

Preview Compito pari - turno 1

1 

Dati studente

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

Cognome

Nome

Matricola

2 

Memory management

Considera un sistema con page buffering e considera il problema del trattamento di una pagina X appartenente ad un processo P. Supponi che X venga modificata da P e poi venga tolta dal suo resident set.

1. Possiamo riassegnare subito il frame di X al processo P? spiega.
2. Possiamo riassegnare subito il frame di X ad un processo diverso da P? spiega.
3. Descrivi una politica di cleaning che pensi possa essere conveniente e motiva la risposta.

Answer:

3 

I/O

Descrivi l'algoritmo di disk scheduling **cyclic-elevator** e metti in evidenza i vantaggi rispetto all'algoritmo elevator semplice?

Answer:

Scheduling delle attivita' all'interno del SO

Considera un sistema con architettura del kernel "execution within user process". In tale 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 60ms.

- A è cpu bound: genera un page fault ogni 20 ms e ciascun page fault è servito in 30ms.
- B è cpu bound: nessun page fault.
- C è cpu bound: genera page faults ogni 300ms.

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	disptatching																		
	system call per I/O																		
	interrupt handler per page fault																		
	interrupt handler per I/O																		
	interrupt handler per quanto scaduto																		
stati processi	running	A																	
	ready	B C																	
	block																		
note tempi																			
altre note																			

Pratica Unix

Considera la directory /usr/include che contiene gli header delle librerie C. Il comando

```
find directory -name "pattern"
```

permette di cercare file il cui nome sia conforme al pattern specificato.


- Mostra un comando che dia la lista dei file che iniziano per la lettera "s" contenuti in un punto qualsiasi del sottoalbero /usr/include (consulta la documentazione

Answer:

Scripting

Il file `router_configuration.txt` contiene il dump di una configurazione di un router. Il file si compone di vari blocchi (pensali come record) separati da due o più linee vuote.

Per svolgere l'esercizio non è necessario conoscere il significato di tutti i campi. Suggerimenti: alcune volte, ma non sempre, conviene processare tale file con awk usando `RS=""` (stringa vuota) e `FS="\n"`; ricorda che, in awk, `gsub()` è un efficace strumento di sostituzione.


- 6  Una parte del file contiene la specifica di rotte, inserite a mano dall'amministratore, dette *rotte statiche*. Tali rotte statiche sono esplicitate nelle righe del file di configurazione della forma

```
ip route <indirizzo-IP-destinazione> <netmask> <indirizzo-interfaccia-router>
```

Seleziona le righe del file di configurazione che iniziano con "ip route" e in cui l'indirizzo-IP-destinazione ha il terzo byte con un valore compreso tra 7 e 11.

esempio: `ip route 20.30.10.2 255.255.255.0 35.1.1.1`

Answer:

- 7  Una parte del file contiene la configurazione delle interfacce del router, su più righe, che inizia con

```
interface <nome-interfaccia>
```

Le righe seguenti nel file (fino alla linea vuota) contengono alcune informazioni sulla configurazione dell'interfaccia specificata.

```
speed 1000M
media-type rj45
negotiation auto
```

in tabella apparirà
GigabitEthernet0/0 rj45

Answer:

8 

Hai di fronte a te un file list.h contenente esclusivamente il seguente codice.

```
struct element
{
    struct element* next;
    double num;
};

struct list
{
    struct element* first;
    struct element* last;
};
```

List.h viene incluso da vari altri .h nel progetto. Pensi che questo sia un corretto stile di impostare il progetto? Discuti quello che può succedere lasciando le cose come stanno e suggerisci le opportune modifiche .

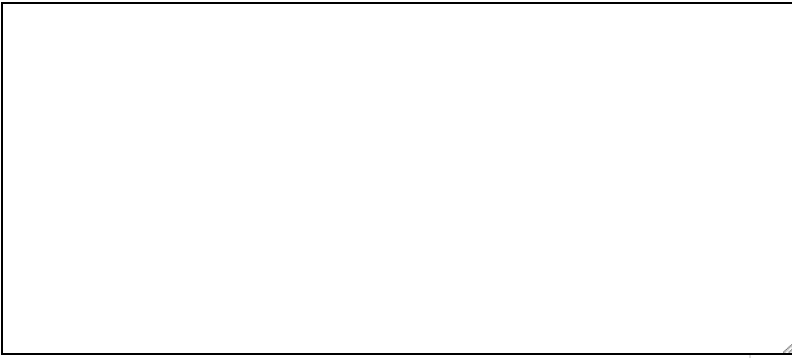
Answer:

9 

Considera il codice del seguente progetto [prj.tar.gz](#). Compila tutti i file con il comando

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

Considera una esecuzione di fib con parametro **21**. Considera il 300-esima volta che la funzione **fib()** sta per ritornare. Quale è il valore che fib ritorna? Sempre in



 [Moodle Docs for this page](#)

You are logged in as [Admin User](#) ([Logout](#))

SO_2010.02.03