Ressource R3.05 - TD 4

Signaux

1. Soit le programme suivant :

```c
int main()
{
  printf("PID %d waiting for a signal.\n", getpid());
  pause();
  printf("PID %d out.\n", getpid());
  return 0;
}
```

Le processus a pour numéro 1234.

(a) Que se passe-t-il si on tape la commande `kill -USR1 1234` au terminal ?

(b) Ajoutez du code pour que le deuxième message s'affiche.

2. Un processus reçoit les signaux `SIGSTOP, SIGTERM, SIGCONT` dans l'ordre indiqué. Quel est à votre avis le comportement de ce processus? Vérifiez votre reponse en TP.

3. Un programme `foo.c` engendre un certain nombre de processus (chacun vit éternellement). On donne la commande

`ps a n -o pid,ppid,pgrp,session,comm --forest`

```
    PID  PPID  PGRP  SESS COMMAND
   2970  2939  2970  2970 bash
  11766  2970 11766  2970  \_ foo
  11767 11766 11767  2970      \_ foo
  11768 11767 11767  2970      |   \_ foo
  11769 11768 11767  2970      |       \_ foo
  11772 11769 11767  2970      |           \_ foo
  11770 11766 11766  2970      \_ foo
  11771 11770 11766  2970          \_ foo
```

(a) Combien y-a-t-il de sessions ? de groupes ? Combien de processus dans chaque groupe ?

(b) Après avoir exécuté une certaine commande la situation change et on voit

```
    PID  PPID  PGRP  SESS COMMAND
   2970  2939  2970  2970 bash
  11766  2970 11766  2970  \_ foo
  11767 11766 11767  2970      \_ foo <defunct>
  11770 11766 11766  2970      \_ foo
  11771 11770 11766  2970          \_ foo
  11768     1 11767  2970 foo
  11769 11768 11767  2970  \_ foo
  11772 11769 11767  2970      \_ foo
```

Quelle est cette commande ?

(c) Proposez une version de `foo.c`.

4. Soit le programme suivant :

```c
void chldhand(int s)
{
  if (waitpid(-1,NULL,WNOHANG)>0) sleep(3);
}
int main(int argc, char ** argv)
{
  int pid,pid1,pid2,pid3;
  struct sigaction sa={0};
  sa.sa_handler=chldhand;
  sigemptyset(&sa.sa_mask);
  sa.sa_flags=SA_RESTART;
  assert (sigaction(SIGCHLD,&sa,NULL) !=-1 );
  pid=fork();
  if (pid<0) { perror("fork0"); exit(1); }
  if (pid==0) { sleep(1); exit(0); }
  pid1=fork();
  if (pid1<0) { perror("fork1"); exit(1); }
  if (pid1==0) { sleep(2); exit(0); }
  pid2=fork();
  if (pid2<0) { perror("fork2"); exit(1); }
  if (pid2==0) { sleep(2); exit(0); }
  pid3=fork();
  if (pid3<0) { perror("fork3"); exit(1); }
  if (pid3==0) { sleep(2); exit(0); }
  while(1) sleep(2);
}
```

Combien de processus en tout engendre ce programme ? Combien d'entre eux vivront éternellement, mourront, deviendront zombie ?