

# SISTEMI OPERATIVI

## 24 febbraio 2011

Cognome: \_\_\_\_\_ Nome: \_\_\_\_\_  
Matricola: \_\_\_\_\_

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

### ESERCIZI RELATIVI ALLA PARTE DI TEORIA DEL CORSO

#### ESERCIZIO 1 (5 punti)

Un sistema operativo adotta un algoritmo di scheduling preemptive. Quattro processi arrivano al tempo indicato e consumano la quantità di CPU indicata nella tabella sottostante)

Processo	T. di arrivo	Burst
P1	0	11
P2	2	6
P3	5	2
P4	9	2

a)  
Quali sono il turnaround medio e il waiting time medio migliori (ossia ottimali) che potrebbero essere ottenuti per lo scheduling dei quattro processi riportati?. **RIPORTATE IL DIAGRAMMA DI GANTT USATO PER IL CALCOLO.** (lasciate pure i risultati numerici sotto forma di frazione)

#### **Turnaround medio (usando SJF):**

(0)....P1 ...(2) .... P2....(5)....P3....(7)....P2....(10)....P4...(12)....P1....(21)  
 $P1 = 21$ ;  $P2 = 8$ ;  $P3 = 2$ ;  $P4 = 3$ ;  $34/4 = 8,5$

#### **Waiting time medio (usando SJF):**

(basta ricordarsi di sottrarre al turnaround di ogni processo, la durata del suo burst):  
 $P1 = 10$ ;  $P2 = 2$ ;  $P3 = 0$ ;  $P4 = 1$ ;  $13/4 = 3,25$

b)  
Nel caso reale il sistema operativo potrà dei risultati migliori, uguali o peggiori di quelli ottenuti alla risposta precedente (motivate la vostra risposta)?

Peggiori, perché SJF non è implementabile

c)  
Descrivete brevemente un algoritmo di scheduling a code multiple con retroazione

Usa più code di ready, gestite ciascuna con una diversa politica di scheduling. Un processo può essere spostato da una coda all'altra in base a come si è comportato l'ultima volta che gli è stata assegnata la CPU.

d)

Indicate lo pseudocodice di tre processi Pa, Pb e Pc che eseguono rispettivamente le procedure A, B e C rispettando le seguenti condizioni: A deve essere eseguito prima di B, e B deve essere eseguito prima di C. Indicate gli eventuali semafori che usate e il loro valore di inizializzazione.

sem1 = 0; sem2 = 0

Pa:      <A>  
         signal (sem1)

Pb: wait (sem1)  
     <B>  
     signal(sem2)

Pc: wait(sem2)  
     <C>

## **ESERCIZIO 2 (5 punti)**

Un sistema con memoria paginata usa un TLB con un hit-ratio del 90%, e un tempo di accesso di 10 nanosecondi. Un accesso in RAM richiede invece 0,09 microsecondi.

- a) Qual è, in nanosecondi, il tempo medio di accesso in RAM (esplicitate i calcoli che fate)?

$$T_{\text{medio}} = 0,9 * (90+10) + 0,1 * (2*90 + 10) = 90 + 19 = 109 \text{ nanosecondi}$$

- b) Il sistema viene ora dotato di memoria virtuale, usando come algoritmo di rimpiazzamento quello della *seconda chance migliorato*.

Il sistema può soffrire dell'anomalia di Belady?

Sì, perché nel caso peggiore l'algoritmo si comporta come FIFO, che soffre dell'anomalia di Belady.

- c) Quali informazioni conterrà ciascuna entry di una page table di un sistema che usa l'algoritmo della seconda chance migliorato?

Numero di un frame, bit di validità, bit di riferimento, dirty bit.

- d) Descrivete brevemente come avviene la prevenzione del thrashing nel sistema Windows.

Windows fissa una soglia minima di frame liberi che devono sempre essere presenti nel sistema. Ad ogni processo sono associati un insieme di lavoro minimo e massimo. Se nel sistema il numero di frame liberi scende sotto la soglia minima, ad ogni processo vengono rimosse tutte le pagine in eccesso rispetto al suo insieme di lavoro minimo (le pagine vengono scelte con l'algoritmo della seconda chance, o con FIFO).

## **ESERCIZIO 3 (4 punti)**

Un hard disk ha la capacità di 32 gigabyte, è formattato in blocchi da 200 (esadecimale) byte, e usa una qualche forma di allocazione indicizzata per memorizzare i file su disco. Sull'hard disk è memorizzato un file A grande 150 Kbyte. Nel rispondere alle domande sottostanti, specificate sempre le assunzioni che fate.

- a) Quante operazioni di I/O su disco sono necessarie per portare in RAM l'ultimo blocco del file A, assumendo che inizialmente sia presente in RAM solo la copia del file directory che "contiene" il file?

L'hard disk contiene  $2^{35}/2^9 = 2^{26}$  blocchi, e sono quindi necessari 4 byte per scrivere in numero di un blocco. Un blocco indice può quindi contenere al massimo  $512/4 = 128$  numeri di blocco. La risposta dipende poi dal tipo di allocazione indicizzata assunta:

- 1) Allocazione indicizzata a schema concatenato: sono necessari 3 blocchi indice per tenere traccia di tutti i blocchi del file. Se assumiamo che sia già in RAM il numero del primo blocco indice, sono necessarie 4 operazioni di I/O: lettura dei tre blocchi indice più lettura del blocco del file.
  - 2) Allocazione indicizzata a più livelli. È sufficiente usare uno schema a due livelli. Se assumiamo che sia già in RAM il numero del blocco indice esterno, sono necessarie 3 operazioni di I/O: lettura del blocco indice esterno, lettura di un blocco indice interno, lettura del blocco del file.
  - 3) Allocazione indicizzata Unix: Assumendo già in RAM il numero dell'index-node, sono necessarie 4 operazioni di I/O: le tre del punto 2) precedute dalla lettura dell'index-node.
- b) Nel caso di accesso ai dati di file molto piccoli, è più efficiente l'implementazione scelta in Windows con NTFS o quella scelta da Unix con gli index-node? (motivate la vostra risposta)

NTFS. Infatti per file molto piccoli, l'elemento che contiene gli attributi del file può contenere anche i dati del file stesso.

- c) Che cosa indica il campo *link counter* di un index-node?

Il numero di hard link ad un file in un file system Unix.

- d) Spiegate almeno una ragione per cui in Unix sono stati introdotti i link simbolici.

Per permettere i link tra cartelle e per realizzare link tra file system diversi.