

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 politiche di ordinamento, in generale minimizza il tempo medio di attesa dei processi:

1. FCFS
2. Round-Robin con valutazione dell'attributo di priorità dei processi
3. Round-Robin senza valutazione dell'attributo di priorità dei processi
4. Shortest Job First.

[1.B]: Sia dato un sistema di memoria con indirizzi virtuali suddivisi in 4 campi:  $a$ ,  $b$ ,  $c$ ,  $d$ , i primi 3 dei quali siano utilizzati per indirizzare tre livelli gerarchici di tabelle delle pagine e il quarto campo rappresenti l'*offset* entro la pagina selezionata. Indicare dall'ampiezza di quali campi dipende il numero di pagine indirizzate nel sistema:

1. da quella di tutti e quattro i campi
2. da quella del campo  $d$
3. da quella del campo  $a$  e  $d$
4. da quelle dei campi  $a$ ,  $b$ ,  $c$ .

[1.C]: Un semaforo binario può:

1. assumere solo valori pari
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".

[1.D] Quale tra le seguenti affermazioni, fatte osservando un grafo di allocazione delle risorse, è certamente vera in generale:

1. se vi sono percorsi chiusi allora vi è situazione di stallo
2. se non vi sono percorsi chiusi allora non vi è situazione di stallo
3. se in un percorso chiuso rilevato si trovano solo risorse a molteplicità unaria, occorre analizzare il caso per decidere
4. nessuna delle precedenti tre possibili risposte.

**RISPOSTE AL QUESITO 1:**

A \_\_\_\_\_

B \_\_\_\_\_

C \_\_\_\_\_

D \_\_\_\_\_

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

- A) Round Robin (divisione di tempo, con priorità, senza prerilascio per priorità, e con quanto di tempo di ampiezza 2)
- B) Fair Priority Scheduling

Per evitare attesa infinita la politica di *Fair Priority Scheduling* prevede che, a seguito di due unità di tempo **consecutive** di esecuzione, la priorità del processo in esecuzione scenda di un punto. (*Esempio 1.* Se il processo è in esecuzione per 4 unità di tempo consecutive, la priorità di tale processo scende di 1 punto dopo le prime due unità temporali e di 1 altro punto dopo le ultime due unità temporali.) (*Esempio 2.* Se un processo è in esecuzione per 3 unità di tempo consecutivamente, la priorità di tale processo scende di 1 punto dopo le prime due unità temporali e basta; l'altra unità temporale di esecuzione non concorre in alcun modo, nemmeno successivamente, a far decrementare la priorità del processo in considerazione.) Infine, la priorità di un processo non risale mai e non può scendere sotto lo zero.

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*.

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

Nel caso di arrivi simultanei di processi allo stato di pronto, fatta salva l'eventuale considerazione del rispettivo valore di priorità, si dia la precedenza ai processi usciti dallo stato di esecuzione rispetto a quelli appena arrivati.

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

**[2.A]: RR** (divisione di tempo, con priorità, senza 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			

**[2.B]: Fair Priority Scheduling** (con priorità 3, 5, 2, 4, 1 per i processi A, B, C, D, E, dove 5 è la massima priorità)

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			

### Quesito 3 (4 punti):

Si consideri un sistema dotato di memoria virtuale, con memoria fisica divisa in 4 page frame. Il tempo di caricamento, tempo di ultimo accesso e i *bit* di R (*Referred*) e M (*Modified*) per ogni pagina sono come mostrato nella tabella sottostante.

pagina	tempo caricamento	ultimo riferimento	R	M
0	132	286	1	0
1	236	251	0	1
2	154	267	0	0
3	91	301	1	1

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

Si supponga che il sistema abbia bisogno di caricare in memoria una nuova pagina. Giustificando (molto brevemente) la risposta, quale di quelle in tabella sarà rimpiazzata se si adotta una politica:

- a) NRU
- b) FIFO
- c) LRU
- d) second chance

**Quesito 4 – (8 punti):**

I “filosofi a cena” è un classico problema di sincronizzazione tra più processi.

**a)** Lo studente descriva brevemente di che problema si tratta e le analogie con scenari inerenti i sistemi operativi.

**b)** Come visto in aula, lo studente utilizzi i semafori per scrivere una procedura `Filosofo`. Tali procedure dovranno poter essere eseguite concorrentemente (come fossero un gruppo di filosofi a cena, appunto) evitando il *deadlock* del sistema (anche le soluzioni con possibile *starvation*, ma comunque senza *deadlock*, saranno considerate corrette).

Si consideri un tavolo con  $N$  filosofi ed  $N$  forchette.

ATTENZIONE: lo studente si ricordi di inizializzare i valori delle variabili semaforo usate nella sua soluzione.

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

**Quesito 5 (6 punti):**

In una chiavetta USB da 2 GB con filesystem FAT32 e blocchi da 4K vengono registrati 100 files da 1000 blocchi ciascuno.

Considerando che:

il tempo di lettura di un blocco è 100 microsecondi;

il tempo medio di accesso (tempo per raggiungere qualsiasi blocco da un altro qualsiasi, purché non contiguo altrimenti il tempo è nullo) è di 5 millisecondi;

si presuma che la FAT sia precaricata in memoria (si trascurino pertanto i tempi di accesso alla stessa).

Calcolare il tempo necessario a leggere tutti i files, nelle ipotesi

**a)** di minima frammentazione;

**b)** di massima frammentazione;

Si calcoli inoltre

**c)** il tempo di scanning dello spazio libero, nel caso a)

**Quesito 6 (4 punti):**

In aula si è discusso di come l'*algoritmo del banchiere* faccia procedere il sistema attraverso una serie di stati sicuri (*safe*).

Data la situazione schematizzata nelle tabelle seguenti, quante istanze della risorsa R1 devono essere disponibili affinché il sistema sia *safe*? (Commentare la scelta)

processi	risorse allocate			max ris. richieste		
	R1	R2	R3	R1	R2	R3
<b>P1</b>	2	1	1	4	2	2
<b>P2</b>	3	1	0	4	2	2
<b>P3</b>	1	3	0	1	6	1
<b>P4</b>	0	1	1	2	1	1

risorse disponibili		
R1	R2	R3
?	1	1

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

## Soluzione

### Soluzione al Quesito 1

[1.A]: risposta 4

[1.B]: risposta 4

[1.C]: risposta 4

[1.D]: risposta 2

### Soluzione al Quesito 2

a) • RR con quanto di tempo di ampiezza 2

processo A	AAAAAAAAAAAAA	LEGENDA DEI SIMBOLI
processo B	-BBBBBBBB	- non ancora arrivato
processo C	--ccccccccccCC	x (minuscolo) attesa
processo D	-----dddDDD	X (maiuscolo) esecuzione
processo E	-----eeeeeeeeE	. coda vuota
CPU	AABBBBBBBDDDACCE	
coda	.baaaadddaaacee.	
	..ccccaaaccce...	
	.....ccceee.....	
	.....ee.....	

processo	risposta	tempo di attesa	turn-around
A	0	10	$10 + 3 = 13$
B	1	1	$1 + 7 = 8$
C	11	11	$11 + 2 = 13$
D	3	3	$3 + 3 = 6$
E	8	8	$8 + 1 = 9$
<b>medie</b>	<b>4,60</b>	<b>6,60</b>	<b>9,80</b>

b) • Fair Priority Scheduling

processo A	AAAAAAAAAA	LEGENDA DEI SIMBOLI
processo B	-BBBBBbbbbBB	- non ancora arrivato
processo C	--ccccccccccCC	x (minuscolo) attesa
processo D	-----DDD	X (maiuscolo) esecuzione
processo E	-----eeeeeeeeE	. coda vuota
CPU	ABBBBBDDDAABBCCE	
coda	.aaaaaaaabbccee.	
	..cccbbbccee...	
	.....ccceee.....	
	.....ee.....	

Nota:

- All'istante 3, il valore di priorità di B passa da 5 a 4.

- All'istante 5, il valore di priorità di B passa da 4 a 3 (e viene dunque pririlasciato all'ingresso di D).

- All'istante 8, il valore di priorità di D passa da 4 a 3.

processo	risposta	tempo di attesa	turn-around
A	0	8	$8 + 3 = 11$
B	0	5	$5 + 7 = 12$
C	11	11	$11 + 2 = 13$
D	0	0	$0 + 3 = 3$
E	8	8	$8 + 1 = 9$
<b>medie</b>	<b>3,80</b>	<b>6,40</b>	<b>9,60</b>

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

**Soluzione al Quesito 3**

- NRU rimuove ovvero la pagina 2 perché è l'unica che abbia  $R = 0$  e  $M = 0$ .
- FIFO rimuove la prima pagina che è stata caricata, ovvero la pagina 3.
- LRU rimuove la pagina 1 perché è quella riferita meno di recente tra quelle con  $R = 0$ .
- second chance rimuove la pagina più vecchia tra quelle con  $R = 0$ , ovvero la pagina 2.

**Soluzione al Quesito 4**

Varie soluzioni possibili, ad esempio: introdurre un filosofo mancino come illustrato nelle slide del corso, nella parte riguardante la sincronizzazione dei processi ed esercizi correlati.

**Soluzione al Quesito 5**

**a)** nel caso di minima frammentazione, tutti i blocchi dei files sono contigui e disposti uno di seguito all'altro. Si tratta perciò di leggere  $100 \times 1000 = 100.000$  blocchi;

per ognuno servono 100 microsecondi, dunque il risultato è  $100.000 \times 100 \text{ microsecondi} = 10 \text{ secondi}$ .

(e' considerato corretto anche il risultato che conta 5 ms iniziali per l'accesso al primo blocco:  $10 \text{ s} + 5 \text{ ms}$ )

**b)** nel caso di massima frammentazione, dato che ci sono in tutto  $2^{31} / 2^{12}$  blocchi = 524.288 blocchi, è possibile considerare che fra ogni blocco di dati ce ne sia uno libero. Pertanto i files sono tutti "polverizzati". Ad ogni lettura va sommato quindi un "salto" (perché non si tratta di blocchi contigui):

tempo totale = 10 secondi (dal punto precedente) +  $(100000-1) \times 5 \text{ ms} = 8 \text{ minuti e } 30 \text{ secondi}$ .

**c)** i blocchi liberi (escludendo lo spazio per la FAT stessa) sono  $524288 - 100000 = 424288$ , tutti contigui; pertanto si tratta di circa 42 secondi.

**Soluzione al Quesito 6**

La differenza tra il numero massimo di risorse richieste dai processi e il numero di risorse ad essi allocati è:

processi	max - allocate		
	R1	R2	R3
<b>P1</b>	2	1	1
<b>P2</b>	1	1	2
<b>P3</b>	0	3	1
<b>P4</b>	2	0	0

Innanzitutto è possibile notare che per essere sicuri di terminare P2 o P3 occorrerebbe liberare delle risorse R2 ed R3, ad esempio terminando P1 o P4.

Per essere in grado di terminare sicuramente P4 occorrono 2 risorse R1.

Per essere in grado di terminare sicuramente P1 occorrono 2 risorse R1, 1 risorsa R2 (disponibile), 1 risorsa R3 (disponibile).

Dunque perché il sistema sia *safe* occorrono almeno 2 risorse R1 disponibili.