

Sistemi Operativi 2015.09.24 (computer pari)

[Home](#) ► [Courses](#) ► [Miscellaneous](#) ► [SOpari20150924](#) ► [General](#) ► [Compito dispari - turno 1 - 6 cfu](#) ► [Preview](#)

Quiz navigation

[1](#) [2](#) [3](#) [4](#) [i](#) [5](#)

[6](#) [7](#) [8](#) [9](#)

[Finish attempt ...](#)

[Start a new preview](#)

Administration

Quiz administration

- [Edit settings](#)
- [Group overrides](#)
- [User overrides](#)

Edit quiz

Preview

Results

- [Locally assigned roles](#)
- [Permissions](#)
- [Check permissions](#)
- [Filters](#)
- [Logs](#)
- [Backup](#)
- [Restore](#)
- [Question bank](#)

Course administration

[Switch role to...](#)

[My profile settings](#)

[Site administration](#)

[Search](#)

Question 1

Not yet answered

Not graded

[Flag question](#)

[Edit question](#)

Dati studente

Inserisci qui i tuoi dati, **compila subito questa parte.**

Quanti CFU?

5 cfu

6 cfu

Cognome

Nome

Matricola

email

Numero Computer

Ordinamento (509, 270, erasmus,ecc.)

Question 2

Not yet answered

Not graded

[Flag question](#)

[Edit question](#)

Memory management

Rispondi, **brevemente e punto per punto**, alle seguenti domande sul concetto di "demand paging".

1. Che cosa si intende con demand paging?
2. Quali sono i vantaggi del demand paging?
3. Quali sono gli svantaggi del demand paging?
4. In che relazione è l'approccio demand paging con il meccanismo di mappatura in memoria di un file (cioè il servizio realizzato dalla system call mmap)?

Paragraph

1.

2.

3.

4.

Path: p

Question 3

Not yet answered

Not graded

[Flag question](#)

[Edit question](#)

I/O

Rispondi brevemente, ma **punto per punto**, alle seguenti domande sul filesystem dei sistemi unix.

1. che cosa è un inode
2. che cosa è un hardlink

3. che cosa è un softlink
4. può un hardlink essere non valido? (cioè puntare a qualcosa che non esiste)
5. può un softlink essere non valido?
6. può un hardlink puntare ad un file fuori dalla partizione in cui è contenuto?
7. può un softlink puntare ad un file fuori dalla partizione in cui è contenuto?

Paragraph

1.

2.

3.

4.

5.

6.

7.

Path: p

Question 4
 Not yet answered
 Not graded
 Flag question
 Edit question

Scheduling di processi e kernel

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** cpu bound ma genera un page fault ogni 20ms. Ciascun page fault è servito in 30ms.
- **B** cpu bound non genera page faults.
- **C** genera system call bloccanti ogni 20ms, 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.

	A	X	<input type="checkbox"/>									
user mode	B		<input type="checkbox"/>									
	C		<input type="checkbox"/>									
	mode switch		<input type="checkbox"/>									
kernel mode	disptatching		<input type="checkbox"/>									
	system call per I/O		<input type="checkbox"/>									
	interrupt handler per page fault		<input type="checkbox"/>									
	interrupt handler per I/O		<input type="checkbox"/>									
	interrupt handler per quanto scaduto		<input type="checkbox"/>									

	running	A	<input type="checkbox"/>																
stati processi	ready	B	<input type="checkbox"/>																
		C	<input type="checkbox"/>																
	block		<input type="checkbox"/>																
note tempi			<input type="checkbox"/>																
altre note	<input type="text"/>																		

Information

-  Flag question
-  Edit question

Grep e Awk

Il file di testo [ubuntu_packages.txt.gz](#) (devi decomprimerlo usando gunzip) contiene un record per ciascun pacchetto software della distribuzione linux ubuntu. Nel file ciascun record è separato da una linea vuota, i campi sono su linee distinte e ciascuna linea inizia con il nome del campo seguito da ":". 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".

Question 5

Not yet answered
Not graded

-  Flag question
-  Edit question

Nel file ubuntu_packages.txt le righe che iniziano per "Size:" contengono un numero intero. Dai un comando **bastato su grep o egrep** che selezioni di tali righe quelle il cui contenuto rispetti **entrambe** le seguenti regole:

- il numero sia maggiore di 3 200 000
- l'ultima cifra del numero sia maggiore o uguale a 5

 Paragraph

Copia e incolla il comando e l'output (almeno le prime righe, usa head se necessario).

Path: p » em

Question 6

Not yet answered
Not graded

-  Flag question
-  Edit question

Considera il campo **Homepage**, nel file ubuntu_packages.txt. Contiene l'URL dell'homepage in cui è ospitato il con progetto di sviluppo del software del pacchetto. Dell'URL ci interessa il nome del server. Ad esempio, in

<http://oss.sgi.com/projects/xfs/>

il nome del server è **oss.sgi.com**.

Dai una riga di comando che mostri la classifica dei primi dieci server per numero di progetti software ospitati. Esempio:

484 launchpad.net
 271 search.cpan.org
 152 www.mono-project.com
 126 www.libreoffice.org
 118 gcc.gnu.org

Paragraph

Copia e incolla il comando e il suo output (almeno le prime righe).

Path: p » em

Question 7

Not yet answered

Not graded

 Flag question

 Edit question
Debugging

Considera il codice del seguente progetto prj3.tar.gz. Compila tutti i file con il comando

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

Considera una esecuzione di **fib** con parametro **21**. Considera la **duecentounesima** volta in cui **fib()** è stata chiamata.

- Mostra lo **stack** in quell'istante.
- Conta quante volte **fib()** è ritornata fino a quell'istante.
- nel contesto di **init_list()** mostra il penultimo elemento della lista **L**
- Esprimi in una formula la relazione che lega il numero di frame relativi a **fib()** nello stack, le chiamate a **fib()**, e i ritorni da **fib()**

Paragraph

Path: p

Question 8

Not yet answered

Not graded

 Flag question

 Edit question
Pratica Unix

Rispondi alle seguenti domande circa segnali e processi.

1. Cosa è un segnale in unix?
2. Qual è il modo più generale per inviare 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 foreground? e a quelli in

background?

 Paragraph						
1. 2. 3. 4.						
Path: p						

Question 9

Not yet answered

Not graded

 Flag question [Edit question](#)**Windows vs. Unix (solo per chi fa 6 cfu)**

Rispondi alle seguenti domande che confrontano Windows e Unix.

1. In Unix gran parte delle informazioni di configurazioni sono in /etc, in Windows dove sono?
2. In Unix gran parte delle system call operano su file descriptor, descrivi il concetto che gioca lo stesso ruolo in Windows.
3. Unix organizza i processi ad albero, in Windows come sono organizzati i processi?

 Paragraph						
1. 2. 3.						
Path: p						

[Next](#) [Moodle Docs for this page](#)You are logged in as [Maurizio Pizzonia](#) ([Log out](#))

SOpari20150924