

Sistemi Operativi 2017.06.26 (computer dispari)

[Home](#) ► [Courses](#) ► [Miscellaneous](#) ► [SODispari20170626](#) ► [General](#) ► [Compito dispari - 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.**

Cognome

Nome

Matricola

email

Numero Computer

Ordinamento (509, 270, erasmus, ecc.)

Quanti CFU?

5 cfu

6 cfu

Question 2

Not yet answered

Marked out of 6.00

[Flag question](#)

[Edit question](#)

Gestione della memoria: working set

Considera un processo P a cui vogliamo applicare il metodo del working set. L'immagine di P è composta da 110 pagine partizionate in 3 gruppi come segue

gruppo	numero di pagine
A	20
B	40
C	50
totale	110

L'esecuzione di P avviene per fasi. Ciascuna fase dura un certo numero di istanti di tempo ed accede a certi gruppi di pagine. Ecco il susseguirsi delle fasi.

fase	durata	inizio	fine	gruppi acceduti
1	400	0	399	B
2	400	400	799	BC
3	400	800	1199	AB
4	400	1200	1599	ABC

Supponi che sia $\delta=100$ e che **gli accessi nel tempo siano uniformemente distribuiti sulle pagine**. Rispondi brevemente alle seguenti domande.

1. Fase 2, quant'è la taglia del WS a regime?
2. Transitorio tra fasi 1 e 2. Quanti istanti devono trascorrere perché WS arrivi regime? Spiega.
3. Fase 3, quant'è la taglia del WS a regime?
4. Transitorio tra fasi 2 e 3. Quanti istanti devono trascorrere perché WS arrivi regime? Spiega
5. Transitorio tra fasi 2 e 3. Qual'è la **taglia massima** raggiunta dal WS durante il transitorio? A che istante si raggiunge? Spiega
6. Fase 4. Qual'è la taglia del WS durante la fase 4? Spiega



Paragraph



Rispondi punto per punto

- 1.
- 2.
- 3.
- 4.
- 5.
- 6.

Path: p » em

Question 3

Not yet answered
Marked out of 6.00

- Flag question
- Edit question

I/O: read vs. write

Considera le operazioni read e write su disco. Rispondi alle seguenti domande circa la comparazione delle due operazioni.

1. Le system call read e write sono bloccanti o no? Descrivi il loro comportamento motivando.
2. Quando una read o una write viene effettivamente conclusa dal disco si genera un interrupt. Cosa fa l'interrupt handler nel caso della read? e nel caso della write?
3. Read e write hanno comportamenti molto diversi per quanto riguarda la fairness rispetto agli altri processi. Spiega.

Paragraph

Rispondi punto per punto

- 1.
- 2.
- 3.

Path: p » em

Question 4

Not yet answered
Marked out of 6.00

- Flag question
- Edit question

Sequenza degli eventi nel SO

In un sistema sono presenti tre processi: A, B, C. La politica di scheduling è **preemptive con priorità (A>B>C)**.

- **A, inizialmente in blocco, è cpu bound.** Genera un major page fault ogni 40ms. E' inizialmente in blocco a causa di un page fault che verrà servito a tempo t=10ms.
- **B, inizialmente in blocco, è I/O bound.** Ciascun I/O viene servito in 50ms. E' inizialmente in blocco a causa di un I/O che verrà servito a tempo t=30 ms.
- **C, inizialmente running,** è cpu bound e non genera page faults.

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		<input type="checkbox"/>							
	B		<input type="checkbox"/>							
	C	X	<input type="checkbox"/>							
mode switch			<input type="checkbox"/>							
kernel mode	sched./disp.		<input type="checkbox"/>							
	system call		<input type="checkbox"/>							
	interrupt handler per		<input type="checkbox"/>							

Fai copia e incolla nella form **dello script, del comando, per eseguirlo e del suo output**

Paragraph

*Fai copia e incolla nella form **dello script, del comando, per eseguirlo e del suo output***

Path: p » em

Question 7

Not yet answered

Marked out of 6.00

 Flag question

 Edit question

Debugging

Si consideri il file [fattoriale.c](#) che contiene l'implementazione di del fattoriale nelle versioni iterativa, `fact_iter()`, e ricorsiva, `fact_rec()`.

Si compili con il comando: `gcc -Wall -g fattoriale.c -o fattoriale`

Si consideri l'esecuzione di `./fattoriale` con parametro `10`. Eseguire le seguenti attività nel debugger.

1. imporre un breakpoint all'inizio di `fact_iter()`
2. eseguire con parametro 10
3. imporre un watchpoint che blocchi l'esecuzione all'interno di `fact_iter()` non appena `r>100`
4. continuare l'esecuzione
5. quando il watchpoint ferma l'esecuzione, mostrare il valore di `r` e di `k`
6. mostrare lo stack
7. imporre un breakpoint sul "return" di `fact_rec()` (riga 13)
8. condizionare il breakpoint appena inserito in modo che si fermi solo se `r>100`. Il comando da usare è "condition" consultare "help condition" per i dettagli.
9. continuare l'esecuzione
10. quando il breakpoint ferma l'esecuzione mostrare lo stack e i valori di `n` ed `r`
11. spostarsi sul frame #3 e mostrare i valori di `n` ed `r`

Fai copia e incolla nella form **dell'intera sessione di debugging (errori compresi!)**

Paragraph

*Fai copia e incolla nella form **dell'intera sessione di debugging (errori compresi!)***

Path: p » em

Next

 [Moodle Docs for this page](#)

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

S0dispari20170626