Descripción de actividades en la evaluación de los indicadores 19I

Oscar Herrera Alcántara	1151018	Sistemas Operativos

AE1	Identificar, formular y resolver problemas de ingeniería con base en los fundamentos de las ciencias básicas y los principios de la ingeniería.
CD5	El alumno Integra conocimientos, de ciencias básicas o de ingeniería, para formular y resolver problemas
I1	50 % de los alumnos Integra conocimientos, de ciencias básicas o de ingeniería, para formular y resolver problemas..
Calificación	100

Actividad: Práctica con programa y reporte sobre la comunicación local y en red de varios procesos para intercambiar información de cambios en un árbol de directorios de GNU/Linux.

Descripción:

Codifique un programa en C que comunica varios procesos a efecto de:

1. Solicitar al usuario un archivo o directorio al cual se realicen operaciones de creación, eliminación, renombrar o consultar dentro de una carpeta dada del directorio home
2. Ejecutar la instrucción de crear, eliminar, renombrar un archivo o directorio
3. Monitorear los cambios en el sistema de archivos
4. Sincronizar el árbol de directorios en un servidor y en otro cliente

Genere un reporte de la actividad realizada y responda las siguientes preguntas:

1. ¿Cuántos procesos creó con la llamada fork?
2. ¿Cuáles mecanismos de intercomunicación de procesos usó?

El reporte debe incluir, además:

3. El código fuente del programa solución e indique claramente las llamadas al sistema utilizadas
4. Un diagrama ilustrativo de la arquitectura del programa implementado

Evidencia: Reporte de actividad atendiendo los 4 numerales previos.

Reporte

1. **¿Cuántos procesos creó con la llamada fork?**

   En el programa del servidor, se utilizaron 3 llamadas a fork(), el primer hijo es quien monitorea la carpeta introducida por el usuario, el segundo hijo es quien atiende al primer cliente, lo escucha y le responde, el tercer hijo es quien atiende al segundo cliente, lo escucha y le responde.
   En el programa cliente, se usaron también 3 llamadas a fork(), el primer hijo es quien atiende al usuario y le escribe al padre en la tubería, el segundo hijo es quien monitorea la carpeta introducida por el usuario y el tercer hijo es quien realiza la conexión con el servidor y recibe sus mensajes, el proceso padre es quien envía los mensajes al servidor.

2. **¿Cuáles mecanismos de intercomunicación de procesos usó?**

   Se usaron tuberías en el código del cliente para comunicar al hijo que escribe en tubería con el padre que lee de la tubería y comunicación mediante send() entre el cliente y el servidor.

3. **El código fuente del programa solución e indique claramente las llamadas al sistema utilizadas**

//////////////////////////Código del cliente////////////////////////////////

```
#include <string.h>
#include <stdlib.h>
#include <stdio.h>
#include <sys/types.h>
#include <sys/stat.h>
#include <sys/wait.h>
#include <unistd.h>
#include <sys/fcntl.h>
#include <sys/inotify.h>
#include <sys/socket.h>
#include <arpa/inet.h>
#include <netdb.h>
#include <netinet/in.h>
#include <sys/mman.h>
#define EVENT_SIZE  ( sizeof (struct inotify_event) )
#define EVENT_BUF_LEN     ( 1024 * ( EVENT_SIZE + 16 ) )
#define MAX 256
#define port 3553
int main (int argc, char** argv)
{
        //modo de lectura y escritura
        mode_t mode = S_IRUSR | S_IWUSR;
        //ruta donde se modificarán los archivos o directorios
        char path[50],path2[50],pathx[50];
        char ipServer[50];
        if(argc!=3){
                printf("\nForma de uso %s <ruta> <ip>, ej \n%s /home/victor/Documents/
127.0.0.1\n",argv[0],argv[0]);
                exit(-1);
        }
        strcpy(path,argv[1]);
        strcpy(path2,argv[1]);
        strcpy(pathx,argv[1]);
        strcpy(ipServer,argv[2]);
```

```
        //printf("%s,%s",path,ip);
        //identificador de la opción del usuario
        char id[MAX];
        //variable auxiliar para mandar la ruta + nombre del archivo
        char aux[MAX];
        //variable auxiliar2 para mandar la ruta + nombre del archivo
        char aux2[MAX];
        char aux3[MAX];
        //variable donde se leerá el nombre del archivo o dir a c/e
        char nombre[MAX];
        //variable donde se leerá el renombrado de un archivo
        char nombre2[MAX];
        //tubería[0]=lectura, tuberia[1]=escritura
        int tuberia[2];
int pid;
int pid1;
int pid2;
        int pid3;
        //creación de la tubería
        pipe(tuberia);
        //Código del padre
        //ficheros descriptores
int nBytes,fd,fds;

        //estructura que recibirá información sobre el nodo remoto
struct hostent *he;
        //información sobre la dirección del servidor
struct sockaddr_in server;
        //llamada a gethostbyname()
if((he = gethostbyname(ipServer)) == NULL)
{
   printf("\nError al obtener el hostname\n");
   exit(-1);
}

//Definición del socket
        //llamada a socket()
if((fds = socket(AF_INET, SOCK_STREAM, 0)) == -1)
{
   printf("\nError en la definición del socket\n");
   exit(-1);
}

//Datos del servidor
server.sin_family = AF_INET;
server.sin_port = htons(port);
        //he->h_addr pasa la información de ``*he'' a "h_addr"
server.sin_addr = *((struct in_addr*)he->h_addr);
bzero(&(server.sin_zero), 8);

//Conexión con el servidor
if(connect(fds, (struct sockaddr*)&server, sizeof(struct sockaddr)) == -1)
{
   printf("\nError de conexión\n");
   exit(-1);
}
        //Comunicación con el servidor
/*if((nBytes = recv(fds, aux, MAX, 0)) == -1)
{
```

```
   perror("\nError de comunicación\n");
   exit(-1);
}*/

//aux[nBytes] = '\0';
        //Conectado con el servidor
//printf("\nConectado al servidor %d\n",fds);
        // CREACIÓN DEL HIJO1
        if((pid1 = fork()) == -1)
{
   perror("Error al crear el hijo1");
   exit(-1);
}//HIJO1
else if (pid1==0){
                int op=0;
                //se cierra la tuberia de lectura, la cual no se va a ocupar aqui
                close(tuberia[0]);
                while(op!=6){
                        printf("\n----------------Menú----------------\n");
                        printf("1 Crear un archivo");
                        printf("\n2 Eliminar un archivo");
                        printf("\n3 Crear un directorio");
                        printf("\n4 Eliminar un directorio");
                        printf("\n5 Renombrar un archivo o directorio");
                        printf("\n6 Salir");
                        printf("\nOpción: ");
                        scanf("%d", &op);
                        printf("\n");
                        //while (getchar() != '\n');
                        if(op==1){
                                //Se copia en id "1"=crear archivo
                                strcpy(id,"1");
                                //lo escribe en la tuberia
                                write(tuberia[1],id,strlen(id)+1);
                                printf("\nDame el nombre del archivo a crear: ");
                                //lee el nombre del nuevo archivo
                                scanf("%s",nombre);
                                write(tuberia[1],nombre,strlen(nombre)+1);
                                while(getchar() != '\n');
                                printf("%s,%s",id,nombre);
                        }else if(op==2){
                                strcpy(id,"2");
                                write(tuberia[1],id,strlen(id)+1);
                                printf("\nDame el nombre del archivo a eliminar: ");
                                scanf("%s",nombre);
                                write(tuberia[1],nombre,strlen(nombre)+1);
                                while(getchar() != '\n');
                                printf("%s,%s",id,nombre);
                        }else if(op==3){
                                strcpy(id,"3");
                                write(tuberia[1],id,strlen(id)+1);
                                printf("\nDame el nombre del directorio a crear: ");
                                scanf("%s",nombre);
                                write(tuberia[1],nombre,strlen(nombre)+1);
                                while(getchar() != '\n');
                                printf("%s,%s",id,nombre);
                        }else if(op==4){
                                strcpy(id,"4");
                                write(tuberia[1],id,strlen(id)+1);
```

```
                                        printf("\nDame el nombre del directorio a eliminar: ");
                                        scanf("%s",nombre);
                                        write(tuberia[1],nombre,strlen(nombre)+1);
                                        while(getchar() != '\n');
                                        printf("%s,%s",id,nombre);
                                }else if(op==5){
                                        strcpy(id,"5");
                                        write(tuberia[1],id,strlen(id)+1);
                                        printf("\nDame el nombre del archivo o directorio a renombrar: ");
                                        scanf("%s",nombre);
                                        write(tuberia[1],nombre,strlen(nombre)+1);
                                        while(getchar() != '\n');
                                        printf("\nDame el nuevo nombre: ");
                                        scanf("%s",nombre2);
                                        while(getchar() != '\n');
                                        write(tuberia[1],nombre2,strlen(nombre2)+1);
                                        printf("%s,%s,%s",id,nombre,nombre2);
                                }else if(op==6){
                                        strcpy(id,"6");
                                        write(tuberia[1],id,strlen(id)+1);
                                        strcpy(aux,path);
                                        write(tuberia[1],aux,strlen(aux)+1);
                                        printf("\nPrograma finalizado\n");
                                        kill(pid2,SIGTERM);
                                        kill(pid3,SIGTERM);
                                        exit(0);
                                }else{
                                        printf("\nOpción no válida");
                                }
                        }
                        //se cierra la tubería de escritura
                        close(tuberia[1]);
                        //exit(0);
                }else{
                        //PADRE
                        //CREACIÓN DEL HIJO2
                        if((pid2 = fork()) == -1)
{
   perror("fork");
   exit(-1);
}
                        //HIJO2
                        else if(pid2==0){
                                //cierra las tuberías porque no las va a ocupar

                                close(tuberia[1]);
                                int length, i = 0;
                        int fdi;
                        int wd;
                        char buffer[EVENT_BUF_LEN];
                                fdi = inotify_init();
                                /*checking for error*/
                                if ( fdi < 0 ) {
                                        perror( "inotify_init" );
                                }
                                /*adding the directory into watch list*/
                                wd = inotify_add_watch( fdi, path, IN_CREATE | IN_DELETE | IN_MOVED_FROM );
                                /*read to determine the event change happens on directory.*/
                                while(1){
```

```
                                i=0;
                                length = read( fdi, buffer, EVENT_BUF_LEN );
                                /*checking for error*/
                                if ( length < 0 ) {
                                        perror( "read" );
                                }
                                /*actually read return the list of change events happens. Here, read the change
event
                                one by one and process it accordingly.*/
                                while ( i < length ) {
                                        struct inotify_event *event = ( struct inotify_event * )& buffer[ i ];
                                        if ( event->len ) {
                                                if ( event->mask & IN_CREATE ) {
                                                        if ( event->mask & IN_ISDIR ) {
                                                                printf( "--------MENSAJE DEL HIJO2 DEL
CLIENTE--------\nNuevo directorio de nombre= %s creado.\n",
                                                                event->name );
                                                        }else{
                                                                printf( "--------MENSAJE DEL HIJO2 DEL
CLIENTE--------\nNuevo archivo de nombre=  %s creado.\n",
                                                                event->name );
                                                        }
                                                }else if ( event->mask & IN_DELETE ) {
                                                        if ( event->mask & IN_ISDIR ) {
                                                                printf( "--------MENSAJE DEL HIJO2 DEL
CLIENTE--------\nDirectorio %s eliminado.\n", event->name );
                                                        }else{
                                                                printf( "--------MENSAJE DEL HIJO2 DEL
CLIENTE--------\nArchivo %s eliminado.\n", event->name );
                                                        }
                                                }else if ( event->mask & IN_MOVED_FROM ) {
                                                        if ( event->mask & IN_ISDIR ) {
                                                                printf( "--------MENSAJE DEL HIJO2 DEL
CLIENTE--------\nDirectorio %s renombrado.\n", event->name );
                                                        }else{
                                                                printf( "--------MENSAJE DEL HIJO2 DEL
CLIENTE--------\nArchivo %s renombrado.\n", event->name );
                                                        }
                                                }
                                        }
                                        i += EVENT_SIZE + event->len;
                                }
                                }//while(read(tuberia[0],id,MAX)>0 && strcmp("6",id)==0);
                                /*Se quita el directorio de la lista observada*/
                                inotify_rm_watch( fdi, wd );
                                /*se cierra la instancia INOTIFY*/
                                close( fdi );
                                close(tuberia[0]);
                                exit(0);
                        }else{

                                //CREACIÓN DEL HIJO 3
                                if((pid3 = fork()) == -1)
        {
        perror("Error al crear el hijo 3");
        exit(-1);
        }
                                //HIJO3
                                else if(pid3==0){
```

```
                                        close(tuberia[0]);
                                        close(tuberia[1]);
                                while(1){
                                        int st;

                                        //Comunicación con el servidor
                                        if((st = recv(fds, aux, MAX, 0)) == -1)
        {
           //printf("\nA la espera de mensajes del servidor\n");
           //close(fds);
           //fds = -1;
           //exit(-1);
        }
        else if(st == 0)
        {
           fprintf(stdout, "\nServidor desconectado\n");
           close(fds);
           fds = -1;
           exit(-1);
        }
        else  //Descripción del mensaje a enviar
        {
           //aux[st] = '\0';
                                                        //CREAR ARCHIVO-----------------------------------------------
----------
           if(strcmp(aux, "1") == 0)
           {
                                                                //Comunicación con el servidor
              if((st = recv(fds, aux2, MAX, 0)) == -1)
              {
                 printf("\nError al recibir mensaje del servidor CA\n");
                 close(fds);
                 exit(-1);
              }
              else if(st == 0)
              {
                 printf("\nServidor desconectado\n");
                 close(fds);
                 exit(-1);
              }
              else
              {
                 //aux2[st] = '\0';
                 strcat(pathx,aux2);
                                                                creat(pathx,mode);
                                                                strcpy(pathx,argv[1]);

              }
                                                        //FIN CREAR ARCHIVO-------------------------------------------
----------------------

                                                        //ELIMINAR ARCHIVO--------------------------------------------
----------------------
           }
           else if(strcmp(aux, "2") == 0)
           {
                                                                //Comunicación con el servidor
              if((st = recv(fds, aux2, MAX, 0)) == -1)
              {
                 printf("\nError de comunicación con el servidor EA\n");
```

```
                    close(fds);
                    exit(-1);
                }
                else if(st == 0)
                {
                    printf("\nServidor desconectado\n");
                    close(fds);
                    exit(-1);
                }
                else
                {
                    //aux2[st] = '\0';
                    strcat(pathx,aux2);
                                                                    unlink(pathx);
                                                                    strcpy(pathx,argv[1]);
                }
                                                        //FIN ELIMINAR ARCHIVO-----------------------------------
---------------------------

                                                        //CREAR DIRECTORIO---------------------------------------
--------------------------
            }
            else if(strcmp(aux, "3") == 0)
            {
                if((st = recv(fds, aux2, MAX, 0)) == -1)
                {
                    printf("\nError de comunicación con el servidor CD\n");
                    close(fds);
                    exit(-1);
                }
                else if(st == 0)
                {
                    fprintf(stdout, "\nServidor desconectado\n");
                    close(fds);
                    exit(-1);
                }
                else
                {
                    //aux2[st] = '\0';
                    strcat(pathx,aux2);
                    mkdir(pathx,mode);
                    strcpy(pathx,argv[1]);
                }
                                                        //FIN CREAR DIRECTORIO-----------------------------------
----------------------

                                                        //ELIMINAR DIRECTORIO------------------------------------
---------------------------
            }
            else if(strcmp(aux, "4") == 0)
            {
                if((st = recv(fds, aux2, MAX, 0)) == -1)
                {
                    printf("\nError de comunicación con el servidor ED\n");
                    close(fds);
                    exit(-1);
                }
                else if(st == 0)
                {
```

```
                    fprintf(stdout, "\nServidor desconectado\n");
                    close(fds);
                    exit(-1);
                }
                else
                {
                    //aux2[st] = '\0';
                    strcat(pathx,aux2);
                    rmdir(pathx);
                    strcpy(pathx,argv[1]);
                }
                                                        //FIN ELIMINAR DIRECTORIO--------------------------------
-----------------------------

                                                        //RENOMBRAR ARCHIVO/DIRECTORIO--------------------
-------------------------------------
            }
            else if(strcmp(aux, "5") == 0)
            {
                if((st = recv(fds, aux2, MAX, 0)) == -1)
                {
                    printf("\nError de comunicación con el servidor R A/D\n");
                    close(fds);
                    exit(-1);
                }
                else if(st == 0)
                {
                    printf("\nServidor desconectado\n");
                    close(fds);
                    exit(-1);
                }
                else
                {
                    //aux2[st] = '\0';

                    if((st = recv(fds, aux3, MAX, 0)) == -1)
                    {
                        printf("\nError de comunicación con el servidor NAR\n");
                        close(fds);
                        exit(-1);
                    }
                    else if(st == 0)
                    {
                        fprintf(stdout, "\nServidor desconectado\n");
                        close(fds);
                        exit(-1);
                    }
                    else
                    {
                        //aux3[st] = '\0';
                                                                        strcat(path2,aux2);
                                                                        strcat(pathx,aux3);

                        rename(path2,pathx);
                        strcpy(path2,argv[1]);
                                                                        strcpy(pathx,argv[1]);

                    }
                }
            }//FIN RENOMBRAR ARCHIVO/DIRECTORIO-----------------------------------------------------
                                                        else{
```

```
                                                printf("\n%s...\n",aux);
                                        }
                }//fin else descripción de mensaje a enviar
        }
        //exit(0);
                        }else{
                                //PADRE
                                close(tuberia[1]);
                                //mientras haya algo que leer en la tubería
                                //printf("\nEstoy entrando al Código del padre\n");
                                while(read(tuberia[0],id,MAX)>0){
                                        printf("entre al while");
                                        //si id="1" entonces se crea un archivo con el nombre a
recuperar de la tuberia
                                        if(strcmp(id,"1")==0){
                                                //mientras se pueda ler el nombre
                                                while(read(tuberia[0],nombre,MAX)>0){
                                                        //printf("\nNombre del archivo:
%s\n",nombre);
                                                        strcat(pathx,nombre);
                                                        //se crea el archivo con los permisos dados
                                                        fd=creat(pathx,mode);
                                                        send(fds,id,MAX,0);
                                                        send(fds,nombre,MAX,0);
                                                        strcpy(pathx,argv[1]);
                                                        //mensaje por si falla la creación del
archivo
                                                        if(fd==-1)
                                                                printf("Error al crear el archivo");
                                                        close(fd);
                                                        break;
                                                }
                                        }else if(strcmp(id,"2")==0){
                                                while(read(tuberia[0],nombre,MAX)>0){
                                                        //printf("\nNombre del archivo:
%s\n",nombre);
                                                        strcat(pathx,nombre);
                                                        fd=unlink(pathx);
                                                        send(fds,id,MAX,0);
                                                        send(fds,nombre,MAX,0);
                                                        strcpy(pathx,argv[1]);
                                                        if(fd==-1)
                                                                printf("Error al eliminar el
archivo");
                                                        close(fd);
                                                        break;
                                                }
                                        }else if(strcmp(id,"3")==0){
                                                while(read(tuberia[0],nombre,MAX)>0){
                                                        //printf("\nNombre del archivo:
%s\n",nombre);
                                                        strcat(pathx,nombre);
                                                        fd=mkdir(pathx,mode);
                                                        send(fds,id,MAX,0);
                                                        send(fds,nombre,MAX,0);
                                                        strcpy(pathx,argv[1]);
                                                        if(fd==-1)
                                                                printf("Error al crear el
directorio");
```

```
                                                close(fd);
                                                break;
                                        }
                                }else if(strcmp(id,"4")==0){
                                        while(read(tuberia[0],nombre,MAX)>0){
                                                //printf("\nNombre del archivo:
%s\n",nombre);

                                                strcat(pathx,nombre);
                                                fd=rmdir(pathx);
                                                send(fds,id,MAX,0);
                                                send(fds,nombre,MAX,0);
                                                strcpy(pathx,argv[1]);
                                                if(fd==-1)
                                                        printf("Error al eliminar el
directorio");

                                                close(fd);
                                                break;
                                        }
                                }else if(strcmp(id,"5")==0){
                                        while(read(tuberia[0],nombre,MAX)>0){
                                                //printf("\nNombre del archivo:
%s\n",nombre);

                                                strcat(path2,nombre);
                                                while(read(tuberia[0],nombre2,MAX)>0){
                                                        //printf("\nNombre del archivo:
%s\n",nombre2);

                                                        strcat(pathx,nombre2);
                                                        fd=rename(path2,pathx);
                                                        send(fds,id,MAX,0);
                                                        send(fds,nombre,MAX,0);
                                                        send(fds,nombre2,MAX,0);
                                                        strcpy(path2,argv[1]);
                                                        strcpy(pathx,argv[1]);
                                                        if(fd==-1)
                                                                printf("Error al
renombrar el archivo");

                                                        close(fd);
                                                        break;
                                                }
                                                break;
                                        }
                                }else if(strcmp(id,"6")==0){
                                        while(read(tuberia[0],aux,MAX)>0){
                                                //pid=wait(NULL);
                                                //pid=wait(NULL);
                                                send(fds,id,MAX,0);
                                                kill(pid2,SIGTERM);
                                                kill(pid3,SIGTERM);
                                        }
                                }
                        }
                        kill(pid2,SIGTERM);
                        kill(pid3,SIGTERM);
                        close(tuberia[0]);
                }
        }
}
```

```
        return 0;
}
```

//////////////////////////Código del servidor////////////////////////////////

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <unistd.h>
#include <arpa/inet.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <sys/fcntl.h>
#include <sys/inotify.h>
#include <sys/types.h>
#include <sys/stat.h>
#include <sys/mman.h>
#include <sys/wait.h>
#define MAX 256
#define port 3553
#define EVENT_SIZE (sizeof(struct inotify_event))
#define EVENT_BUF_LEN (1024 * (EVENT_SIZE + 16))
#define BACKLOG 2


int main(int argc, char* argv[]){
        char path[50],path2[50];
        mode_t mode = S_IRUSR | S_IWUSR;

        if(argc != 2)
  {
    printf("\nForma de uso %s <carpeta>\nEjemplo %s /home/victor/Documents/\n",argv[0],argv[0]);
    exit(-1);
  }
  strcpy(path, argv[1]);
  strcpy(path2, argv[1]);
        //Hijo 1 (Proceso que va a monitorear la carpeta y avisa de cambios en los archivos)
        int pid1,pid2,pid3;

        //Rutas que va a recibir el programa donde hará las modificaciones correspondientes
        char aux[MAX];
  char aux2[MAX];
  char aux3[MAX];

  //Creación del hijo 1
  if((pid1 = fork()) == -1)
  {
    printf("\nError al crear el hijo 1\n");
    exit(-1);
  }
  else if (pid1 == 0)
  {
    //HIJO 1, MONITOR DE CARPETA

    int length, i = 0;
        int fd;
        int wd;
        char buffer[EVENT_BUF_LEN];
                fd = inotify_init();
```

```
                //checking for error
                if ( fd < 0 ) {
                        perror( "inotify_init" );
                }
                /*Agrega el directorio a la lista observada*/
                wd = inotify_add_watch( fd, path, IN_CREATE | IN_DELETE | IN_MOVED_FROM );
                /*lee todos los eventos que suceden en el directorio*/
                while(1){
                        i=0;
                        length = read( fd, buffer, EVENT_BUF_LEN );
                        /*checa por errores*/
                        if ( length < 0 ) {
                                printf( "\nError en el read\n" );
                        }
                        /*regresa la lista de cambios que suceden*/
                        while ( i < length ) {
                                struct inotify_event *event = ( struct inotify_event * )& buffer[ i ];
                                if ( event->len ) {
                                        if ( event->mask & IN_CREATE ) {
                                                if ( event->mask & IN_ISDIR ) {
                                                        printf( "--------MENSAJE DEL HIJO2 DEL
SERVIDOR--------\nNuevo directorio de nombre= %s creado.\n",
                                                        event->name );
                                                }else{
                                                        printf( "--------MENSAJE DEL HIJO2 DEL
SERVIDOR--------\nNuevo archivo de nombre=  %s creado.\n",
                                                        event->name );
                                                }
                                        }else if ( event->mask & IN_DELETE ) {
                                                if ( event->mask & IN_ISDIR ) {
                                                        printf( "--------MENSAJE DEL HIJO2 DEL
SERVIDOR--------\nDirectorio %s eliminado.\n", event->name );
                                                }else{
                                                        printf( "--------MENSAJE DEL HIJO2 DEL
SERVIDOR--------\nArchivo %s eliminado.\n", event->name );
                                                }
                                        }else if ( event->mask & IN_MOVED_FROM ) {
                                                if ( event->mask & IN_ISDIR ) {
                                                        printf( "--------MENSAJE DEL HIJO2 DEL
SERVIDOR--------\nDirectorio %s renombrado.\n", event->name );
                                                }else{
                                                        printf( "--------MENSAJE DEL HIJO2 DEL
SERVIDOR--------\nArchivo %s renombrado.\n", event->name );
                                                }
                                        }
                                }
                                i += EVENT_SIZE + event->len;
                        }
                }//while(read(tuberia[0],id,MAX)>0 && strcmp("6",id)==0);
                //Se quita el directorio de la lista observada
                inotify_rm_watch( fd, wd );
                //se cierra la instancia INOTIFY
                close( fd );
                //exit(0);
        }
        else//INICIO ELSE CÓDIGO PADRE
        {
                //PADRE CONFIGURA EL SERVIDOR
           int fds,fdcl1=-1,fdcl2=-1,sin_sizecl1,sin_sizecl2;
```

```
    struct sockaddr_in server, cl1,cl2;
    server.sin_family = AF_INET;
    server.sin_port = htons(port);
            /* INADDR_ANY coloca nuestra dirección IP automáticamente */
    server.sin_addr.s_addr = INADDR_ANY;
            /* escribimos ceros en el reto de la estructura */
    bzero(&(server.sin_zero), 8);
    //Definición del socket
    if((fds = socket(AF_INET, SOCK_STREAM, 0)) == -1)
    {
      printf("\nError en el socket\n");
      exit(-1);
    }
    /* A continuación la llamada a bind() */
    if(bind(fds, (struct sockaddr*)&server, sizeof(struct sockaddr)) == -1)
    {
      printf("\nError en el bind\n");
      exit(-1);
    }
    /* llamada a listen() */
    if(listen(fds, 5) == -1)
    {
      printf("\nError en el listen\n");
      exit(-1);
    }
    printf("\nSERVIDOR ON\n");

  sin_sizecl1=sizeof(struct sockaddr_in);
            if((fdcl1=accept(fds,(struct sockaddr*)&cl1, &sin_sizecl1))==-1)
                    printf("\nError en el accept\n");

            printf("\nCliente 1 se ha conectado: %s\n", (char*)inet_ntoa(cl1.sin_addr));
            printf("\nESPERANDO POR CLIENTE 2\n");

            sin_sizecl2=sizeof(struct sockaddr_in);
            if((fdcl2=accept(fds,(struct sockaddr*)&cl2, &sin_sizecl2))==-1)
                    printf("\nError en el accept\n");

            printf("\nCliente 2 se ha conectado: %s\n", (char*)inet_ntoa(cl2.sin_addr));
            printf("\nESPERANDO MENSAJES\n");
    //-------------------------------------------------------------------------------------------------------------------
    //while(1){
                    if((pid2 = fork()) == -1)
                    {
                            printf("\nError al crear el hijo 2\n");
                            exit(-1);
                    }
                    else if (pid2 == 0)
                    {
                            int st,fd;
                            //MANDA UN MENSAJE DE CONEXIÓN EXITOSA AL CLIENTE
                            //printf("\nCliente 1 se ha conectado: %s\n",
(char*)inet_ntoa(cl1.sin_addr));
                            //printf("\nCliente 2 se ha conectado: %s\n",
(char*)inet_ntoa(cl2.sin_addr));
                            //printf("\nESPERANDO POR CLIENTE 2\n");
                            /*strcpy(aux, "Conexion exitosa al servidor...");
  send(fdcl1, aux, MAX, 0);*/
  //send(fdcl2, aux, MAX, 0);
```

```
                                        while(1){//while(1) ciclo que se mantiene a la espera de mensajes del
cliente
                                                //send(fdcl2, "0", MAX, 0);
                                                if((st = recv(fdcl1, aux, MAX, 0)) == -1)
                                                {//inicio if st
                                                //printf("\nEsperando mensajes de cliente 1...\n");
                                                        close(fdcl1);
                                                        fdcl1 = -1;
                                                        exit(-1);
                                        }//fin if st
                                        else if(st == 0)
                                        {//inicio else if st
                                                printf("\nEl cliente 1 %s se ha desconectado\n",
(char*)inet_ntoa(cl1.sin_addr));
                                                close(fdcl1);
                                                fdcl1 = -1;
                                                exit(-1);
                                        }//fin if st
                                        else
                                        {//inicio else
                                                send(fdcl2, aux, MAX, 0);
                                                //-------------------------------------------------------------------------------
-------------------------------
                                                //INICIO CREAR ARCHIVO
                                                if(strcmp(aux, "1") == 0)
                                                {//inicio if(strcmp(aux,"1")==0)
                                                        if((st = recv(fdcl1, aux2, MAX, 0)) == -1)
                                                        {//inicio if ((st = recv(fdcl1, aux2, MAX, 0)) == -1)
                                                                printf("\nError en la comunicación con
cliente 1 al recibir el segundo mensaje crear archivo\n");
                                                                close(fdcl1);
                                                                fdcl1 = -1;
                                                                exit(-1);
                                                        }//fin if ((st = recv(fdcl1, aux2, MAX, 0)) == -1)
                                                        else if(st == 0)
                                                        {//inicio else if(st ==0)
                                                                printf("\nEl cliente 1 %s se ha
desconectado...\n", (char*)inet_ntoa(cl1.sin_addr));
                                                                close(fdcl1);
                                                                fdcl1 = -1;
                                                                exit(-1);
                                                        }//fin else if(st ==0)
                                                        else
                                                        {//inicio del else
                                                                send(fdcl2, aux2, MAX, 0);
                                                                printf("\nRecibido de cliente 1 en crear
archivo: %s,%s\n",aux,aux2);
                                                  //aux2[st] = '\0';
                                                  strcpy(path,argv[1]);
                                                                        strcat(path,aux2);
                                                                        printf("\nCrear archivo en
%s",path);
                                                                        creat(path,mode);
                                                        }//fin del else
                                                }//fin if(strcmp(aux,"1")==0)
                                                //FIN CREAR ARCHIVO
                                                //-------------------------------------------------------------------------------
-------------------------------
                                                //INICIO ELIMINAR ARCHIVO
```

```
                                        else if(strcmp(aux, "2") == 0)
                                        {//inicio if(strcmp(aux,"2")==0)
                                                if((st = recv(fdcl1, aux2, MAX, 0)) == -1)
                                                {//inicio if ((st = recv(fdcl1, aux2, MAX, 0)) == -1)
                                                        printf("\nError en la comunicación con
cliente 1 al recibir el segundo mensaje eliminar archivo\n");
                                                        close(fdcl1);
                                                        fdcl1 = -1;
                                                        exit(-1);
                                                }//fin if ((st = recv(fdcl1, aux2, MAX, 0)) == -1)
                                                else if(st == 0)
                                                {//inicio else if(st ==0)
                                                        printf("\nEl cliente 1 %s se ha
desconectado...\n", (char*)inet_ntoa(cl1.sin_addr));
                                                        close(fdcl1);
                                                        fdcl1 = -1;
                                                        exit(-1);
                                                }//fin else if(st ==0)
                                                else
                                                {//inicio del else
                                                        send(fdcl2, aux2, MAX, 0);
                                                        printf("\nRecibido de cliente 1 en eliminar
archivo: %s,%s\n",aux,aux2);
                                        //aux2[st] = '\0';
                                        strcpy(path,argv[1]);
                                                                strcat(path,aux2);
                                        printf("\nEliminar archivo en %s",path);
                                                                unlink(path);
                                                }//fin del else
                                         }//fin else if(strcmp(aux,"2")==0)
                                         //FIN ELIMINAR ARCHIVO
                                         //-------------------------------------------------------------------------------
-------------------------------
                                         //INICIO CREAR CARPETA
                                         else if(strcmp(aux, "3") == 0)
                                         {//inicio if(strcmp(aux,"3")==0)
                                                if((st = recv(fdcl1, aux2, MAX, 0)) == -1)
                                                {//inicio if ((st = recv(fdcl1, aux2, MAX, 0)) == -1)
                                                        printf("\nError en la comunicación con
cliente 1 al recibir el segundo mensaje crear carpeta\n");
                                                        close(fdcl1);
                                                        fdcl1 = -1;
                                                        exit(-1);
                                                }//fin if ((st = recv(fdcl1, aux2, MAX, 0)) == -1)
                                                else if(st == 0)
                                                {//inicio else if(st ==0)
                                                        printf("\nEl cliente 1 %s se ha
desconectado...\n", (char*)inet_ntoa(cl1.sin_addr));
                                                        close(fdcl1);
                                                        fdcl1 = -1;
                                                        exit(-1);
                                                }//fin else if(st ==0)
                                                else
                                                {//inicio del else
                                                        send(fdcl2, aux2, MAX, 0);
                                                        printf("\nRecibido de cliente 1 en crear
carpeta: %s,%s\n",aux,aux2);
                                        //aux2[st] = '\0';
                                        strcpy(path,argv[1]);
```

```
                                                                strcat(path,aux2);
                                                                printf("\nCrear carpeta en %s\n",
path);
                                                                fd=mkdir(path,mode);
                                                        }//fin del else
                                                }//fin else if(strcmp(aux,"3")==0)
                                                //FIN CREAR CARPETA
                                                //-------------------------------------------------------------------------------
-------------------------------
                                                //INICIO ELIMINAR CARPETA
                                                else if(strcmp(aux, "4") == 0)
                                                {//inicio if(strcmp(aux,"4")==0)
                                                        if((st = recv(fdcl1, aux2, MAX, 0)) == -1)
                                                        {//inicio if ((st = recv(fdcl1, aux2, MAX, 0)) == -1)
                                                                printf("\nError en la comunicación con
cliente 1 al recibir el segundo mensaje eliminar carpeta\n");
                                                                close(fdcl1);
                                                                fdcl1 = -1;
                                                                exit(-1);
                                                        }//fin if ((st = recv(fdcl1, aux2, MAX, 0)) == -1)
                                                        else if(st == 0)
                                                        {//inicio else if(st ==0)
                                                                printf("\nEl cliente 1 %s se ha
desconectado...\n", (char*)inet_ntoa(cl1.sin_addr));
                                                                close(fdcl1);
                                                                fdcl1 = -1;
                                                                exit(-1);
                                                        }//fin else if(st ==0)
                                                        else
                                                        {//inicio del else
                                                                send(fdcl2, aux2, MAX, 0);
                                                                printf("\nRecibido de cliente 1 en eliminar
carpeta: %s,%s\n",aux,aux2);
                                                        //aux2[st] = '\0';
                                                        strcpy(path,argv[1]);
                                                                strcat(path,aux2);
                                                                printf("\nEliminar carpeta en
%s\n", path);
                                                                rmdir(path);
                                                        }//fin del else
                                                }//fin else if(strcmp(aux,"4")==0)
                                                //FIN ELIMINAR CARPETA
                                                //-------------------------------------------------------------------------------
-------------------------------
                                                //INICIO RENOMBRAR ARCHIVO/CARPETA
                                                else if(strcmp(aux, "5") == 0)
                                                {//inicio if(strcmp(aux,"5")==0)
                                                        if((st = recv(fdcl1, aux2, MAX, 0)) == -1)
                                                        {//inicio if ((st = recv(fdcl1, aux2, MAX, 0)) == -1)
                                                                printf("\nError en la comunicación con
cliente 1 al recibir el segundo mensaje renombrar arch/carp\n");
                                                                close(fdcl1);
                                                                fdcl1 = -1;
                                                                exit(-1);
                                                        }//fin if ((st = recv(fdcl1, aux2, MAX, 0)) == -1)
                                                        else if(st == 0)
                                                        {//inicio else if(st ==0)
                                                                printf("\nEl cliente1 %s se ha
desconectado...\n", (char*)inet_ntoa(cl1.sin_addr));
```

```
                                                close(fdcl1);
                                                fdcl1 = -1;
                                                exit(-1);
                                        }//fin else if(st ==0)
                                        else
                                        {//inicio del else
                                                send(fdcl2, aux2, MAX, 0);
                                                if((st = recv(fdcl1, aux3, MAX, 0)) == -1)
                                                        {//inicio if ((st = recv(fdcl1, aux3,
MAX, 0)) == -1)
                                                                printf("\nError en la
comunicación con cliente 1 al recibir el tercer mensaje\n");
                                                                close(fdcl1);
                                                                fdcl1 = -1;
                                                                exit(-1);
                                                        }//fin if ((st = recv(fdcl1, aux3,
MAX, 0)) == -1)
                                                        else if(st == 0)
                                                        {//inicio else if(st ==0)
                                                                printf("\nEl cliente 1 %s
se ha desconectado...\n", (char*)inet_ntoa(cl1.sin_addr));
                                                                close(fdcl1);
                                                                fdcl1 = -1;
                                                                exit(-1);
                                                        }//fin else if(st ==0)
                                                        else
                                                        {//inicio del else
                                                                send(fdcl2, aux3, MAX,
0);
                                                                printf("\nRecibido de
cliente 1 en renombrar archivo/carpeta: %s,%s,%s\n",aux,aux2,aux3);
                                                //aux2[st] = '\0';
                                                strcpy(path,argv[1]);
                                                                strcpy(path2,argv[1]);
                                                                strcat(path,aux2);
                                                                strcat(path2,aux3);
                                                                printf("\nRenombrar
arch/carp %s,%s\n", path,path2);
                                                                rename(path,path2);
                                                        }//fin del else
                                                }//fin del else
                                        }//fin else if(strcmp(aux,"5")==0)
                                        //FIN RENOMBRAR ARCHIVO/CARPETA
                                        //-------------------------------------------------------------------------------
-------------------------------
                                        else if(strcmp(aux,"6"))
                                        {
                                                send(fdcl2, aux, MAX, 0);
                                                kill(pid1,SIGTERM);
                                                kill(pid2,SIGTERM);
                                                kill(pid3,SIGTERM);
                                        }
                                }//fin else
                        }//fin segundo ciclo while(1)
                }//fin else if(pid2==0)
                else
                {//INICIA CÓDIGO DEL PADRE
                        //ENTRA CÓDIGO PARA OTRO CLIENTE
                        //while(1){
```

```
/*sin_sizecl2=sizeof(struct sockaddr_in);
if((fdcl2=accept(fds,(struct sockaddr*)&cl2, &sin_sizecl2))==-1)
printf("\nError en el accept\n");*/
if((pid3 = fork()) == -1)
{
        printf("\nError al crear el hijo 3\n");
        exit(-1);
}
else if (pid3 == 0)
{
        int st,fd;
        //MANDA UN MENSAJE DE CONEXIÓN EXITOSA AL
CLIENTE
        //printf("\nCliente 2 se ha conectado: %s\n",
(char*)inet_ntoa(cl2.sin_addr));
        //printf("\nESPERANDO MENSAJES\n");
        /*strcpy(aux, "Conexion exitosa al servidor...");
send(fdcl2, aux, MAX, 0);*/
        while(1){//while(1) ciclo que se mantiene a la espera
de mensajes del cliente
                //send(fdcl1, "0", MAX, 0);
                if((st = recv(fdcl2, aux, MAX, 0)) == -1)
                {//inicio if st
                        //printf("\nEsperando mensajes
de cliente2...\n");
                        close(fdcl2);
                        fdcl2 = -1;
                        exit(-1);
                }//fin if st
                else if(st == 0)
                {//inicio else if st
                        printf("\nEl cliente 2 %s se ha
desconectado\n", (char*)inet_ntoa(cl2.sin_addr));
                        close(fdcl2);
                        fdcl2 = -1;
                        exit(-1);
                }//fin if st
                else
                {//inicio else
                        send(fdcl1, aux, MAX, 0);
                        //-------------------------------------
---------------------------------------------------------------
                        //INICIO CREAR ARCHIVO
                        if(strcmp(aux, "1") == 0)
                        {//inicio if(strcmp(aux,"1")==0)
                                if((st = recv(fdcl2, aux2,
MAX, 0)) == -1)
                                {//inicio if ((st =
recv(fdcl2, aux2, MAX, 0)) == -1)

        printf("\nError en la comunicación con cliente 1 al recibir el segundo mensaje crear archivo\n");
                                        close(fdcl2);
                                        fdcl2 = -1;
                                        exit(-1);
                                }//fin if ((st = recv(fdcl2,
aux2, MAX, 0)) == -1)
                                else if(st == 0)
                                {//inicio else if(st ==0)
```

```
                                                                printf("\nEl
cliente 2 %s se ha desconectado...\n", (char*)inet_ntoa(cl2.sin_addr));
                                                                close(fdcl2);
                                                                fdcl2 = -1;
                                                                exit(-1);
                                                        }//fin else if(st ==0)
                                                        else
                                                        {//inicio del else
                                                                send(fdcl1,
aux2, MAX, 0);

        printf("\nRecibido de cliente 2 en crear archivo: %s,%s\n",aux,aux2);
                                                                //aux2[st] = '\0';

        strcpy(path,argv[1]);

        strcat(path,aux2);
                                                                printf("Crear
archivo en %s",path);

        creat(path,mode);
                                                                }//fin del else
                                                        }//fin if(strcmp(aux,"1")==0)
                                                        //FIN CREAR ARCHIVO
                                                        //-----------------------------------------
-------------------------------------------------------------------
                                                        //INICIO ELIMINAR ARCHIVO
                                                        else if(strcmp(aux, "2") == 0)
                                                        {//inicio if(strcmp(aux,"2")==0)
                                                                if((st = recv(fdcl2, aux2,
MAX, 0)) == -1)
                                                                {//inicio if ((st =
recv(fdcl2, aux2, MAX, 0)) == -1)

        printf("\nError en la comunicación con cliente 2 al recibir el segundo mensaje eliminar archivo\n");
                                                                close(fdcl2);
                                                                fdcl2 = -1;
                                                                exit(-1);
                                                        }//fin if ((st = recv(fdcl2,
aux2, MAX, 0)) == -1)
                                                        else if(st == 0)
                                                        {//inicio else if(st ==0)
                                                                printf("\nEl
cliente 2 %s se ha desconectado...\n", (char*)inet_ntoa(cl2.sin_addr));
                                                                close(fdcl2);
                                                                fdcl2 = -1;
                                                                exit(-1);
                                                        }//fin else if(st ==0)
                                                        else
                                                        {//inicio del else
                                                                send(fdcl1,
aux2, MAX, 0);

        printf("\nRecibido de cliente 2 en eliminar archivo: %s,%s\n",aux,aux2);
                                                                //aux2[st] = '\0';

        strcpy(path,argv[1]);

        strcat(path,aux2);
```

```
        printf("\nEliminar archivo en %s",path);
                                                                    unlink(path);
                                                                }//fin del else
                                                            }//fin else if(strcmp(aux,"2")==0)
                                                            //FIN ELIMINAR ARCHIVO
                                                            //-----------------------------------------
-----------------------------------------------------------------
                                                            //INICIO CREAR CARPETA
                                                            else if(strcmp(aux, "3") == 0)
                                                            {//inicio if(strcmp(aux,"3")==0)
                                                                if((st = recv(fdcl2, aux2,
MAX, 0)) == -1)
                                                                {//inicio if ((st =
recv(fdcl2, aux2, MAX, 0)) == -1)

        printf("\nError en la comunicación con cliente 2 al recibir el segundo mensaje crear carpeta\n");
                                                                    close(fdcl2);
                                                                    fdcl2 = -1;
                                                                    exit(-1);
                                                                }//fin if ((st = recv(fdcl2,
aux2, MAX, 0)) == -1)
                                                                else if(st == 0)
                                                                {//inicio else if(st ==0)
                                                                    printf("\nEl
cliente 2 %s se ha desconectado...\n", (char*)inet_ntoa(cl2.sin_addr));
                                                                    close(fdcl2);
                                                                    fdcl2 = -1;
                                                                    exit(-1);
                                                                }//fin else if(st ==0)
                                                                else
                                                                {//inicio del else
                                                                    send(fdcl1,
aux2, MAX, 0);

        printf("\nRecibido de cliente 2 en crear carpeta: %s,%s\n",aux,aux2);
                                                                    //aux2[st] = '\0';

        strcpy(path,argv[1]);

        strcat(path,aux2);

        printf("\nCrear carpeta en %s",path);

        mkdir(path,mode);
                                                                }//fin del else
                                                            }//fin else if(strcmp(aux,"3")==0)
                                                            //FIN CREAR CARPETA
                                                            //-----------------------------------------
-----------------------------------------------------------------
                                                            //INICIO ELIMINAR CARPETA
                                                            else if(strcmp(aux, "4") == 0)
                                                            {//inicio if(strcmp(aux,"4")==0)
                                                                if((st = recv(fdcl2, aux2,
MAX, 0)) == -1)
                                                                {//inicio if ((st =
recv(fdcl2, aux2, MAX, 0)) == -1)

        printf("\nError en la comunicación con cliente 2 al recibir el segundo mensaje eliminar carpeta\n");
```

```
                                                                close(fdcl2);
                                                                fdcl2 = -1;
                                                                exit(-1);
                                                        }//fin if ((st = recv(fdcl2,
aux2, MAX, 0)) == -1)
                                                        else if(st == 0)
                                                        {//inicio else if(st ==0)
                                                                printf("\nEl
cliente 2 %s se ha desconectado...\n", (char*)inet_ntoa(cl2.sin_addr));
                                                                close(fdcl2);
                                                                fdcl2 = -1;
                                                                exit(-1);
                                                        }//fin else if(st ==0)
                                                        else
                                                        {//inicio del else
                                                                send(fdcl1,
aux2, MAX, 0);

        printf("\nRecibido de cliente 2 en eliminar carpeta: %s,%s\n",aux,aux2);
                                                                  //aux2[st] = '\0';

        strcpy(path,argv[1]);

        strcat(path,aux2);

        printf("\nEliminar carpeta en %s",path);
                                                                rmdir(path);
                                                        }//fin del else
                                                }//fin else if(strcmp(aux,"4")==0)
                                                //FIN ELIMINAR CARPETA
                                                //-----------------------------------------
---------------------------------------------------------------------
                                                //INICIO RENOMBRAR
ARCHIVO/CARPETA
                                                else if(strcmp(aux, "5") == 0)
                                                {//inicio if(strcmp(aux,"5")==0)
                                                        if((st = recv(fdcl2, aux2,
MAX, 0)) == -1)
                                                        {//inicio if ((st =
recv(fdcl2, aux2, MAX, 0)) == -1)

        printf("\nError en la comunicación con cliente 2 al recibir el segundo mensaje renombrar
arch/carp\n");
                                                                close(fdcl2);
                                                                fdcl2 = -1;
                                                                exit(-1);
                                                        }//fin if ((st = recv(fdcl2,
aux2, MAX, 0)) == -1)
                                                        else if(st == 0)
                                                        {//inicio else if(st ==0)
                                                                printf("\nEl
cliente 2 %s se ha desconectado...\n", (char*)inet_ntoa(cl2.sin_addr));
                                                                close(fdcl2);
                                                                fdcl2 = -1;
                                                                exit(-1);
                                                        }//fin else if(st ==0)
                                                        else
                                                        {//inicio del else
```

```
                                                                    send(fdcl1,
aux2, MAX, 0);
                                                                    if((st =
recv(fdcl2, aux3, MAX, 0)) == -1)
                                                                    {//inicio if ((st
= recv(fdcl2, aux3, MAX, 0)) == -1)
        printf("\nError en la comunicación con cliente 2 al recibir el tercer mensaje\n");
        close(fdcl2);
        fdcl2 = -1;
        exit(-1);
                                                                    }//fin if ((st =
recv(fdcl2, aux3, MAX, 0)) == -1)
                                                                    else if(st == 0)
                                                                    {//inicio else
if(st ==0)
        printf("\nEl cliente 2 %s se ha desconectado...\n", (char*)inet_ntoa(cl2.sin_addr));
        close(fdcl2);
        fdcl2 = -1;
        exit(-1);
                                                                    }//fin else if(st
==0)
                                                                    else
                                                                    {//inicio del
else
        send(fdcl1, aux3, MAX, 0);
        printf("\nRecibido de cliente 2 en renombrar archivo/carpeta: %s,%s,%s\n",aux,aux2,aux3);
                                                                            //aux2[st] = '\0';
        strcpy(path,argv[1]);
        strcpy(path2,argv[1]);
        strcat(path,aux2);
        strcat(path2,aux3);
        printf("\nRenombrar arch/carp %s,%s\n", path,path2);
        rename(path,path2);
                                                                    }//fin del else
                                                                }//fin del else
                                                            }//fin else if(strcmp(aux,"4")==0)
                                                            //FIN RENOMBRAR
ARCHIVO/CARPETA
                                                            //-----------------------------------------
-----------------------------------------------------------------
                                                            else if(strcmp(aux,"6"))
                                                        {
```

```
                                                                    kill(pid1,SIGTERM);
                                                                        kill(pid2,SIGTERM);
                                                                        kill(pid3,SIGTERM);
                                                                        send(fdcl2, aux, MAX,
0);
                                                                    }
                                                                    }//fin else
                                                                }//fin segundo ciclo while(1)
                                                            }//fin else if(pid2==0)
                                                        //}//fin primer while(1)
                                                    }//FINALIZA CÓDIGO DEL PADRE
                                                //}//fin primer while(1)

   }//FIN ELSE CODIGO PADRE


            return 0;
}
```

4. Un diagrama ilustrativo de la arquitectura del programa implementado