INSTITUTO POLITÉCNICO NACIONAL
ESCUELA SUPERIOR DE CÓMPUTO
SISTEMAS OPERATIVOS
GRUPO: 2CM8

ALUMNO:
NAVA VIVAS ANA PAOLA

ENVIO DE FOTO

Esta práctica consistía en realizar el envío de algún archivo a travéz de sockets.

Cliente:

```c
#include <sys/socket.h>
#include <sys/types.h>
#include <netinet/in.h>
#include <netdb.h>
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <unistd.h>
#include <errno.h>
#include <arpa/inet.h>

int main(int argc,char * argv[])
{
  if(argc!=4){printf("USO: [cliente] <puerto> <IP servidor> <archivo>\n");exit(1);}
  int socket_id;
  int bytes = 0;
  char recvBuff[256];
  struct sockaddr_in serv_addr;

  memset(recvBuff, '0', sizeof(recvBuff));

  if((socket_id = socket(AF_INET, SOCK_STREAM, 0))==-1){perror("Socket");exit(1);}

  serv_addr.sin_family = AF_INET;
  serv_addr.sin_port = htons(atoi(argv[1]));
  serv_addr.sin_addr.s_addr = inet_addr(argv[2]);

  if(connect(socket_id, (struct sockaddr *)&serv_addr, sizeof(serv_addr))<0)
    {perror("Connect");exit(1);}

  FILE *fp;
  fp = fopen(argv[3], "wb");
  if(NULL == fp){printf("Error opening file");exit(1);}
```

El cliente hace la petición al servidor para que le envíe una imagen. Este una vez con el socket listo recibe la imagen y la guarda en un bufer que en turno la guarda directo en un archivo nuevo.

```c
  if(NULL == fp){printf("Error opening file");exit(1);}

  while((bytes = read(socket_id, recvBuff, 256)) > 0)
    {
      fwrite(recvBuff, 1,bytes,fp);
    }

  if(bytes < 0) perror("Lectura");


  return 0;
}
```

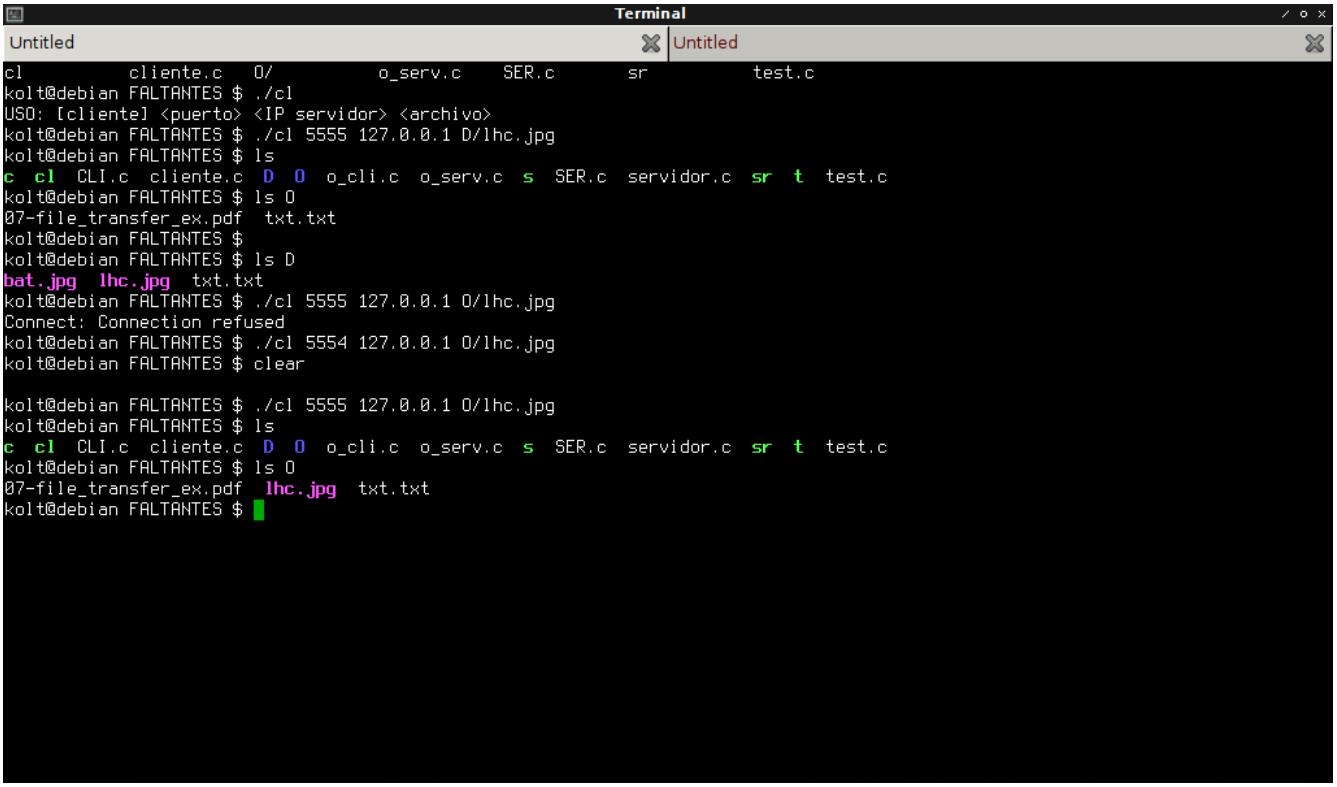
El programa detecta cuando el archivo ya no está escribiendo paquetes completos, lo que significa que no está recibiendo más datos:

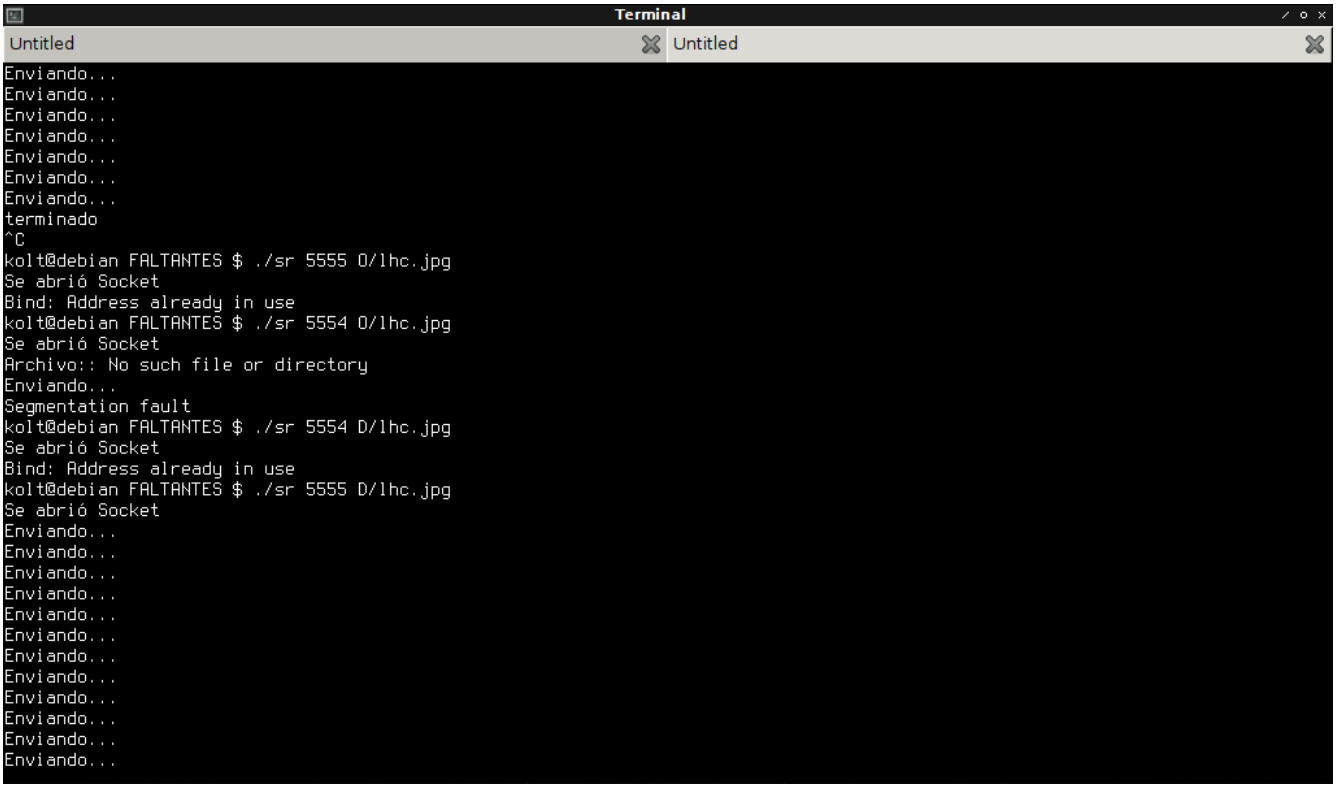
Servidor:

El servidor “resuelve la petición del cliente y busca la imagen que le pide para así fragmentarla. Se puede pensar como un proceso inverso al de recibir los datos.

```c
#include <netinet/in.h>
#include <arpa/inet.h>
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <errno.h>
#include <string.h>
#include <sys/types.h>

#define BACKLOG 5

int main(int argc, char *argv[])
{
  if(argc!=3){printf("USO: [servidor] <puerto> <archivo>\n");exit(1);}
  int socket_id;
  int id_canal;
  struct sockaddr_in servidor;
  char sendBuff[1025];
  int numrv;
  unsigned char buff[256];

  if((socket_id = socket(AF_INET, SOCK_STREAM, 0))==-1){perror("Socket");exit(1);}

  printf("Se abrió Socket\n");

  //Se inicializan variables
  memset(&servidor, '0', sizeof(servidor));
  memset(sendBuff, '0', sizeof(sendBuff));
  memset(buff, '0', sizeof(buff));
  //Se inicializa estructura
  servidor.sin_family = AF_INET;
  servidor.sin_addr.s_addr = htonl(INADDR_ANY);
  servidor.sin_port = htons(atoi(argv[1]));
```

Después el servidor cierra el archivo de donde está copiando los datos.


Terminal
Untitled
Untitled
cl cliente.c O/ o_serv.c SER.c sr test.c
kolt@debian FALTANTES $ ./cl
USO: [cliente] <puerto> <IP servidor> <archivo>
kolt@debian FALTANTES $ ./cl 5555 127.0.0.1 D/lhc.jpg
kolt@debian FALTANTES $ ls
c cl CLI.c cliente.c D O o_cli.c o_serv.c s SER.c servidor.c sr t test.c
kolt@debian FALTANTES $ ls O
07-file_transfer_ex.pdf txt.txt
kolt@debian FALTANTES $
kolt@debian FALTANTES $ ls D
bat.jpg lhc.jpg txt.txt
kolt@debian FALTANTES $ ./cl 5555 127.0.0.1 O/lhc.jpg
Connect: Connection refused
kolt@debian FALTANTES $ ./cl 5554 127.0.0.1 O/lhc.jpg
kolt@debian FALTANTES $ clear
kolt@debian FALTANTES $ ./cl 5555 127.0.0.1 O/lhc.jpg
kolt@debian FALTANTES $ ls
c cl CLI.c cliente.c D O o_cli.c o_serv.c s SER.c servidor.c sr t test.c
kolt@debian FALTANTES $ ls O
07-file_transfer_ex.pdf lhc.jpg txt.txt
kolt@debian FALTANTES $


Terminal
Untitled
Untitled
Enviando...
Enviando...
Enviando...
Enviando...
Enviando...
Enviando...
Enviando...
terminado
^C
kolt@debian FALTANTES $ ./sr 5555 O/lhc.jpg
Se abrió Socket
Bind: Address already in use
kolt@debian FALTANTES $ ./sr 5554 O/lhc.jpg
Se abrió Socket
Archivo:: No such file or directory
Enviando...
Segmentation fault
kolt@debian FALTANTES $ ./sr 5554 D/lhc.jpg
Se abrió Socket
Bind: Address already in use
kolt@debian FALTANTES $ ./sr 5555 D/lhc.jpg
Se abrió Socket
Enviando...
Enviando...
Enviando...
Enviando...
Enviando...
Enviando...
Enviando...
Enviando...
Enviando...
Enviando...
Enviando...
Enviando...

CONCLUSIÓN:

Es importante hacer notar que ambos archivos fueron abiertos, escritos, leídos y cerrados en forma binaria. Esto se hace porque en modo texto pueden existir caracteres que el sistema operativo no reconozca. Es muy útil tener conocimientos sobre envío de datos a través de los sockets, ya que son el principio de las comunicaciones que utilizamos y que no nos percatamos ni de cómo funcionan la mayor parte del tiempo.