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 45 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.
Il candidato riporti generalità e matricola negli spazi indicati in alto e inserisca le proprie risposte interamente su questi fogli.

Quesito 1:

DOMANDA	**Vero/Falso**
In un sistema di memoria a paginazione, il *Translation Lookaside Buffer* (TLB) velocizza la traduzione di indirizzi fisici in indirizzi virtuali	
La gestione della memoria con segmentazione può ridurre il consumo di memoria, in quanto consente a più processi di condividere blocchi di codice e di dati	
Nella gestione della memoria con paginazione, il fenomeno della frammentazione interna è tanto meno rilevante quanto più la lunghezza media dei programmi è grande rispetto alla dimensione della pagina	
Molti *page fault* su un processo non modificano le prestazioni degli altri processi	
Il nome e la dimensione sono due attributi di un file	
NTFS è il file system più utilizzato dai sistemi operativi GNU/Linux	
Con NTFS è possibile che il file system scriva il contenuto di file di piccola dimensione (es. <1KB) direttamente nel record dell'MFT	
`rmdir` è un comando GNU/Linux per rinominare directory	

Quesito 2:

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 *t0* i contatori siano inizializzati come segue:

contatore pagina 1: **010** contatore pagina 2: **111** contatorepagina 3: **001** contatore pagina 4: **100**

All'istante *t1* avviene uno *sweep*. Tra *t0* e *t1* è stata eseguita la seguente sequenza di accessi a memoria, nell'ordine:

pagina 1, pagina 3, pagina 1, pagina 3.

[2.A] Che valore avranno i contatori dopo lo *sweep* in *t1*?

contatore pagina 1: ____ contatore pagina 2: ____ contatore pagina 3: ____ contatore pagina 4: ____

[2.B] Supponendo che immediatamente dopo *t1* si verifichi un *page fault*, quale pagina sarebbe rimpiazzata? Perché?

[2.C] Supponendo invece che (al posto del caso precedente) tra *t0* e *t1* fossero stati eseguiti i seguenti accessi in memoria in sequenza:

pagina 1, pagina 3, pagina 4, pagina 3, pagina 2, pagina 3

In caso di *page fault* immediatamente dopo *t1*, quale pagina sarebbe rimpiazzata? (indicare anche come si arriva alla risposta)

Cognome e nome: _____ Matricola: _____ Posto: ____ ____

Quesito 3:

Sia data una partizione di disco ampia 128 GB organizzata in blocchi dati di ampiezza 1 KB. Si considerino degli indirizzi di dimensione minima (ma multipla di 8 bit) per indirizzare i blocchi di tale partizione. Si determini quindi l'ampiezza massima di file ottenibile per l'architettura di file system ext2fs nel caso pessimo di contiguità nulla, assumendo i-node ampi 512 B, i-node principale contenente 13 indici di blocco e 1 indice di I, II e III indirezione ciascuno. Si determini poi il rapporto inflattivo che ne risulta, ossia l'onere proporzionale dovuto alla memorizzazione delle strutture di rappresentazione rispetto a quella dei dati veri e propri.

Effettuati tali calcoli si discuta se e con quale rapporto inflattivo le architetture FAT e NTFS rispettivamente possano rappresentare file di tale ampiezza nella partizione data, sotto le medesime ipotesi di contiguità nulla. Per l'architettura NTFS si assumano record ampi 512 B, 208 B riservati all'attributo dati nel record principale e 400 B nei record di estensione.

Cognome e nome: _____ **Matricola:** _____ **Posto:** ____ ____

Quesito 4:
Si consideri un sistema con paginazione della memoria virtuale, pagine di 2^6 bytes e la seguente page table dove la riga più in alto corrisponde alla entry 0 e quella più in basso corrisponde alla entry 7

In/Out	Frame
In	00101
Out	01011
In	00001
Out	11010
In	00011
Out	10101
Out	11111
In	10101

Dire se i seguenti indirizzi logici genereranno un *page fault*. In caso negativo, scrivere l'indirizzo fisico corrispondente.
(In caso la tabella delle pagine non sia sufficiente a rispondere in alcuni casi, lo si dichiari)
a) 0000001101001

b) 0000010010110

c) 0000100000101

d) 0000010000100

Quesito 5:
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*? ____

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

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

Cognome e nome: _____ **Matricola:** _____ **Posto:** ____ ____

Soluzione

Soluzione al Quesito 1

DOMANDA	Vero/Falso
In un sistema di memoria a paginazione, il *Translation Lookaside Buffer* (TLB) velocizza la traduzione di indirizzi fisici in indirizzi virtuali	F
La gestione della memoria con segmentazione può ridurre il consumo di memoria, in quanto consente a più processi di condividere blocchi di codice e di dati	V
Nella gestione della memoria con paginazione, il fenomeno della frammentazione interna è tanto meno rilevante quanto più la lunghezza media dei programmi è grande rispetto alla dimensione della pagina	V
Molti *page fault* su un processo non modificano le prestazioni degli altri processi	F
Il nome e la dimensione sono due attributi di un file	V
NTFS è il file system più utilizzato dai sistemi operativi GNU/Linux	F
Con NTFS è possibile che il file system scriva il contenuto di file di piccola dimensione (es. <1KB) direttamente nel record dell'MFT	V
`rmdir` è un comando GNU/Linux per rinominare directory	F

Soluzione al Quesito 2

[2.A] contatore pagina 1: _101_ contatore pagina 2: _011_ contatore pagina 3: _100_ contatore pagina 4: _010_

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

[2.C] Sostituirebbe la pagina 3 perché ha il valore di contatore più basso fra tutti (101, 111, **100**, 110)

Soluzione al Quesito 3

In questa soluzione useremo la notazione informatica tradizionale, con prefissi che denotano potenze di 2.
Essendo la memoria secondaria ampia 128 GB e i blocchi dati ampi 1 KB, è immediato calcolare
che sono necessari: $\left\lceil \frac{128GB}{1KB} \right\rceil$ = 128 M = $2^7 \times 2^{20} = 2^{27}$ indici, la cui rappresentazione binaria banalmente richiede 27 bit.
Stante l'ovvio vincolo che la dimensione dell'indice debba essere un multiplo di un "ottetto" (8 bit), otteniamo la dimensione di 32 bit (4 B).

File system di tipo ext2fs:
Sotto queste ipotesi, il file di massima dimensione rappresentabile dall'architettura ext2fs fissata dal quesito sarà composto da:

- 13 blocchi, risultanti dall'utilizzo dei corrispondenti indici diretti presenti nell'i-node principale, al costo di 1 i-node, pari a 512 B
- $\left\lfloor \frac{512B}{4B} \right\rfloor$ = 128 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, pari a 512 B
- $128^2 = 2^{14}$ = 16 K blocchi, risultanti dall'utilizzo dell'indice di II indirezione, al costo di 1 + 128 = 129 i-node, pari a: 129 × 512B = (4.096 + 128)B = (2^{16} + 512) B = 66.048 B
- $128^3 = 2^{21}$ = 2 M blocchi, risultanti dall'utilizzo dell'indice di III indirezione, al costo di 1 + 128 + 128^2 = 16.513 i-node, pari a: 16.513 × 512 B = 8.454.656 B

corrispondenti a 13 + 128 + 16.384 + 2.097.152 = 2.113.677 blocchi ampi 1 KB, al costo complessivo di 1 + 1 + 129 + 16.513 = 16.644 i-node
i-node ampi 128 B, per un rapporto inflattivo di: $\frac{16.644 \times 512\,\text{B}}{2.113.677 \times 1\,\text{KB}} = \frac{16.644}{2.113.677 \times 2} = 0{,}39\%$.
Vediamo ora di determinare se e in che modo le architetture di file system FAT e NTFS siano in grado di rappresentare file di tale ampiezza sotto le ipotesi fissate dal quesito.

Cognome e nome: _____ **Matricola:** _____ **Posto:** ____ ____

File system di tipo FAT:

La struttura FAT, che rappresenta la vista dell'intera partizione in termini di blocchi dati, sarà composta da $\left\lceil \frac{128GB}{1KB} \right\rceil$ = 128 M celle ampie 4 B, una per indice di blocco: di queste, il file che dobbiamo rappresentare ne occuperà 2.113.677, per un rapporto inflattivo — calcolato considerato che l'architettura FAT concettualmente usa l'intera struttura per ogni singolo file — pari a: $\frac{128\,\mathrm{M}\times 4\,\mathrm{B}}{2.113.677\ \mathrm{KB}} = \frac{512\,\mathrm{MB}}{2.113.677\,\mathrm{KB}} = \frac{524.288\,\mathrm{KB}}{2.113.677\,\mathrm{KB}} = 24{,}80\%$

Nota: anche il caso 28 bit (anziché 32 bit = 4B è stata considerata corretta per quanto discusso in aula sulla FAT-32).

File system di tipo NTFS:

Dei 208 B riservati all'attributo dati nel record principale, 2 × 4 B = 8 B saranno riservati alla coppia {base, indice}, mentre i rimanenti 208 − 8 = 200 B potranno essere utilizzati per denotare le sequenze contigue che, sotto le ipotesi di contiguità nulla fissate del quesito, sono tutte ampie 1 blocco. Poiché ciascuna sequenza di tipo {inizio, fine} richiede 8 B, il record principale potrà ospitare: $\left\lfloor \frac{200B}{8B} \right\rfloor = 25$, mentre un singolo *record* di estensione dispone di 400 B per la memorizzazione di $\left\lfloor \frac{400B}{8B} \right\rfloor = 50$ ulteriori sequenze. Ne segue che, per rappresentare un file dell'ampiezza data, l'architettura NTFS necessiterà, in prima approssimazione, di: $1 + \left\lceil \frac{2.113.677 - 25 blocchi}{50 blocchi/record} \right\rceil = 1 + \left\lceil \frac{2.113.652}{50} \right\rceil = 1 + 42.274 = 42.275$ *record*.

In conclusione, un rapporto inflattivo pari a: $\frac{42.275\times\ 512\mathrm{B}}{2.113.677\,\mathrm{KB}} = 1\ \%$.

Soluzione al Quesito 4

Vista la dimensione di pagina allora gli ultimi 6 bit sono l'offset all'interno della pagina mentre i precedenti indicano la pagina. Dunque potremmo riscrivere gli indirizzi logici separando le due parti

a) 0000001 101001
b) 0000010 010110
c) 0000100 000101
d) 0000010 000100

Usando i primi sette bit come selettori nella tabella delle pagine otteniamo che

a) pagina 1, è out, quindi page fault.
b) pagina 2 – pagina valida, (sostituisco prima parte con quanto dentro la tabella delle pagine) diventa 00001010110
c) pagina 4 – pagina valida, (sostituisco prima parte con quanto dentro la tabella delle pagine) diventa 00011000101
d) pagina 2 – pagina valida, (sostituisco prima parte con quanto dentro la tabella delle pagine) diventa 00001000100

Soluzione al Quesito 5

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**
1	**2**	**3**	**4**	4	4	**5**	**6**	**2**	**1**	1	**3**	**7**	**6**	6	**2**	**1**	1	**3**	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**
1	**2**	**3**	**4**	2	1	**5**	**6**	2	1	2	**3**	**7**	**6**	3	2	**1**	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**
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