

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 45 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 3 punti nel Quesito 1 e almeno 16 punti in totale.

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

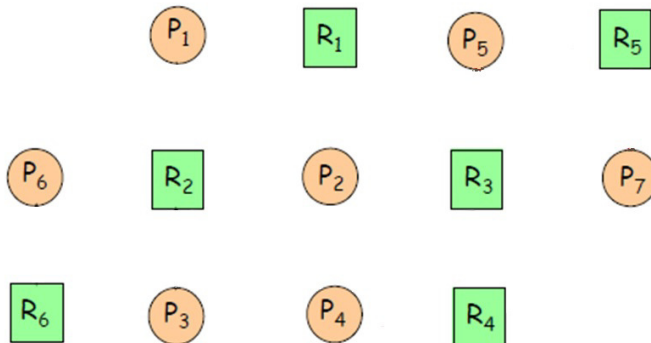
DOMANDA	Vero/Falso
Una <i>system call</i> dà sempre luogo ad un <i>mode switch</i> tra modalità utente e modalità <i>kernel</i>	✓
Un <i>process switch</i> tra processi utente avviene sempre contestualmente a 2 <i>mode switch</i> (utente->kernel, kernel->utente)	✓
Un <i>interrupt</i> viene gestito in modalità utente	F
Se un processo inizialmente attivo esegue il seguente codice, il sistema si troverà con 8 processi che eseguono InstrX, di cui uno è il processo iniziale.  <pre>fork(); fork(); fork(); fork(); InstrX;</pre>	
Se un processo è in blocco da 10 ms significa che 10 ms fa ha eseguito una <i>system call</i>	
Ogni <i>interrupt</i> è associato ad un processo che ha richiesto una operazione di I/O	F
L'inversione di priorità è una tecnica utilizzata per evitare la <i>starvation</i> dei processi a bassa priorità	✓
Un processo per lanciare un nuovo processo deve fare una <i>system call</i>	✓

**Quesito 2:**

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 3:**

Lo studente mostri con una figura i possibili stati di avanzamento di un processo.

**Quesito 4:**

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:

<i>Processo</i>	<i>Allocate</i>	<i>Massimo</i>
<i>A</i>	1 0 2 1 1	3 1 2 1 3
<i>B</i>	2 0 1 1 1	3 3 4 2 1
<i>C</i>	1 1 0 1 0	2 1 4 1 0
<i>D</i>	1 1 1 1 0	1 1 3 2 1

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

[4.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?

Cognome e nome: \_\_\_\_\_ Matricola: \_\_\_\_\_ Posto: \_\_\_\_

**Quesito 5:**

**[5.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*)

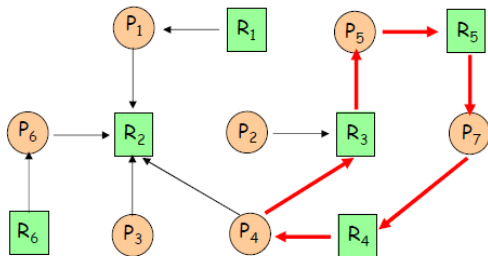
<pre> <b>void</b> Lettore (void) {      <b>while</b> (true) {          <b>P</b>(mutex);          numeroLettori++;          <b>if</b> (numeroLettori==1) <b>V</b>(database);          <b>V</b>(mutex);          // leggi il dato          numeroLettori--;          <b>if</b> (numeroLettori==0) <b>V</b>(database);          // usa il dato letto      }  } </pre>	<pre> <b>void</b> Scrittore (void) {      <b>while</b> (true) {          // prepara il dato da scrivere          <b>V</b>(database);          // scrivi il dato          <b>P</b>(database);      }  } </pre>
--	---

**[5.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**

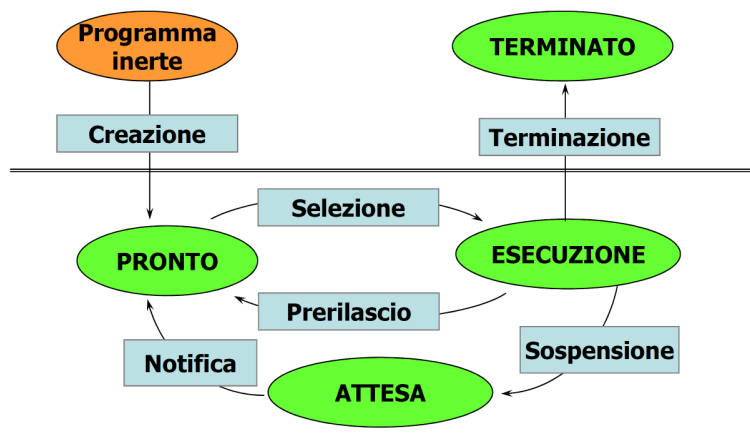
DOMANDA	Vero/Falso
Una system call dà sempre luogo ad un mode switch tra modalità utente e modalità kernel	V
Un process switch tra processi utente avviene sempre contestualmente a 2 mode switch (utente->kernel, kernel->utente)	V
Un interrupt viene gestito in modalità utente	F
Se un processo inizialmente attivo esegue il seguente codice, il sistema si troverà con 8 processi che eseguono InstrX, di cui uno è il processo iniziale. <pre>fork(); fork(); fork(); fork(); InstrX;</pre>	F
Se un processo è in blocco da 10 ms significa che 10ms fa ha eseguito una system call	V
Ogni interrupt è associato ad un processo che ha richiesto una operazione di I/O	F
L'inversione di priorità è una tecnica utilizzata per evitare la <i>starvation</i> dei processi a bassa priorità	F
Un processo per lanciare un nuovo processo deve fare una system call	V

**Soluzione al Quesito 2**

$P4 \rightarrow R3 \rightarrow P5 \rightarrow R5 \rightarrow P7 \rightarrow R4 \rightarrow P4$  sono in deadlock.

**Soluzione al Quesito 3**

Varie soluzioni possibili. Ad esempio:



Cognome e nome: \_\_\_\_\_ Matricola: \_\_\_\_\_ Posto: \_\_\_\_\_

**Soluzione al Quesito 4****[4.A]** La matrice delle necessità (massimo numero di risorse richieste dal processo - risorse allocate al processo) è la seguente:

```

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

```

Il proc. D potrebbe essere eseguito fino alla fine. Quando ha finito, il vettore delle risorse disponibili è [1 1 4 2 2].

Il proc. C potrebbe dunque essere eseguito e al suo completamento, il vettore delle risorse disponibili diverrebbe [2 2 4 3 2].

Questo permetterebbe di eseguire e terminare il processo A ottenendo [3 2 6 4 3] come vettore delle risorse disponibili.

Questo non permetterebbe però di eseguire il processo B in quanto mancherebbe una risorsa di tipo R2.

Il sistema NON è quindi in uno stato sicuro.

**[4.B]** L'Algoritmo del Banchiere (Banker's Algorithm)**Soluzione al Quesito 5****[5.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);
    }
}

```

**[5.B]**

Non importa la sintassi...

```

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

```