

----server----

```
#include<stdio.h>
#include<stdlib.h>
#include<unistd.h>
#include<sys/types.h>
#include<sys/types.h>
#include<fcntl.h>
#include<string.h>

int main()
{
    int n;
    puts("Server");
    char strMessage[5000];//[] = {"welcome", "to", "the", "module.", "This",
"will", "now", "stop"};
    int fd = open("fifo6.txt", O_WRONLY);
    int fd2 = open ("fifo7.txt", O_RDONLY);
    if(fd == -1)
    {
        perror("cannot open fifo6");
        return EXIT_FAILURE;
    }
    if(fd2 == -1)
    {
        perror("cannot open fifo7");
        return EXIT_FAILURE;
    }
    puts("FIFO OPEN");
    //read string up to(5000 characters)
    char stringBuffer[5000];
    memset(stringBuffer, 0, 5000);
    int res;
    char Len;
    //while(1)
    {
        printf("\n\n\tEnter the Message to be passed (hitting ENTER
without any string will terminate program): ");
        fgets(strMessage, 100, stdin);
        char L = (char) strlen(strMessage);
        //printf("\n\tLength of the given string: %d\n", (L-1));

        write(fd, &L, 1);
        write(fd, strMessage, strlen(strMessage));
        fflush(stdin);
        strMessage[0] = 0;//reseting the character array
        //if(L==1)//since null counts 1
        //    //break;
        int len2;
        res = read(fd2, &len2, 1);
        //if(len2 == 1)//since null counts 1
        //    //break;

        read(fd2, stringBuffer, 5000); //Read String Characters

        printf("\nServer Received: %s\n", stringBuffer);
        stringBuffer[(int)len2] = 0;
    };
    //printf("\n\nCLIENT CLOSED\n")
    //return 0;
}
```

----client----

```

#include<stdio.h>
#include<stdlib.h>
#include<sys/types.h>
#include<sys/stat.h>
#include<unistd.h>
#include<fcntl.h>
#include<string.h>

int main()
{
    puts("\n\tClient - Listening\n");
    int code6 = mkfifo("fifo6.txt", 0666);
    int code7 = mkfifo("fifo7.txt", 0666);
    char strMessage[5000];
    if(code6 == -1)
        perror("\n\tmkfifo6 returned an error-file any already exist\n");
    if(code7 == -1)
        perror("\n\tmkfifo7 returned an error-file any already exist\n");

    int fd = open("fifo6.txt", O_RDONLY);
    int fd2 = open("fifo7.txt", O_WRONLY);
    if(fd == -1)
    {
        perror("Cannot open FIFO6 for read");
        return EXIT_FAILURE;
    }
    if(fd2 == -1)
    {
        perror("Cannot open FIFO7 for write");
        return EXIT_FAILURE;
    }
    puts("FIFO OPEN");
    //read string up to(5000 characters)
    char stringBuffer[5000];
    memset(stringBuffer, 0, 5000);

    int res;
    char Len;
    //while(1)
    {
        res = read(fd, &Len, 1);
        //if(Len == 1)//since null counts 1
        //break;

        read(fd, stringBuffer, Len); //Read String Characters
        stringBuffer[(int)Len] = 0;
        printf("\nClient Received: %s\n", stringBuffer);
        int j = 0, w=0, line = 0;
        while(stringBuffer[j]!='\0'){
            char ch = stringBuffer[j];
            if((ch==' ')||(ch=='\n')){
                w++;
                if(ch=='\n')
                    line++;
            }
            j++;
        }
        char LC = (char) strlen(strMessage);
        char str1[256];
        char str2[256];
        char str3[256];
        sprintf(str1, " No.of Words : %d:::", w); strcat(strMessage,str1);
        sprintf(str2, " No.of Charecters: %d:::", (j-1));
        strcat(strMessage,str2);
    }
}

```

```

        sprintf(str3," No.of Lines: %d",line); strcat(strMessage,str3);
        strcat(strMessage,"\"0");
            printf("\n\tString: %s",strMessage);
        write(fd2, &LC, 1);
        write(fd2, strMessage, strlen(strMessage));
        fflush(stdin);
        strMessage[0] = 0;//reseting the character array
        //if(LC==1)
        //break;

    }
    printf("\n");
    puts("CLIENT CLOSED");
    puts("SERVER CLOSED");
    close(fd);
    close(fd2);
    return 0;
}

```

OUTPUT:

```

avccoe@avccoe-HP-ProDesk-400-G1-SFF:~/Pr7/FIFO/fifo$ gcc server.c
avccoe@avccoe-HP-ProDesk-400-G1-SFF:~/Pr7/FIFO/fifo$ ./a.out
Server
FIFO OPEN

```

Enter the Message to be passed (hitting ENTER without any string will terminate program): hello

Server Received: No.of Words : 1::: No.of Charecters: 5::: No.of Lines: 1

```

avccoe@avccoe-HP-ProDesk-400-G1-SFF:~/Pr7/FIFO/fifo$ gcc client.c
avccoe@avccoe-HP-ProDesk-400-G1-SFF:~/Pr7/FIFO/fifo$ ./a.out

```

Client - Listening

FIFO OPEN

Client Received: hello

```

String: No.of Words : 1::: No.of Charecters: 5::: No.of Lines: 1
CLIENT CLOSED
SERVER CLOSED

```