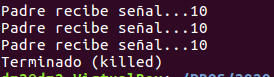
**ACTIVIDAD 10**

**🗶**

**Enunciado**

Realiza un programa C en donde un hijo envíe 3 señales SIGUSR1 a su padre y después envíe una señal SIGKILL para que el proceso padre termine.

**Resultado esperado:**



**Solución:**

Incluir aquí el código del programa Actividad10.c

#include <stdio.h>

#include <stdlib.h>

#include <signal.h>

#include <unistd.h>

// Metodo que devuelve los mensajes que saldran por consola

void signal\_handler(int signo) {

if (signo == SIGUSR1) {

printf("Padre recibe señal...10\n");

} else if (signo == SIGKILL) {

printf("Terminado (killed)\n");

exit(0);

}

}

int main() {

// Configurar el manejador de señales para SIGUSR1

signal(SIGUSR1, signal\_handler);

pid\_t child\_pid;

child\_pid = fork();

if (child\_pid < 0) {

// si el pid es negativo, da error

perror("Error al crear el proceso hijo");

exit(1);

} else if (child\_pid == 0) {

// Esperar para que el proceso padre esté listo para recibir señales

sleep(2);

// Enviar 3 señales SIGUSR1 al proceso padre

for (int i = 0; i < 3; i++) {

kill(getppid(), SIGUSR1);

// un segundo de espera

sleep(1);

}

// Utilizamos SIGKILL para matar el proceso padre

kill(getppid(), SIGKILL);

}

return 0;

}

**Resolución de problemas:**

**CRITERIOS DE CORRECCIÓN**

1. Nivel de ajuste a la nomenclatura de objetos definida en el enunciado de la actividad y nivel de ajuste al almacenamiento de la información requerida en las subcarpetas definidas para la actividad y unidad didáctica correspondientes (15%).
2. Correcto funcionamiento de los programas desarrollados en C (35%).
3. Ajuste a los resultados esperados (35%).
4. Claridad del código C (15%).
5. Explicación detallada de cómo se han resuelto problemas que hayan podido surgir en la realización de la actividad (hasta un 15% adicional sobre la nota obtenida).
6. Avisos o errores de compilación (-15%).
7. Nivel de comentarios en el código fuente (-15%).