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: Cinque processi *batch*, identificati dalle lettere A, B, C, D, E, arrivano all'elaboratore agli istanti 0, 1, 2, 6, 7 rispettivamente. Essi hanno un tempo di esecuzione stimato di 4, 7, 2, 3, 1 unità di tempo rispettivamente e priorità 3, 5, 2, 3, 1 rispettivamente (dove 5 è la massima priorità e 0 è la minima). Per ognuna delle seguenti politiche di ordinamento:

A) *Round Robin* con priorità (divisione di tempo, con priorità, con prerilascio per priorità e con quanto di tempo di ampiezza 2)

B) *Shortest Remaining Time Next (SRTN)* ovvero SJF con prerilascio

Determinare, trascurando i ritardi dovuti allo scambio di contesto: (i) il tempo medio di risposta; (ii) il tempo medio di attesa; (iii) il tempo medio di *turn around*.

Nel caso di due processi aventi la stessa priorità, implicita o esplicita, di cui uno in esecuzione, si dia la precedenza a quello in esecuzione.

[4.A]: *RR* (divisione di tempo, con priorità, con prerilascio per priorità, e con quanto di tempo di ampiezza 2)

Proc. A																								
Proc. B																								
Proc. C																								
Proc. D																								
Proc. E																								

CPU																								
Coda																								

processo	t. risposta	t. attesa	*turn-around*
A			
B			
C			
D			
E			
medie			

[3.B]: *SRTN*

Proc. A																								
Proc. B																								
Proc. C																								
Proc. D																								
Proc. E																								

CPU																								
Coda																								

processo	t. risposta	t. attesa	*turn-around*
A			
B			
C			
D			
E			
Medie			

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

Quesito 4:
Lo studente riporti le 4 condizioni necessarie e sufficienti affinché possa verificarsi lo stallo (deadlock) di un sistema.
Diversamente dal solito modo di svolgere gli esercizi d'esame, è sufficiente elencarle, senza spiegarle.

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:
Nel retro di questo foglio, lo studente realizzi una soluzione al problema dei "filosofi a cena" utilizzando i *semafori*. Tale soluzione deve essere la più semplice possibile e non comportare rischi di *deadlock* (stallo) o *starvation*.
(Per coloro che avessero studiato solo sul libro di testo: *P*, corrisponde a *down*, *V* corrisponde a *up*)
Lo studente si ricordi di riportare l'indicazione di **tipo e 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

[3.A]

Proc. A	A	a	a	a	a	a	a	a	A	A	a	a	A												
Proc. B	-	B	B	B	B	B	B	B																	
Proc. C	-	-	c	c	c	c	c	c	c	c	c	c	c	c	C	C									
Proc. D	-	-	-	-	-	-	d	d	d	d	D	D	d	D											
Proc. E	-	-	-	-	-	-	-	e	e	e	e	e	e	e	e	e	E								

CPU	A	B	B	B	B	B	B	B	A	A	D	D	A	D	C	C	E								
Coda	-	a	a	a	a	a	a	a	d	d	a	a	d	c	e	e									
			c	c	c	c	d	d	c	c	c	c	c	e											
							c	c	e	e	e	e	e												
								e																	

processo	**t. risposta**	**t. attesa**	***turn-around***
A	0	9	13
B	0	0	7
C	12	12	14
D	4	5	8
E	9	9	10
medie	**5**	**7**	**10.4**

[3.B]

Proc. A	A	A	A	A																					
Proc. B	-	b	b	b	b	b	b	b	b	b	B	B	B	B	B	B	B								
Proc. C	-	-	c	c	C	C																			
Proc. D	-	-	-	-	-	-	D	d	D	D															
Proc. E	-	-	-	-	-	-	-	E																	

CPU	A	A	A	A	C	C	D	E	D	D	B	B	B	B	B	B	B								
Coda		b	c	c	b	b	b	d	b	b															
			b	b				b																	

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

processo	t. risposta	t. attesa	*turn-around*
A	0	0	4
B	9	9	16
C	2	2	4
D	0	1	4
E	0	0	1
medie	**2.2**	**2.4**	**5.8**

Soluzione al Quesito 4
- Accesso esclusivo a risorsa condivisa
- Accumulo di risorse
- Inibizione di prerilascio
- Condizione di attesa circolare

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)

Soluzione al Quesito 6
Varie soluzioni possibili, ad esempio quella del filosofo mancino:

```
int semaforo f[i] = 1;
Filosofo(i) {
  while(1) {
    <pensa>
    if(i == X) {
      P(f [(i+1)%N]);
      P(f [i]);
    } else {
      P(f [i]);
      P(f [(i+1)%N]);
    }
    <mangia>
    V(f [i]);
    V(f [(i+1)%N]);
  }
}
```