NOMBRE:CHRISTOFER FABIÁN CHÁVEZ CARAZAS

- Escriba un programa en el cual se generen procesos y uno de ellos quede huérfano. Muestre y explique la evidencia que demuestre la existencia del proceso huerfano.

```c
int main(int argc, char *argv[]){
    pid_t pid;
    if ((pid=fork()) == 0 ){
        printf("1->Soy el hijo (%d, hijo de %d)\n", getpid(), getppid());
        sleep(30);
        printf("2->Soy el hijo (%d, hijo de %d)\n", getpid(), getppid());
    }
    else{
        printf("1->Soy el padre (%d, mi hijo es %d)\n", getpid(), pid);
        sleep(10);
        printf("2->Soy el padre (%d, mi hijo es %d)\n", getpid(), pid);
    }
    return 0;
}
```

```
xnpio@xnpio-Satellite-U40t-A:~/Documentos/Xnpio/SO/sesion8$ ./ejem
1->Soy el padre (5025, mi hijo es 5026)
1->Soy el hijo (5026, hijo de 5025)
2->Soy el padre (5025, mi hijo es 5026)
xnpio@xnpio-Satellite-U40t-A:~/Documentos/Xnpio/SO/sesion8$ 2->Soy el hijo (5026
, hijo de 1602)
```

- Escriba un programa en el cual se creen procesos y uno de ellos se encuentre en estado zombie. Muestre y explique la evidencia que demuestre la existencia del proceso zombie.

```c
int main(int argc, char *argv[]){
    pid_t pid;
    if ((pid=fork()) == 0 ){
        printf("1->Soy el hijo (%d, hijo de %d)\n", getpid(), getppid());
        sleep(10);
        printf("2->Soy el hijo (%d, hijo de %d)\n", getpid(), getppid());
    }
    else{
        printf("1->Soy el padre (%d, mi hijo es %d)\n", getpid(), pid);
        sleep(30);
        printf("2->Soy el padre (%d, mi hijo es %d)\n", getpid(), pid);
    }
    return 0;
}
```

```
top - 19:32:04 up  1:37,  2 users,  load average: 1,26, 1,19, 1,02
Tareas: 257 total,   1 ejecutar,  255 hibernar,    0 detener,    1 zombie
%Cpu(s): 16,0 usuario,  5,2 sist,  0,1 adecuado, 76,6 inact,  2,0 en espera,  0,
KiB Mem :  6022144 total,   756668 free,  2774828 used,  2490648 buff/cache
KiB Swap:  8063996 total,  8063996 free,        0 used.  2428204 avail Mem

  PID USUARIO   PR  NI    VIRT    RES    SHR S  %CPU %MEM     HORA+ ORDEN
 2397 xnpio     20   0 2572832 1,274g 130892 S  38,5 22,2  45:16.38 firefox
 1105 root      20   0  517596 144344 129152 S  15,6  2,4   5:51.82 Xorg
 1945 xnpio     20   0 1629316 123396  64932 S   9,6  2,0   4:14.47 compiz
 5610 xnpio     20   0  424004  22168  18700 S   3,0  0,4   0:00.19 gnome-scre+
 1897 xnpio      9 -11  577832  14996  11212 S   2,0  0,2   2:02.47 pulseaudio
  771 root      20   0    4400   1360   1264 S   0,7  0,0   0:06.43 acpid
 2962 xnpio     20   0 1160668  77032  58036 S   0,7  1,3   0:28.90 unity-cont+
    6 root      20   0       0      0      0 S   0,3  0,0   0:03.71 kworker/u8+
    7 root      20   0       0      0      0 S   0,3  0,0   0:12.79 rcu_sched
  799 message+  20   0   44400   5220   3428 S   0,3  0,1   0:02.05 dbus-daemon
 1033 debian-+  20   0   78864  34792  12408 S   0,3  0,6   0:03.24 tor
 1335 root      20   0  169760  14256   8752 S   0,3  0,2   0:21.28 teamviewerd
 3233 xnpio     20   0 1553288 188804  68960 S   0,3  3,1   0:36.41 Stremio-ru+
 3265 xnpio     20   0 1291408 224604 100300 S   0,3  3,7   0:42.51 Stremio-ru+
 5168 root      20   0       0      0      0 S   0,3  0,0   0:00.03 kworker/2:0
    1 root      20   0  185216   5800   3920 S   0,0  0,1   0:01.56 systemd
    2 root      20   0       0      0      0 S   0,0  0,0   0:00.00 kthreadd
```

- Escriba un programa que cree la sgte jerarquía de procesos:

  P : el proceso padre debe de esperar a que sus hijos terminen para terminarse.

  H1 : el hijo 1, debe de mostrar su PID y el de su padre, además de enviarle cada 20 segundos una señal SIGUSR1 a H2.

  H2 : el hijo 2 se activa cuando recibe la señal de H1, y debe de mostrar su PID como respuesta a la señal SIGUSR1 que recibe, para luego volverse a suspender.

  H3 : el hijo 3 ejecuta un script que le de la bienevenida al usuario activo, considerando el horario correspondiente.

```c
#include <sys/types.h>
#include <signal.h>
#include <stdlib.h>
#include <unistd.h>
#include <stdio.h>

void alarma();
void despertar();

pid_t pidH1, pidH2 , pidH3;

int main(int argc, char *argv[]){
    int status1, status2;
    if((pidH2 = fork()) ==  0){
        while(1){
            signal(SIGUSR1,despertar);
            pause();
        }
    }
    else{
        if((pidH1 = fork()) == 0){
            if((pidH3 = fork()) == 0){
                system("sh script.sh");
            }
            else{
                printf("Soy el hijo1 -> %d y mi padre es %d\n",getpid(),getppid());
                signal(SIGALRM,alarma);
                alarm(20);
                while(1);
            }

        }
        else{
            waitpid(pidH1,&status1,0);
            waitpid(pidH2,&status2,0);
        }
    }
}

void alarma(){
    signal(SIGALRM,SIG_IGN);
    kill(pidH2,SIGUSR1);
    signal(SIGALRM,alarma);
    alarm(20);
}

void despertar(){
    printf("Soy el hijo2 -> %d\n",getpid());
}
```

```
xnpio@xnpio-Satellite-U40t-A:~/Documentos/Xnpio/SO/sesion8$ ./ejem
Soy el hijo1 -> 13760 y mi padre es 13758
Buenas noches, xnpio
Soy el hijo2 -> 13759
Soy el hijo2 -> 13759
Soy el hijo2 -> 13759
Soy el hijo2 -> 13759
Soy el hijo2 -> 13759
Soy el hijo2 -> 13759
Soy el hijo2 -> 13759
Soy el hijo2 -> 13759
Soy el hijo2 -> 13759
Soy el hijo2 -> 13759
Soy el hijo2 -> 13759
Soy el hijo2 -> 13759
```