***Université Larbi Ben M'hidi - Oum El Bouaghi -***
DEPARTEMENT DE MATHEMATIQUE ET D'INFORMATIQUE

**3ème année LMD SI** **Examen S5 Matière : SE2** **21 Janvier 2024**

---

# Corrigé type

## Exercice N° : 01 *(04 pts : 01+ 01 + 02)*

1. Lorsque le thread principal se termine, les threads secondaires s'arrêtent
2. Lorsque le processus père se termine, les processus fils continuent l'exécution
3. hello hello How are you How are you How are you How are you

## Exercice N° : 01 *(05 pts)*

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

void fonct1 () {printf ("Hello\n") ; }
void fonct2 () {compt++ ;}
void fonct3 () { printf ("Hi\n") ;
                 signal (SIGTERM, SIG_IGN);}

int compt=0 ;

void main ( ) {
  signal (SIGINT, fonct1) ;
  signal (SIGTSTP, fonct2);
  signal (SIGTERM, fonct3);

  while (1) pause();
  }
```

3 pts

1,5 pts

0.5 pt

## Exercice N° : 02 (05 pts )

Semaphores
Mutex [n] = {1, 1, ….,1},
Vide [n] = {Max, Max, …, Max},
Plein [n] = {0, 0, ., 0} ;

01 pt

**Producteur ( )**

```
{
Repeter
{
  Produir(element);
  pour i=0, n- 1, pas 1 faire
    wait(Vide[i]) ;
    wait(Mutex[i]) ;
   deposer(element);
   Signal(Mutex[i]) ;
   Signal(Plein[i]) ;
 }tant que vrai ;
}
```

02 pts

**Consommateur( int i)**

```
{
Repeter
{
    wait(Plein[i]) ;
    wait (Mutex[i]) ;
    Retirer(element);
    Signal(Mutex[i]) ;
    Signal(Vide[i]) ;
   Consommer(element)
}tant que vrai ;
}
```

02 pts

## Exercice N° : 04 *(06 pts)*

1/ L'état Ti est décrit par :

**Available** = (3, 3, 2) et **Need** = $\begin{pmatrix} 7 & 4 & 3 \\ 1 & 2 & 2 \\ 6 & 0 & 0 \\ 0 & 1 & 1 \\ 4 & 3 & 1 \end{pmatrix}$

1,5 pt

2/ <P1, P3, P4, P0, P2> est une séquence saine => l'état Ti est sain. 1,5 pt

3/
R1(1, 0, 2) ≤ Need1(1, 2, 2)
R1(1, 0, 2) ≤ Available(3, 3, 2)

1 pt

La requête R1 est accordée immédiatement parce que : P1 est le premier processus de la séquence saine.

2 pts