**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

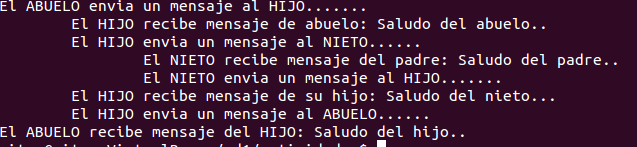
ABUELO –--------> HIJO –---------> NIETO

←------- < ----------

fd2 fd1

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

**Resultado esperado:**



**Solución:**

**#include <stdlib.h>**

#include <unistd.h>

#include <stdio.h>

#include <sys/wait.h>

void imprimir\_mensaje(const char \*mensaje) {

printf("%s", mensaje);

fflush(stdout);

}

int main() {

int fd1[2], fd2[2];

char buffer[100];

pid\_t pid\_hijo, pid\_nieto;

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

perror("Error al crear los pipes");

exit(-1);

}

pid\_hijo = fork();

if (pid\_hijo == -1) {

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

exit(-1);

}

if (pid\_hijo == 0) {

// Proceso nieto

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

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

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

imprimir\_mensaje(buffer);

char mensaje\_nieto[] = "El NIETO envia un mensaje al HIJO.....\n";

write(fd2[1], mensaje\_nieto, sizeof(mensaje\_nieto));

exit(0);

} else {

pid\_nieto = fork();

if (pid\_nieto == -1) {

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

exit(-1);

}

if (pid\_nieto == 0) {

// Proceso hijo

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

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

char mensaje\_hijo[] = "El HIJO envia un mensaje al NIETO.....\n";

write(fd1[1], mensaje\_hijo, sizeof(mensaje\_hijo));

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

imprimir\_mensaje(buffer);

char mensaje\_nieto2[] = "El NIETO envia un mensaje al HIJO.....\n";

write(fd1[1], mensaje\_nieto2, sizeof(mensaje\_nieto2));

exit(0);

} else {

// Proceso abuelo

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

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

char mensaje\_abuelo[] = "El ABUELO envia un mensaje al HIJO....\n";

write(fd1[1], mensaje\_abuelo, sizeof(mensaje\_abuelo));

wait(NULL);

wait(NULL);

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

imprimir\_mensaje(buffer);

exit(0);

}

}

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%).