Nome_____ Cognome _____N. di matricola (10 cifre) _____ Riga ___ Col _____

UNIVERSITA' DEGLI STUDI DI BOLOGNA - CORSO DI LAUREA IN INFORMATICA
CORSO DI SISTEMI OPERATIVI - ANNO ACCADEMICO 2009/2010
CONCORRENZA – 12 luglio 2010

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.

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

```
Processo[i]:
     priocoop.init(prio)
     …..
     priocoop.yield(prio)
     …..
     priocoop.yield(prio)
     …..
     priocoop.fini()
```

La funzione yield deve quindi attivare un altro processo e bloccare il chiamante. Scrivere un monitor chiamato `priocoop` che preveda tre "procedure entry": init, yield, fini. Init viene chiamata da un processo che vuole iniziare la propria esecuzione e ha un parametro che rappresenta la priorita' del processo (nel range 0..9), 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 avvicendare i processi, deve attivare quello a priorita' massima fra quelli in attesa, FIFO fra processi aventi la stessa priorita'.

Esercizio 2: Message passing con priorita'. Sia dato un servizio di message passing asincrono. Si implementi un servizio di message passing con priorita' **senza fare uso di processi server**. Il nuovo servizio ha due chiamate psend(dest, prio, msg) e precv(sender). La precv consente la ricezione da chiunque (sender==*). Quando un processo P chiama la precv questa deve restituire il messaggio di massima priorita' fra quelli spediti dal sender specificato a P o sospendere il processo se non ce ne sono in attesa. (suggerimento: se serve si puo' usare la funzione getpid() che restituisce l'identificativo del processo chiamante).

Esercizio 3: La seguente classe twomult può essere usata al posto della Test&Set?

```
Class twomult {
   private int a;
   private int b;
   void twomult(void) {a=1;b=1}
   atomic int op (int x) {
      int result=b=a*b;
      a=b*x;
      return result;
  }
}
```

Produrre una dimostrazione (suggerimento: usare +1 e -1 per non fare "esplodere" i valori delle moltiplicazioni).

Nome_____ Cognome _____N. di matricola (10 cifre) _____ Riga ___ Col _____

UNIVERSITA' DEGLI STUDI DI BOLOGNA - CORSO DI LAUREA IN INFORMATICA
CORSO DI SISTEMI OPERATIVI - ANNO ACCADEMICO 2009/2010
PARTE GENERALE – 16 giugno 2010

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 identita' valido. La consultazione o anche solo la disponibilita' di altro materiale comportera' l'annullamento del compito (con la stessa penalizzazione di punteggio della grave insufficienza per la prossima esercitazione scritta).

Esercizio 1: Si confrontino le esecuzioni degli algoritmi SJN (non preemptive) e SRTF (preemptive) nell'esecuzione dei seguenti programmi.

```
P1:    for (i=0;i<4;i++) {
            read(1ms);
            compute(7ms);
            write(1ms);
       }
P2:    for (i=0;i<4;i++) {
            read(2ms);
            compute(2ms);
            write(2ms);
       }
```

Le letture e le scritture avvengono su un unico device (FIFO).
I tempi dei context switch e di valutazione del for si intendono tralasciabili ai fini didattici.

Esercizio 2: Si consideri il seguente scenario per un banchiere multivaluta.
(a) Per quali valori di x lo stato e' safe? Motivare la risposta.
(b1) Scegliete un valore x per il quale lo stato sia safe e mostrate una sequenza di allocazioni che porterebbe il sistema in uno stato unsafe in due mosse (allocazioni). Cosa accade in un sistema reale se tale sequenza e' ricevuta dal banchiere?
(b2) Scegliere un valore x per il quale lo stato sia unsafe e mostrare una sequenza di deallocazioni che porterebbe il sistema in uno stato safe in due mosse (deallocazioni).

	Valuta 1			Valuta 2		
	COH=3			COH=20		
i	Ci	Pi	Ni	Ci	Pi	Ni
	Max	Curr.	Residuo	Max	Curr.	Residuo
1	10	6	4	30+x	x	30
2	5	2	3	15	10	5
3	13	5	8	22	2	20
4	1	1	0	60	20	40

Esercizio 3: Se N e' il numero di matricola calcolare M=N%5. Prendere l'oggetto M e descrivere:
a) cosa e', a cosa serve
b) esempi reali

oggetto0: buffer overflow attack
oggetto1: working set
oggetto2: macchine virtuali
oggetto3: microkernel
oggetto4: generazione del kernel