

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. Per superare l'esame il candidato deve acquisire almeno 2 punti nel Quesito 1 e un totale di almeno 18 punti su tutti i quesiti, inserendo le proprie risposte interamente su questi fogli. Il candidato riporti generalità e matricola negli spazi indicati in alto.

**Quesito 1: (+0,5 punti per ogni risposta esatta; -0,25 punti per ogni risposta sbagliata; 0 punti se lasciata in bianco)**

DOMANDA	Vero/Falso
Se non vi sono percorsi chiusi in un grafo di allocazione allora non vi è situazione di stallo	
In un sistema a paginazione la traduzione degli indirizzi mantiene l'offset intatto	
Il TLB consente di ridurre gli accessi alla tabella delle pagine	
La segmentazione consente a due processi di condividere un segmento	
L'ampiezza dello spazio di indirizzamento virtuale dipende dalla dimensione in bit degli indirizzi	
La strategia di allocazione contigua può comportare frammentazione esterna	
Per deframmentare un disco occorre prima riformattarlo	
La politica di scheduling <i>Round Robin</i> non è soggetto al fenomeno di <i>starvation</i>	
Il meccanismo di <i>context-switch</i> è operato dallo scheduler dei processi	
Un processo che esegue in spazio kernel non può accedere al suo <i>stack</i> in spazio utente	

**Quesito 2:**

Consideriamo un file ampio 1MB con blocchi di dimensione 512B, allocato su un disco di dimensione 1GB con una struttura a blocchi anche essi ampi 512B. Si considerino le tre strategie di allocazione del file, contigua, concatenata e indicizzata, e si supponga che le informazioni che riguardano il file, in ognuno dei tre casi, siano già in memoria centrale. L'ultimo blocco letto dal file è il blocco 3, Il prossimo blocco da leggere è il blocco 8.

[N.B: stiamo sotto il vincolo che lo spazio indirizzamento deve essere una potenza di 2]

[2.A] Per ognuna delle strategie di allocazione (contigua, concatenata e indicizzata), si calcoli quanti accessi a disco sono necessari per la lettura del blocco 8 e si motivi la risposta.

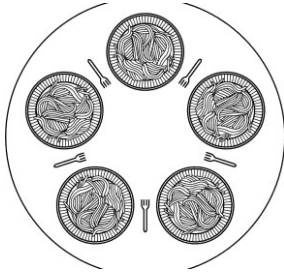
[2.B] Determinare le caratteristiche di una struttura di tipo nodo indice necessaria per poter tracciare il file assumendo che la struttura a nodo indice abbia la stessa dimensione di un blocco disco e contenga 15 puntatori diretti e 3 di indirizzamento indiretto di 1, 2, 3 livello rispettivamente (ovvero quanti puntatori/indici/blocchi saranno necessari per memorizzare il file?). Infine, si calcoli il rapporto inflativo dovuto all'uso di tale struttura.

processo	t. risposta	t. attesa	<i>turn-around</i>
A			
B			
C			
D			
E			
medie			



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

**Quesito 6:**



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 *deadlock* del sistema o *starvation* di filosofi.

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

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

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

## Soluzione

### Soluzione al Quesito 1

DOMANDA	Vero/Falso
Se non vi sono percorsi chiusi in un grafo di allocazione allora non vi è situazione di stallo	V
In un sistema a paginazione la traduzione degli indirizzi mantiene l'offset intatto	V
Il TLB consente di ridurre gli accessi alla tabella delle pagine	V
La segmentazione consente a due processi di condividere un segmento	V
L'ampiezza dello spazio di indirizzamento virtuale dipende dalla dimensione in bit degli indirizzi	V
La strategia di allocazione contigua può comportare frammentazione esterna	V
Per deframmentare un disco occorre prima riformattarlo	F
La politica di scheduling <i>Round Robin</i> non è soggetto al fenomeno di <i>starvation</i>	V
Il meccanismo di <i>context-switch</i> è operato dallo scheduler dei processi	F
Un processo che esegue in spazio kernel non può accedere al suo <i>stack</i> in spazio utente	F

### Soluzione al Quesito 2

[2.A] Da notare le corrispondenza 1:1 di blocchi disco e blocchi file. Per cui nei rispettivi casi avremo:

- allocazione contigua: 1 accesso al disco di massa permetterà di reperire direttamente il blocco 8.
- allocazione concatenata: 5 accessi al disco, in quanto i blocchi che separano il terzo dall'ottavo devono essere letti sequenzialmente.
- allocazione indicizzata: due accessi alla memoria, uno al blocco indice, se non presente in memoria centrale, per reperire il puntatore al blocco di dati richiesto, l'altro al blocco di dati. Vale anche la risposta "1 accesso alla memoria" se si considera il blocco indice relativo al blocco considerato già presente in memoria.

[2.B] Dai dati del problema abbiamo un disco ampio  $D_{\text{disco}} = 2^{30} \text{ B}$  con struttura a blocchi di dimensione  $D_{\text{blocco}} = 512 \text{ B} = 2^9 \text{ B}$ . Pertanto l'intero disco ha un numero blocchi pari a  $N_{\text{blocchi\_disco}} = 2^{30} / 2^9 = 2^{21}$ . Ne risulta che sono sufficienti 21 bit di indirizzo per poter indirizzare tutti i blocchi disco. Stando sotto l'ipotesi che un indirizzo memoria dovrebbe essere una potenza di due, ne risulta che lo spazio di indirizzamento è ampio  $A_{\text{si}} = 32 \text{ b} = 2^2 \text{ B}$ .

Inoltre, il file ha dimensione  $D_{\text{file}} = 1 \text{ MB} = 2^{20} \text{ B}$ . Pertanto il file occupa un numero blocchi disco pari a:

$$N_{\text{blocchi\_file}} = 2^{20} / 2^9 = 2^{11}$$

La struttura a nodo indice avrà le seguenti caratteristiche:

- 15 puntatori diretti che servono a tracciare 15 blocchi
- Non bastando i blocchi diretti, bisogna utilizzare il blocco di prima indizione. Tale blocco contiene  $D_{\text{blocco}} / A_{\text{si}} = 2^9 / 2^2 = 2^7$  indirizzi memoria. Risulta che  $2^{11} - (15 + 2^7)$  sono scoperti e non tracciabili, indicando il bisogno d'utilizzo di blocchi di seconda indizione.

- Essendo  $2^7$  la quantità indirizzi tracciabili con un singolo blocco, risulta che servono  $\sup((2^{11} - 15 - 2^7) / 2^7) = 15$  blocchi di seconda indizione.

Il rapporto inflativo che ne deriva è  $R_{\text{infl}} = [(1 + 1 + 1 + 15) * 512] \text{ B} / 2^{20} \text{ B} = 0.8\%$

### Soluzione al Quesito 3

L'ampiezza della memoria virtuale è di  $4 \text{ GB} = 2^{32} \text{ B}$ . Per poter indirizzare tutti i byte virtuali servirebbero 32b indirizzo. Avendo una pagina di  $4 \text{ KB} = 2^{12} \text{ B}$ , risulta che l'offset indirizzo fisico/virtuale sia di 12b, i restanti bit servono per indirizzare la pagina virtuale. Pertanto si ha questa configurazione dello spazio indirizzi virtuali:

$$I_{\text{virtuale}} = \{20 \text{ b \#pagina virtuale, 12 bit offset}\}$$

<b>processo</b>	<b>t. risposta</b>	<b>t. attesa</b>	<b><i>turn-around</i></b>
A	0	0	4
B	9	9	16
C	2	2	4
D	0	1	4
E	0	0	1
<b>medie</b>	<b>2.2</b>	<b>2.4</b>	<b>5.8</b>

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

### Soluzione al Quesito 5

[5.A,B,C] Abbiamo una memoria virtuale che può indirizzare un massimo di  $512 \text{ B} = 2^9 \text{ B}$ . Pertanto bastano 9b per coprire lo spazio virtuale. Di questi, 2 servono per l'offset, questo visto che abbiamo pagine di  $4 \text{ B} = 2^2 \text{ B}$ . Pertanto si ha la seguente configurazione dello spazio indirizzamento virtuale:

$$I_{\text{virtuale}} = \{7\text{b \#pagina}, 2\text{b offset}\}$$

In modo analogo, per poter indirizzare tutti i byte fisici serve un indirizzo fisico ampio,  $2^8 \text{ B} = 256 \text{ B}$ , 8b. Pertanto si ha la seguente configurazione dello spazio indirizzamento fisico:

$$I_{\text{fisico}} = \{6\text{b \#frame}, 2\text{b offset}\} \text{ o anche } \{7\text{b \#frame}, 2\text{b offset}\} \text{ dove però il primo bit rimarrà sempre 0}$$

[5.D] Indirizzo logico (0):  $0000000\{00\} \rightarrow 0001100\{00\}$ : pagina 0  
 Indirizzo logico (22):  $0000101\{10\} \rightarrow 0010000\{10\}$ : pagina 5  
 Indirizzo logico (4):  $0000001\{00\} \rightarrow 0000001\{00\}$ : pagina 1  
 Indirizzo logico (9):  $0000010\{01\} \rightarrow 0010001\{01\}$ : pagina 2  
 Indirizzo logico (19):  $0000100\{11\} \rightarrow 0001011\{11\}$ : pagina 4  
 Indirizzo logico (15):  $0000011\{11\} \rightarrow 0010110\{11\}$ : pagina 3

### 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]);
    }
}
```