

UNIVERSITA' DEGLI STUDI DI BOLOGNA - CORSO DI LAUREA IN INFORMATICA  
CORSO DI SISTEMI OPERATIVI - ANNO ACCADEMICO 2008/2009  
CONCORRENZA – 1 settembre 2009

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

**Esercizio 0:** Scrivere correttamente nome, cognome, matricola e posizione prima di svolgere ogni altro esercizio.

**Esercizio 1:** Uno scheduler cooperativo prevede che ogni processo chiami la funzione yield() quando puo' cedere il controllo agli altri processi.

La funzione yield deve quindi attivare un altro processo e bloccare il chiamante.

Scrivere un monitor chiamato coopsched che preveda tre "procedure entry":

init, yield, fini tutte prive di parametri. Init viene chiamata da un processo che vuole iniziare la propria esecuzione, yield e' quella descritta sopra e fini viene chiamata da un processo subito prima di terminare.

Un solo processo alla volta deve essere in esecuzione (fra quelli che hanno chiamato init e non ancora fini), yield deve fare avvicinare i processi in ordine ciclico (come se fossero in una coda circolare, il processo che fa yield deve essere riattivato dopo che tutti quelli in attesa hanno avuto modo di essere eseguiti e chiamare nuovamente yield o fini).

**Esercizio 2:** I semafori binari a V persa sono semafori binari (possono assumere solo i valori 0 e 1) che non bloccano il chiamante se viene richiamata una V quando il valore del semaforo e' 1, semplicemente in questo caso la V non produce alcun effetto (viene appunto persa).

I semafori binari a V persa hanno lo stesso potere espressivo dei semafori?

N.B. La risposta richiede una dimostrazione.

**Esercizio 3:** Sia data una funzione atomica che scambia due numeri interi **contigui** in memoria, cioe' scambia tra loro i valori registrati in due voci successive di memoria (come per esempio due elementi di un array di interi).

```
Atomic void swap2words (int *x) {  
    int tmp=*x;  
    *x= *(x+1);  
    *(x+1)=tmp;  
}
```

Questa funzione consente di realizzare un servizio di mutua esclusione fra due processi. Mostrare come.

Questa funzione ha necessita' di un processo gestore per poter fornire un servizio di mutua esclusione fra un numero arbitrario di processi, mostrare come.



UNIVERSITA' DEGLI STUDI DI BOLOGNA - CORSO DI LAUREA IN INFORMATICA  
 CORSO DI SISTEMI OPERATIVI - ANNO ACCADEMICO 2008/2009  
 PARTE GENERALE - 1 settembre 2009

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

**Esercizio 0:** Scrivere correttamente nome, cognome, matricola e posizione prima di svolgere ogni altro esercizio.

**Esercizio 1:** Sia data una stringa di riferimenti riferita a tre processi a,b,c. Si consideri un sistema che applichi il meccanismo del Working set in modo globale e non processo per processo. Il Working set ha durata equivalente a 15 riferimenti in memoria e il numero di frame globale del sistema e' pari a 10.

Sia data quindi la seguente stringa di riferimenti (a1 significa che il processo a accede alla pagina 1).

a1 b2 c3 a1 b4 c5 a6 b7 c5 a8 b9 c10 a1 b4 c5 a1 c3 b2 c11 (\*) a12 b7 a1 b4 b2 a1 b9

Calcolare i working set ad ogni istante. Cosa succede e cosa fa/puo' fare il sistema nel punto (\*)?

**Esercizio 2:** Sia dato un sistema con scheduler round robin per la CPU (time slice = 3ms) e un disco gestito con la politica C-LOOK con velocità di seek 1ms per cilindro e tempo di lettura della traccia trascurabile (tempo nullo ai fini dell'esercizio). La testina è posizionata inizialmente sul cilindro 5. Siano dati i seguenti tempi

p1. 4ms CPU, I/O traccia 2, 3 ms CPU, I/O traccia 8, 3 ms CPU

p2. 5ms CPU, I/O traccia 5, 3 ms CPU, I/O traccia 1, 2 ms CPU

p3. 3ms CPU, I/O traccia 10, 3 ms CPU, I/O traccia 3, 3 ms CPU

p4. 4ms CPU, I/O traccia 7, 3 ms CPU, I/O traccia 1, 5 ms CPU

Mostrare il diagramma di gantt della CPU e del disco.

**Esercizio 3:** Se N e' il numero di matricola calcolare  $M=N\%5$ . Prendere l'oggetto M e descrivere:

a) a cosa serve

b) esempi reali di utilizzo

1)

oggetto0: autotools (autoconf, automake)

oggetto1: buddy list

oggetto2: linker dinamico

oggetto3: sistemi di controllo di versione distribuiti (ad esempio: cvs (concurrent version system), svn (subversion), git (fast version control system))

oggetto4: sottosistema di ambiente



