

# Sistemi Operativi

## 14 giugno 2010

### Compito

Si risponda ai seguenti quesiti, giustificando le risposte.

1. Si discutano i criteri di valutazione degli algoritmi di scheduling della CPU.

**Risposta:** I criteri più comunemente utilizzati per valutare la bontà degli algoritmi di scheduling della CPU sono i seguenti:

- Utilizzo della CPU: misurazione del carico della CPU.
- Throughput: conteggio del numero di processi completati nell'unità di tempo.
- Tempo di turnaround: misurazione del tempo totale impiegato per l'esecuzione di un processo.
- Tempo di attesa: misurazione del tempo che un processo passa nella coda ready.
- Tempo di risposta: misurazione del tempo impiegato da quando una richiesta viene inviata a quando si ottiene la prima risposta (criterio pensato per i sistemi time-sharing).
- Varianza del tempo di risposta: valutazione della variabilità del tempo di risposta.

Naturalmente, in generale, non esiste una soluzione ottima sotto tutti gli aspetti e per tutti i sistemi: occorre sempre valutare l'algoritmo in base alla natura ed agli scopi del sistema operativo (ad esempio, dei criteri che possono essere adatti alla valutazione degli algoritmi in un sistema time-sharing potrebbero portare a dei risultati non ottimali in un sistema batch o real-time).

2. (a) Si descriva il problema del barbiere.  
(b) Si completi il seguente codice che fornisce una soluzione al problema del barbiere (ogni riga corrisponde ad un'istruzione):

```
customers = 0    % conta il numero di clienti nel negozio
mutex = Semaphore(1) % protegge la variabile customers
customer = Semaphore(0)
barber = Semaphore(0)
```

Codice del cliente

```
mutex.down();
if customers = n+1 then mutex.up();
customers = customers +1;
...
customer.up();
...
getHairCut();
...
customers= customers -1;
...
```

Codice del barbiere

```
...
...
cutHair();
```

**Risposta:**

- (a) Il problema del barbiere è uno degli esempi paradigmatici di programmazione concorrente. La descrizione prevede che in un negozio ci sia un solo barbiere, una sedia da barbiere e  $n$  sedie per l'attesa e le seguenti regole:

- Quando non ci sono clienti, il barbiere dorme sulla sedia.
- Quando arriva un cliente, questo sveglia il barbiere se sta dormendo.
- Se la sedia è libera e ci sono clienti, il barbiere fa sedere un cliente e lo serve.
- Se un cliente arriva e il barbiere sta già servendo un cliente, si siede su una sedia di attesa se ce ne sono di libere, altrimenti se ne va.

# Sistemi Operativi

## 14 giugno 2010

### Compito

La soluzione del problema consiste nel programmare il barbiere e i clienti filosofi in modo da garantire l'assenza di deadlock e di starvation.

(b)

```
customers = 0    % conta il numero di clienti nel negozio
mutex = Semaphore(1)    % protegge la variabile customers
customer = Semaphore(0)
barber = Semaphore(0)
```

Codice del cliente

```
mutex.down();
if customers = n+1 then mutex.up();
else {
customers = customers +1;
mutex.up();
customer.up();
barber.down();
getHairCut();
mutex.down();
customers= customers -1;
mutex.up();
}
```

Codice del barbiere

```
while(true) { customer.down();
barber.up();
cutHair();
}
```

3. Un computer ha 4 frame, i cui istanti di caricamento, ultimo riferimento e i reference bit sono riportati nella seguente tabella:

Frame	Caric.	Rifer.	R
2	141	304	1
1	258	260	1
0	178	269	0
3	230	287	1

Dire quale pagina verrebbe liberata dall'algoritmo FIFO, da LRU e da CLOCK (in questo caso si supponga che l'ultimo frame controllato sia 3).

**Risposta:** L'algoritmo FIFO considera l'istante di caricamento e quindi libererebbe la pagina 2. LRU invece considera l'istante relativo all'ultimo riferimento alle pagine e quindi libererebbe la pagina 1. Infine CLOCK libererebbe la pagina 0 dato che è la prima che incontra con reference bit impostato a zero.

4. (a) Cos'è un'interruzione precisa?  
(b) Cosa comporta per una macchina avere delle interruzioni imprecise?

**Risposta:**

(a) Un'interruzione si dice precisa quando gode delle seguenti quattro proprietà:

1. il program counter viene salvato in un posto noto,
2. tutte le istruzioni che precedono quella puntata dal program counter sono state completamente eseguite,
3. nessuna delle istruzioni che seguono quella puntata dal program counter è stata eseguita,
4. lo stato di esecuzione dell'istruzione puntata dal program counter è noto.

# Sistemi Operativi

## 14 giugno 2010

### Compito

- (b) Nel caso in cui si abbiano interruzioni imprecise è difficile riprendere l'esecuzione in modo esatto in hardware: la CPU si limita a "riversare" sullo stack tutta l'informazione relativa allo stato corrente, lasciando che sia il sistema operativo a capire che cosa debba essere fatto. In questo modo rallentano le fasi di ricezione degli interrupt e di context-switch/ripristino dell'esecuzione, provocando grosse latenze.
5. Relativamente all'allocazione dei blocchi nei dischi, si spieghi brevemente come funziona lo schema di allocazione concatenata, illustrandone pregi e difetti. Soffre di frammentazione esterna?

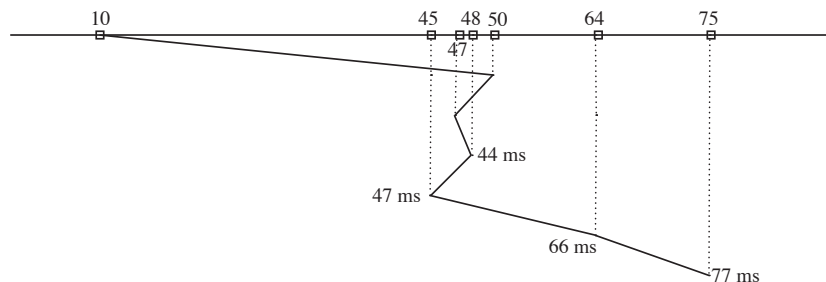
**Risposta:** Lo schema di allocazione concatenata prevede che ogni blocco dati abbia una parte di esso riservata alla memorizzazione dell'indirizzo del blocco successivo (l'ultimo blocco viene marcato con un indirizzo non valido, e.g., -1). In questo modo l'implementazione di un file consiste in una lista di blocchi concatenati ed è sufficiente memorizzare per ogni file nella directory entry il blocco iniziale e quello finale. I pregi consistono essenzialmente nella semplicità dell'implementazione, mentre gli aspetti negativi sono la fragilità dello schema (se si corrompe un puntatore della catena, si perde tutta la parte del file da quel punto in poi) e la mancanza di supporto dell'accesso diretto (seek); infatti per accedere al blocco i-esimo bisogna prima scorrere la catena di puntatori dei precedenti i-1 blocchi.

Questo schema non soffre di frammentazione esterna, dato che i blocchi assegnati ad un file possono essere sparsi ovunque nel disco (non devono necessariamente essere contigui); infatti, per aggiungere un blocco ad un file è sufficiente aggiornare il puntatore dell'ultimo blocco in modo che punti ad uno dei blocchi liberi su disco (marcando l'indirizzo di quest'ultimo come indirizzo finale, e.g., -1 ed aggiornando la directory entry corrispondente).

6. Si consideri un disco gestito con politica SSTF. Inizialmente la testina è posizionata sul cilindro 10; lo spostamento ad una traccia adiacente richiede 1 ms. Al driver di tale disco arrivano richieste per i cilindri 64, 45, 48, 75, rispettivamente agli istanti 0 ms, 40 ms 43 ms, 50 ms. Si trascuri il tempo di latenza.
1. In quale ordine vengono servite le richieste?
  2. Il tempo di attesa di una richiesta è il tempo che intercorre dal momento in cui è sottoposta al driver a quando viene effettivamente servita. Qual è il tempo di attesa medio per le quattro richieste in oggetto?

**Risposta:**

1. Come si evince dal grafico seguente, l'ordine di servizio delle quattro richieste è 48, 45, 64, 75.



2. Il tempo di attesa medio per le quattro richieste in oggetto è  $\frac{(44-43)+(47-40)+(66-0)+(77-50)}{4} = \frac{1+7+66+27}{4} = \frac{101}{4} = 25,25 \text{ ms}$ .