

UNIVERSITA' DEGLI STUDI DI BOLOGNA - CORSO DI LAUREA IN INFORMATICA  
CORSO DI SISTEMI OPERATIVI  
ANNO ACCADEMICO 2010/2011 - PROVA DI CONCORRENZA

17 giugno 2011

**Esercizio -1:** essersi iscritti correttamente per svolgere questa prova.

**Esercizio 0:** Scrivere correttamente nome, cognome, matricola e posizione in tutti i fogli prima di svolgere ogni altro esercizio. Scrivere esclusivamente a penna senza abrasioni. E' vietato l'uso delle penne cancellabili, della matita, dei coprenti bianchi per la correzione (bianchetto) e la scrittura in colore rosso (riservato alla correzione). Il compito e' formato da due fogli, quattro facciate compresa questa. Le soluzioni che si vogliono sottoporre per la correzione devono essere scritte negli spazi bianchi di questi fogli. Non verranno corretti altri supporti. E' obbligatorio consegnare il compito, e' possibile chiedere che esso non venga valutato scrivendo "NON VALUTARE" in modo ben visibile nella prima facciata.

**Esercizio 1:** Nell'allocazione gerarchica a ogni le classe di risorse e' associata una prioritaa'. Un processo non puo' allocare risorse di una classe a prioritaa' n se ha risorse gia' assegnate di prioritaa' m,  $m \geq n$ .

Per allocare risorse di prioritaa' n si deve quindi rilasciare tutte le risorse di prioritaa' maggiore o uguale a n e allocarle nuovamente in seguito.

Sia dato un sistema con 10 classi di risorse e 10 istanze in ogni classe di risorse. Le classi sono numerate da 0 a 9 e l'ordine numerico corrisponde alla prioritaa' delle classi (9 prioritaa' massima, 0 prioritaa' minima)

Scrivere un monitor hierarchical\_allocator che fornisca le due procedure entry:

allocate(class, num)

e

deallocate(class, num)

che servono per allocare e deallocare rispettivamente "num" risorse della classe "class".  $1 \leq \text{num} \leq 10$ ,  $0 \leq \text{class} \leq 9$ .

La funzione allocate deve automaticamente rilasciare le risorse gia' ottenute dal processo nelle classi maggiori o uguali a class, allocare le nuove risorse e acquisire nuovamente le risorse rilasciate prima di poter proseguire.

Hint: occorrera' tenere una "contabilitaa'" delle risorse allocate da ogni processo in una struttura dati indicizzata dal pid del processo chiamante.

Esercizio 2: Sia dato il seguente programma:

```
Semaphore S1(2);
Semaphore S2(2);
Semaphore S3(1);
Int val=1;
```

```
process P1:
while(1):
  S1.P(); S1.P();
  S2.P(); S2.P();
  val+=2;
  S1.V(); S1.V();
  S2.V(); S2.V();
```

```
process P2:
while(1):
  S2.P(); S1.P();
  S1.P(); S2.P();
  val*=2;
  S1.V(); S2.V();
  S2.V(); S1.V();
```

Quali sono i valori che puo' assumere la variabile val quando il programma va in deadlock? (dimostrare il risultato).

Esercizio 3: Dato un servizio di message passing asincrono implementare un servizio di message passing generale in grado di operare come message passing sincrono, asincrono e completamente asincrono.

Le funzioni da implementare sono:

```
void gsend(boolean sync, pid_t dst, msgT msg)
msgT genrecv(boolean sync, pid_t sender)
```

il campo booleano sync indica se la chiamata debba essere bloccante o meno.

Se sync e' vero gsend deve aspettare che il messaggio sia stato recapitato per proseguire (servizio sincrono). Se sync e' falso gsend si comporta come un servizio di message passing asincrono.

Per genrecv, se sync e' vero il chiamante deve attendere il primo messaggio dal mittente specificato, se sync e' falso la chiamata non e' mai bloccante, restituisce NULL se non vi sono messaggi corrispondenti alle richieste.

UNIVERSITA' DEGLI STUDI DI BOLOGNA - CORSO DI LAUREA IN INFORMATICA  
CORSO DI SISTEMI OPERATIVI  
ANNO ACCADEMICO 2010/2011 – PARTE GENERALE  
17 giugno 2011

**Esercizio -1:** essersi iscritti correttamente per svolgere questa prova.

**Esercizio 0:** Scrivere correttamente nome, cognome, matricola e posizione in tutti i fogli prima di svolgere ogni altro esercizio. Scrivere esclusivamente a penna senza abrasioni. E' vietato l'uso delle penne cancellabili, della matita, dei coprenti bianchi per la correzione (bianchetto) e la scrittura in colore rosso (riservato alla correzione). Il compito e' formato da due fogli, quattro facciate compresa questa. Le soluzioni che si vuole sottoporre per la correzione devono essere scritte negli spazi bianchi di questi fogli. Non verranno corretti altri supporti.

E' obbligatorio consegnare il compito, e' possibile chiedere che esso non venga valutato scrivendo "NON VALUTARE" in modo ben visibile nella prima facciata.

Per svolgere questo compito occorre solo una penna e un documento di identità valido. La consultazione o anche solo la disponibilità di altro materiale comporterà l'annullamento del compito (con la stessa penalizzazione di punteggio della grave insufficienza per la prossima esercitazione scritta).

**Esercizio 1:** Si consideri un disco con 10 tracce e un tempo di seek di 1ms/traccia.

(i) Mostrare una sequenza di richieste di traccia (con relativi tempi di arrivo) tale per cui l'algoritmo FCFS impieghi almeno il doppio a servire tutte le richieste rispetto ad un algoritmo LOOK.

(ii) Mostrare una sequenza di richieste di traccia (con relativi tempi di arrivo) tale per cui l'algoritmo FCFS impieghi esattamente lo stesso tempo a servire tutte le richieste rispetto ad un algoritmo LOOK.

In entrambi i casi, le sequenze devono contenere almeno 10 richieste.

**Esercizio 2:** Due studiosi stanno discutendo se ampliare l'ampiezza del blocco dati di un file system ext2 da 1024 a 2048 byte. Nell'inode vi sono 10 indici ai blocchi, un indice indiretto, uno due volte indiretto ed infine uno a tripla indizione. Gli indici dei blocchi dati sono interi a 64 bit.

Seguiamo la loro discussione:

a- l'ampiezza della tabella di i-node sara' maggiore

b- i link simbolici occupano il doppio di spazio

c- anche per i link fisici si spreca piu' spazio

d- l'ampiezza massima dei file viene raddoppiata

e- ci saranno meno blocchi indiretti, quindi tutti i file occuperanno un numero uguale o minore di byte su disco

f- al contrario tutti i file occuperanno un numero maggiore o uguale di byte perche' aumenta la frammentazione interna

Confermare o confutare ogni affermazione, spiegando perche' l'affermazione sia da considerare vera o falsa.

**Esercizio 3:** Detti rispettivamente R il numero di riga e C il numero di colonna, soddisfare la richiesta corrispondente a  $(R+C)\%5$  (esercizio obbligatorio):

0) Cosa e' un microkernel e quale e' la differenza con un kernel monolitico

1) Cosa e' una macchina virtuale e in quali casi si utilizza.

2) Cosa e' un modulo del kernel e in quali casi viene utilizzato.

3) Cosa significa generare (compilare) un kernel e a cosa serve generare kernel specifici

4) Quali sono le caratteristiche di un kernel per sistemi multiprocessore. Cosa significa che un kernel e' di tipo SMP (symmetric multi processor).

NOTA: tutti gli esercizi verranno valutati solo se le risposte saranno corredate da motivazioni e dimostrazioni scritte in Italiano o in Inglese (corretto o almeno comprensibile). La presenza nella soluzione di un esercizio di solo codice sorgente/tabelle/scarabocchi/simboli vari comporta la non valutazione dell'esercizio.

