

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

Università degli Studi di Padova - Facoltà di Scienze MM.FF.NN. - Corso di Laurea in Informatica

**Regole dell'esame**

Il presente esame scritto deve essere svolto in forma individuale in un tempo massimo di 90 minuti 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 e la sessione orale avverrà in data e ora comunicate dal docente durante la prova scritta; i risultati saranno esposti sul sito del docente entro il giorno precedente gli orali.

Per superare l'esame il candidato deve acquisire almeno 1.5 punti nel Quesito 1 e un totale di almeno 18 punti su tutti i quesiti, inserendo le proprie risposte interamente su questi fogli. Riportare generalità e matricola negli spazi indicati.

Per la convalida e registrazione del voto finale il docente si riserva di proporre al singolo candidato una prova orale.

**Quesito 1 (punti 4): 1 punto per risposta giusta, diminuzione di 0,33 punti per risposta sbagliata, 0 punti per risposta vuota**

[1.A] Quale tra le seguenti affermazioni concernenti la politica di ordinamento *Round-Robin* è corretta:

1. il tempo di attesa di un processo è sempre maggiore o uguale del suo tempo di risposta
2. il tempo di attesa di un processo è sempre minore o uguale del suo tempo di risposta
3. il tempo di attesa di un processo è sempre uguale al suo tempo di risposta
4. il tempo di attesa di un processo e il suo tempo di risposta non hanno alcun legame prefissato.

[1.B] In quale tra i seguenti sistemi operativi è più conveniente l'utilizzo di *Inverted Page Tables*:

1. nessuno dei seguenti, il vantaggio è pari per tutti
2. sistemi a 16 bit
3. sistemi a 32 bit
4. sistemi a 64 bit

[1.C] Una *system call* bloccante causa sempre un *context switch*:

1. Sempre
2. Mai
3. Sì ma solo se la macchina ha più di un processore
4. Sì ma solo se c'è qualche altro processo attivo

[1.D]: Un semaforo binario può:

1. assumere solo valori discreti
2. gestire solo l'accesso a due risorse condivise
3. gestire solo le richieste di accesso provenienti da due processi
4. assumere solo i valori 0 e 1, con essi denotando "risorsa occupata" e "risorsa libera".

**RISPOSTE AL QUESITO 1:**

A \_\_\_\_\_ B \_\_\_\_\_ C \_\_\_\_\_ D \_\_\_\_\_

**Quesito 2 – (4 punti):**

Si consideri l'algoritmo AGING di *page replacement* con contatore (o stimatore) di 3 bit, e una memoria di 4 *frames* contenenti rispettivamente le pagine 1 2 3 e 4 di un certo processo. Si supponga che subito dopo uno *sweep* (aggiornamento del contatore) all'istante  $t_0$  i contatori siano inizializzati come segue:

contatore pagina 1: **110**      contatore pagina 2: **111**      contatore pagina 3: **101**      contatore pagina 4: **100**

All'istante  $t_1$  avviene uno *sweep*. Tra  $t_0$  e  $t_1$  è stata eseguita la seguente sequenza di accessi a memoria, nell'ordine: pagina 4, pagina 2; pagina 4, pagina 2.

[2.A] Che valore avranno i contatori dopo lo *sweep* in  $t_1$ ?

contatore pagina 1: \_\_\_\_\_ contatore pagina 2: \_\_\_\_\_ contatore pagina 3: \_\_\_\_\_ contatore pagina 4: \_\_\_\_\_

[2.B] Supponendo invece che subito dopo  $t_0$  fosse avvenuto un *page fault*, di quale pagina avrebbe causato la sostituzione? Perché?

1. Round Robin (divisione di tempo, senza priorità e con quanto di tempo di ampiezza 2)
2. Priority Scheduling (con prerilascio)

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

[illegible][illegible]

Processo	t. risposta	t. attesa	<i>turn-around</i>
A			
B			
C			
D			
E			
<b>Medie</b>			

[illegible][illegible]

Processo	t. risposta	t. attesa	<i>turn-around</i>
A			
B			
C			
D			
E			
<b>Medie</b>			

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

**Quesito 4 – (4 punti):**

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

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

[3.B] 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 la durata  $c$  può essere espressa come  $c = aq + b$  (ovvero  $a$  quanti di tempo  $+ b$ ; dove  $b$  è un tempo  $< q$  e corrisponde al tempo nell'ultimo quanto)

**Quesito 5 – (7 punti):**

Il problema dei lettori e scrittori è un classico problema di sincronizzazione tra più processi che accedono concorrentemente a una risorsa condivisa. **A)** Si descriva e discuta molto brevemente questo problema. **B)** Inoltre, utilizzando i *semafori* si proponga una procedura *reader* e una procedura *writer* in grado di rappresentarne una soluzione.

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

**Quesito 6 – (6 punti):**

Un sistema di controllo di una cisterna misura la quantità d'acqua contenuta nella stessa misurandone l'altezza ogni secondo facendo uso di un apposito sensore. Grazie a questa misura una centralina prende le dovute decisioni di svuotamento o riempimento. Il sensore comunica con la centralina per mezzo di una linea seriale asincrona; il sensore invia ogni misura corredandola di alcuni dati come segue:

- 2 byte per il proprio identificativo univoco;
- 4 byte per contenere un timestamp;
- 2 byte per la misura vera e propria;
- 1 byte contenente un dato ausiliario (crc).

Assumendo che:

la misura viene effettuata e spedita alla centralina ogni secondo; l'invio consiste nella spedizione da parte del sensore alla centralina del pacchetto di dati come sopra descritto; la linea seriale asincrona è configurata per funzionare ad una velocità di 1200 baud, "8N1" (8 bit di dati, nessuna parità, un solo bit di stop);

**a)** Calcolare la percentuale di utilizzo della linea (tale percentuale è il rapporto fra la banda realmente usata e la velocità massima permessa).

Assumendo invece che:

la centralina deve loggare le misure su una memoria persistente per 30 giorni; la centralina bufferizza i dati ricevuti e li scrive tal quali (=in formato grezzo) all'interno di un file binario, registrandolo su base oraria (ogni ora cambia file); i files vengono registrati su un filesystem basato su i-node, blocchi da 1KB, assumendo i-node ampi 128 B, record da 32 bit, i-node principale contenente 12 indici di blocco e 1 indice di I, II e III indizione ciascuno.

**b)** Calcolare il rapporto inflattivo nel caso in cui i files siano lasciati come files singoli contenenti i dati orari (pertanto in tutto:  $24 \times 30 = 720$  files)

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

**Soluzione****Soluzione al Quesito 1**

[1.A]: risposta 1

[1.B]: risposta 4

[1.C]: risposta 4

[1.D]: risposta 4

**Soluzione al Quesito 2**

[2.A] contatore pagina 1: \_011\_      contatore pagina 2: \_111\_      contatore pagina 3: \_010\_      contatore pagina 4: \_110\_

[2.B] Sostituirebbe la pagina 4 perché ha il valore di contatore più basso fra tutti.

**Soluzione al Quesito 3**

a) • RR (divisione di tempo, senza priorità e con quanto di tempo di ampiezza 2)

processo A	AAaaA	LEGENDA DEI SIMBOLI
processo B	-bBBbbbBBbbbBBbB	- non ancora arrivato
processo C	--cccCC	x (minuscolo) attesa
processo D	-----dddDDdddD	X (maiuscolo) esecuzione
processo E	-----eeeeE	. coda vuota
CPU	AABBACCBBDDDEBBDB	
coda	.baacbbddeebddb. ..ccb.deebbd.... ..... .....	

processo	risposta	tempo di attesa	turn-around
A	0	2	$2 + 3 = 5$
B	1	8	$8 + 7 = 15$
C	3	3	$3 + 2 = 5$
D	3	6	$6 + 3 = 9$
E	4	4	$4 + 1 = 5$
<b>medie</b>	<b>2,20</b>	<b>4,60</b>	<b>7,80</b>

b) • Priority Scheduling (con valori di priorità espliciti e con prerilascio)

processo A	AaaaaaaaaaAA	LEGENDA DEI SIMBOLI
processo B	-BBBBBBB	- non ancora arrivato
processo C	--ccccccccccCC	x (minuscolo) attesa
processo D	-----ddDDD	X (maiuscolo) esecuzione
processo E	-----eeeeeeeeE	. coda vuota
CPU	ABBBBBBBDDDAACCE	
coda	.aaaaaddaaaccee. ..ccccaacccee... .....ccee..... .....e.....	

processo	risposta	tempo di attesa	turn-around
A	0	10	$10 + 3 = 13$
B	0	0	$0 + 7 = 7$
C	11	11	$11 + 2 = 13$
D	2	2	$2 + 3 = 5$
E	8	8	$8 + 1 = 9$
<b>medie</b>	<b>4,20</b>	<b>6,20</b>	<b>9,40</b>

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

**Soluzione al Quesito 4****[3.A]** Un processo aspetta  $(N-1)c$  per ottenere la CPU.**[3.B]** L'utente aspetta  $Nqa + (N-1)b$  oppure anche scrivibile come  $c(N-1) + aq$ **Soluzione al Quesito 5**

Spiegazione e soluzione sono disponibili sul libro di testo e sulle slide dedicate agli esercizi sulla sincronizzazione.  
Segue un esempio di possibile soluzione.

```
typedef int semaphore;          /* use your imagination */
semaphore mutex = 1;           /* controls access to 'rc' */
semaphore db = 1;              /* controls access to the database */
int rc = 0;                     /* # of processes reading or wanting to */

void reader(void)
{
    while (TRUE) {              /* repeat forever */
        down(&mutex);           /* get exclusive access to 'rc' */
        rc = rc + 1;            /* one reader more now */
        if (rc == 1) down(&db); /* if this is the first reader ... */
        up(&mutex);             /* release exclusive access to 'rc' */
        read_data_base();       /* access the data */
        down(&mutex);           /* get exclusive access to 'rc' */
        rc = rc - 1;            /* one reader fewer now */
        if (rc == 0) up(&db);    /* if this is the last reader ... */
        up(&mutex);             /* release exclusive access to 'rc' */
        use_data_read();        /* noncritical region */
    }
}

void writer(void)
{
    while (TRUE) {              /* repeat forever */
        think_up_data();        /* noncritical region */
        down(&db);              /* get exclusive access */
        write_data_base();      /* update the data */
        up(&db);                /* release exclusive access */
    }
}
```

**Soluzione al Quesito 6**

**a)** Per ogni byte bisogna inviare 10 bit: 1 bit di start (sempre presente), 8 bit di dati (v. "8"N1), nessun bit di parità (v. 8"N"1), un (solo) bit di stop (v. 8"N"1").

Ogni misura consiste di  $2+4+2+1 = 9$  byte = 90 bit da spedire. Ad una misura al secondo è pertanto necessaria una velocità di almeno 90 baud.

La linea è ben dimensionata essendo da 1200 baud; la percentuale di utilizzo è  $90 / 1200 = 7,5 \%$

**b)** Un file contenente un'ora di dati grezzi occupa  $9 \times 3600 = 32400$  byte;

$32400 \text{ byte} / 1024 = 31,6 \text{ KB} = 32$  blocchi da 1KB;

Pertanto ogni file per la sua mappatura richiede l'i-node principale impegnando i 12 indici di blocco diretti e solo l'i-node di I indirezione dato che sono necessari 20 blocchi < 32 blocchi max. ( $32 = 128 \text{ B}$  ampiezza dell'i-node /  $4 \text{ B}$  ampiezza del record).

In totale si hanno  $24 \times 30 = 720$  files (un file all'ora per ogni giorno del mese)

Il rapporto inflattivo globale è definito dal rapporto fra lo spazio richiesto dagli i-node di mappatura e lo spazio utilizzato dai dati:

$2 \text{ i-node/file} \times 720 \text{ file} \times 128 \text{ B} / \text{i-node}$

----- = 0,78%

$32 \text{ blocchi} / \text{file} \times 1 \text{ K} / \text{blocco} \times 720 \text{ file}$