

TP2 : La fonction fork()

Vous pouvez programmer en C++ s'il est installé sur la machine sinon langage C.

Exercice 1: Ecrire sur votre éditeur le programme affichant ‘Salam’.

Compiler puis exécuter : combien de messages affichés ?

Exercice 2: Modifier le programme précédent comme suit :

```
#include <unistd.h> //pour fork
#include <stdio.h>
using namespace std;
int main()
{
    fork();
    printf ('salam');
    return 0;
}
```

Compiler puis exécuter : combien de messages affichés ? Expliquez.

Exercice 3: Dans le programme précédent, ajouter 2 fork() après le 1^{er} fork.

Compiler puis exécuter : combien de messages affichés ? Expliquez.

Exercice 4: Ecrire sur votre éditeur le programme suivant :

```
#include <unistd.h>
#include <stdio.h>
using namespace std;
int main()
{
    printf ('salam');
    fork();
    printf ('Bonjour');
    return 0 ;
}
```

Compiler puis exécuter et interpréter les résultats.

Exercice 5 : Ecrire sur votre éditeur le programme suivant :

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

```
using namespace std;
int main()
{
    pid_t pid ;
    pid = fork() ;
    cout << " pid = "<<pid<<endl;
    return 0 ;
}
```

Compiler puis exécuter et interpréter les résultats.

Exercice 6 :

getpid() : fonction qui retourne le pid du process appelant

getppid(): fonction qui retourne le pid du père du process appelant.

Modifier le programme précédent de la manière suivante :

- déclarer une variable i = 1.
- le processus fils ajoute 5 à i puis affiche son pid et le pid de son père
- le processus père ajoute 10 à i puis affiche son pid et celui de son fils.

Indication:

```
/*---*/
int main()
{
    int i = 1;
    pid_t pid;
    pid = fork();
    if (pid == 0)
    {
        i=i+5;
        cout << " je suis le processus fils, mon pid = "<<?????? << ", le pid de mon père =
" <<?????????<<"i = "" i<<endl;
    }
    else //pid!=0
    {
        i=i+10 ;
        ????????????
    }
    return 0 ;
}
```

Compiler puis exécuter et interpréter les résultats.