

TD 1

Module : Programmation système - Section ICE3

Enseignants : Chiheb Ameer ABID  
Islem DENDEN

Année universitaire : 2023-2024

**Exercice 1.**

1) Donner le schéma d'exécution pour chacun des programmes suivants :

```
int main() {
    fork();
    printf("hello!\n");
    exit(0);
}

int main() {
    fork();
    fork();
    printf("hello!\n");
    exit(0);
}
```

```
int main() {
    fork();
    fork();
    fork();
    printf("hello!\n");
    exit(0);
}
```

2) Combien de hello! imprime chacun des programmes ? Combien de processus sont créés ?

a)

```
int main {
    int i;
    for (i=0 ;i<2 ;i++) fork() ;
    printf("hello!\n") ;
    exit(0) ;
}
```

b)

```
void process() {
    fork() ;
    fork() ;
    printf("hello!\n");
}
```

```
int main () {
    process() ;
    printf("hello !\n") ;
    exit(0) ;
}
```

c)

```
int main () {
```

```
if(fork()) fork() ;
printf("hello !\n") ;
exit(0) ;
}
```

d)

```
int main () {
    if (fork()==0) {
        if (fork()) {
            printf("hello !\n") ;
        }
    }
}
```

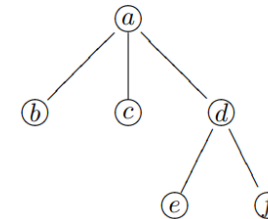
**Exercice 2.**

Proposez un programme qui crée N processus fils en parallèle, puis attend leur terminaison. Chaque processus créé doit afficher son pid, le pid de son pere, et son numéro d'ordre de création.

**Exercice 3.**

1) Écrire un programme qui crée 10 processus fils. Chacun d'entre eux devra afficher dix fois d'affilé son numéro d'ordre entre 0 et 9.

2) Écrire un programme qui engendre l'arbre généalogique suivant :



**Exercice 4.**

On souhaite trouver toutes les occurrences d'un élément dans un tableau donné. Pour cela, si le tableau est long (plus qu'une constante TAILLE\_MIN), on peut le couper en deux et effectuer la recherche indépendamment dans les deux moitiés, en les confiant à deux processus différents travaillant en parallèle. Écrire un programme effectuant la recherche de cette manière ; le programme devra afficher un message pour chaque occurrence trouvée, puis afficher le nombre total d'occurrences de l'élément dans le tableau. Pour cela, on pourra utiliser les valeurs de retour des processus fils (ces valeurs de retour sont codées sur un octet).

**Exercice 5.**

Écrire une fonction `forkn(int n, void (**tab)())` prenant en argument un entier n et un tableau de n pointeurs de fonctions, et lançant n processus fils ; le ième fils doit exécuter la ième fonction avant de terminer. Comment faire pour que les fils s'exécutent en parallèle ? et l'un après l'autre ?