

Matricola: _____

Nome: _____

Cognome: _____

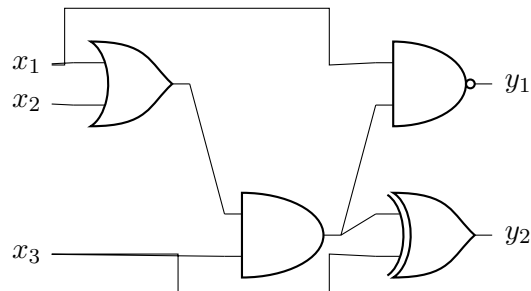
ESAME - Programmazione e Architetture (Modulo B)

25 Maggio 2021 (Simulazione)

L'esame consiste di 15 domande a risposta multipla sugli argomenti del corso. Affinché una risposta sia considerata valida la scelta *deve essere motivata*.

Domanda 1

Si consideri il seguente circuito:



Supponendo che gli input siano $x_1 = 0$, $x_2 = 1$ e $x_3 = 1$, i valori degli output y_1 e y_2 saranno:

$y_1 = 0, y_2 = 0$

$y_1 = 0, y_2 = 1$

$y_1 = 1, y_2 = 0$

$y_1 = 1, y_2 = 1$

Domanda 2

Si consideri il seguente valore in esadecimale: 5F. Convertito in base 10 il numero è:

95

65

-32

90

Domanda 3

Supponendo di avere una ALU che può svolgere 8 diverse operazioni, di quanti bit potrebbe essere composto l'opcode che ci permette di selezionare che operazione eseguire?

2

La ALU non necessita di opcode

Almeno 3

Non meno di 8

Domanda 4

Ricordando che ARM-v7a ha 13 registri di 32 bit R0, ..., R12, e supponendo che il registro R1 contenga il valore 3 e che il registro R2 contenga il valore 2, quale è il valore contenuto nel registro R0 al termine dell'esecuzione del seguente programma?

```
ciclo    MOV          R0, #0
         CMP          R2, #0
         BEQ          fine
         ADD          R0, R0, R1
         SUB          R2, R2, #1
         B           ciclo
```

fine

Si ricorda che CMP effettua una comparazione tra due valori e che BEQ effettua un salto condizionale quando l'ultima comparazione era tra valori uguali.

0

5

Il programma non termina

6

Domanda 5

Quale approccio è possibile per segnalare alla CPU il completamento di una operazione di I/O senza che la CPU debba continuamente controllare lo stato dell'operazione?

interrupt

polling

operazioni di load e store

È possibile solo col port-mapped I/O

Domanda 6

Si considerino tre processi P_1 , P_2 e P_2 in un sistema con scheduling di tipo Shortest Job First in cui tutti i processi arrivano nello stesso momento. Supponendo che P_1 abbia un tempo di esecuzione di 500ms, P_2 di 600ms e P_3 di 200ms. Dopo quanto tempo termineranno i processi P_1 , P_2 e P_3 ?

P_1 : 500ms, P_2 : 1100ms, P_3 : 1300ms

P_1 : 700ms, P_2 : 200ms, P_3 : 1300ms

P_1 : 500ms, P_2 : 200ms, P_3 : 600ms

P_1 : 1300ms, P_2 : 800ms, P_3 : 600ms

Domanda 7

Per tenere traccia dei blocchi occupati dal contenuto dei file il filesystem unix "classico" (i.e., come quello visto a lezione) utilizza:

- Solamente i blocchi puntati dagli inode Una data bitmap in cui ogni bit rappresenta un blocco occupato
- Non serve che ne tenga traccia il filesystem,
 è il dispositivo a blocchi che gestisce lo spazio libero un file salvato nel superblocco
-
-
-

Domanda 8

Si consideri un sistema in cui la memoria libera è divisa nei seguenti intervalli:

- (a) da 100 a 200
- (b) da 400 a 450
- (c) da 670 a 700
- (d) da 810 a 890

Supponendo che un processo richieda 80 indirizzi contigui, secondo le policy "first fit" e "best fit" dove avverrebbero le allocazioni?

- first fit: (a), best fit: (d) first fit: (d), best fit: (d)
- first fit: (a), best fit: (a) first fit: (b), best fit: (c)
-
-
-

Domanda 9

Si consideri il seguente comando per la shell Unix:

```
cat foo.txt | sort | head > bar.txt
```

Supponendo che il file `foo.txt` esista e non vi siano problemi di permessi, il file `bar.txt` al termine dell'esecuzione del comando conterrà:

- Le prime 10 righe del file `foo.txt`
 - Le ultime 10 righe del file `foo.txt`
 - Le prime 10 righe in ordine lessicografico del contenuto del file `foo.txt`
 - Le righe del file `foo.txt` private dei duplicati
-
-
-

Domanda 10

Si consideri il seguente codice:

```
#include <stdio.h>
#include <unistd.h>

int main(int argc, char * argv[])
{
    pid_t pid = fork();
    while (pid != 0) {
        pid = fork();
    }
    return 0;
}
```

Quale delle seguenti affermazioni è vera?

- Vengono creati due processi
 - Vengono creati tre processi
 - Nessun processo terminerà
 - Il programma continuerà a creare nuovi processi
-
-
-

Domanda 11

Si supponga che il processo P_1 invii un segnale `SIGUSR1` al processo P_2 . Allora:

- | | |
|---|---|
| <input type="checkbox"/> P_2 potrà leggere il segnale con <code>read</code> | <input type="checkbox"/> Verrà invocato il gestore di segnale relativo a SIGUSR1 in P_1 |
| <input type="checkbox"/> Verrà chiamata la funzione <code>void * SIGUSR1 (void * arg)</code> in un nuovo thread | <input type="checkbox"/> Verrà invocato il gestore di segnale relativo a SIGUSR1 in P_2 |
-
-
-

Domanda 12

Si supponga di avere tre thread t_1 , t_2 e t_3 e tre lock L_1 , L_2 e L_3 . L'ordine di acquisizione (e rilascio) dei lock da parte dei thread è il seguente:

- t_1 : L_1
- t_2 : L_1, L_2
- t_3 : L_2, L_3

Quale delle seguenti affermazioni è vera?

- | | |
|---|---|
| <input type="checkbox"/> Non si può verificare un deadlock | <input type="checkbox"/> Si può verificare un deadlock dovuto all'ordine di acquisizione dei lock |
| <input type="checkbox"/> Per non avere un deadlock t_1 dovrebbe acquisire prima L_3 | <input type="checkbox"/> Per non avere un deadlock t_3 dovrebbe acquisire L_1 dopo L_3 |
-
-
-

Domanda 13

Si consideri il seguente frammento di codice:

```
pthread_mutex_t mylock = PTHREAD_MUTEX_INITIALIZER;  
  
void * f(void * arg)  
{  
    pthread_lock(&mylock);  
    for (int i = 1; i < 3; i++) {  
        printf("%d_", i);  
    }  
}
```

```

    pthread_unlock(&mylock);
    return NULL;
}

int main(int argc, char * argv[])
{
    pthread_t p1;
    pthread_t p2;
    pthread_create(&p1, NULL, f, NULL);
    pthread_create(&p2, NULL, f, NULL);
    return 0;
}

```

Quale dei seguenti è un possibile output?

1 1 2 2

1 2 1 2

1 2 2 1

Vi è un possibile deadlock

Domanda 14

Si consideri il seguente frammento di codice, supponendo esistano il lock `mylock` e la variabile globale `x` e che sia creato un pthread che esegue `g`:

```

void * g(void * arg)
{
    pthread_lock(&mylock);
    x = x + 1;
    return NULL;
}

```

Quale delle seguenti affermazioni è vera?

- Avremmo dovuto proteggere `x` con una variabile condizione
 - Non ci sono situazioni nelle quali sarebbe necessario proteggere `x` con un lock
 - `g` non ha la signature corretta per essere passata come argomento a `pthread_create`
 - Non rilasciamo il lock acquisito
-
-
-

Domanda 15

Quali dei seguenti protocolli a livello di trasporto utilizzeremo se siamo interessati solo a multiplexing e non ad avere un canale affidabile?

- TCP
 - HTTPS
 - UDP
 - IP
-
-
-