

Sistemi Operativi 2016.02.22 (computer pari)

You are logged in as [Maurizio Pizzonia](#) (Log out)[Home](#) ► [Courses](#) ► [Miscellaneous](#) ► [SOpari20160222](#) ► [General](#) ► [Compito pari - turno 1](#) ► [Preview](#)

Quiz navigation

[1](#) [2](#) [3](#) [4](#) [5](#) [6](#)[7](#)[Finish attempt ...](#)[Start a new preview](#)

Question 1

Not yet answered

Not graded

[Flag question](#)[Edit question](#)

Dati studente

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

Quanti CFU?

☐

6 cfu

☐

5 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 ma punto per punto**, a ciascuna delle seguenti domande circa i **page fault**.

1. Cosa succede al processo running quando si ha un minor page fault?
2. Elenca almeno 2 situazioni diverse in cui la CPU rileva un page fault e il kernel non uccide il processo
3. Come fa la CPU a capire che una pagina **non** è residente?
4. Mostra come sia possibile che una istruzione macchina **senza operandi** possa produrre 2 o più page fault quando la page table è ad **un livello**.

Paragraph

- 1.
- 2.
- 3.
- 4.

Path: p

Question 3

Not yet answered

Not graded

[Flag question](#)[Edit question](#)

I/O

Considera la sequenza di eventi che avvengono dopo una system call di **lettura da disco (read)** e rispondi **brevemente ma punto per punto** alle seguenti domande.

1. Spesso diciamo che la read è bloccante, ma in pratica esistono vari casi che rendono la

2. Supponi che la read sia stata bloccante, descrivi cosa succede dopo che il disco ha effettivamente finito la lettura.
3. Come tratta le read un disk scheduler anticipatorio? a che scopo?

Question 4

Not yet answered

Not graded

Flag question

























































































 [Edit question](#)

Sequenza degli eventi nel SO

In un sistema sono presenti tre processi: A, B, C. La politica di scheduling è **round robin** con quanto di tempo **q=30ms**.

- **A** è cpu bound, **a t=0 running**, nessun page fault .
- **B** è I/O bound, **a t=0 in blocco** per una lettura che verrà servita a **t=20ms**.
- **C** è 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.

user mode	A	X								
	B									
	C									
mode switch										
kernel mode	sched./disp.									
	system call									
	interrupt handler per page fault									
	interrupt handler per I/O									
	interrupt handler per quanto scaduto									
stati processi	running	A								
	ready	C								

	block	B								
note tempi										
altre note										

Question 5

Not yet answered

Not graded

Flag question

Edit question

SED

Nel file di testo `passwd.txt` i campi sono separati dai : (duepunti). Usare `sed` per **sostituire**, in tutte le righe che contengano "Daemon" nel quinto campo, il "false" a fine riga (se c'è) con "STOP". Non c'è bisogno di sostituire nel file, basta inviare il risultato su standard output.

Paragraph

Copia e incolla il comando e l'output.

Path: p » em

Question 6

Not yet answered

Not graded

Flag question

Edit question

AWK

Si consideri il file di testo `misurazioni.txt` i cui record hanno il seguente schema

`nome_persona : misurazione1 : misurazione2 : ...`

Ciascun record in tale file contiene le misurazioni effettuate da una persona. In ogni record, i campi sono separati da ":". Il primo campo è una stringa (il nome della persona) mentre gli altri campi sono valori interi compresi tra 0 e 30 (le misurazioni). Per ogni persona c'è un solo record. Non tutte le persone eseguono lo stesso numero di misurazioni, quindi il numero di campi per ciascun record non è costante.







Ad esempio

Mario:25:27:30:30
Carlo:22
Claudio:30:23:22
Andrea:28:0:2:4:5
Giovanni:30:21
Elsa:20

Scrivi uno script in awk (anche su un file separato, se vuoi) che produca in output un report con la media delle misurazioni **per ciascuna persona** e il **valore massimo** tra le medie ottenute, come nell'esempio

```
____MEDIE____  
Mario: 28  
Carlo: 22  
Claudio: 25  
Andrea: 7,8  
Giovanni: 25,5  
Elsa: 20  
  
Max: 28
```



___ FINE ___

	Paragraph					
<i>Copia e incolla lo script e l'output</i>						
Path: p » em						

Question 7

Not yet answered

Not graded

 Flag question Edit question**Debugging**

Considera il codice del seguente file [test.c](#) e compila tale file con il comando

```
gcc -g -o test test.c
```

Non ci sono errori (solo alcuni warning) nella compilazione, ma eseguendo il programma con input "8 9" (otto spazio nove) l'esecuzione termina improvvisamente.

```
pizzonia@pisolo$ ./test
8 9
Floating point exception (core dumped)
```

Esegui il debugger tramite il comando

```
gdb test
```

ed esegui i seguenti passi nell'ambiente di debugging nell'ordine specificato

- Inserisci un breakpoint su "main"
- fai "single step" fino all'input e inserisci come input "8 9"
- fai "single step" fino alla chiamata di "div" e entra dentro
- continua (senza single step!) finché il programma non dà errore
- stampa i valori di x e y nel contesto di div
- mostra lo stack
- stampa i valori di x e y nel contesto di main

	Paragraph					
<i>Fai copia e incolla di tutta la sessione.</i>						
Path: p » em » span						

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

Administration

SOpari20160222

