

## Exercice 1

Soient les trois programmes P1, P2 et P3 dont les codes sont donnés dans le tableau ci-dessous :

P1	P2	P3
<pre>int main() {     int i;      for (i = 0; i &lt; 3; i++)         fork();     printf("plop!\n");     exit(0); }</pre>	<pre>void doit() {     fork();     fork();     printf("plop\n");     fork(); }  int main() {     doit();     printf("plop\n");     exit(0); }</pre>	<pre>int main() {     if (fork()    fork())         fork();     printf("plop!\n");     exit(0); }</pre>

Pour chacun de programmes ci-dessus :

- 1- Donner l'arborescence générée par le programme.
- 2- Déterminer le nombre des lignes «plop!» imprimées.

## Exercice 2

1. Écrire un programme C "**prog.c**" qui permet de créer un processus fils. Le fils exécute la commande "**ls -l / home**".
2. Modifier le programme précédent de telle manière à faire passer la commande exécutée par le fils lors du lancement du processus.

Exemple d'exécution: "**./prog ls-l /home**"

## Exercice 3

On considère les deux programmes suivants :

P1	P2
<pre> void traitement_bis(char *chaine){     printf("%s\n", chaine); }  int main(){     int pid=0, i;     for (i = 0; i &lt; 3 &amp;&amp; !pid ; i++){         switch (pid=fork()){             case -1 :                 printf("Erreur de fork\n"); exit(-1);             case 0 :                 traitement_bis("Bonjour");             default :                 wait(NULL);         }     }     if (pid != 0)         printf("Au revoir\n");     return 0; } </pre>	<pre> void traitement(char *chaine){     printf("%s\n", chaine);     exit(0); }  int main(){     int i;     for (i = 0; i &lt; 3; i++){         switch (fork()){             case -1 :                 printf("Erreur de fork\n"); exit(-1);             case 0 :                 traitement("Bonjour");             default :                 wait(NULL);         }     }     printf("Au revoir\n");     return 0; } </pre>

Indiquer pour chaque programme :

1. Le nombre de processus créés.
2. Le nombre total de processus.
3. Leur relation de parenté (l'arborescence des processus)
4. L'affichage obtenu (combien de "Bonjour" et de "Au revoir", et pourquoi).
5. Etant donné le diagramme d'états/transitions suivant, citer la/les transition(s) qui doivent être supprimée(s) si on utilise un algorithme d'ordonnancement sans réquisition (non préemptif). Justifier votre réponse.

