

Compito dispari - turno 1

☐ persi

☐ disponibili

Parte degradata:

☐ tutto

☐ solo in parte

☐ niente

☐ non applicabile

Parte impegnata nel rebuilding

☐ tutto

☐ tutto ciò che è degradato

☐ parte di ciò che è degradato

☐ niente

☐ non applicabile

Situazione B. Supponi che sia il primo che il secondo disco del terzo array siano in fault, cosa poi dire circa la condizione dell'array?

Dati:

☐ persi

☐ disponibili

Parte degradata:

☐ tutto

☐ solo in parte

☐ niente

☐ non applicabile

Parte impegnata nel rebuilding

☐ tutto

☐ tutto ciò che è degradato

☐ parte di ciò che è degradato

☐ niente

☐ non applicabile

Questa configurazione, nel caso più favorevole, può sopportare la rottura di quanti dischi ?

Spiega brevemente:

Considera la situazione A. Supponi che la configurazione fosse dotata di 1 hot-spare, cosa poi dire circa la condizione dell'array?

Dati:

☐ persi

☐ disponibili

Parte degradata:

☐ tutto

☐ solo in parte

☐ niente

☐ non applicabile

Parte impegnata nel rebuilding

☐ tutto

☐ tutto ciò che è degradato

☐ parte di ciò che è degradato

☐ niente

☐ non applicabile

Spiega brevemente

Considera la situazione B. Supponi che la configurazione fosse dotata di 2 hot-spare e che i fault siano molto vicini nel tempo, cosa poi dire circa la condizione dell'array complessivo?

Dati:

☐ persi

☐ disponibili

Parte degradata:

☐ tutto

☐ solo in parte

☐ niente

☐ non applicabile

Parte impegnata nel rebuilding

☐ tutto

☐ tutto ciò che è degradato

☐ parte di ciò che è degradato

☐ niente

☐ non applicabile

Spiega brevemente

Question 4

Not yet answered
Not graded

Flag question
Edit question

Scheduling delle attività in kernel/user mode

In un sistema sono presenti tre processi: **A, B, C**. La politica di scheduling è **round robin** con quanto **q=30ms**. inizialmente **A** è running, **B** è ready e **C** è in blocco.

- A** è cpu-bound e genera un page fault ogni 10ms di esecuzione, ciascuno servito in 40ms.
- B** è cpu-bound, nessun page fault.
- C** è inizialmente in blocco, per una lettura che verrà servita a t=20ms, poi continuerà come 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.

	A	X								
--	---	---	--	--	--	--	--	--	--	--



Compito dispari - turno 1

user mode	B		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	C		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
mode switch			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
kernel mode	sched./disp.		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	system call		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	interrupt handler per page fault		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	interrupt handler per I/O		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	interrupt handler per quanto scaduto		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
stati processi	running	A	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	ready	B	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	block	C	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
note tempi		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
altre note		<input type="text"/>							

Question 5

Not yet answered
Not graded

Flag question
Edit question

Espressioni Regolari (grep/egrep)

Considera il file di testo di esempio indirizzi.txt. Assumi che in tale file certe righe contengano esclusivamente un indirizzo di posta elettronica nella forma account@dominio e che possano essere identificate dal fatto che contengano il carattere '@'. Fornire una linea di comando basata su grep o egrep per stampare tutte le righe che contengono indirizzi di posta elettronica tali che tutte le seguenti regole siano rispettate

- il dominio finisca con la stringa .org
- l'account sia lungo 3 caratteri
- il primo carattere dell'account sia una lettera compresa tra 'a' e 'd'
- il secondo carattere dell'account sia un numero compreso tra '1' e '4'

Sul file d'esempio il comando deve stampare:

a3z@example.org
d10@example.org

Fai copia e incolla nella form del comando e del suo output

Paragraph

Fai copia e incolla nella form del comando e del suo output

Path: p » em

Question 6

Not yet answered

Si consideri il file passwords.txt in cui ogni riga segue il seguente schema:

Compito dispari - turno 1

Not graded

Flag question
Edit question

User: *:UID:GID:USER_DESCRIPTION:Home_Directory:Shell

Si scriva uno script (o riga di comando) basato su awk che elabori tale file per produrre un output come nell'esempio in fondo. Per produrre tale output si seguano le seguenti regole

- solo le righe da 2 a 6 del file passwords.txt sono considerate
- solo i campi User, UID, GID, Home_Directory e Shell sono considerati
- usa printf per l'output formattato (consulta man awk se necessario) e scegli tu la lunghezza dei campi

Le statistiche in fondo devono mostrare

- per i ogni GID quanti user che usano la shell "/usr/bin/false" hanno quel GID.

Ecco l'output che ci si aspetta.

User	UID	GID	Home Directory	Shell
root	0	0	/var/root	/bin/sh
daemon	1	1	/var/root	/usr/bin/false
_uucp	4	4	/var/spool/uucp	/usr/sbin/uucico
_taskgated	13	13	/var/empty	/usr/bin/false
_networkd	24	13	/var/networkd	/usr/bin/false

GID: 13 with 2 users

GID: 1 with 1 users

Fai copia e incolla nella form **del comando e del suo output**

Paragraph

Fai copia e incolla nella form **del comando e del suo output**

Path: p » em

Question 7

Not yet answered

Not graded

Flag question
Edit question

Debugging

Si consideri il file [fattoriale.c](#) che contiene due funzioni che implementano il fattoriale di un numero sia nella versione ricorsiva che nella versione iterativa.

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

Si esegua quindi: `./fattoriale 6`

ottenendo:

`[6] ric: 720 iter: 1440`

la versione iterativa non ritorna il risultato corretto poiché nella riga 12 di `fattoriale.c` la variabile `r` è inizializzata al valore 2. Far partire il debugging con `gdb fattoriale` ed eseguire quanto segue

- 1) imporre un breakpoint sulla riga 14
- 2) imporre un breakpoint sulla riga 16
- 3) imporre un breakpoint sulla riga 26
- 4) eseguire con l'input 6
- 5) stampare il valore della variabile `n`
- 6) continuare con l'esecuzione
- 7) stampare il valore della variabile `r`
- 8) continuare con l'esecuzione
- 9) stampare di nuovo il valore della variabile `r`
- 10) avanzare di uno step
- 11) avanzare di un altro step
- 12) stampare il valore della variabile `k`
- 13) continuare con l'esecuzione
- 14) mostrare breakpoint attivi
- 15) togliere il breakpoint sulla riga 14 indicando proprio il numero di riga del file `fattoriale.c`

- 16) continuare con l'esecuzione
- 17) mostrare informazioni sullo stack
- 18) mostrare breakpoint attivi
- 19) eliminare il breakpoint sulla riga 16 tramite il num. indicato dalle info dei breakpoint attivi
- 20) mostrare breakpoint attivi
- 21) continuare con l'esecuzione
- 22) rimuovere l'ultimo breakpoint attivo tramite il num. indicato dalle info dei breakpoint attivi
- 23) mostrare breakpoint attivi
- 24) uscire da gdb

*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)

SOpari20170131