

Sistemi Operativi
Prova Scritta PAP 2018

Nome e Cognome:

Luciano Moreira

Tempo a disposizione: 90 minuti / Documentazione: 1 foglio di appunti A5 manoscritto.

Vero o falso? [10 punti]

Domanda 1 L'algoritmo di scheduling I/O *Elevator* è inutile nel caso di una memoria SSD

1/1



Vero



Falso

Domanda 2 Il journaling di tipo *ordered* non garantisce la consistenza dei dati utente

1/1



Vero



Falso

Domanda 3 La dimensione dei frame è legata alla quantità di memoria fisica disponibile

1/1



Vero



Falso

Domanda 4 Il comando TRIM obbliga il controller del disco SSD ad effettuare un'operazione di *garbage collection*

1/1



Vero



Falso

Domanda 5 Il numero di settori per ogni cilindro è sempre uguale

0/1



Vero



Falso

Domanda 6 Il bit R nella tabella delle pagine viene messo a 0 dalla MMU

0/1



Vero



Falso

Domanda 7 La dimensione delle pagine è legata alla quantità di memoria virtuale disponibile

1/1



Vero



Falso

Domanda 8 La struttura della tabella delle pagine dipende dall'architettura hardware del sistema

0/1



Vero



Falso

Domanda 9 Su un disco rigido *tradizionale* il *seek time* è legato alla velocità di rotazione del disco

1/1



Vero



Falso

Domanda 10 Il wear-out di una memoria SSD è aggravato se il sistema operativo utilizza il TRIM

1/1



Vero



Falso



Gestione della memoria [40 punti]

► **Domanda** Supponiamo di avere un sistema con 5 frame di pagina che utilizza l'**algoritmo della seconda opportunità**. Quando le pagine vengono caricate in memoria hanno il bit $R=0$. Supponiamo che il bit R venga messo a 1 solo dall'algoritmo. Vengono effettuati i seguenti riferimenti alle pagine:

6 3 0 5 2 6 4 7 8 7

- Qual è il numero totale di page fault? [5 punti]

5/5

☐ 5 ☐ 6 ☐ 7 ☒ 8 ☐ 9

- Quali pagine si troveranno in memoria dopo l'ultimo riferimento? [5 punti]

5/5

☒ [6, 4, 7, 8, 2] ☐ [0, 5, 6, 7, 8] ☐ [3, 6, 4, 2, 8] ☐ [5, 2, 6, 4, 7]

► **Domanda** Supponiamo di avere un sistema che utilizza l'**algoritmo NRU**. Sono stati effettuati diversi accessi alle pagine, e la situazione attuale nella tabella delle pagine è:

Pagina	Caricata al tempo	R	M
0	32ms	1	0
1	200ms	0	0
2	80ms	0	1
3	145ms	1	1

- Qual è l'ordine di rimpiazzamento delle pagine (da quella che verrebbe rimossa per prima a quella che verrebbe rimossa per ultima)? [5 punti]

5/5

☐ [0,2,3,1] ☐ [1,2,3,0] ☒ [1,2,0,3] ☐ [0,1,2,3]

► **Domanda** Supponiamo di avere un sistema con 4 frame che utilizza l'**algoritmo LRU in hardware (con una matrice)**. Vengono effettuati i seguenti accessi ai frame:

2 3 0 1 0 0 2 3 3 2

- Qual è il contenuto finale della matrice? [5 punti]

5/5

☐

0	1	0	0
1	0	0	0
1	1	0	0
1	1	0	0

☐

0	1	0	1
0	0	0	1
1	1	0	0
1	1	1	0

☒

0	1	0	0
0	0	0	0
1	1	0	1
1	1	0	0

- Quale frame verrebbe liberato per primo? [2 punti]

2/2

☐ 0 ☒ 1 ☐ 2 ☐ 3



► **Domanda** Supponiamo di avere un sistema con 4 pagine in memoria che utilizza l'algoritmo **NFU (modificato con invecchiamento)**. All'inizio i contatori (di **lunghezza 4 bit**) sono tutti a **zero**. Al primo ciclo di clock i bit R sono **1101** (1 per la pagina 0, 1 per la pagina 1, 0 per la pagina 2 e 1 per la pagina 3). Successivamente i valori sono **0001**, **0010**, **0011**, **0100**, **1101**, **0110** e infine **1011**.

- Qual è il valore dei contatori dopo l'ultimo ciclo di clock ? **[4 punti]**

4/4

☐

1	0	1	1
0	0	0	1
1	1	0	0
1	0	1	0

☒

1	0	1	0
0	1	1	1
1	1	0	0
1	0	1	0

☐

1	0	1	1
0	0	0	1
1	0	1	0
1	0	1	0

► **Domanda** Su computer che utilizza la segmentazione 8086 (Real Mode) scrivo il valore (esadecimale) **0x1a** nel registro selettore; successivamente accedo all'indirizzo virtuale (esadecimale) **0x408**. Quale indirizzo fisico sto accedendo? **[2 punti]**

0/2

☐ 1058
 ☒ 1448
 ☒ 16538

► **Premessa alle prossime 5 domande** In un sistema che utilizza la paginazione, le pagine hanno una dimensione di **16 Bytes**. La tabella delle pagine che consideriamo è:

	V?	P?	Frame
0	1	1	5
1	1	0	
2	0	0	
3	1	1	3
4	1	1	0
5	1	1	2
6	1	0	
7	1	1	1

Domanda 11 Dalla traduzione dell'indirizzo virtuale **0x7A** ottengo... **[1 punto]**

1/1

☐ 0x5A
 ☒ 0x1A
 ☐ Invalid page fault
 ☐ Major page fault

Domanda 12 Dalla traduzione dell'indirizzo virtuale **0x23** ottengo... **[1 punto]**

0/1

☐ 0x23
 ☒ 0x21
 ☒ Invalid page fault
 ☐ Major page fault

Domanda 13 Dalla traduzione dell'indirizzo virtuale **0x5D** ottengo... **[1 punto]**

0/1

☐ 0x5D
 ☒ 0x2D
 ☒ Invalid page fault
 ☐ Major page fault

Domanda 14 Dalla traduzione dell'indirizzo virtuale **0x3C** ottengo... **[1 punto]**

0/1

☐ 0x33
 ☒ 0x3C
 ☐ Invalid page fault
 ☒ Major page fault



Domanda 15 Dalla traduzione dell'indirizzo virtuale **0x6B** ottengo... [1 punto]

2/1

☐

0x61

☒

0x6B

☐

Invalid page fault

☒

Major page fault

Domanda 16 Considera un sistema con 64 KiBytes (65536 Bytes) di memoria che utilizza l'astrazione **base-limite**. La seguente combinazione di **base** e **limite** è valida? Processo A: base 16000, limite 4000; Processo B: base 22000, limite 8000; Processo C: base 48000, limite 14000 [2 punti]

0/2

☒

non valido

☒

valido

► **Premessa alle prossime 2 domande** Considera un sistema con architettura a 32 bit e pagine di dimensione 4 KiBytes (4096 Bytes) che utilizza un sistema di paginazione con due livelli. La tabella di primo livello contiene 1024 riferimenti a 32 bit, mentre ogni riga di traduzione nelle tabelle di secondo livello occupa 32 bit.

Domanda 17 Quanto spazio verrà utilizzato dalle tabelle delle pagine per un processo che alloca una sola pagina virtuale? [2.5 punti]

2.5/2.5

☐

4096 Bytes

☐

32768 Bytes

☒

8192 Bytes

Domanda 18 Quanto spazio verrà utilizzato dalle tabelle delle pagine per un processo che alloca tutto lo spazio di indirizzamento virtuale? [2.5 punti]

0/2.5

☒

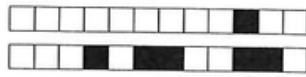
4198400 Bytes

☒

4194304 Bytes

☐

33587200 Bytes



Virtualizzazione [Risposta multipla 4x2.5 = 10 punti]

L'emulazione completa in software (*full software emulation*) non può essere considerata una tecnica di virtualizzazione perché...

0/2.5

- ☒ non vale il principio di efficienza ✗
- ☒ non vale il principio di equivalenza
- ☐ non permette l'esecuzione di un sistema operativo completo ✗
- ☒ non è possibile garantire la completa isolamento del sistema emulato
- ☐ Nessuna risposta è giusta.

La paravirtualizzazione è una tecnica che...

1.25/2.5

- ☐ comporta la traduzione dinamica delle istruzioni privilegiate del sistema ospite
- ☐ sta alla base dei software come VirtualBox o VMWare
- ☒ permette di migliorare l'efficienza nell'accesso ai dispositivi hardware
- ☒ necessita della modifica del sistema operativo ospite *guest*

La *shadow page table*...

2.5/2.5

- ☐ permette di evitare i *page fault* del sistema ospite (*guest*)
- ☒ viene gestita dall'hypervisor
- ☐ viene gestita dal sistema operativo ospite (*guest*)
- ☒ viene utilizzata dalla MMU del processore

Il meccanismo denominato *trap-and-emulate*...

1.5/2.5

- ☒ risolve il problema dell'esecuzione di istruzioni privilegiate a partire dal sistema ospite (*guest*)
- ☒ impedisce l'esecuzione di istruzioni privilegiate a partire dal sistema ospite (*guest*)
- ☐ viene implementato tramite la MMU del processore
- ☐ necessita di un sistema di virtualizzazione di tipo 1 (*bare metal*) ✗



Filesystem [20 punti]

► **Domanda** Consideriamo un sistema che implementa un algoritmo di schedulazione del braccio di un disco con 10 cilindri (cilindri da 1 a 10). La coda delle richieste dell'algoritmo di schedulazione è limitata a un massimo di 3 cilindri (cioè decide sulla base di tre richieste future al massimo): non appena una richiesta viene soddisfatta viene tolta dalla coda e un'altra entra nella coda. Le richieste che arrivano al kernel coinvolgono, nell'ordine, i seguenti cilindri:

7 4 5 2 1 2 8 6 3 5

- Qual è l'ordine con cui queste richieste verranno servite se la testina si trova inizialmente sul cilindro 6 e l'algoritmo utilizzato è SSTF (*Shortest Seek Time First*)? Nota: in caso di *parità* viene sempre scelto il cilindro con indice più piccolo [5 punti]

5/5

☐ [6,5,5,4,3,2,2,1,7,8]

[5,4,2,2,1,6,7,8,5,3]



[7,5,4,2,2,1,3,5,6,8]

- Qual è l'ordine con cui queste richieste verranno servite se la testina si trova inizialmente sul cilindro 6 e l'algoritmo utilizzato è LOOK, e il braccio si muove correntemente verso i cilindri con indice più grande? [5 punti]

5/5

☐ [8,6,5,3,1,2,2,4,5,7]

[7,5,4,2,2,1,3,5,6,8]



[7,4,5,2,1,2,8,6,3,5]

► **Domanda** Consideriamo un sistema che implementa un algoritmo di schedulazione del braccio di un disco con 10 cilindri (cilindri da 1 a 10). La coda delle richieste dell'algoritmo di schedulazione è limitata a un massimo di 3 cilindri (cioè decide sulla base di tre richieste future al massimo): non appena una richiesta viene soddisfatta viene tolta dalla coda e un'altra entra nella coda. Le richieste che arrivano al kernel coinvolgono, nell'ordine, i seguenti cilindri:

5 4 2 2 1 6 7 8 5 3

- Qual è l'ordine con cui queste richieste verranno servite se la testina si trova inizialmente sul cilindro 3 e l'algoritmo utilizzato è SSTF (*Shortest Seek Time First*)? Nota: in caso di *parità* viene sempre scelto il cilindro con indice più piccolo [5 punti]

5/5

☐ [7,5,4,2,2,1,3,5,6,8]

[2,2,1,4,5,6,5,3,7,8]



[6,5,5,4,3,2,2,1,7,8]

- Qual è l'ordine con cui queste richieste verranno servite se la testina si trova inizialmente sul cilindro 3 e l'algoritmo utilizzato è SCAN, e il braccio si muove correntemente verso i cilindri con indice più grande? [5 punti]

5/5

☐ [8,6,5,3,1,2,2,4,5,7]

[4,5,1,2,2,6,7,8,3,5]



[4,5,2,2,1,6,7,8,5,3]