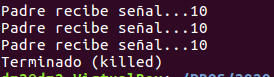
**ACTIVIDAD 10**

**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>

#include <sys/wait.h>

int count = 0;

// Función para manejar la señal SIGUSR1 en el padre

void sigusr1\_handler(int signo) {

if (signo == SIGUSR1) {

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

}

}

int main() {

// Registra el manejador de señal para SIGUSR1 en el padre

signal(SIGUSR1, sigusr1\_handler);

pid\_t child\_pid;

// Crea un proceso hijo

if ((child\_pid = fork()) < 0) {

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

exit(1);

}

if (child\_pid == 0) {

// Este es el proceso hijo

sleep(1); // Espera un segundo para asegurarse de que el padre esté listo

// Envía tres señales SIGUSR1 al proceso padre

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

kill(getppid(), SIGUSR1);

sleep(1);

}

// Envía una señal SIGKILL para terminar el proceso padre

kill(getppid(), SIGKILL);

} else {

// Este es el proceso padre

while (1) {

// Espera a recibir señales

pause();

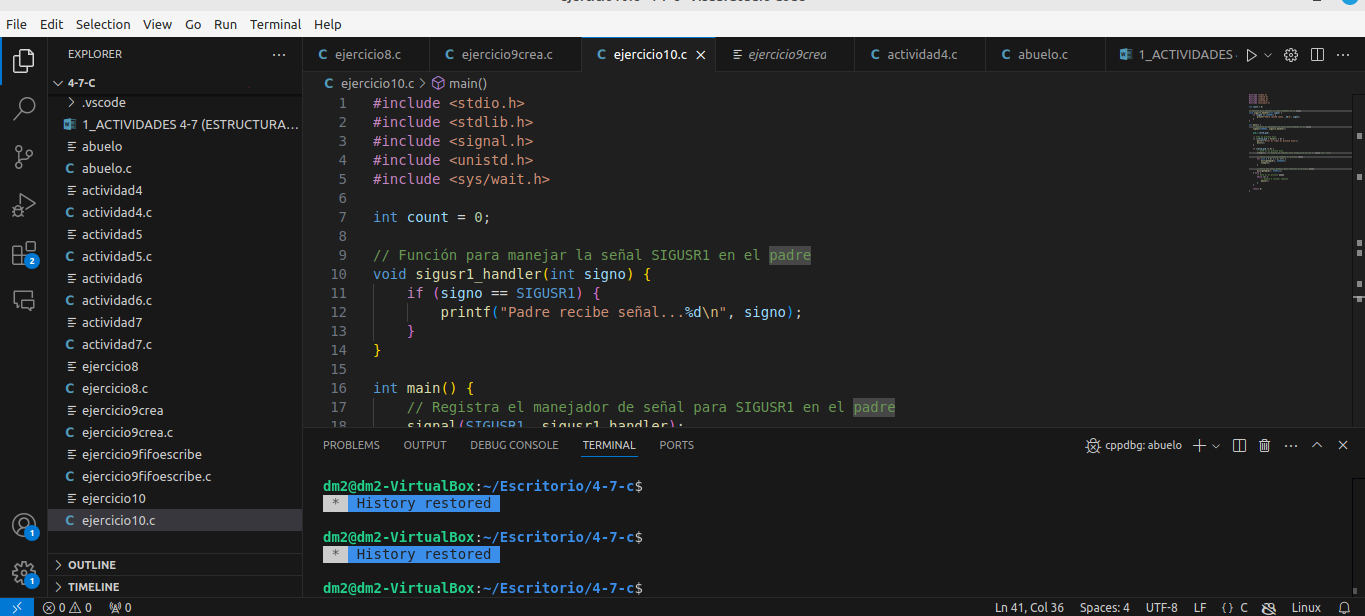
}

}

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%).