Cognome e nome: _____ Matricola: _____ Posto: ____ ____

Università degli Studi di Padova - Corso di Laurea in Informatica

Regole dell'esame

Il presente esame scritto deve essere svolto in forma individuale in un tempo massimo di 60 min dalla sua presentazione.
Non è consentita la consultazione di libri o appunti in forma cartacea o elettronica, né l'uso di palmari e telefoni cellulari.
La correzione avverrà in data e ora comunicate dal docente; i risultati saranno esposti sul sito del docente.
Il candidato riporti generalità e matricola negli spazi indicati in alto e inserisca le proprie risposte interamente su questi fogli.

Per avere accesso al secondo compitino il candidato deve acquisire almeno 2 punti nel Quesito 1 e almeno 16 punti in totale.

Quesito 1: ***0,5 punti per risposta giusta, diminuzione di 0,25 punti per ogni sbaglio, 0 punti per risposta vuota***

DOMANDA	**Vero/Falso**
Una *system call* dà sempre luogo ad un *mode switch* tra modalità utente e modalità *kernel*	
Se un processo è in blocco da 10 ms significa che 10 ms fa ha eseguito una *system call*	
Ogni *interrupt* può essere associato ad un processo che ha richiesto una operazione di I/O	
Un processo per lanciare un nuovo processo deve fare una *system call*	
L'algoritmo di scheduling Shortest Remaining Time Next (SRTN) minimizza il tempo di risposta	
La possibilità di effettuare prerilascio è necessaria al funzionamento dello scheduling Round Robin	
Con scheduling First Come First Served (FCFS) senza valutazione dell'attributo di priorità, il tempo di attesa è sempre uguale al tempo di risposta	
Un semaforo binario può gestire le richieste di accesso solo se provenienti da massimo due processi	
L'inversione di priorità è una tecnica utilizzata per evitare la *starvation* dei processi a bassa priorità	
Gli interrupt sono asincroni mentre le chiamate di sistema (trap) sono sincrone	

Quesito 2:
Si consideri la politica di *scheduling Round Robin* di quanto q e si supponga che in un sistema ci siano N processi interattivi tutti con lo stesso comportamento. Ciascuna interazione dà luogo ad un CPU *burst* che richiede la CPU per un tempo c.

[2.A] Se $c < q$ quanto tempo aspetta al più un processo in coda *ready* prima di ottenere la CPU?

[2.B] Se $c < q$ quanto tempo aspetta al più l'utente prima che la CPU finisca di elaborare l'interazione?

[2.C] Se $c > q$ quanto tempo aspetta l'utente prima di iniziare l'ultimo quanto di interazione per l'ultimo processo rimasto ancora attivo? (Si consideri che c può essere espresso come $c = aq + b$; ovvero a quanti di tempo + $b<c$ tempo nell'ultimo quanto)

Cognome e nome: _____ Matricola: _____ Posto: ____ ____

Quesito 3:

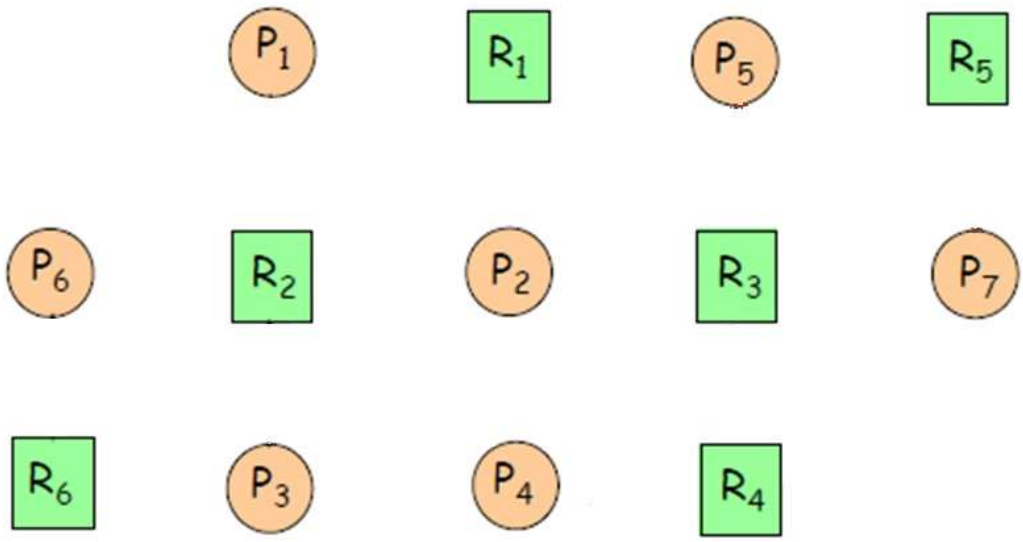
Lo studente riporti le 4 condizioni necessarie e sufficienti affinché possa verificarsi lo stallo (deadlock) di un sistema.

Quesito 4:

Un sistema è composto da sette processi P1… P7 e da sei risorse condivise R1… R6 ciascuna diversa dalle altre, presente in singola istanza e ad accesso mutuamente esclusivo. La situazione corrente del sistema è la seguente:

- P1 occupa R1 e richiede R2;
- P2 non occupa risorse e richiede R3;
- P3 non occupa risorse e richiede R2;
- P4 occupa R4 e richiede sia R2 sia R3;
- P5 occupa R3 e richiede R5;
- P6 occupa R6 e richiede R2;
- P7 occupa R5 e richiede R4;

Si determini, utilizzando il grafo di allocazione delle risorse, se il sistema sia in stallo (deadlock) e, in caso affermativo, quali siano i processi e le risorse coinvolti.



Cognome e nome: _____ Matricola: _____ Posto: ____ ____

Quesito 5:

Un sistema ha 4 processi (A, B, C, D) e 5 risorse (R1, R2, R3, R4, R5) da ripartire. L'attuale allocazione e i bisogni massimi sono i seguenti:

Processo	*Allocate*	*Massimo*
A	1 0 2 1 1	1 1 2 1 4
B	2 0 1 1 1	3 3 4 2 1
C	1 1 0 1 0	2 1 4 1 0
D	1 1 1 1 0	1 1 3 2 1

[5.A] Considerando il vettore delle risorse disponibili uguale a [0 0 3 1 2], si discuta se il sistema sia in uno stato sicuro.

[5.B] Il procedimento di verifica dello stato sicuro è uno dei passi ripetuti da un noto algoritmo che assegna risorse ai processi solo se l'assegnazione fa rimanere il sistema in uno stato sicuro. Come si chiama questo algoritmo?

Quesito 6:

[6.A] La seguente soluzione del problema dei lettori-scrittori contiene alcuni errori e mancanze. Lo studente ne modifichi il codice tramite aggiunte, cancellazioni e correzioni. Il risultato dovrà rappresentare una versione corretta, realizzata apportando il minor numero possibile di modifiche all'originale qui di seguito.
(Per coloro che avessero studiato solo sul libro di testo: *P*, corrisponde a *down*, *V* corrisponde a *up*)

```
void Lettore (void) {

  while (true) {

    P(mutex);

    numeroLettori++;

    if (numeroLettori==1) V(database);

    V(mutex);

    // leggi il dato

    numeroLettori--;

    if (numeroLettori==0) V(database);

    // usa il dato letto

  }

}
```

```
void Scrittore (void) {

  while (true) {

    // prepara il dato da scrivere

    V(database);

    // scrivi il dato

    P(database);

  }

}
```

[6.B] Lo studente riporti qua sotto l'indicazione del tipo e del valore iniziale di ciascuna variabile.

Cognome e nome: _____ Matricola: _____ Posto: ____ ____

Soluzione

Soluzione al Quesito 1

DOMANDA	Vero/Falso
Una *system call* dà sempre luogo ad un *mode switch* tra modalità utente e modalità *kernel*	V
Se un processo è in blocco da 10 ms significa che 10 ms fa ha eseguito una *system call*	V
Ogni *interrupt* può essere associato ad un processo che ha richiesto una operazione di I/O	F
Un processo per lanciare un nuovo processo deve fare una *system call*	V
L'algoritmo di scheduling Shortest Remaining Time Next (SRTN) minimizza il tempo di risposta	F
La possibilità di effettuare prerilascio è necessaria al funzionamento dello scheduling Round Robin	V
Con scheduling First Come First Served (FCFS) senza valutazione dell'attributo di priorità, il tempo di attesa è sempre uguale al tempo di risposta	V
Un semaforo binario può gestire le richieste di accesso solo se provenienti da massimo due processi	F
L'inversione di priorità è una tecnica utilizzata per evitare la *starvation* dei processi a bassa priorità	F
Gli interrupt sono asincroni mentre le chiamate di sistema (trap) sono sincrone	V

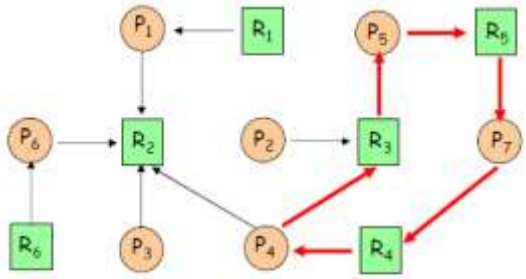
Soluzione al Quesito 2

[2.A] Un processo aspetta *(N-1)c* per ottenere la CPU.
[2.B] L'utente aspetta *(N-1)c + c = Nc*
[2.C] L'utente aspetta *N qa + (N-1) b oppure anche scrivibile come c(N-1) + aq*

Soluzione al Quesito 3

Risposta disponibile sulle slide dell'insegnamento.

Soluzione al Quesito 4



P4 → R3 → P5 → R5 → P7 → R4 → P4 sono in deadlock.

Soluzione al Quesito 5

[5.A] La matrice delle necessità (massimo numero di risorse richieste dal processo - risorse allocate al processo) è la seguente:

0 1 0 0 3
1 3 3 1 0
1 0 4 0 0
0 0 2 1 1

Il processo D può essere eseguito fino alla fine. Quando ha finito, il vettore delle risorse disponibili è [1 1 4 2 2].
Il processo C può dunque essere eseguito. Dopo il suo completamento, il vettore delle risorse disponibili diventa [2 2 4 3 2].
Purtroppo questo non questo permette di eseguire e completare né A (manca una risorsa di tipo R5), né B (manca una risorsa di tipo R2).
Il sistema NON è quindi in uno stato sicuro.

[5.B] L'Algoritmo del Banchiere (Banker's Algorithm)

Cognome e nome: _____ Matricola: _____ Posto: ____ ____

Soluzione al Quesito 6

[6.A]

```
void Lettore (void) {
  while (true) {
    P(mutex);
    numeroLettori++;
    if (numeroLettori==1) P(database);
    V(mutex);
    // leggi il dato
    P(mutex);
    numeroLettori--;
    if (numeroLettori==0) V(database);
    V(mutex);
    // usa il dato letto
  }
}
```

```
void Scrittore (void) {
  while (true) {
    // prepara il dato da scrivere
    P(database);
    // scrivi il dato
    V(database);
  }
}
```

[6.B]

Non importa la sintassi…

int numeroLettori = 0
semaforo mutex = 1
semaforo database = 1