**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

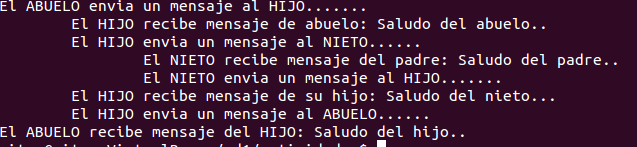
ABUELO –--------> HIJO –---------> NIETO

←------- < ----------

fd2 fd1

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

**Resultado esperado:**



**Solución:**

// ejercicio8

// Joseba Martinez

#include <stdlib.h>

#include <unistd.h>

#include <stdio.h>

#include <sys/wait.h>

void main (){

pid\_t pid,pid2;

int status;

int fd1[2];

int fd2[2];

char buffer[25];

pipe(fd1);

pipe(fd2);

pid = fork();

switch (pid){

case -1:

printf("No se a podido crear el proceso hijo\n");

exit(-1);

break;

case 0:

pid2 = fork();

switch (pid2){

case -1:

printf("No se a podido crear el proceso nieto\n");

exit(-1);

break;

// Nieto

case 0:

read(fd2[0],buffer,18);

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

write(fd1[1],"Saludo del NIETO..",18);

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

exit(0);

// Hijo

default:

read(fd1[0],buffer,19);

printf("\tEl HIJO recibe mensaje de abuelo: %s\n",buffer);

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

write(fd2[1],"Saludo del PADRE..\n",18);

// Espera a que el nieto responda, lee el mesnaje y escribe al padre

waitpid(pid2,&status,0);

read(fd1[0],buffer,18);

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

write(fd2[1],"Saludo del HIJO..\n",18);

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

exit(0);

}

// Padre

default:

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

write(fd1[1],"Saludo del ABUELO..\n",19);

// Espera a que el hijo le escriba y lee el mensaje

waitpid(pid,&status,0);

read(fd2[0],buffer,18);

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

break;

}

exit(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%).