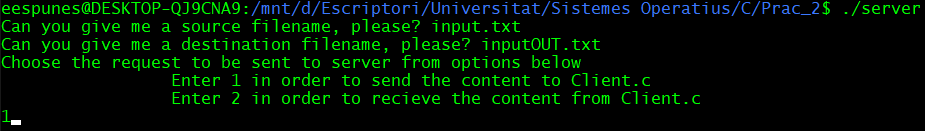
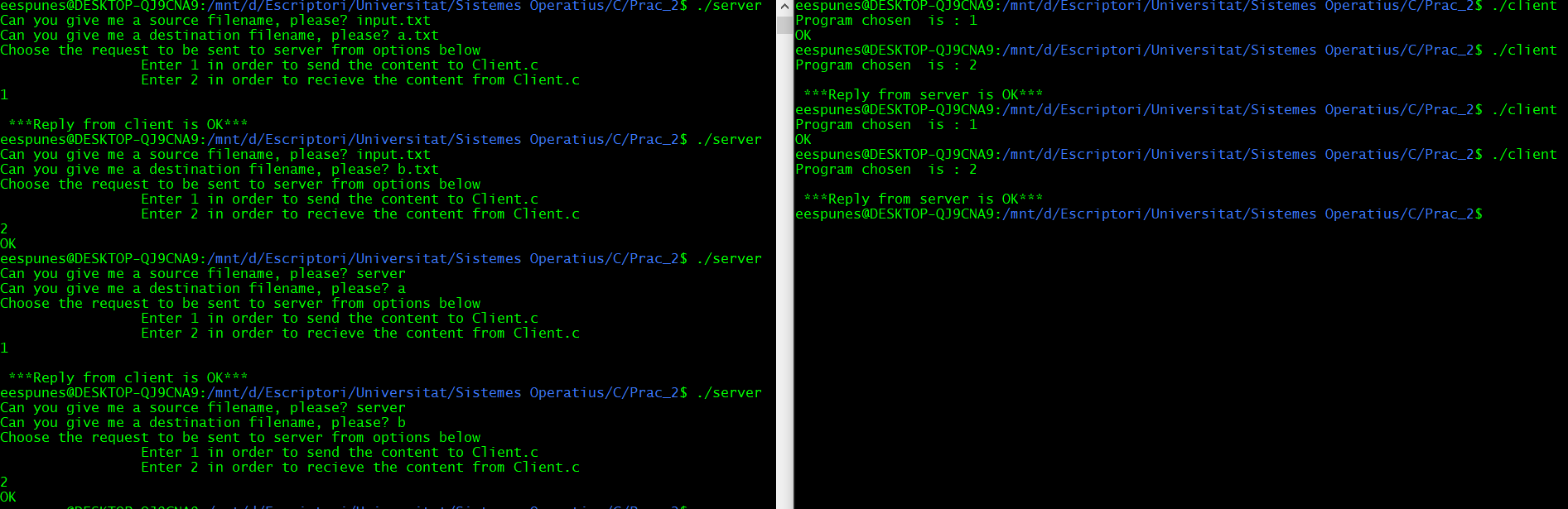
**How to Execute**

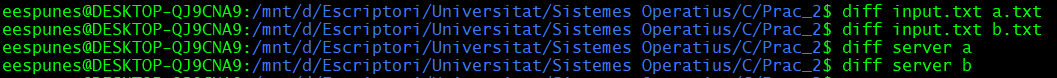
In order to execute the programs, first, you have to open two terminals in your Raspberry or in Ubuntu. In one of them, you’ll execute the Server and in the other the client.

When you have done that you execute Server.c when you enter all the data to the server(like the image below),

Then you execute Client.c.

Finally, you’ll have all done.

**EXECUTION**



**Server.c**

**CODE**

//Authors: Sofian Ben Ayata && Erik Espuñes Juberó  
  
#include <stdio.h>  
#include <stdlib.h>  
#include <string.h>  
#include <fcntl.h>  
#include <sys/stat.h>  
#include <sys/types.h>  
#include <unistd.h>  
  
**int** main() {  
 **char** name\_fifo\_server[32] = "/tmp/fifo\_server", name\_fifo\_client[32] = "/tmp/fifo\_client";  
 **int** fifo\_server=-1, fifo\_client=-1, file=-1;  
 **char** srcfilename[100]="", dstfilename[100]="";  
 **int** choice=0;  
  
 //Ask for source filename, destination filename and which program use  
 printf("Can you give me a source filename, please? ");  
 scanf("%s", srcfilename);  
 printf("Can you give me a destination filename, please? ");  
 scanf("%s", dstfilename);  
  
 printf("Choose the request to be sent to server from options below");  
 printf("\n\t\t Enter 1 in order to send the content to Client.c\n \  
 Enter 2 in order to recieve the content from Client.c\n");  
 scanf("%d", &choice);  
  
 //delete the previous pipes  
 unlink(name\_fifo\_server);  
 unlink(name\_fifo\_client);  
 **if** (choice == 1 || choice == 2) {  
 //if the program chose ins between 1 or 2 means that are the correct values to choose the program, and creates the two pipes, one for to read from server and the other to read from client  
 fifo\_server = mkfifo(name\_fifo\_server, 0666);  
 **if** (fifo\_server < 0) {  
 perror("Unable to create a fifo\_server");  
 exit(-1);  
 }  
 fifo\_client = mkfifo(name\_fifo\_client, 0666);  
 **if** (fifo\_client < 0) {  
 perror("Unable to create a fifo\_client");  
 exit(-1);  
 }  
 //We open the pipes  
 fifo\_server = open(name\_fifo\_server, O\_RDWR);  
 **if** (fifo\_server < 0) {  
 perror("Error in opening fifo\_server");  
 exit(-1);  
 }  
 fifo\_client = open(name\_fifo\_client, O\_RDWR);  
  
 //We write the chosen program  
 write(fifo\_client, &choice, 1);  
  
 **if** (fifo\_client < 0) {  
 perror("Error in opening fifo\_client");  
 exit(-1);  
 }  
 **if** (choice == 1) {  
 //If program is 1 we open the source file and send the destination filename to the client  
 **if** ((file = open(srcfilename, O\_RDONLY)) < 0) {  
 perror("Error! opening source file from Server.c");  
 exit(-1);  
 }  
 write(fifo\_client, dstfilename, **sizeof**(dstfilename));  
 //We write to the client pipe the information that it's in the source file  
 **char** line[256];  
 **int** size;  
 **while** ((size=read(file, line, **sizeof**(line))) > 0) {  
 write(fifo\_client, line, size\* **sizeof**(**char**));  
 }  
 //We wait for the result from client  
 **char** \*buf = malloc(10 \* **sizeof**(**char**));  
 read(fifo\_server, buf, 10 \* **sizeof**(**char**));  
 printf("\n \*\*\*Reply from client is %s\*\*\*\n", buf);  
 } **else** {  
 //else it's the other program and we open destination file and we send to the client the source filename  
 **if** ((file = open(dstfilename, O\_WRONLY|O\_CREAT|O\_TRUNC)) < 0) {  
 perror("Error! opening destination file from Server.c");  
 exit(-1);  
 }  
  
 write(fifo\_client, srcfilename, **sizeof**(srcfilename));  
 //We write to the destination file what's inside the server pipe  
  
 **char** line[256]="";  
 **int** size;  
 **while** ((size=read(fifo\_server, line, **sizeof**(line))) > 0) {  
 write(file, line, size\* **sizeof**(**char**));  
 **if**(size<256)  
 **break**;  
 }  
 write(fifo\_client, "OK", 10 \* **sizeof**(**char**));  
 printf("OK\n");  
 }  
 close(file);  
 close(fifo\_server);  
 close(fifo\_client);  
 } **else** {  
 perror("Error! the number must be 1 or 2");  
 exit(1);  
 }  
 **return** 0;  
}

**Client.c**

**CODE**

//Authors: Sofian Ben Ayata && Erik Espuñes Juberó  
  
#include <stdio.h>  
#include <stdlib.h>  
#include <string.h>  
#include <fcntl.h>  
#include <sys/stat.h>  
#include <sys/types.h>  
#include <unistd.h>  
  
**int** main() {  
 **char** name\_fifo\_server[32] = "/tmp/fifo\_server", name\_fifo\_client[32] = "/tmp/fifo\_client", filename[100];  
 **int** fifo\_server=-1, fifo\_client=-1, file=-1;  
 **int** choice = 0;  
 **char** \*buf = "OK";  
  
 //We open the pipes  
 fifo\_client = open(name\_fifo\_client, O\_RDWR);  
 **if** (fifo\_client < 0) {  
 perror("Error opening fifo\_client");  
 exit(-1);  
 }  
 fifo\_server = open(name\_fifo\_server, O\_RDWR);  
 **if** (fifo\_server < 0) {  
 perror("Error opening fifo\_server");  
 }  
 //Read the program chosen and the filename  
 read(fifo\_client, &choice, 1);  
 read(fifo\_client, filename, **sizeof**(filename));  
 printf("Program chosen is : %d\n", choice);  
 **if** (choice == 1) {  
 //if it's program 1 we open destination file  
 **if** ((file = open(filename, O\_WRONLY|O\_CREAT|O\_TRUNC)) < 0) {  
 buf = "ERROR";  
 write(fifo\_client, buf, **sizeof**(buf));  
 perror("Error! opening destination file from Client.c");  
 exit(-1);  
 }  
 //we write to the destination file what it's inside the pipe  
 **char** line[256]="";  
 **int** size;  
 **while** (((size=read(fifo\_client, line, **sizeof**(line))) > 0)) {  
 write(file, line, size\* **sizeof**(**char**));  
 **if**(size<256)  
 **break**;  
 }  
 printf("%s\n", buf);  
 //We tell the server if the program was successful or wrong  
 write(fifo\_server, buf, **sizeof**(buf));  
 } **else** {  
 //else it's program 2 and we open source file  
 **if** ((file = open(filename, O\_RDONLY)) < 0) {  
 buf = "ERROR";  
 write(fifo\_server, buf, **sizeof**(buf));  
 perror("Error! opening source file from Client.c");  
 exit(-1);  
 }  
 //we write to the server pipe, what's in the source file  
 **char** line[256]="";  
 **int** size;  
 **while** ((size=read(file, line, **sizeof**(line))) > 0) {  
 write(fifo\_server, line, size\* **sizeof**(**char**));  
 }  
 //Finally we wait to the Server result  
 buf = malloc(10 \* **sizeof**(**char**));  
 read(fifo\_client, buf, 10 \* **sizeof**(**char**));  
 printf("\n \*\*\*Reply from server is %s\*\*\*\n", buf);  
 }  
 close(file);  
 close(fifo\_server);  
 close(fifo\_client);  
 **return** 0;  
}