

Cognome e nome: _____ Matricola: _____ Posto: _____

Quesito 3

Sia data una partizione di disco ampia **64 GB** organizzata in blocchi dati di ampiezza **1 KB** e indici di dimensione **32 bit**. In caso serva, si consideri l'ipotesi di contiguità **nulla** di un file (ciascun blocco si trova su disco in posizione non adiacente al blocco precedente e a quello successivo nella composizione del file).

[3.A] Si determini l'ampiezza massima di file ottenibile per l'architettura di file system **ext2fs** assumendo i-node ampi esattamente un blocco, i-node principale contenente **12** indici di blocco, **1** indice di I indirezione e **1** indice di II indirezione.

[3.B] Si determini la quantità di spazio occupata dalla struttura di i-node necessaria a rappresentare un tale file.

[3.C] Si determini la quantità di spazio occupata dal totale della struttura FAT in caso il file system usato per rappresentare il file calcolato in [3.A] nel sistema sopra descritto sia basato su FAT invece che su i-node (ext2fs).

Quesito 4:

Un sistema di allocazione della memoria ha le seguenti pagine libere, in questo ordine:
8KB, 14KB, 3KB, 11KB, 5KB, 7KB, 20KB, 18KB.

Si considerino tre richieste di allocazione che arrivano, una di seguito all'altra, nel seguente ordine:

A) 11KB; B) 6KB; C) 13KB.

Indicare a quali pagine vengono assegnate le tre richieste sequenziali A, B e C considerando le politiche *First Fit*, *Next Fit*, *Best Fit* e *Worst Fit*.

Nota: si assuma che, qualora un blocco libero venga assegnato a seguito di una richiesta di dimensione inferiore, il blocco libero sia comunque interamente assegnato.

	A)	B)	C)
<i>First Fit</i>			
<i>Next Fit</i>			
<i>Best Fit</i>			
<i>Worst Fit</i>			

Cognome e nome: _____ Matricola: _____ Posto: _____

Quesito 5

Il problema del “produttore/consumatore” è un classico problema di sincronizzazione tra più processi che accedono concorrentemente a risorse condivise. Lo studente utilizzi i **monitor** per scrivere due procedure chiamate `Producer` e `Consumer` che possano essere eseguite concorrentemente al fine di risolvere il problema evitando il *deadlock* del sistema.

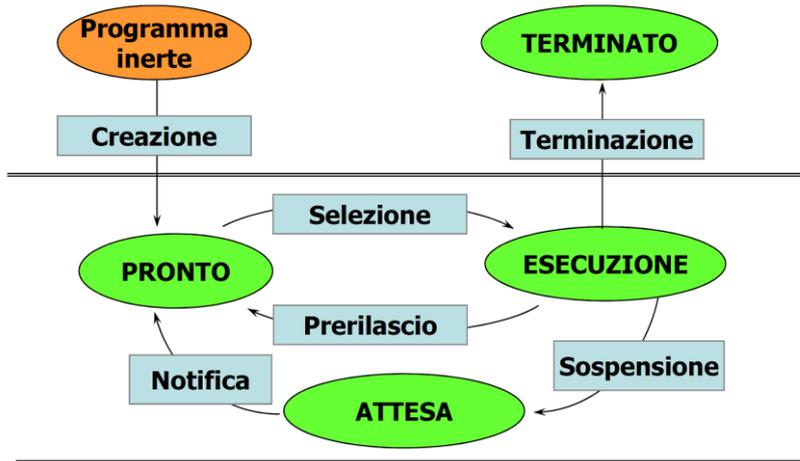
(Si consideri il caso in cui le risorse prodotte e non ancora consumate possano essere al massimo N).

Cognome e nome: _____ Matricola: _____ Posto: _____

Soluzione

Soluzione al Quesito 1

Varie soluzioni possibili. Ad esempio:



Soluzione al Quesito 2

Politica di rimpiazzo **FIFO**; totale *page fault*? **13** (quelli in grassetto)

r1	r2	r3	r4	r2	r1	r5	r6	r2	r1	r2	r3	r7	r6	r3	r2	r1	r6	r6	r2
1	2	3	4	4	4	5	6	2	1	1	3	7	6	6	2	1	1	1	1
	1	2	3	3	3	4	5	6	2	2	1	3	7	7	6	2	2	2	2
		1	2	2	2	3	4	5	6	6	2	1	3	3	7	6	6	6	6
			1	1	1	2	3	4	5	5	6	2	1	1	3	7	7	7	7

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	r6	r6	r2
1	2	3	4	2	1	5	6	2	1	2	3	7	6	3	2	1	6	6	2
	1	2	3	4	2	1	5	6	2	1	2	3	7	6	3	2	1	1	6
		1	2	3	4	2	1	5	6	6	1	2	3	7	6	3	2	2	1
			1	1	3	4	2	1	5	5	6	1	2	2	7	6	3	3	3

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	r6	r6	r2
1	2	3	4	4	4	5	6	6	6	6	6	7	7	7	7	1	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

[3.A] In questa soluzione useremo la notazione informatica tradizionale, con prefissi che denotano potenze di 2.

Sotto le ipotesi date, la dimensione di un indice è 32 bit = 4 B e il *file* di massima dimensione rappresentabile dall'architettura ext2fs fissata dal quesito sarà composto da:

- 12 blocchi, risultanti dall'utilizzo dei corrispondenti indici diretti presenti nell'i-node principale, al costo di 1 i-node, pari a 1 KB
- $\left\lfloor \frac{1024B}{4B} \right\rfloor = 256$ blocchi, risultanti dall'utilizzo dell'intero i-node secondario denotato dall'indice di I indirezione presente nell'i-node principale, al costo di 1 i-node aggiuntivo, pari a 1 KB
- $256^2 = 2^{16}$ blocchi, risultanti dall'utilizzo dell'indice di II indirezione, al costo di $1 + 256 = 257$ i-node aggiuntivi, pari a: $257 \times 1 \text{ KB} = 257 \text{ KB}$

In totale avremo dunque un ammontare di $12 + 256 + 65536 = 65804$ blocchi di dati ampi 1 KB, corrispondenti a 65804 KB (dimensione massima dei dati in un file)

Cognome e nome: _____ Matricola: _____ Posto: _____

[3.B] Per quanto scritto sopra è evidente che il costo complessivo in termini di spazio occupato dagli i-node è di $1 + 1 + 257 = 259$ i-node ampi 1 KB

[3.C] La FAT è unica per ogni partizione di disco e non dipende dalla quantità o dimensione dei file in essa contenuti. Essendo la memoria secondaria in questo esercizio ampia 64 GB e i blocchi dati ampi 1 KB, è immediato calcolare che la memoria secondaria è composta di $\left\lceil \frac{64GB}{1KB} \right\rceil = 64 \text{ M} = 2^{26}$ blocchi ognuno dei quali deve avere una entry nella FAT. Siccome ciascuna entry ha dimensione uguale a quella di un indice, ovvero 4 Byte, ne risulta che la dimensione totale della FAT è calcolabile come $= 2^{26} \times 2^2 = 2^{28} = 256 \text{ MB}$.

Soluzione al Quesito 4

	A)	B)	C)
<i>First Fit</i>	14KB	8KB	20KB
<i>Next Fit</i>	14KB	11KB	20KB
<i>Best Fit</i>	11KB	7KB	14KB
<i>Worst Fit</i>	20KB	18KB	14KB

Soluzione al Quesito 5

Varie soluzioni possibili, ad esempio:

monitor *ProducerConsumer*

condition *full, empty;*

integer *count;*

procedure *insert(item: integer);*

begin

if *count = N* **then** **wait**(*full*);

insert_item(item);

count := count + 1;

if *count = 1* **then** **signal**(*empty*)

end;

function *remove: integer;*

begin

if *count = 0* **then** **wait**(*empty*);

remove = remove_item;

count := count - 1;

if *count = N - 1* **then** **signal**(*full*)

end;

count := 0;

end monitor;

procedure *producer;*

begin

while *true* **do**

begin

item = produce_item;

ProducerConsumer.insert(item)

end

end;

procedure *consumer;*

begin

while *true* **do**

begin

item = ProducerConsumer.remove;

consume_item(item)

end

end;