

YYY02 Esercizi per Teoria

Esercizi per preparazione alla prova scritta

Esercizio YYY02_02 - Busy Waiting

In linguaggio ANSI C, siano dichiarate le seguenti variabili, e siano queste variabili debitamente inizializzate:

```
pthread_mutex_t    mutex;  
pthread_cond_var   cond;
```

considerare poi la seguente porzione di codice, in cui condizione sia una espressione da valutare:

```
while ( condizione ) {  
    pthread_cond_wait( &cond, &mutex );  
}  
fai_qualcosa();
```

La porzione di codice, sopra riportata, effettua busy-waiting ?

Esercizio YYYY02_02

Una stessa struttura dati ha la stessa dimensione in tutti i processori? Perché ?

Esercizio YYYY02_03

In un programma in C scrivo, in una funzione dichiaro un vettore di elementi, in cui gli elementi sono un tipo di struttura dati struct A. Guardando come viene collocato in memoria quel vettore, può accadere che tra un elemento ed il successivo, dello stesso vettore, ci siano dei byte non occupati dalla struttura dati di tipo struct A? Perché?

Esercizio YYYY02_04

In linguaggio C, se un programma è costituito da più moduli, è possibile che una stessa funzione (cioè una funzione con stesso nome, stessi tipi di argomenti e stesso tipo di dato restituito) sia implementata in due diversi moduli senza che questo provochi un errore nella generazione del programma eseguibile? Perché ?

Esercizio YYYY02_05

La seguente istruzione si trova nella funzione main di un modulo scritto in C.

```
ris = PRODOTTO ( 3 - 4 , SOMMA ( 2 , 7 ) );
```

Precedentemente, nello stesso file, le due macro sono state così definite:

```
#define PRODOTTO( X, Y ) ( (X)*(Y) )
```

```
#define SOMMA( X , Y )      ( (X)+(Y) )
```

Se quel modulo viene processato dal **preprocessore** del compilatore gcc, in che modo viene tradotta quella istruzione dal preprocessore ?

Esercizio YYY02_06

Supponiamo di inserire le seguente due istruzioni, in un modulo scritto in C, in una zona fuori da tutte le funzioni del modulo;

```
#define NUM 10  
int vet [ 10 ];
```

Se quel modulo viene processato dal **preprocessore** del compilatore gcc, in che modo viene tradotta quella istruzione dal preprocessore ?

Esercizio YYY02_07

Spiegare cosa si intende con Race Condition.

Esercizio YYY02_08

Spiegare cosa si intende quando si dice che una sequenza di istruzioni viene eseguita in maniera "atomica".

Esercizio YYY02_09

Quando un processo esegue una fork, le sue pagine di memoria vengono immediatamente duplicate Un Task puo' avere piu' stack a sua disposizione? Perche' si o perche' no ?

Esercizio YYY02_10

Spiegare come dovrebbe funzionare l' "algoritmo ottimale" di sostituzione delle pagine.

Esercizio YYYY02_11

Spiegare come funziona l' algoritmo "Least Recently Used (LRU)" di sostituzione delle pagine.

Esercizio YYYY02_12

Consideriamo una finestra del working set di ampiezza 6,
e consideriamo un un programma che accede alle proprie pagine in questa sequenza,
3 5 7 2 8 5 6 5 4 5 6 7 1 5 6 4.

Supponiamo ora che sia stata appena acceduta l'ultima pagina della sequenza indicata.
Indicare qual'e' il working set.

Esercizio YYYY02_13

Che cos'e' il thrashing ?

Esercizio YYYY02_14

Nell'ambito della paginazione per la memoria virtuale, cosa si intende con "allocazione dei frame ai processi".

Esercizio YYYY02_15

Spiegare a cosa serve il procedimento di sostituzione delle pagine nell'ambito della gestione della memoria virtuale.

Esercizio YYYY02_16

In che contesto si parla di "dirty bit" e a che cosa serve ?

Esercizio YYY02_17

Spiegare cosa avviene allo scatenarsi di un page fault.

Esercizio YYY02_18

Spiegare cosa sono i processi "I/O bound" e quelli "CPU bound".

Esercizio YYY02_19

Spiegare la differenza tra "scheduling preemptive" e "scheduling non preemptive".

Esercizio YYY02_20

L'algoritmo di scheduling FIFO soffre di starvation ? Perché ?

Esercizio YYY02_21

Cos'è la formattazione fisica o formattazione di basso livello?

Esercizio YYY02_22

Spiegare come avviene il bootstrap nei sistemi moderni.

Esercizio YYY02_23

A cosa serve il programma **fdisk** ?

Esercizio YYY02_24

A cosa serve il programma **ln** ?

Esercizio YYYY02_25

Devo implementare una delle politiche di scheduling del disco magnetico descritte a lezione, ma ancora non so quale.

Che informazioni mi servono per potere implementare una qualunque di quelle politiche?

Esercizio YYYY02_26

Perche' non parliamo di politiche di scheduling del disco quando abbiamo a che fare con dischi a stato solido?

Esercizio YYYY02_27

Ipotizziamo che i cilindri di un disco magnetico siano indicizzati in maniera crescente dalla periferia verso il centro.

Supponiamo che la testina di lettura di un disco si trovi sul cilindro di indice 10.

Ipotizziamo che la testina, per arrivare alla posizione attuale, si sia spostata verso il centro del disco.

Supponiamo che l'elenco delle richieste pendenti sia la seguente, ordinata in ordine di tempo di arrivo delle richieste.

12, 15, 6, 3, 6, 11, 13, 17, 20, 1.

Se uso un algoritmo di scheduling Shortest Seek Time First (SSTF), quale sara' la sequenza temporale dei prossimi accessi a disco?

Esercizio YYYY01_28

Nelle stesse ipotesi dell'esercizio precedente, quale sara' la sequenza temporale dei prossimi accessi a disco se invece uso una politica di scheduling del disco di tipo C-LOOK?

Esercizio YYYY02_29

Spiegare la differenza tra Hard link e Soft link in ambito Linux.

Esercizio YYYY02_30

Cos'e' un inode ?

Esercizio YYYY02_31

Che struttura si utilizza, in ambiente Linux, per memorizzare i blocchi di un file?