

**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 2 ore 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 è sempre maggiore del tempo di risposta
2. il tempo di attesa è sempre minore del tempo di risposta
3. il tempo di attesa è sempre uguale al tempo di risposta
4. il tempo di attesa e il tempo di risposta non hanno alcun legame prefissato.

**[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]:** Quale tra i seguenti costituisce un criterio valido di valutazione di una politica di ordinamento di processi:

1. la capacità di trattare anche processi di lunga durata
2. il numero di processi completati per unità di tempo
3. il numero di processi in esecuzione per unità di tempo
4. il numero di processi in attesa di essere eseguiti.

**[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):**

Si consideri la seguente serie di riferimenti a pagine di memoria: 1, 2, 3, 4, 2, 1, 5, 6, 2, 1, 2, 3, 7, 6, 3, 2, 1, 2, 3, 6.

Si considerino le seguenti politiche di rimpiazzo:

- FIFO
- LRU
- Optimal

Quanti *page fault* avvengono considerando una RAM con solo 4 *page frame* ed inizialmente vuota?

Si completino inoltre le tabelle mostrando ad ogni istante il contenuto dei 4 page frame di cui è composta la RAM (non è necessario che lo studente mantenga un preciso ordine delle pagine virtuali nelle *page frame*).

Nota: nella tabella la prima riga indica la pagina di memoria virtuale riferita in quell'istante.

Politica di rimpiazzo **FIFO**; totale *page fault*? \_\_\_\_\_[illegible]

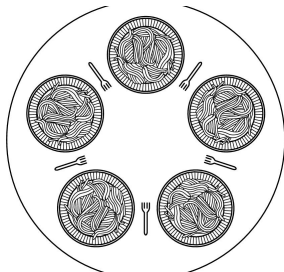
Cognome e nome: \_\_\_\_\_ Matricola: \_\_\_\_\_ Posto: \_\_\_\_\_

Politica di rimpiazzo **LRU**; totale *page fault*? \_\_\_\_\_

r1	r2	r3	r4	r2	r1	r5	r6	r2	r1	r2	r3	r7	r6	r3	r2	r1	r2	r3	r6

Politica di rimpiazzo **Optimal**; totale *page fault*? \_\_\_\_\_

r1	r2	r3	r4	r2	r1	r5	r6	r2	r1	r2	r3	r7	r6	r3	r2	r1	r2	r3	r6

**Quesito 3 – (6 punti):****Lo studente risolva, a scelta, UNO SOLO fra i quesiti 3.1 e 3.2****Quesito 3.1 (alternativa):**

I “filosofi a cena” è un classico problema di sincronizzazione tra più processi (i filosofi) che accedono concorrentemente a risorse condivise (le forchette).

Come visto in aula, lo studente utilizzi i semafori per scrivere una procedura *Filosofo* che cerchi a fasi alterne di pensare e mangiare. Tali procedure dovranno poter essere eseguite concorrentemente (come fossero un gruppo di filosofi a tavola) 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.

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

**Quesito 3.2 (alternativa):**

E' noto che esistano 4 condizioni che concorrono all'insorgere di situazioni di stallo. Lo studente ne citi 2 (illustrandole brevemente, ovvero in non più di due righe ciascuna).

**Inoltre:** Dato il sistema descritto dalla seguente rappresentazione insiemistica delle assegnazioni delle risorse:

$$\begin{aligned}
 P &= \{P_1, P_2, P_3, P_4, P_5\} \\
 R &= \{R_1^2, R_2^1, R_3^1, R_4^2\} \\
 E &= \{P_2 \rightarrow R_2, P_3 \rightarrow R_4, P_3 \rightarrow R_2, P_1 \rightarrow R_3, P_4 \rightarrow R_3, \\
 &\quad R_1 \rightarrow P_1, R_1 \rightarrow P_2, R_2 \rightarrow P_5, R_3 \rightarrow P_2, R_4 \rightarrow P_4, R_4 \rightarrow P_5\}
 \end{aligned}$$

si rappresenti il relativo grafo di allocazione delle risorse verificando anche se il sistema si trovi, nell'istante rappresentato, in condizione di stallo. Successivamente si discuta l'effetto che l'eventuale richiesta da parte del processo  $P_2$  della risorsa  $R_4$  causerebbe sullo stato corrente del sistema.

**Quesito 4 – (9 punti):**

Alcune applicazioni per PC fanno uso di un sistema di protezione basato su una cosiddetta “dongle” hardware, ovvero un dispositivo che viene connesso al PC e con il quale il programma da proteggere interagisce durante il funzionamento. Nella fattispecie, la *dongle* è pensata per non poter essere copiata (clonata), e l'algoritmo che contiene non viene reso noto; mentre la grande varietà di messaggi scambiati rende inutile una registrazione della conversazione per poi riproporla.

Ciò premesso, si consideri per semplicità di calcolo una *dongle* connessa ad un PC tramite una porta seriale (di fatto, le *dongle* moderne sono connesse tramite il bus USB, ma molte di esse vengono comunque rilevate come dispositivi seriali virtuali). La *dongle* oggetto del presente compito si occupa di ricevere dei dati dal PC sotto forma di stringhe di caratteri crittografati ed encodati di lunghezza fissa, e li restituisce al PC “in chiaro”, ovvero dopo averli decodificati e decrittati, sempre tramite messaggi di lunghezza fissa.

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

Esempio:

PC -> dongle: "1A6hK9051s68dS+Kw8e4f20Eg0B584wf+eNqFvA/7bnf/g4Sn165K+8LJdrgHe==" (64 byte)  
 (...elaborazione...)

dongle -> PC (testo in chiaro): "un proverbio: per un punto Martin perse la cappa" (48 byte)

Si considerino i seguenti dati tecnici:

- a.. i messaggi crittografati hanno tutti una lunghezza fissa di 64 byte;
- b.. la risposta in chiaro è lunga sempre 48 byte;
- c.. il tempo di decodifica e decrittazione si assuma fisso a 278 microsecondi.

**Parte 1) (4 punti)** Considerando una velocità della linea di comunicazione di 115200 baud e una configurazione della porta seriale "8N1" (8 dit di dati, nessuna parità, un bit di stop), calcolare quanti messaggi al secondo la dongle è in grado di processare.

**Parte 2) (2 punti)** Nel processo di un messaggio, calcolare il rapporto fra il tempo di sola decodifica/decrittazione e il tempo totale, nell'ipotesi di velocità 115200, 230400, 921600 baud (suggerimento: per semplificare i calcoli, si può sfruttare il fatto che si tratta di velocità multiple di 115200).

**Parte 3) (3 punti)** Considerando che il sistema riceva un milione di messaggi in una giornata (24h) da processare, calcolare qual è il rapporto di *idle* della *dongle*, ovvero il rapporto fra il tempo in cui è in azione (riceve i dati - li processa - li ritrasmette decrittati) e il tempo di pura attesa di nuovi dati da processare. Tale rapporto è utile per vedere se la *dongle* scelta (nonché la relativa configurazione di comunicazione) è adatta al carico di lavoro previsto: fra 0 e 10%: quasi scarica; fra 10% e 30%: normale carico di lavoro; oltre: sovraccarica.

**Quesito 5 – (5 punti):**

Una "chiavetta USB" da 2 GB è formattata con un *filesystem* di tipo FAT con blocchi da 1kB.

**Parte 1) (2 punti)** Calcolare la dimensione di ogni record della FAT (scegliendo fra lunghezze che siano potenze di 2; ovvero di 8, 16, 32 e 64 bit, non altri valori intermedi) e la dimensione totale della FAT.

**Parte 2) (3 punti)** Graficare l'andamento del rapporto inflattivo della FAT considerando che condivide la chiavetta con un file di dimensione minima 1B e massima uguale al massimo possibile (il rapporto inflattivo è definito come l'onere proporzionale dovuto alla memorizzazione delle strutture di rappresentazione rispetto a quella dei dati veri e propri). E' sufficiente una soluzione che mostri chiaramente il dominio e il codominio della funzione e ne definisca il comportamento intuitivo (es. retta, parabola, iperbole, altro...).

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

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

[1.A]: risposta 4 (a seconda di come si interpreta la frase nel quesito e in particolare la parola “maggiore” le risposte potevano essere diverse; pertanto, sono state considerate corrette anche la risposta 4 e “nessuna di queste”)

[1.B]: risposta 4

[1.C]: risposta 2

[1.D]: risposta 2

**Soluzione al Quesito 2**

Politica di rimpiazzo **FIFO**; totale *page fault*? **14** (quelli in grassetto)

r1	r2	r3	r4	r2	r1	r5	r6	r2	r1	r2	r3	r7	r6	r3	r2	r1	r2	r3	r6
<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	4	4	<b>5</b>	<b>6</b>	<b>2</b>	<b>1</b>	1	<b>3</b>	<b>7</b>	<b>6</b>	6	<b>2</b>	<b>1</b>	1	<b>3</b>	3
	1	2	3	3	3	4	5	6	2	2	1	3	7	7	6	2	2	1	1
		1	2	2	2	3	4	5	6	6	2	1	3	3	7	6	6	2	2
			1	1	1	2	3	4	5	5	6	2	1	1	3	7	7	6	6

Politica di rimpiazzo **LRU**; totale *page fault*? **10** (quelli in grassetto)

r1	r2	r3	r4	r2	r1	r5	r6	r2	r1	r2	r3	r7	r6	r3	r2	r1	r2	r3	r6
<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	2	1	<b>5</b>	<b>6</b>	2	1	2	<b>3</b>	<b>7</b>	<b>6</b>	3	2	<b>1</b>	2	3	6
	1	2	3	4	2	1	5	6	2	1	2	3	7	6	3	2	1	2	3
		1	2	3	4	2	1	5	6	6	1	2	3	7	6	3	3	1	2
			1	1	3	4	2	1	5	5	6	1	2	2	7	6	6	6	1

Politica di rimpiazzo **Optimal**; totale *page fault*? **8** (quelli in grassetto)

r1	r2	r3	r4	r2	r1	r5	r6	r2	r1	r2	r3	r7	r6	r3	r2	r1	r2	r3	r6
<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	4	4	<b>5</b>	<b>6</b>	6	6	6	6	<b>7</b>	<b>7</b>	<b>7</b>	<b>7</b>	<b>1</b>	1	1	1
	1	2	3	3	3	3	3	3	3	3	3	6	6	6	6	6	6	6	6
		1	2	2	2	2	2	2	2	2	2	3	3	3	3	3	3	3	3
			1	1	1	1	1	1	1	1	1	2	2	2	2	2	2	2	2

**Soluzione al Quesito 3.1**

Varie soluzioni possibili.

Ad esempio:

Lezioni Prof. Palazzi - Vedi slide numero 30 di [E01b - Esercizi Sincronizzazione.pdf \(Online\)](#)

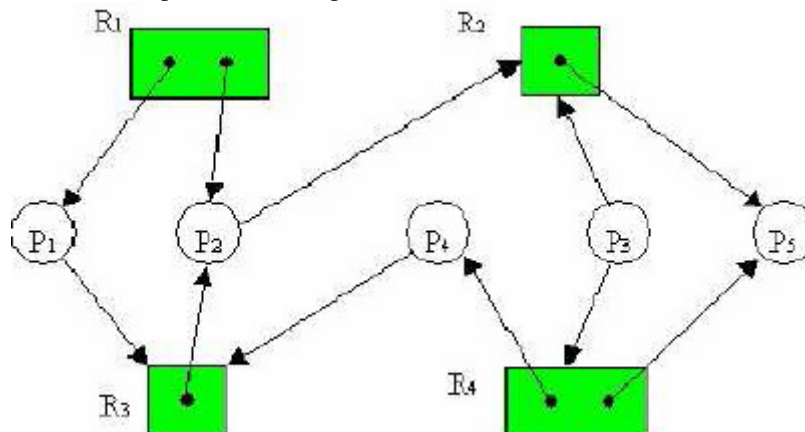
Lezioni Prof. Venier - Vedi slide numero 32 di [S01-Sincronizzazione.pdf \(Online\)](#)

**Soluzione al Quesito 3.2**

Le quattro condizioni sono mutua esclusione, attesa circolare, accumulo di risorse e inibizione del prerilascio. Una spiegazione di due righe può essere desunta dal libro di testo o dalle slide del corso (online).

La figura sottostante mostra graficamente la rappresentazione insiemistica del sistema proposta dal quesito.

In tale rappresentazione non si rilevano percorsi chiusi, per cui il sistema non si trova attualmente in condizione di stallo.



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

Qualora, in tale situazione, il processo P2 richiedesse la risorsa R4, la richiesta genererebbe invece il percorso chiuso:

P2 -> R4 -> P4 -> R3 -> P2.

In tale percorso chiuso è presente una risorsa a molteplicità multipla (R4) il che ci impedisce di fare affermazioni a priori sull'eventualità di stallo, richiedendo invece un'analisi puntuale del problema in esame.

E' facile vedere che P5 ha tutte le risorse di cui necessita al momento e potrebbe dunque eseguire e liberare R2 e una istanza di R4. A questo punto se entrambe le risorse venissero affidate a P2, oppure entrambe a P3, questi processi potrebbero eseguire, terminare, e liberare a loro volta le risorse in loro possesso (e così via): non si può quindi desumere che vi sia stallo.

Se invece le risorse R2 e R4 liberate da P5 vengono distribuite una a testa tra P2 e P3, si forma un ciclo che non può essere sciolto da altri. Il sistema è dunque in stallo.

Si noti che se si fosse seguito un ordine cronologico di richieste basato sull'ordine di questo testo si avrebbe che P2 ha chiesto R2 prima di P3, il quale però ha chiesto R4 prima di P2. Mantenendo le assegnazioni secondo quest'ordine si avrebbe stallo.

#### Soluzione al Quesito 4

##### Parte 1)

Per trasmettere 1 byte data la configurazione 8N1 devo di fatto trasmettere 10 bit; per ogni messaggio il volume di dati scambiati è PC-dongle + dongle-PC = 64 + 48 = 112 byte; 112 \* 10 = 1120 bit; a 115200 baud tale scambio impiega 9,722 millisecondi, ovvero 9722 microsecondi; a questo tempo dobbiamo sommare il tempo di decodifica e decrittazione del messaggio che è noto e pari a 278 microsecondi = 0,278 millisecondi; in tutto il tempo è 10 millisecondi.

Pertanto in un secondo il sistema può processare  $1 / 0,010 = 100$  messaggi.

##### Parte 2)

Esprimendo tutto in microsecondi (us):

velocità (baud)	tempo di decodifica/decriptazione (us)	tempo di scambio dati	tempo totale	rapporto %
115200	278 us (è fisso)	9722 us	10000 us	2.8 %
230400 (=115200 x 2)	278 us (è fisso)	$9722 / 2 = 4861$ us	5139 us	5.4 %
921600 (=115200 x 8)	278 us (è fisso)	$9722 / 8 = 1215$ us	1493 us	18.6 %

##### Parte 3)

Abbiamo calcolato che un messaggio in totale impiega 10 millisecondi per essere processato; pertanto un milione di messaggi in tutto impiegano  $0,01 \times 1000000 = 10000$  secondi per essere processati. In un giorno ci sono  $24 \times 3600 = 86400$  secondi; pertanto il rapporto è  $(10000 / 86400) = 0,116 = 11.6 \%$ ; il dispositivo funziona nel range di normale carico di lavoro.

#### Soluzione al Quesito 5

##### Parte 1)

totale blocchi = 2 GB / 1kB =  $2^{31} / 2^{10} = 2^{21}$  blocchi; servono 21 bit per indirizzare ogni blocco -> record da 32 bit.

totale dimensione della FAT =  $2^{21}$  record per 4 byte (ovvero 32 bit) = 8388608 byte.

##### Parte 2)

Per semplicità nel grafico riportiamo 8MB per indicare i 8388608 byte.

