



**Université d'AIN TEMOUCHENT – Algérie**  
Faculté des Sciences et de la Technologie  
Département de Mathématiques et d'Informatique



S1 – CySIA

Matière: **Système d'exploitation Avancé**

2025-2026

**Rappels sur les Processus et les Threads**

**TD n° : 00**

**Exercice 01:** *Création d'un processus fils et exécution en parallèle avec le père*

Soit le programme suivant:

```
/* perefils.c */
#include<stdio.h>
#include<unistd.h>
int main(){
    int pid;
    if((pid=fork())==0)
    {
        printf("le fils: mon pid est %d, le pid de mon père est %d\n", getpid(), getppid());
        printf("fork a retourné au fils la valeur de %d\n", pid);
        sleep(2);
    }
    else if(pid>0)
    {
        printf("le père: mon pid est %d, le pid de mon père est %d\n", getpid(), getppid());
        printf("fork a retourné au père la valeur de %d\n", pid);
    }
    return 0;
}
```

- 1) Après exécution de ce programme, quel est le sort du père et de son fils ?
- 2) Modifier le programme de façon à faire passer le processus fils dans l'état Zombi.

**Exercice 02:** *Processus Zombi et Orphelin*

Écrire un programme qui crée deux fils, dont l'un deviendra zombi, et l'autre deviendra orphelin

**Exercice 03:** *Primitives fork, exec, wait*

Dessiner les deux arborescences de ces deux pseudo-codes en montrant les résultats affichés pour chacun d'entre eux. Calcul est un exécutable qui additionne les deux entiers transmis en paramètre et affiche le résultat.

```
pour i de 1 à 2
{
    ret=fork();
    if(ret==0)
        ecrire(i);
    else
        ecrire(ret);
}
```

**programme (a)**

```
pour i de 1 à 2
{
    ret=fork();
    if(ret==0)
        execl("/home/calcul", "calcul", "5", "7", null);
    }
    wait();
}
```

**programme (b)**

**Exercice 04:**

Soit le programme suivant :

```
void *f(void *arg)
{
    exit(1);
}
int main()
{
    pthread_t tid;
    pthread_create(&tid, NULL, f, NULL);
    sleep(5);
    printf("FIN");
    return 0;
}
```

- 1) Combien de messages « FIN » sont affichés ? Pourquoi ?
- 2) Modifier le programme pour que l'affichage du message « FIN » soit continuellement affiché.

**Exercice 05:**

Écrire un programme C qui permet de créer deux threads, l'un affichera les nombres pairs de 0 à 10 et l'autre les nombres impairs, de sorte que l'affichage soit comme suit : 0 1 2 3 4 5 ... 10