**ENUNCIADO**

**Actividad 9**

Modificar los programas C de nombre actividad9fifocrea.c y actividad9fifoescribe.c, para que cuando se ejecute el programa actividad9fifoescribe, envíe al programa actividad9fifocrea un mensaje para que éste lo visualice.

**//actividad9fifocrea.c**

#include <stdio.h>

#include <stdlib.h>

#include <sys/types.h>

#include <sys/stat.h>

#include <fcntl.h>

#include <unistd.h>

int main (void)

{

int fp;

int p, bytesleidos;

char saludo[] = "Un saludo !!!!!\n", buffer [10];

fp = open ("FIFO2", 0); /// abrimos fifo con permiso de lectura

if (p== -1) {

printf("Ha ocurrido un error: Recuerda borrar la pipe la proxima vez.... \n"); // recuerda borrarlo la segunda vez...

exit (0);

}

while (1) {

fp = open ("FIFO2", 0); //Abrimos el FIFO con permiso de lectura para leer proximos mensajes

bytesleidos = read(fp, buffer, 1);

printf("Obteniendo información...\n");

while (bytesleidos != 0) { //recorremos el mensaje caracter a caracter y lo imprimimos por pantalla

printf("%s", buffer);

bytesleidos = read (fp, buffer, 1);

}

close (fp);

}

return(0);

}

**//actividad9fifoescribe.c**

#define \_DEFAULT\_SOURCE

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include <fcntl.h>

#include <unistd.h>

int main (void)

{

int fp;

int p, bytesleidos;

char saludo[] = "Un saludo !!!!!\n";

fp = open ("FIFO2", 1); /// abrimos fifo con permiso de escritura

if (fp == -1) {

printf("Error al abrir el fichero... \n");

exit (1);

}

printf("Mandando información al FIFO...\n");

write (fp, saludo, sizeof(saludo));

close (fp);

return(0);

}

**Resultado esperado:**

$./actividad9fifocrea

Obteniendo información…Un saludo....

Obteniendo información…Un saludo....

Obteniendo información…Un saludo....

$./actividad9fifoescribe

Mandando información al FIFO…

$./ actividad9fifoescribe

Mandando información al FIFO…

$./ actividad9fifoescribe

Mandando información al FIFO…

**Solución:**

//actividad9fifocrea.c

#include <stdio.h>

#include <stdlib.h>

#include <sys/types.h>

#include <sys/stat.h>

#include <fcntl.h>

#include <unistd.h>

int main (void)

{

int fp;

int p, bytesleidos;

char saludo[] = "Un saludo !!!!!\n", buffer [10];

fp = open ("FIFO2", 0); /// abrimos fifo con permiso de lectura

if (p== -1) {

printf("Ha ocurrido un error: Recuerda borrar la pipe la proxima vez.... \n"); // recuerda borrarlo la segunda vez...

exit (0);

}

while (1) {

fp = open ("FIFO2", 0); //Abrimos el FIFO con permiso de lectura para leer proximos mensajes

bytesleidos = read(fp, buffer, 1);

printf("Obteniendo información...\n");

while (bytesleidos != 0) { //recorremos el mensaje caracter a caracter y lo imprimimos por pantalla

printf("%s", buffer);

bytesleidos = read (fp, buffer, 1);

}

close (fp);

}

return(0);

}

//actividad9fifoescribe.c

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include <fcntl.h>

#include <unistd.h>

int main (void)

{

int fp;

int p, bytesleidos;

char saludo[] = "Un saludo !!!!!\n";

p=mknod("FIFO2", S\_IFIFO|0666, 0); /// creamos FIFO con permisos de lectura y escritura.

//Como el FIFO se crea en el archivo de escritura, habria que ejecutar este en primer lugar

fp = open ("FIFO2", 1); /// abrimos fifo con permiso de escritura

if (fp == -1) {

printf("Error al abrir el fichero... \n");

exit (1);

}

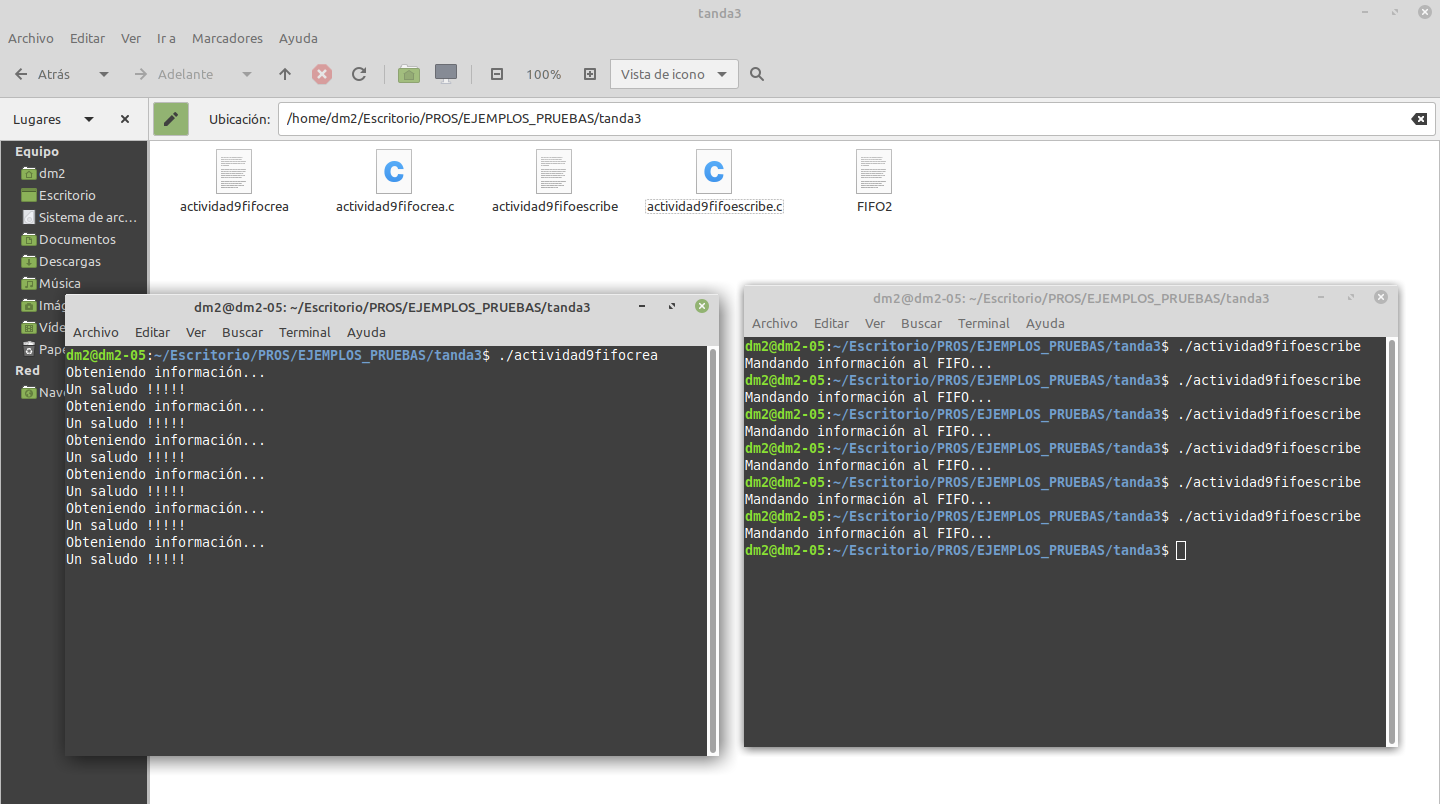
printf("Mandando información al FIFO...\n");

write (fp, saludo, sizeof(saludo)); //Escribimos el mensaje en el FIFO

close (fp);

return(0);

}

****

**Resolución de problemas:**

**REQUISITOS**

1. Entregar la solución de la actividad en un único archivo comprimido donde tendremos:
   1. Un archivo de extensión .odt o .docx que contendrá
      1. Una copia del código del programa
      2. Una captura de pantalla del resultado de la ejecución del programa
   2. El archivo con el código fuente de extensión .c
2. El único archivo con la solución contendrá el nombre original de la actividad, seguido de un guión bajo y del primer apellido del alumno.

**CRITERIOS DE CORRECCIÓN**

1. Nivel de cumplimiento de los Requisitos (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%).