**ENUNCIADO**

**Actividad 8**

Realiza un programa en C donde los procesos abuelo, hijo y nieto se envíen mensajes de acuerdo al siguiente gráfico:

fd1 fd2

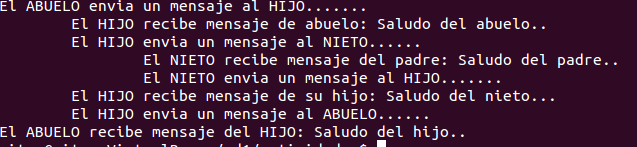
ABUELO –--------> HIJO –---------> NIETO

←------- < ----------

fd2 fd1

Nota: como se puede observar en el gráfico, se utilizan 2 pipes.

**Resultado esperado:**



**Solución:**

**#include <stdio.h>**

**#include <stdlib.h>**

**#include <unistd.h>**

**#include <string.h>**

**int main() {**

**int fd1[2], fd2[2];**

**pid\_t p;**

**// Crear los pipes**

**if (pipe(fd1) == -1) {**

**fprintf(stderr, "Pipe Failed");**

**return 1;**

**}**

**if (pipe(fd2) == -1) {**

**fprintf(stderr, "Pipe Failed");**

**return 1;**

**}**

**p = fork();**

**// Proceso abuelo**

**if (p > 0) {**

**char buff[100];**

**close(fd1[0]); // Cerrar el extremo de lectura del primer pipe**

**write(fd1[1], "Saludo del abuelo", strlen("Saludo del abuelo") + 1);**

**printf("El ABUELO envía un mensaje al HIJO...\n");**

**close(fd1[1]);**

**wait(NULL); // Esperar a que el hijo termine**

**close(fd2[1]); // Cerrar el extremo de escritura del segundo pipe**

**read(fd2[0], buff, sizeof(buff));**

**printf("El ABUELO recibe mensaje del HIJO: %s\n", buff);**

**close(fd2[0]);**

**}**

**// Proceso hijo**

**else {**

**pid\_t p2 = fork();**

**if (p2 > 0) {**

**char buff[100];**

**close(fd1[1]); // Cerrar el extremo de escritura del primer pipe**

**read(fd1[0], buff, sizeof(buff));**

**printf("\tEl HIJO recibe mensaje del ABUELO: %s\n", buff);**

**close(fd1[0]);**

**close(fd2[0]); // Cerrar el extremo de lectura del segundo pipe**

**write(fd2[1], "Saludo del padre", strlen("Saludo del padre") + 1);**

**printf("\tEl HIJO envía un mensaje al NIETO...\n");**

**wait(NULL); // Esperar a que el nieto termine**

**read(fd1[0], buff, sizeof(buff));**

**printf("\tEl HIJO recibe mensaje de su hijo: %s\n", buff);**

**close(fd1[0]);**

**write(fd2[1], "Saludo del hijo", strlen("Saludo del hijo") + 1);**

**printf("\tEl HIJO envía un mensaje al ABUELO...\n");**

**close(fd2[1]);**

**}**

**// Proceso nieto**

**else {**

**char buff[100];**

**close(fd2[1]); // Cerrar el extremo de escritura del segundo pipe**

**read(fd2[0], buff, sizeof(buff));**

**printf("\t\tEl NIETO recibe mensaje del HIJO: %s\n", buff);**

**close(fd2[0]);**

**write(fd1[1], "Saludo del nieto", strlen("Saludo del nieto") + 1);**

**printf("\t\tEl NIETO envía un mensaje al HIJO...\n");**

**close(fd1[1]);**

**}**

**}**

**return 0;**

**}**

**Resolución de problemas:**

El problema era que el proceso "hijo" cierra el extremo de escritura de fd2 antes de que el "abuelo" tenga la oportunidad de leer el mensaje final.

**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%).