

SISTEMI OPERATIVI
13 settembre 2019
corso A nuovo ordinamento

Cognome: _____ Nome: _____
Matricola: _____

ESERCIZI RELATIVI ALLA PARTE DI TEORIA DEL CORSO

ESERCIZIO 1 (7 punti)

- a) 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 il semaforo (o i semafori) mancante/i e il relativo 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);  
}
```

b1) Riportate, a vostra scelta, il diagramma di stato o il diagramma di accodamento di un sistema time sharing **che usa un algoritmo di rimpiazzamento delle pagine**;

b2) Facendo riferimento al sistema del punto b1, elencate le ragioni per cui un processo può passare dallo stato *waiting* allo stato *ready to run*

b1) Per i diagrammi si vedano i lucidi della sezione 3.1.2 (p. 6), 3.2.1 (p. 18) e il lucido a p. 9 del cap. 9.

b2) è stata completata una operazione di I/O; signal sul semaforo su cui il processo era addormentato; è arrivata in ram la pagina mancante che aveva causato un page fault del processo.

c) riportate lo pseudocodice relativo alla corretta implementazione dell'operazione di *signal* e dite cosa fa la system call usata nel codice della signal.

Si vedano i lucidi della sezione 6.5.2.

d) Che svantaggio da l'uso dei processi al posto dei thread?

Un insieme di peer thread condivide lo spazio di indirizzamento. Per questa ragione, il context switch tra peer thread e la creazione di un nuovo peer thread richiedono molto meno tempo delle corrispondenti operazioni sui processi.

e) 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: $\text{Burst_P1} \leq \text{Burst_P2} \leq \text{Burst_P3}$

ESERCIZIO 2 (7 punti)

In un sistema operativo un indirizzo fisico è scritto su 29 bit, e un frame è grande 8 Kbyte. E' noto che la tabella delle pagine più grande del sistema occupa esattamente un frame.

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

Poiché 8 Kbyte = 2^{13} byte, il numero di un frame del sistema è scritto su $29 - 13 = 16$ bit, ossia 2 byte, dunque in un singolo frame possiamo scrivere al massimo $4096 = 2^{12}$ entry da due byte, e lo spazio di indirizzamento logico è pari a $2^{12} * 2^{13} = 2^{25}$ byte.

b) Il sistema potrebbe soffrire di thrashing? (motivate la risposta)

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

c) Assumiamo ora che il sistema adotti una Inverted Page Table, e che nel sistema possano essere attivi contemporaneamente non più di 4096 processi, quanto sarebbe grande questa tabella?

Per rispondere alla domanda è necessario ipotizzare il numero di bit usati per scrivere il PID di un processo. Assumiamo ad esempio di usare 10 bit. La IPT ha un numero di entry pari al numero di frame del sistema, e ogni entry contiene il numero di una pagina (12 bit) e il numero di un PID (12 bit). Ne segue che la IPT è grande $2^{16} * 3$ byte.

d) Elencate tre vantaggi dell'uso delle librerie dinamiche

Possono essere condivise tra più processi, per cui occupano meno spazio in RAM di quelle statiche. Vengono caricate in RAM solo se sono chiamate, per cui i processi partono più rapidamente. Se vengono aggiornate non occorre ricompilare i programmi che le usano.

e) E' vera la seguente affermazione (motivate la vostra risposta)? *In un qualsiasi sistema operativo che implementi la memoria virtuale, l'effettivo tempo di esecuzione di un programma dipende molto anche dal modo con cui accede ai propri dati in RAM.*

L'affermazione è vera. Si veda ad esempio la sezione 9.9.5 (struttura dei programmi).

f) In un sistema in funzione si osserva in un dato istante che la CPU ha una percentuale di utilizzo del 90%. Si decide allora di mandare in esecuzione un certo numero di nuovi processi per sfruttare quel 10% di CPU inutilizzata, ma la percentuale di utilizzo della CPU scende invece di salire. Come si può spiegare la cosa?

Il sistema è in thrashing. I nuovi processi hanno sottratto frame a quelli esistenti, aumentando la frequenza di page fault e il tempo passato dai processi stessi nello stato *waiting for page*.

ESERCIZIO 3 (6 punti)

a) In Unix che cosa succede nelle strutture interne al sistema, quando viene creato un nuovo link simbolico ad un file A già esistente?

Una entry viene aggiunta nella cartella in cui è stato creato il nuovo link simbolico, e nella entry viene scritto il numero di un nuovo index-node. In un campo opportuno del nuovo index node viene scritto il path name di A passato come argomento durante la creazione del link.

b) perché sono necessari i link simbolici?

Per permettere di costruire collegamenti fra le directory, tra le quali non sono ammessi i link fisici (in Unix, i link simbolici permettono anche link fra file che stanno su partizioni diverse degli hard disk).

c) In quale caso, e perché, l'allocazione indicizzata dello spazio in memoria secondaria è particolarmente **vantaggiosa**?

Nel caso di file molto grandi che devono essere acceduti in modo diretto. Infatti, l'uso dei blocchi indice assicura un accesso diretto efficiente, e l'uso di uno o più blocchi indice rappresenta un overhead marginale rispetto al numero totale di blocchi occupati dal file.

d) Descrivete le caratteristiche di un sistema RAID nella configurazione 1? (usate un disegno se vi può essere utile per descrivere meglio le caratteristiche del RAID 1)

Usiamo il RAID 1 se abbiamo bisogno della massima affidabilità e velocità di accesso ai dati. Se il RAID è composto da n dischi, n/2 dischi sono usati per i dati, e gli altri n/2 come mirroring. I blocchi logici del disco virtuale sono suddivisi in strip distribuiti sui dischi di dati e replicati sui dischi di mirroring, per cui l'accesso ai dati è velocizzato dalla possibilità di leggere in parallelo strip che risiedono su dischi diversi

(usando eventualmente anche i dischi di mirroring per quegli strip che si trovano sullo stesso disco di dati)

e) Perché il RAID 5 è preferibile al RAID 4?

Perché distribuisce gli strip di parità e dei dati fra tutti i dischi, uniformando così il carico di lavoro e l'usura dei dischi stessi.