

## Ressource R3.05 - TD 3

### Processus

- (a) Combien de processus le programme engendre-t-il ?  
(b) Affecter des PID aux processus et dire ce qu'affiche le programme (Utiliser un graphe).

---

```
1 int main(void){
2     pid_t tab[4]={-1,-1,-1,-1};;
3     int i;
4     for (i=0;i<2;i++){
5         tab[i]=fork();
6     }
7     printf("Process: %d -- Parent: %d --",getpid(),getppid());
8     printf("tab-->(%d,%d,%d,%d) \n",tab[0],tab[1],tab[2],tab[3]);
9 }
```

---

- Donnez le graphe du processus qui exécute le code suivant :

---

```
1 while (1) {
2     pid_t x=fork();
3     switch(x){
4         case (pid_t)-1:
5             printf("Fork failure! \n");
6             exit(1);
7         case (pid_t)0:
8             sleep(3);
9             exit(0);
10            break;
11        default:
12            sleep(2);
13            /*exit(0);*/
14    }
15 }
```

---

Idem en commentant la ligne 9 et décommentant la ligne 13.

Dans les deux cas, aura-t-on des processus zombies ? orphelins ?

3. Dessinez le graphe de processus correspondant à l'exécution du programme suivant (en admettant que chaque appel `fork()` réussisse)

---

```

1  #include <unistd.h>
2  int main(void){
3  fork() && (fork() || fork());
4  while(1);
5  }

```

---

4. Donnez une ligne de commandes qui est conforme à la sortie de la commande `ps` suivante :

```

USER      PID  PPID  TT      PGID  TPGID  SESS  COMMAND
denis    27445 27441 pts/5    27441 27307 27307 sleep 30
denis    27444 27441 pts/5    27441 27307 27307 sleep 30
denis    27442 27438 pts/5    27438 27307 27307 sleep 30
denis    27440 27438 pts/5    27438 27307 27307 sleep 30
denis    27439 27307 pts/5    27439 27307 27307 sleep 30

```

5. Ecrire un programme qui pour  $n \geq 1$  donné sur la ligne de commande, engendre l'arbre généalogique :

```

0|
 | \1
 |  x
 | \
 |  \2
 |   x
 |   ...
 | \
 |  \
 |   \n
 x    x

```

Chaque processus choisit un nombre au hasard entre 0 et 127. Le processus 0 affichera la plus grande valeur choisie par tous les processus.