**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

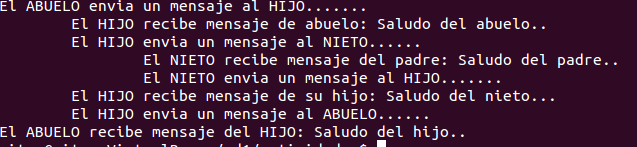
ABUELO –--------> HIJO –---------> NIETO

←------- < ----------

fd2 fd1

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

**Resultado esperado:**



**Solución:**

**#include <stdio.h>**

**#include <stdlib.h>**

**#include <string.h>**

**#include <unistd.h>**

**#include <sys/wait.h>**

**int main() {**

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

**pid\_t hijo, nieto;**

**// Crear los pipes**

**if (pipe(fd1) == -1 || pipe(fd2) == -1) {**

**fprintf(stderr, "Fallo en pipe");**

**return 1;**

**}**

**hijo = fork();**

**if (hijo < 0) {**

**fprintf(stderr, "Fallo en fork");**

**return 1;**

**}**

**if (hijo > 0) { // Proceso Abuelo**

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

**char saludo\_abuelo[] = "Saludo del abuelo";**

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

**write(fd1[1], saludo\_abuelo, strlen(saludo\_abuelo) + 1);**

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

**// Espera a que el proceso hijo termine**

**wait(NULL);**

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

**char buffer[100];**

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

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

**} else { // Proceso Hijo**

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

**char buffer[100];**

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

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

**nieto = fork();**

**if (nieto < 0) {**

**fprintf(stderr, "Fallo en fork");**

**return 1;**

**}**

**if (nieto > 0) { // Proceso Hijo**

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

**char saludo\_hijo[] = "Saludo del hijo";**

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

**write(fd2[1], saludo\_hijo, strlen(saludo\_hijo) + 1);**

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

**// Espera a que el proceso nieto termine**

**wait(NULL);**

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

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

**char respuesta\_hijo[] = "Respuesta del hijo";**

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

**write(fd1[1], respuesta\_hijo, strlen(respuesta\_hijo) + 1);**

**} else { // Proceso Nieto**

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

**char buffer[100];**

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

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

**char respuesta\_nieto[] = "Respuesta del nieto";**

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

**write(fd2[1], respuesta\_nieto, strlen(respuesta\_nieto) + 1);**

**}**

**}**

**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%).