

Preview Compito dispari - turno 1

1

Dati studente

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

Cognome

Nome

Matricola

2

Memory management

Con riferimento ad una architettura con inverted page table. Rispondi alle seguenti domande

1. cosa succede quando si deve tradurre un indirizzo di memoria virtuale in uno fisico?
2. cosa accade quando una nuova pagina viene aggiunta al resident set di un certo processo?

Answer:

1. La CPU esegue le seguenti operazioni. A partire dal pid e dal numero di pagina si calcola un hash f , se la entry f della IPT contiene pagina e PID corretti f è il numero frame cercato. Altrimenti, si segue il campo chain, finché si trova una entry con pagina e PID corretti (nel qual caso l'indice della entry è il numero di frame cercato). Se tale entry non esiste nella chain allora la pagina o non è residente o non è lecitamente accessibile.

2. Il SO esegue le seguenti operazioni. A partire dal PID e dal page number si calcola un hash f . Se la entry f è libera la si assegna alla pagina (e la si inizializza) altrimenti si usa il campo chain per redirigere la ricerca su un'altra entry (e quindi su un altro frame). Questo può essere fatto in vari modi. Il modo più semplice è di scegliere arbitrariamente un nuovo frame (e una nuova entry) libero e di inserirlo in cima alla lista puntata dal chain di f .

3

I/O

Calcola il Mean Time Between Failures di un array raid 0 composto da N dischi identici.

Puoi eseguire lo stesso calcolo per raid 5? Spiega.

Answer:

Per il calcolo vedi materiale didattico.

Per raid 5: il calcolo non ha senso poiché raid 5 è tollerante al primo guasto. Per fare il calcolo bisogna definire un modello in cui si esplicita cosa succede dopo il primo guasto (ad esempio fix immediato, fix entro un tempo t con una certa probabilità, ecc.)

4

Scheduling delle attività 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 30ms.

- **A** è cpu bound: nessun page fault.
- **B** è cpu bound: page fault ogni 10ms servito in 20ms.
- **C** è I/O bound: cpu burst trascurabili, I/O servito in 100ms, 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														X				X
	B						X													
	C										X									
mode switch			X			X		X			X		X			X		X		X
kernel mode	dispatching				X					X					X					
	system call per I/O													X						
	interrupt handler per page fault							X												
	interrupt handler per I/O																	X		
	interrupt handler per quanto scaduto			X																
stati processi	running	A	A	A	B	B	B	B	B	C	C	C	C	C	A	A	A	A	A	A
	ready	B C	B C	B C	C A	C A	C A	C A	C A	A	A	A	A	A					B	B
	block									B	B	B	B	B	B	B	B	B	C	C
note tempi																				
altre note																				

5

Pratica Unix

Esegui le seguenti operazioni su una shell unix e fai copia di ciò che vedi sul terminale (comandi e output).


1. esegui un process grafico (per esempio gedit o xeyes) in foreground
2. stoppalo
3. mostra che il processo ancora esiste
4. fallo ripartire in background
5. uccidilo dalla shell
6. mostra che il processo non esiste più

```
utente@campus3:~$ xeyes
( ctrl-Z )
[1]+  Stopped xeyes
utente@campus3:~$ ps
PID TTY TIME CMD
6471 pts/0 00:00:00 bash
6477 pts/0 00:00:00 xeyes
6492 pts/0 00:00:00 ps
utente@campus3:~$ bg xeyes
[1]+  xeyes &
utente@campus3:~$ kill 6477
utente@campus3:~$ ps
PID TTY TIME CMD
6471 pts/0 00:00:00 bash
6496 pts/0 00:00:00 ps
[1]+  Terminato xeyes
```

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() e' 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 che iniziano con `ip route` in cui il terzo byte dell'indirizzo-interfaccia-router e' minore di o uguale a 3 o maggiore o uguale a 10 (cioè non sia 3<valoreTerzoByte<10).


esempio:

```
ip route 20.30.3.2 255.255.255.0 35.1.2.1
ip route 20.30.3.2 255.255.255.0 35.1.100.1
```

Answer:

```
cat router_configuration.txt | egrep "^ip route .\\.([0-3]([0-9]*[1-9]0))\\.([0-9]*$)"
oppure
cat router_configuration.txt | grep "^ip route" | egrep -v "\.[4-9]\.[0-9]*$"

ip route 10.0.2.0 255.255.255.0 192.168.2.2
ip route 10.0.3.0 255.255.127.0 192.168.10.2
ip route 10.0.4.0 255.255.0.0 192.168.2.2
ip route 10.0.5.0 255.255.255.0 192.168.3.2
ip route 10.0.6.0 255.255.248.0 192.168.2.2
ip route 10.0.7.0 255.255.240.0 192.168.2.2
ip route 10.0.8.0 255.255.0.0 192.168.2.2
ip route 10.6.9.0 255.248.0.0 192.168.2.2
ip route 10.1.1.0 255.255.0.0 192.168.10.2
ip route 10.1.2.0 255.255.0.0 192.168.3.2
ip route 10.1.3.0 255.255.255.0 192.168.3.2
ip route 10.1.4.0 255.255.0.0 192.168.3.2
ip route 10.1.5.0 255.255.255.0 192.168.3.2
ip route 10.1.6.0 255.255.0.0 192.168.3.2
ip route 10.1.7.0 255.255.127.0 192.168.3.2
ip route 10.1.8.0 255.255.0.0 192.168.3.2
ip route 10.1.9.0 255.255.0.0 192.168.2.2
ip route 10.1.10.0 255.255.255.0 192.168.2.2
```

- 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.

Mostra un comando che produca una tabella con una riga per ciascuna interfaccia **per cui esiste il campo duplex**, in cui il primo campo sia il nome dell'interfaccia e il


```
media-type rj45
negotiation auto
```

in tabella apparirà
GigabitEthernet0/0 half

Answer:

```
cat router_configuration.txt | awk -v FS="\n" -v RS="" '/duplex/ {print $0 ;
print"" }' | egrep '^interface|^duplex|^$)' | awk -v FS="\n" -v RS="" '{print $1, $
2}' | awk '{gsub(/interface /, ""); gsub(/duplex /, ""); print $0}'
```

```
FastEthernet0/1 half,full
GigabitEthernet0/0 half
GigabitEthernet0/1 full
GigabitEthernet0/2 auto
GigabitEthernet0/3 half,full
GigabitEthernet0/4 auto
```

8  Stai compilando un codice C non tuo. Ti imbatti in un errore di questo tipo

```
init_list.c:(.text+0x5f): undefined reference to
`XtAppAddActionHook'
```

In che fase si genera questo errore?

In realtà nel tuo sistema c'è tutto il necessario per una compilazione corretta.
Descrivi le operazioni che puoi compiere per ricercare la libreria giusta nel tuo sistema.

Answer:

l'errore è generato nella fase di link. Bisogna specificare
-l<libreria> al compilatore. La libreria va cercata, o nella
documentazione della funzione XtAppAddActionHook() o
direttamente nelle librerie con un comando del tipo

```
objdump -T *.so | less --pattern=XtAppAddActionHook
```

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
quell'istante, che valore ha l'ultimo elemento della lista L in main()?

Answer:

 [Moodle Docs for this page](#)

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

tmp

```
utente@campus22:~/Scrivania/prj$ gdb fib
GNU gdb 6.8-debian
Copyright (C) 2008 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law. Type "show copying"
and "show warranty" for details.
This GDB was configured as "x86_64-linux-gnu"...
(gdb) b fib
Breakpoint 1 at 0x400781: file init_list.c, line 8.
(gdb) r 21
Starting program: /home/utente/Scrivania/prj/fib 21

Breakpoint 1, fib (f1=1) at init_list.c:8
8 if ( f1==1 || f1==2 )
(gdb) l
3 #include "list.h"
4 #include <math.h>
5 long int fib( long int f1 )
6 {
7 long int f;
8 if ( f1==1 || f1==2 )
9 f = 1;
10 else
11 f = fib(f1-1) + fib(f1-2);
12 return f;
(gdb) b 12
Breakpoint 2 at 0x4007be: file init_list.c, line 12.
(gdb) ig 2 299
Will ignore next 299 crossings of breakpoint 2.
(gdb) dis 1
(gdb) r 21
The program being debugged has been started already.
Start it from the beginning? (y or n) y

Starting program: /home/utente/Scrivania/prj/fib 21

Breakpoint 2, fib (f1=5) at init_list.c:12
12 return f;
(gdb) info b
Num Type Disp Enb Address What
1 breakpoint keep n 0x000000000400781 in fib at init_list.c:8
2 breakpoint keep y 0x0000000004007be in fib at init_list.c:12
breakpoint already hit 300 times
(gdb) p f
$1 = 5
(gdb) bt
#0 fib (f1=5) at init_list.c:12
#1 0x0000000004007b6 in fib (f1=7) at init_list.c:11
#2 0x0000000004007a6 in fib (f1=8) at init_list.c:11
#3 0x0000000004007a6 in fib (f1=9) at init_list.c:11
#4 0x0000000004007a6 in fib (f1=10) at init_list.c:11
#5 0x0000000004007a6 in fib (f1=11) at init_list.c:11
#6 0x0000000004007fb in init_list (L=0x601010, n=21) at init_list.c:20
#7 0x00000000040089d in main (argc=2, argv=0x7ffea9551c8) at main.c:21
(gdb) up 7
#7 0x00000000040089d in main (argc=2, argv=0x7ffea9551c8) at main.c:21
21 init_list(L, n);
(gdb) p L.last
$2 = (struct element *) 0x601150
(gdb) p *L.last
$3 = {next = 0x0, num = 10170.28645158315}
```