UNIVERSIDAD NACIONAL SAN AGUSTÍN
INGENIERÍA DE SISTEMAS

Sistemas Operativos: Laboratorio 5

DOCENTE: KARIM GUEVARA PUENTE DE LA VEGA.

Presentado por:

Salazar Taco, Jashin Mario

AREQUIPA 2020

Ejercicio 1.

El código usado en el ejercicio 1 es el siguiente.

```
#include <stdio.h>
#include <sys/types.h>
#include <unistd.h>
#include <stdlib.h>
int main(){
    pid_t pid_hijo;
    pid_t pid_nieto;
    printf("El piddel programa principal es %d\n", (int) getpid());
    pid_hijo = fork();
    if(pid_hijo!=0){
        printf("Este es el proceso padre con ID %d\n", (int)
getpid());
        printf("El ID del hijo es %d\n", (int) pid_hijo);
        sleep(10);
    }else{
        pid_nieto=fork();
        if(pid_nieto!=0){
            printf("Este es el proceso hijo, con ID %d\n", (int)
getpid());
            printf("Mi padre es, con ID %d\n", (int) getppid());
            sleep(10);
        }else{
            printf("Este es el proceso nieto, con ID %d\n", (int)
getpid());
            printf("Mi padre es, con ID %d\n", (int) getppid());
            char* argumentos[] = {"./script01.sh", "","/",NULL} ;
            execvp("./script01.sh", argumentos);

        }
    }
    return 0;
}
```

El ejercicio nos pide crear un hijo de un hijo, basándonos en el ejercicio fork.c, se le agrego una variable denominada pid_nieto, este será creado con un fork() cuando nos encontremos dentro del proceso hijo, este proceso hijo solo mostrara su ID y el de su padre, mientras el proceso nieto ejecutara el archivo “script01.sh” desarrollado en el laboratorio pasado con el comando “execvp”.

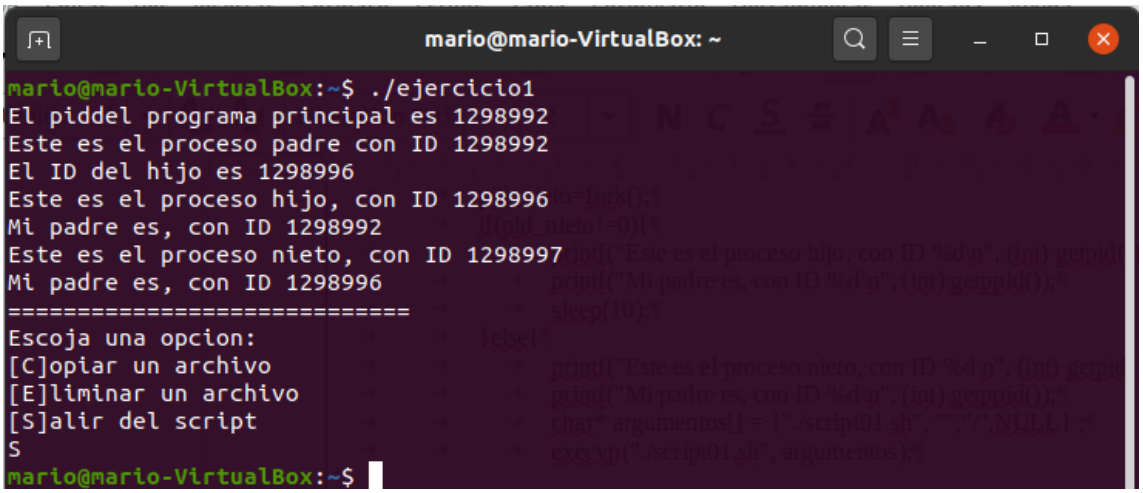
Se uso el comando sleep(10) para hacer dormir a los padres, bien esto no es lo óptimo quizás hubiera sido mejor usar wait() pero por motivos de que el ejercicio no lo indica se dejo asi.

La ejecución del código se muestra a continuación.

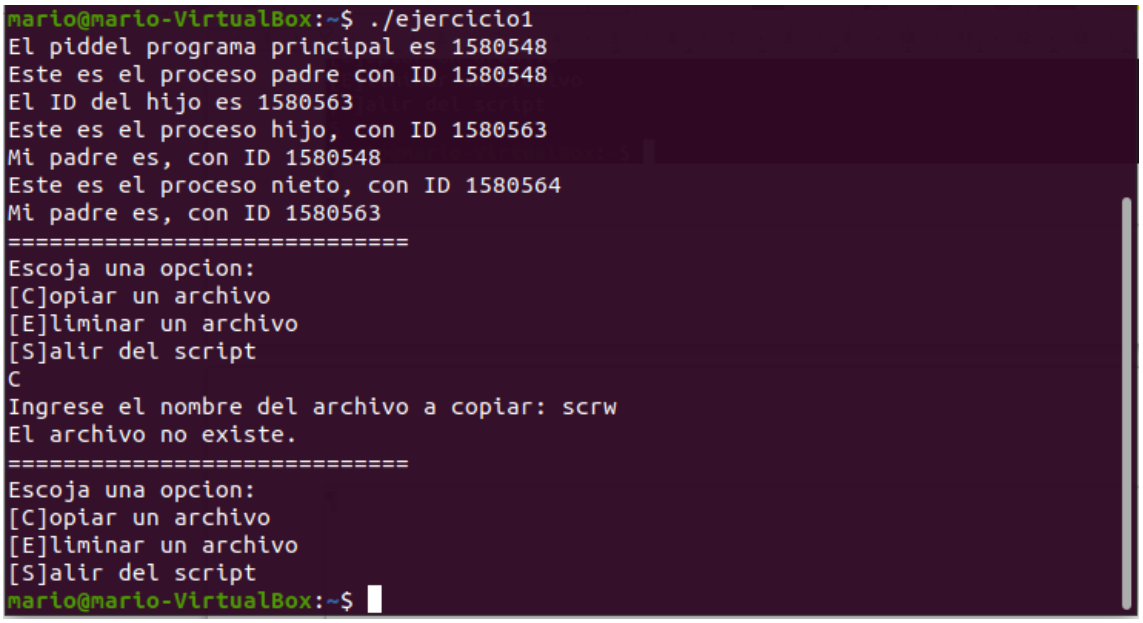
Primero se hizo la respectiva compilación.

```
mario@mario-VirtualBox:~$ gcc ejercicio1.c -o ejercicio1
mario@mario-VirtualBox:~$
```

No mostrando ningún error.



En la ejecución se aprecia que la estructura definida es correcta ya que el padre es 1298992, el hijo es 1298996 y el nieto es 1298997, siendo este ultimo el que ejecuta el script, lo malo seria que al presiona “S” para salir pues aun se esta ejecutando los sleep programados en el código y se tiene que esperar a que estos terminen.



Se muestra también el correcto funcionamiento del “script01.sh”, con el inconveniente de que si se acaba el tiempo de los 20 segundos en total programados en el sleep se terminara todo proceso.

Ejercicio 2.

Para el ejercicio numero 2 se utilizó el siguiente código.

```
#include <signal.h>
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
int ncortes=0;
int bucle=1;
void cowsay();
void cortar();
pid_t pid_hijo;
int main(){
    signal (SIGINT, cortar);
    signal (SIGUSR1, cowsay);
    printf("Ejercicio de signal 2. \n");
    while(bucle);
    signal (SIGINT,SIG_IGN);
    printf("Hasta luego....\n");
    exit(0);
}

void cortar(){
    signal (SIGINT,SIG_IGN);
    ncortes++;
    printf("Has pulsado CTRL C %d veces.\n", ncortes);
    signal (SIGINT, cortar);
}

void cowsay(){
    signal (SIGUSR1, cowsay);
    printf("Â¡cowsay! \n");
    pid_hijo = fork();
    if(pid_hijo==0){
        char* argumentos[] = {"cowsay", "hola","/",NULL} ;
        execvp("cowsay", argumentos);
    }
}
```

Se uso 2 procedimientos uno cortar, el cual contara cuantas veces se preciona CTRL-C y las ira mostrando por pantalla aumentando el contador “ncortes”.

El segundo procedimiento es cowsay, el cual cuando le mandemos la señal desde otra terminal ejecutara el comando “cowsay hola” mediante un proceso hijo creado por fork().

```
mario@mario-VirtualBox:~$ gcc ejercicio2.c -o ejercicio2
mario@mario-VirtualBox:~$ ./ejercicio2
Ejercicio de signal 2.
^CHas pulsado CTRL C 1 veces.
^CHas pulsado CTRL C 2 veces.
^CHas pulsado CTRL C 3 veces.
^CHas pulsado CTRL C 4 veces.
^CHas pulsado CTRL C 5 veces.
^CHas pulsado CTRL C 6 veces.
^CHas pulsado CTRL C 7 veces.
^CHas pulsado CTRL C 8 veces.
^CHas pulsado CTRL C 9 veces.
^CHas pulsado CTRL C 10 veces.
^CHas pulsado CTRL C 11 veces.
^CHas pulsado CTRL C 12 veces.
¡cowsay!
 ________
< hola / >
 --------
        \   ^__^
         \  (oo)\_______
            (__)\       )\/\
                ||----w |
                ||     ||
```

```
mario@mario-VirtualBox: ~
mario@mario-VirtualBox:~$ pgrep ejercicio2
3577
3580
3762
3763
3786
3815
3942
mario@mario-VirtualBox:~$ kill -10 3942
mario@mario-VirtualBox:~$
```