterna più grande contiene 512 numeri di frame. La tabella delle pagine interna del processo più grande è dunque spezzata in 512 frame ognuno dei quali contiene 512 entry.

Lo spazio di indirizzamento logico è quindi di: $2^9 * 2^9 * 2^{11} = 2^{29}$ byte

Lo spazio di indirizzamento fisico è di $2^{32} * 2^{11} = 2^{43}$ byte

b) Supponendo che non si verifichino page fault, assumendo che il sistema sia dotato di un TLB perfetto con un hit rate medio dell'80%, e che il tempo di accesso in RAM sia di 50 microsecondi, qual è il medium access time del sistema descritto sopra? (esplicitate i calcoli che fate)

$$mat = 0,8 * 50 + 0,2 * (50 + 50 + 50) = 40 + 30 = 70 \text{ microsecondi}$$

c) Nel sistema sopra riportato, è necessario stabilire una politica di allocazione (locale o globale che sia) dei frame? (motivate la vostra risposta)

Solo se si vuole implementare la memoria virtuale (cosa non necessaria perché lo spazio fisico è maggiore di quello logico).

d) È possibile che un processo P che gira su un sistema A dotato di memoria virtuale venga eseguito più velocemente che su un sistema B del tutto identico ad A ma non dotato di memoria virtuale? (motivate la vostra risposta)

Sì, ad esempio se P non genera mai page fault e non è necessario caricare in memoria tutto il codice di P per l'esecuzione.

e) perché l'algoritmo della seconda chance è una approssimazione di LRU?

Perché usando il reference bit riesce a discriminare tra pagine riferite di recente e non riferite di recente

f) Per evitare il problema del thrashing conviene aumentare la dimensione della RAM o la dimensione dell'area di SWAP? (motivate la vostra risposta)

La dimensione della RAM perché così diminuisce la probabilità che un processo generi page fault.

ESERCIZIO 3 (6 punti)

Un hard disk ha la capienza di 2^{40} byte, ed è formattato in blocchi da 2048 byte.

a) Quanti accessi al disco sono necessari per leggere l'ultimo blocco di un file A della dimensione di 2^{13} byte, assumendo che sia già in RAM il numero del primo blocco del file stesso e che venga adottata una allocazione concatenata dello spazio su disco? (motivate la vostra risposta)

5. Ogni blocco infatti memorizza 2044 byte di dati più 4 byte di puntatore al blocco successivo (infatti, $2^{40}/2^{11} = 2^{29}$), per cui sono necessari 5 blocchi per memorizzare l'intero file.

b) Quanto sarebbe grande, in megabyte, la FAT di questo sistema? (motivate numericamente la vostra risposta)

La FAT è un array con una entry per ciascun blocco dell'hard disk e che contiene il numero di un blocco, per cui: $2^{29} * 2^2 \text{ byte} = 2 \text{ gigabyte}$

c) Quali sono gli svantaggi nell'uso della FAT?

Per garantire un accesso efficiente ai file deve essere sempre tenuta in RAM, se viene persa si perdono tutte le informazioni sul file system, e deve quindi essere periodicamente salvata su disco.

d) Disegnate la FAT di un hard disk formato da 2^4 blocchi e contenente un unico file memorizzato, nell'ordine, nei blocchi 9, 5, 11, 3. Indicate anche dove è memorizzato il numero del primo blocco del file.

Si veda la figura 11.7 della sezione 11.4.2 dei lucidi.

e) dov'è memorizzato il pathname assoluto di un file?

Da nessuna parte.