



## Compito pari - 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=40ms**. inizialmente **A** è running, **B** è in blocco e **C** ready.

- A** è I/O-bound, ciascun I/O servito in 100ms, nessun page fault.
- B** è inizialmente in blocco a causa di un page fault che verrà servito a  $t=20ms$ , poi prosegue cpu-bound e con page faults ogni 30ms di esecuzione.
- C** è cpu-bound.

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								
--	---	---	--	--	--	--	--	--	--	--

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

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 un recapito telefonico. Tali righe sono così composte: <numero><interno> dove numero ha il formato XXX-XXX-XXXX con X una cifra e interno è una qualsiasi sequenza di caratteri. Fornire una linea di comando basata su grep o egrep per stampare tutte le righe che contengono recapiti telefonici tali che tutte le seguenti regole siano rispettate:

- tutte le cifre del numero siano diverse da 9
- l'interno non deve contenere il carattere 'c'
- l'interno finisce per una stringa di 5 caratteri di cui il primo carattere è la lettera x mentre gli ultimi 4 caratteri sono cifre comprese tra 0 e 9

Usando il file indirizzi.txt il comando deve stampare:  
714-555-5350 x2134

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:  
User:\*:UID:GID:USER\_DESCRIPTION:Home\_Directory:Shell

Not graded

Flag question  
Edit question

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 prime 12 righe del file passwords.txt sono considerate
- solo i campi 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 ogni GID quanti user hanno quel GID
- solo i GID che hanno per lo meno 2 utenti devono essere stampati.

Ecco l'output che ci si aspetta.

User	UID	GID	Home Directory	Shell
nobody	-2	-2	/var/empty	/usr/bin/false
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
_lp	26	13	/var/spool/cups	/usr/bin/false
_scsd	31	31	/var/empty	/usr/bin/false
_ces	32	54	/var/empty	/usr/bin/false
_mcxalr	54	54	/var/empty	/usr/bin/false
_appleevents	55	55	/var/empty	/usr/bin/false
_geod	56	56	/var/db/geod	/usr/bin/false
GID: 13 with 3 users				
GID: 54 with 2 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					
<p><i>Fai copia e incolla nella form <b>dell'intera sessione di debugging (errori compresi!)</b></i></p>						
Path: p » em						

Next

 [Moodle Docs for this page](#)

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

SOpari20170131