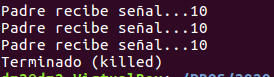
**ACTIVIDAD 10**

**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:**

#include <stdio.h>

#include <stdlib.h>

#include <signal.h>

#include <unistd.h>

#include <sys/wait.h>

void handle\_signal(int signal) {

printf("Padre recibe señal...%d\n", signal);

}

int main() {

pid\_t pid = fork();

if (pid < 0) {

fprintf(stderr, "Error en fork\n");

exit(1);

}

if (pid > 0) { // Proceso padre

signal(SIGUSR1, handle\_signal);

// El proceso padre espera que el hijo termine

wait(NULL);

} else { // Proceso hijo

// Enviar tres señales SIGUSR1 al proceso padre

kill(getppid(), SIGUSR1);

sleep(1);

kill(getppid(), SIGUSR1);

sleep(1);

kill(getppid(), SIGUSR1);

sleep(1);

// Enviar una señal SIGKILL para terminar 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%).