EXAMEN – SEMESTRE 5

EPREUVE : PROGRAMMATION SYSTÈME II

QCM

1)

f) It will sometimes print “0” and sometimes print “1”.

2)

(b) int fd = open("hoola.txt", O_RDWR);
dup2(fd, STDOUT_FILENO);
write(STDOUT_FILENO, "Hello", 5);

3) (d) Whatever is written to the file through fd, can be read using fd2

4) (d) None of the above

5) (c) User stack

6) (b) The zombie child is reaped by the init process.

Problème1

Answer: a= 5, b= 2, c= 4

Problème2

- **Parent: i = 10**
- **Enfant: i = 20**

Problème3

Part 1 :

The output is: `fd3 =` 3

Part 2 :

The output is: `c =` r

Part 1 :

The output is: `fd3 =` f

Part 1 :

The output is: `fd3 =` a

Problème4

A) Number of lines of output = **2^n**

B) Possible output sequences : **acb, abc**

C) output of this program

counter = 2
counter = 0
counter = 2

Exercice 2 :

1. Le programme exécute la commande ls.
2. Après modification du PATH, il exécutera faux ls et lancera un shell.

3. Le risque est une escalade de privilèges, un attaquant pourrait obtenir un shell root.

Exercice 3 :

1. pthread_mutex_lock assure l'exclusion mutuelle et pthread_mutex_unlock libère le mutex.

2. Le programme crée 10 threads, chacun verrouillant et déverrouillant un mutex après un délai aléatoire.

3. Code modifié :

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

pthread_mutex_t mutex = PTHREAD_MUTEX_INITIALIZER;
int x = 0;

void *routine(void *arg) {
    pthread_mutex_lock(&mutex);
    x++;
    printf("Bonjour IPNET CS-GL\n");
    pthread_mutex_unlock(&mutex);
    return NULL;
}

int main() {
    pthread_t threads[10];
    for (int i = 0; i < 10; i++) {
        pthread_create(&threads[i], NULL, routine, NULL);
    }
    for (int i = 0; i < 10; i++) {
        pthread_join(threads[i], NULL);
    }
```

```
    printf("Valeur finale de x: %d\n", x);
    return 0;
}
```

1. Ordre correct du cycle de compilation :
    - Réponse : (d) foo.c -> foo.s -> foo.o -> foo
2.
    - Réponse : (b) .static