

UNIVERSITA' DEGLI STUDI DI BOLOGNA - CORSO DI LAUREA IN INFORMATICA
 CORSO DI SISTEMI OPERATIVI - ANNO ACCADEMICO 2003/2004
 CONCORRENZA - 20 Gennaio 2006

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:

A) Scrivere un monitor bufbuf avente le seguenti funzioni:

```
bufbuf.enqueue(int n, Type *v);
```

```
bufbuf.dequeue(int n, Type *v);
```

La funzione enqueue accoda n ($n \leq \text{NELEM}$) elementi nel buffer con un'unica operazione. Il valore degli elementi e' contenuto nel vettore v.

La funzione dequeue disaccoda n ($n \leq \text{NELEM}$) elementi dal buffer (in ordine FIFO) e restituisce gli n valori all'interno del vettore v.

Se non ci sono abbastanza elementi liberi (o occupati) l'operazione enqueue (o dequeue) diventa bloccante, i.e. deve attendere.

B) Date le specifiche del problema ci puo' essere starvation (perche'?)

C) Date le specifiche del problema ci puo' essere deadlock (perche'?)

Esercizio 2 (Semafori Robin Hood):

Un semaforo Robin Hood e' un tipo di dato astratto su cui sono definite le seguenti operazioni:

init(S,n,m); inizializza a n il valore corrente e a m il valore massimo del semaforo S

Prh(S1,S2); per $S1 \neq S2$, decrementa il semaforo piu' "ricco", ovvero quello con il valore piu' alto; se entrambi i semafori sono a 0, allora il processo si sospende sia sulla coda S1 che sulla coda S2

Vrh(S1,S2); per $S1 \neq S2$, fa una V su entrambi i semafori; un processo risvegliato da una Vrh viene eliminato da entrambe le code sulle quali era sospeso. Se uno dei due semafori ha come valore corrente il massimo consentito, allora la Vrh e' bloccante (il processo si sospende lasciando inalterati i semafori)

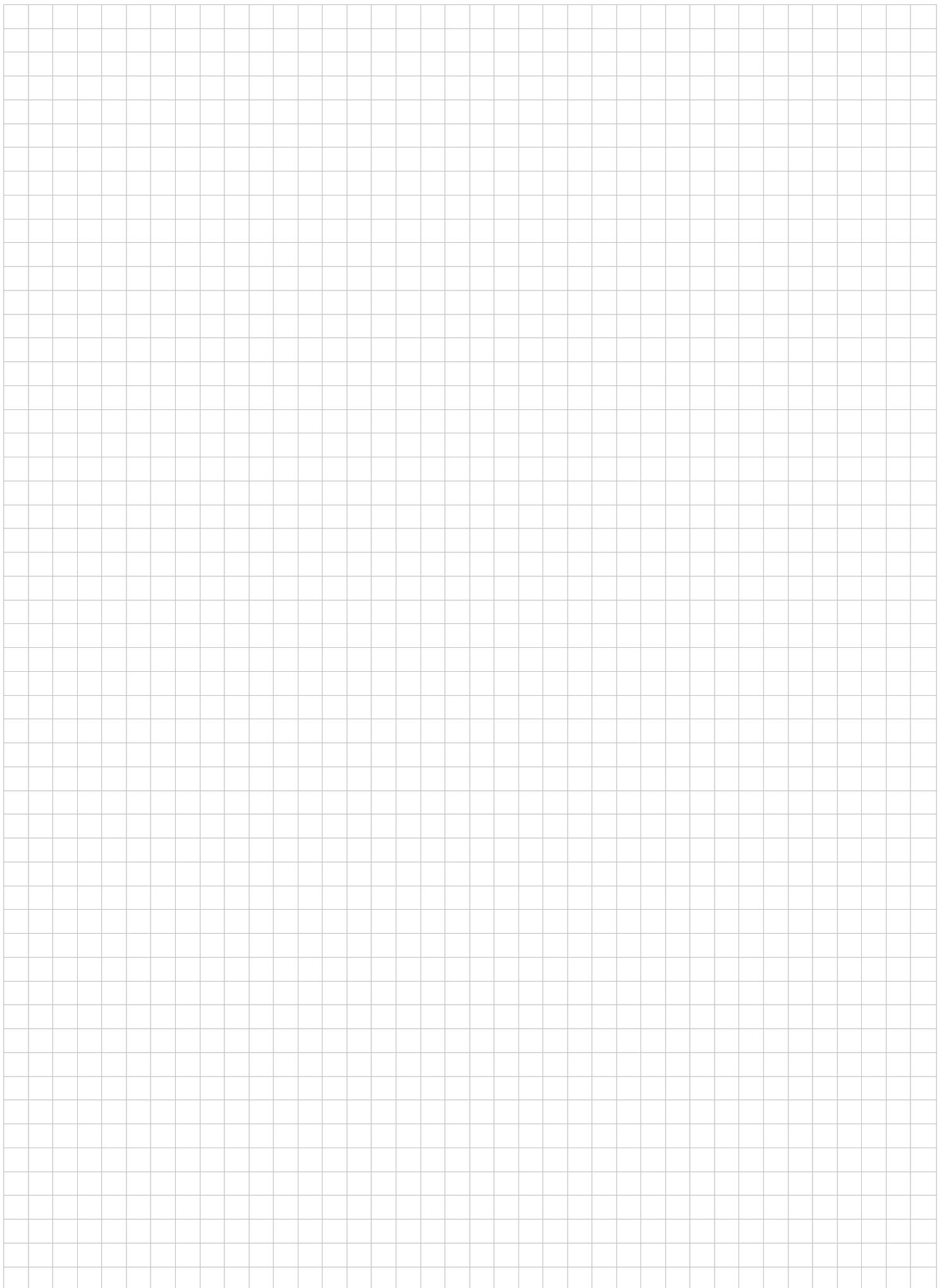
Implementare utilizzando semafori Robin Hood il protocollo di accesso alle sezioni critiche (ovvero la coppia di funzioni [enter_CS] e [exit_CS]).

Esercizio 3:

Considerate il seguente frammento di codice:

```
int in = 0;
int turno = 0;
Semaphore Se = 2;
Semaphore S[2] = {0; 0};
void f(int i)
{ while(1) {
    P(Se);
    turno=i;
    if(in++ == 0) print "entro";
    if(turno == i) print i;
    V(S[(i+1)%2]);
    P(S[i]);
    in=0;
    V(Se);
}
}
main() { f(0) // f(1) }
```

Il sistema puo' andare in deadlock? Perche'? Descrivere tutti gli output possibili.



UNIVERSITA' DEGLI STUDI DI BOLOGNA - CORSO DI LAUREA IN INFORMATICA
 CORSO DI SISTEMI OPERATIVI - ANNO ACCADEMICO 2003/2004
 CONCORRENZA - 20 Gennaio 2006

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:

A) Scrivere un monitor bufbuf avente le seguenti funzioni:

```
bufbuf.enqueue(int n, Type *v);
```

```
bufbuf.dequeue(int n, Type *v);
```

La funzione enqueue accoda n ($n \leq \text{NELEM}$) elementi nel buffer con un'unica operazione. Il valore degli elementi e' contenuto nel vettore v.

La funzione dequeue disaccoda n ($n \leq \text{NELEM}$) elementi dal buffer (in ordine FIFO) e restituisce gli n valori all'interno del vettore v.

Se non ci sono abbastanza elementi liberi (o occupati) l'operazione enqueue (o dequeue) diventa bloccante, i.e. deve attendere.

B) Date le specifiche del problema ci puo' essere starvation (perche'?)

C) Date le specifiche del problema ci puo' essere deadlock (perche'?)

Esercizio 2 (Semafori Robin Hood):

Un semaforo Robin Hood e' un tipo di dato astratto su cui sono definite le seguenti operazioni:

init(S,n,m); inizializza a n il valore corrente e a m il valore massimo del semaforo S

Prh(S1,S2); per $S1 \neq S2$, decrementa il semaforo piu' "ricco", ovvero quello con il valore piu' alto; se entrambi i semafori sono a 0, allora il processo si sospende sia sulla coda S1 che sulla coda S2

Vrh(S1,S2); per $S1 \neq S2$, fa una V su entrambi i semafori; un processo risvegliato da una Vrh viene eliminato da entrambe le code sulle quali era sospeso. Se uno dei due semafori ha come valore corrente il massimo consentito, allora la Vrh e' bloccante (il processo si sospende lasciando inalterati i semafori)

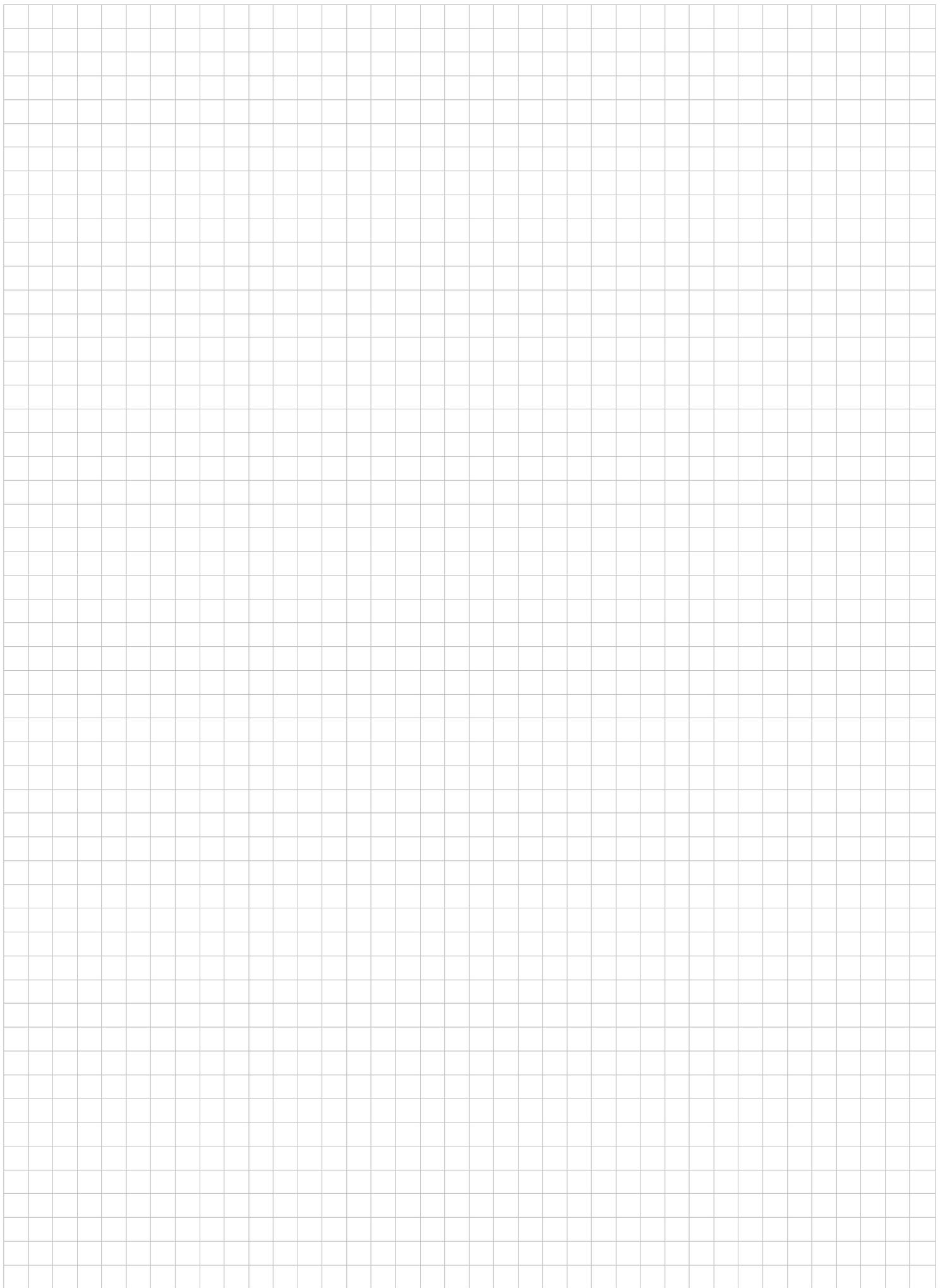
Implementare utilizzando semafori Robin Hood un semaforo binario inizializzato a 1 (nella versione in cui la V su un semaforo gia' a 1 e' bloccante).

Esercizio 3:

Considerate il seguente frammento di codice:

```
int in = 2;
int turno = 0;
Semaphore Se = 2;
Semaphore S[2] = {0; 0};
void f(int i)
{ while(1) {
    P(Se);
    turno=i;
    if(turno == i) print i;
    if(--in == 0) print "esco";
    V(S[(i+1)%2]);
    P(S[i]);
    in=2;
    V(Se);
}
}
main() { f(0) // f(1) }
```

Il sistema puo' andare in deadlock? Perche'? Descrivere tutti gli output possibili.



Nome/cognome _____ N. di matricola (10 cifre) _____ Posizione: Riga ____ Col ____

UNIVERSITA' DEGLI STUDI DI BOLOGNA - CORSO DI LAUREA IN INFORMATICA
CORSO DI SISTEMI OPERATIVI - ANNO ACCADEMICO 2003/2004
PARTE GENERALE - 20 Gennaio 2006

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:

Fornire un esempio di banchiere multivaluta con tre processi P1, P2, P3, ai quali inizialmente non siano state assegnate risorse, e una sequenza di quattro richieste tali per cui le prime tre vengano soddisfatte immediatamente dal banchiere, ma non la quarta.

Esercizio 2:

Sia dato questo programma:

Program P:

```
for (i=0;i<2;i++) {  
    long_compute();  
    io_on_dev(i);  
}  
short_compute();
```

long compute impiega 6 ms, short compute 2 ms e io_on_dev impiega 7ms.

(l'indice della funzione io_on_dev indica il device sul quale viene eseguita la funzione).

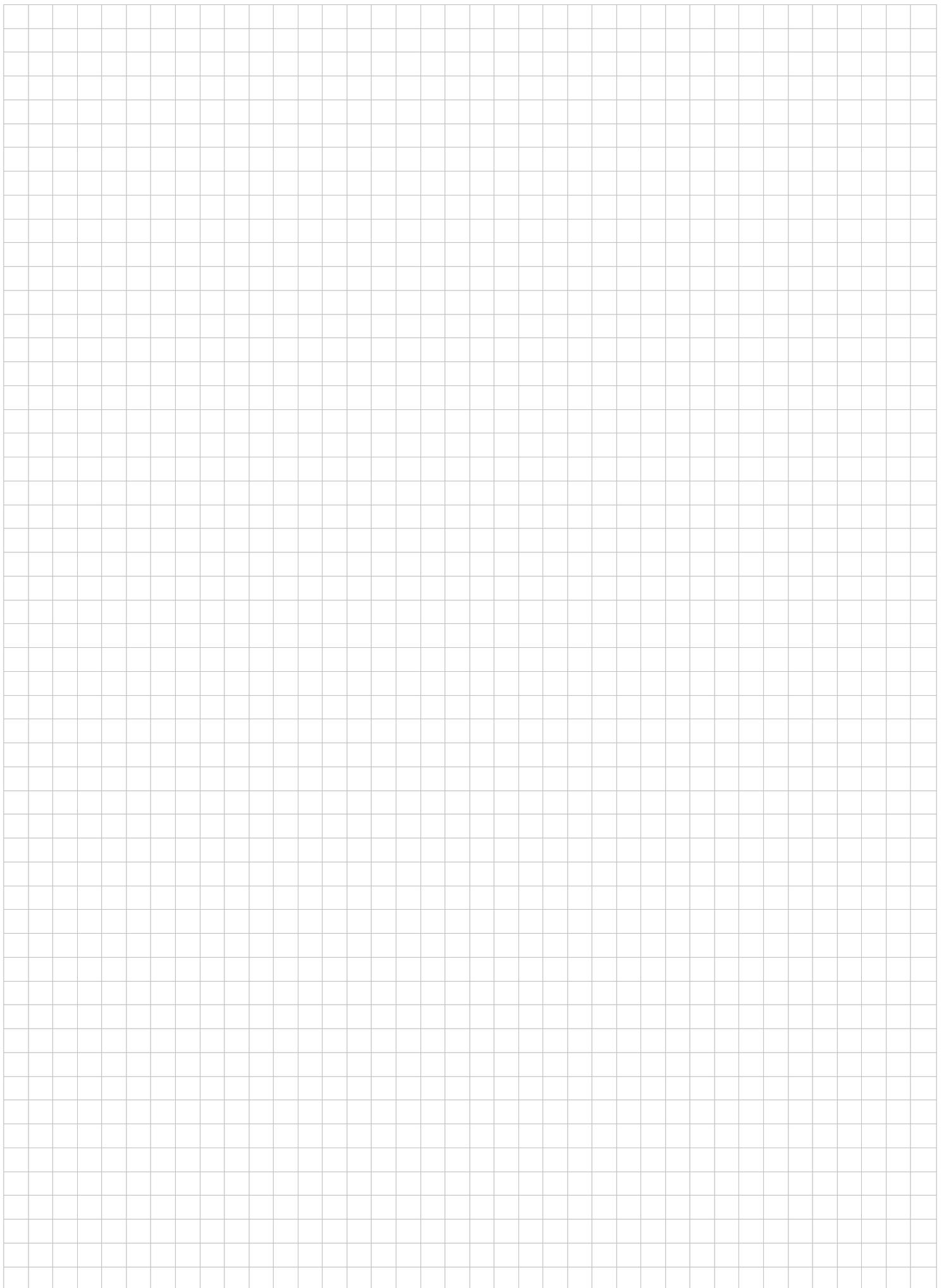
Considerando in un sistema di elaborazione dove sono in esecuzione tre istanze del programma P che sono state attivate ai tempi 0,3ms e 7ms e che il sistema usa uno scheduler round robin per l'accesso alla CPU disegnare il diagramma di Gantt dell'esecuzione. (time slice=2 ms).

Esercizio 3:

Sia x l'ultima e y la penultima cifra del vostro numero di matricola e sia $n = (y*10 + x)\%6$.

Discutere quali effetti comporta sulle funzionalita' e l'efficienza di un sistema operativo l'assenza del meccanismo hardware n:

- 0) modalita' supervisore
- 1) reference bit
- 2) TLB
- 3) DMA
- 4) timer
- 5) MMU



Nome/cognome _____ N. di matricola (10 cifre) _____ Posizione: Riga ____ Col ____

UNIVERSITA' DEGLI STUDI DI BOLOGNA - CORSO DI LAUREA IN INFORMATICA
CORSO DI SISTEMI OPERATIVI - ANNO ACCADEMICO 2003/2004
PARTE GENERALE - 20 Gennaio 2006

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:

Fornire un esempio di banchiere multivaluta con tre processi P1, P2, P3, ai quali inizialmente non siano state assegnate risorse, e una sequenza di quattro richieste tali per cui le prime tre vengano soddisfatte immediatamente dal banchiere, ma non la quarta.

Esercizio 2:

Sia dato questo programma:

Program P:

```
for (i=0;i<2;i++) {  
    long_compute();  
    io_on_dev(i);  
}  
short_compute();
```

long compute impiega 6 ms, short compute 2 ms e io_on_dev impiega 9ms.

(l'indice della funzione io_on_dev indica il device sul quale viene eseguita la funzione).

Considerando in un sistema di elaborazione dove sono in esecuzione tre istanze del programma P che sono state attivate ai tempi 0,3ms e 7ms e che il sistema usa uno scheduler round robin per l'accesso alla CPU disegnare il diagramma di Gantt dell'esecuzione. (time slice=2 ms).

Esercizio 3:

Sia x l'ultima e y la penultima cifra del vostro numero di matricola e sia $n = (y*10 + x)\%6$.

Discutere quali effetti comporta sulle funzionalita' e l'efficienza di un sistema operativo l'assenza del meccanismo hardware n:

- 6) modalita' supervisore
- 7) reference bit
- 8) TLB
- 9) DMA
- 10)timer
- 11)MMU

