

OS Lab 3

1

Server side

```
student@HP-EliteDesk-800-G8-15: ~/os
student@HP-EliteDesk-800-G8-15:~/os$ gcc fifoserver.c -o ser
student@HP-EliteDesk-800-G8-15:~/os$ ./ser
Received string: "Hi there" and length is 8
Received string: "end" and length is 3
student@HP-EliteDesk-800-G8-15:~/os$
```

Client side

```
student@HP-EliteDesk-800-G8-15: ~/os
student@HP-EliteDesk-800-G8-15:~/os$ gcc fifoclient.c -o cli
student@HP-EliteDesk-800-G8-15:~/os$ ./cli
FIFO_CLIENT: Send messages, infinitely, to end enter "end"
Enter string: Hi there
Sent string: "Hi there" and string length is 8
Enter string: How was your day
Sent string: "How was your day" and string length is 16
Enter string: emd
Sent string: "emd" and string length is 3
Enter string: end
Sent string: "end" and string length is 3
student@HP-EliteDesk-800-G8-15:~/os$
```

mknod is a system call which is used to create a special file which is used for named pipe with name MYFIFO.

An infinite while loop is used which terminates only when “end” is typed.

The client program sends a string, the server program receives the string and counts the number of characters in the string and sends the value back to the client. The client prints the string length.

Server file

```

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

#define FIFO_FILE "MYFIFO"
int main(){
    int fd;
    char readbuf[80];
    char end[10];
    int to_end;
    int read_bytes;
    char sbuf[80];
    int stringlen;
    int end_process;

    /*Create the FIFO if it does not exist*/
    mknod(FIFO_FILE, __S_IFIFO|0640,0);
    strcpy(end, "end");
    while (1)
    {
        fd= open(FIFO_FILE, O_RDWR);
        read_bytes= read(fd, readbuf, sizeof(readbuf));
        readbuf[read_bytes]='\0';
        printf("Received string: \"%s\" and length is %d\n", readbuf, (int)strlen(readbuf));
        for (int i=0;i<(int)strlen(readbuf);i++){
            if(readbuf[i]>='a' && readbuf[i]<='z'){
                sbuf[i]=readbuf[i]-32;}
            else{
                sbuf[i]=readbuf[i];}}

        to_end=strcmp(readbuf, end);
        if (to_end==0){
            close(fd);
            break;
        }
    }
}

```

```

    stringlen = strlen(sbuf);
    sbuf[stringlen]='\0';
    end_process=strcmp(sbuf,end);
    //printf("end process is %d\n",end_process);
    if(end_process!=0){
        write(fd,sbuf,strlen(sbuf));
        printf("Sent string: \"%s\" and string length is %d \n", sbuf, (int)strlen(sbuf));
        memset(readbuf,0,strlen(readbuf));
        memset(sbuf,0,strlen(sbuf));
    }
    else{
        write(fd,readbuf,strlen(readbuf));
        printf("Sent string: \"%s\" and string length is %d \n", sbuf, (int) strlen(sbuf));
        close(fd);
        break;
    }
}

return 0;

}

```

Client file

```

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

#define FIFO_FILE "MYFIFO"
int main(){
    int fd;
    int end_process;
    int stringlen;
    char readbuf[80];
    char end_str[5];
    int fd1;
    int read_bytes;
    char sbuf[80];
    printf("FIFO_CLIENT: Send messages , infinitely to end enter\"end\\n");

```

```

fd=open(FIFO_FILE ,O_CREAT|O_RDWR);
strcpy(end_str,"end");
while(1){
    printf("Enter string:");
    fgets(readbuf, sizeof(readbuf),stdin);
    stringlen = strlen(readbuf);
    readbuf[stringlen-1]='\0';
    end_process=strcmp(readbuf,end_str);
    //printf("end process is %d\n",end_process);
    if(end_process!=0){
        write(fd,readbuf,strlen(readbuf));
        printf("Sent string: \"%s\" and string length is %d \n", readbuf, (int)strlen(readbuf));
    }
    else{
        write(fd,readbuf,strlen(readbuf));
        printf("Sent string: \"%s\" and string length is %d \n", readbuf, (int) strlen(readbuf));
        close(fd);
        break;
    }

    read_bytes= read(fd, sbuf, sizeof(sbuf));
    sbuf[read_bytes]='\0';
    printf("Received string: \"%s\" and length is %d\n", sbuf, (int)strlen(sbuf)); }

    return 0;
}

```

Server side

```

student@HP-EliteDesk-800-G8-15: ~/os
student@HP-EliteDesk-800-G8-15:~/os$ gcc fifoserver.c -o ser
student@HP-EliteDesk-800-G8-15:~/os$ ./ser
Received string: "Hi there" and length is 8
Sent string: "HI THERE" and string length is 8
Received string: "How are you" and length is 11
Sent string: "HOW ARE YOU" and string length is 11
Received string: "end" and length is 3
student@HP-EliteDesk-800-G8-15:~/os$ 

```

Client side

```

student@HP-EliteDesk-800-G8-15: ~/os
student@HP-EliteDesk-800-G8-15:~/os$ gcc fifoclient.c -o cli
student@HP-EliteDesk-800-G8-15:~/os$ ./cli
FIFO_CLIENT: Send messages , infinitely to end enter"end"
Enter string:Hi there
Sent string: "Hi there" and string length is 8
Received string: "HI THERE" and length is 8
Enter string:How are you
Sent string: "How are you" and string length is 11
Received string: "HOW ARE YOU" and length is 11
Enter string:end
Sent string: "end" and string length is 3
student@HP-EliteDesk-800-G8-15:~/os$ 

```

Server file

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

#define FIFO_FILE "MYFIFO"
int main(){
    int fd;
    char readbuf[80];
    char end[10];
    int to_end;
    int read_bytes;
    char sbuf[80];
    int stringlen;
    int end_process;
    char message[50];
    int i=0;
    char ch;

    /*Create the FIFO if it does not exist*/
    mknod(FIFO_FILE, __S_IFIFO|0640,0);

    fd= open(FIFO_FILE, O_RDWR);
    read_bytes= read(fd, readbuf, sizeof(readbuf));
    readbuf[read_bytes]='\0';
    printf("Received filename: %s\n", readbuf);

    FILE *fptr;
    fptr = fopen(readbuf, "r");
    if (fptr == NULL)
    { printf("file can't be opened \n");
      strcpy(message,"file can't be opened.\n");
      write(fd, message, strlen(message) + 1);
      close(fd);
      exit(0);}
    else { while ((ch = fgetc(fptr)) != EOF)
```

```

        { message[i] = ch; i++; }
        write(fd, message, strlen(message) + 1); }
        fclose(fptr);

return 0;

}

```

Client file

```

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

#define FIFO_FILE "MYFIFO"
int main(){
    int fd;
    int end_process;
    int stringlen;
    char readbuf[80];
    char end_str[5];
    int fd1;
    int read_bytes;
    char sbuf[80];

    fd=open(FIFO_FILE ,O_CREAT|O_RDWR);

    printf("Enter filename:");
    fgets(readbuf, sizeof(readbuf),stdin);
    stringlen = strlen(readbuf);
    readbuf[stringlen-1]='\0';
    write(fd,readbuf,strlen(readbuf));
    printf("Sent file name: \"%s\"\n",readbuf);
}

```

```

read_bytes= read(fd, sbuf, sizeof(sbuf));
sbuf[read_bytes]='\0';
printf("Received string: %s", sbuf);

return 0;
}

```

Server side

```

student@HP-EliteDesk-800-G8-13:~/Desktop/New Folder 3$ ./ser
Received filename: hello.txt
file can't be opened

```

Server side

```

student@HP-EliteDesk-800-G8-13:~/Desktop/New Folder 3$ ./cli
Enter filename:hello.txt
Sent file name: "hello.txt"
Received string: file can't be opened.
student@HP-EliteDesk-800-G8-13:~/Desktop/New Folder 3$

```

Server side

```

student@HP-EliteDesk-800-G8-15: ~/os
student@HP-EliteDesk-800-G8-15:~/os$ gcc fifoserver.c -o ser
student@HP-EliteDesk-800-G8-15:~/os$ ./ser
Received filename: hello.txt
student@HP-EliteDesk-800-G8-15:~/os$

```

Client side

```

student@HP-EliteDesk-800-G8-15: ~/os
student@HP-EliteDesk-800-G8-15:~/os$ gcc fifoclient.c -o cli
student@HP-EliteDesk-800-G8-15:~/os$ ./cli
Enter filename:hello.txt
Sent file name: "hello.txt"
Received string: Hi there
How are youstudent@HP-EliteDesk-800-G8-15:~/os$

```

hello.txt



