

Start again

Inserisci qui i tuoi dati, compila subito questa parte.

5 cfu

6 cfu

Nome

email

Computer  
Ordinances

(509, 270)

erasmus,

Considera una architettura Pentium-like (4GB di memoria, pagina di 4 KB, **paginazione a due livelli**, pte 4 byte, root page table sempre in memoria).

Considera la seguente istruzione macchina

0x22802FFD store 0x22826DDD

che occupa 5 bytes a partire da **0x22802FFD** e copia il contenuto del registro A nei 4 bytes a partire dall'indirizzo specificato nell'operando.

Calcola quanti page fault può generare al più tale istruzione nelle fasi di **fetch** e di **execute** per **codice** o **dati** non residenti o per **page table** non residenti.

Page faults

	Fetch	Execute
page table		
codice o dati		

Descrivi brevemente il perché di tale risultato

Supponi che un disco di marca X abbia un Mean Time Between Failures (MTBF, cioè il valore atteso del tempo di guasto) pari a T.

- Immagina di avere array raid 0 composto da 2 dischi X. Qual'è il suo MTBF? perché?
- Immagina di avere array raid 5 composto da 4 dischi X. Qual'è il tempo medio (nel senso di valore atteso) per osservare l'array che va in stato degradato? perché?

Answer:

Trebuchet 1 (8 pt) Lang B I U S x<sub>2</sub> x<sup>2</sup>

[List icons] [Text icon] [Link icon] [Image icon] [Table icon] [Smiley icon] [Globe icon] [PDF icon] [Code icon] [Help icon]

[illegible]

Path:

**?** 

In un sistema sono presenti tre processi: A, B, C, inizialmente tutti e tre ready nell'ordine A in testa, poi B, C in coda. La politica di scheduling è **round robin** con quanto di tempo pari a 40ms. Attenzione: le system call possono essere bloccanti e i page fault major o minor.

- **A** I/O bound, I/O burst dura 25ms.
- **B** cpu bound ma genera alternativamente una system call non bloccante e un major page fault ogni 10 ms, ciascun page fault è servito in 50ms.
- **C** cpu bound, nessun page fault.

Il processore esegue di volta in volta A, B, C, e inoltre, con tempi trascurabili, mode switching, dispatching, system call e interrupt handlers. Mostra schematicamente, nella seguente tabella, l'ordine con cui tali attività vengono eseguite (una sola croce per ciascuna colonna). Indica anche quali processi sono running, quali ready e quali bloccati in ciascun istante come indicato nell'esempio.


[illegible]

	interrupt handler per I/O																					interrupt handler per I/O
	interrupt handler per quanto scaduto																					interrupt handler per quanto scaduto
stati processi	running	A																				running
	ready	B C																				ready
	block																					block
note tempi																						
altre note																						

 **Grep e Awk**

Il file di testo [packages.txt](#) contiene un record per ciascun pacchetto software della distribuzione linux debian. Nel file ciascun record è separato da una linea vuota, i campi sono su linee distinte, tranne il campo "Description" che è su più linee. Alcuni campi possono mancare. Per svolgere l'esercizio non è necessario conoscere il significato di tutti i campi.

Suggerimenti: alcune volte conviene processare tale file con awk usando RS="" (stringa vuota) e FS="\n".


5

Nel file packages.txt il campo Suggests contiene una lista di nomi di pacchetti separati da virgole ',' o da pipeline '|' e opzionalmente annotate con la versione tra parentesi.

Dai un comando **bastato su grep** che selezioni le righe relative ai campi "Suggests:" che contengono almeno due pacchetti i cui nomi iniziano per "lib"


ad esempio  
Suggests: automake, autoconf, **libtool**, **libtool-doc**

Answer:

6

Il campo Priority contiene la "priorità" del pacchetto e il campo Size la sua occupazione una volta installato. Considera l'insieme dei pacchetti con priorità "extra" e "optional": dai una linea di comandi che riporti la somma complessiva dei campi Size per tale insieme di pacchetti.

Answer:

7

**Debugging**

Considera il codice del seguente progetto [prj2.tar.gz](#). Compila tutti i file con il comando

```
gcc -g *.c -lm -o fib
```

Considera una esecuzione di fib con parametro **20**. Considera la **trentesima** volta in cui fib() sta per ritornare un valore **maggiore di 40**.

- Mostra lo **stack** in quell'istante.
- Mostra, nel contesto di **init\_list()**, il penultimo elemento della lista **L** in quell'istante.
- Mostra il numero di volte che la funziona fib() e' stata invocata fino a quel momento.

Answer:

8

**Pratica Unix**

Rispondi alle seguenti domande circa la gestione dei processi.

1. Cosa è un segnale in unix?
2. Come si invia un segnale ad un processo?
3. Cosa succede se si digita control-C su un terminale in termini di segnali inviati a processi?
4. Cosa succede se si digita control-Z su un terminale ai processi in background e a quelli in foreground?

Answer:

9 

### Windows: handles (solo per chi fa 6 cfu)

Rispondi brevemente alle seguenti domande.

1. Che cosa sono gli executive objects? Esistono in kernel space o in user space?
2. Fai tre esempi di executive objects.
3. Cosa sono gli handles? Esistono in kernel space o in user space?
4. Cosa è l'object manager? In che rapporto è con gli handle?
5. Cosa deve fare un processo per usare un executive object?

Answer:

Save without submitting

Submit all and finish

 [Moodle Docs for this page](#)

You are logged in as [Maurizio Pizzonia](#) (Logout)

SOpari20130905

[Validate HTML](#)   [Section 508 Check](#)   [WCAG 1 \(2.3\) Check](#)