



VASIREDDY VENKATADRI INSTITUTE OF TECHNOLOGY

Permanently Affiliated to JNTU Kakinada, Approved by AICTE
Accredited by NAAC with 'A' Grade, ISO 9001:2008 Certified
Nambur, Pedakakani (M), Guntur (Dt) - 522508

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
B.Tech Program is Accredited by NBA

CERTIFICATE

Name of the Lab : OPERATING SYSTEMS
Name of the Student : Thota Nagababu
Student Regd. No. : 18BQ1A05K3
CLASS : III B.TECH. I SEM CSE - D
GithubLink: <https://github.com/nagababuthota984/5K3-OS-LAB>



VASIREDDY VENKATADRI INSTITUTE OF TECHNOLOGY

Permanently Affiliated to JNTU Kakinada, Approved by AICTE

Accredited by NAAC with 'A' Grade, ISO 9001:2008 Certified

Nambur, Pedakakani (M), Guntur (Dt) - 522508

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

B.Tech Program is Accredited by NBA

INDEX

S. No	Name of the Experiment	LAST DATE	PAGE NO
5C	INTER PROCESS COMMUNICATION USING FIFO	JAN 12	3-6

Experiment 5c

AIM: Interprocess communication using fifo.

DESCRIPTION:

A named pipe (also known as a **FIFO**) is one of the methods for intern-process communication.

- ▯ It is an extension to the traditional pipe concept on Unix. A traditional pipe is “unnamed” and lasts only as long as the process.
- ▯ A named pipe, however, can last as long as the system is up, beyond the life of the process. It can be deleted if no longer used.
- ▯ Usually a named pipe appears as a file and generally processes attach to it for inter-process communication. A FIFO file is a special kind of file on the local storage which allows two or more processes to communicate with each other by reading/writing to/from this file.
- ▯ A FIFO special file is entered into the filesystem by calling `mkfifo()` in C. Once we have created a FIFO special file in this way, any process can open it for reading or writing, in the same way as an ordinary file. However, it has to be open at both ends simultaneously before you can proceed to do any input or output operations on it.

LIBRARIES:

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

CODE:

Server:

```
#include<stdio.h>
#include<unistd.h>
#include<fcntl.h>
#include<sys/types.h>
#include<string.h>
#include<sys/stat.h>
int main()
{
    char fname[25]="";
    char fcontent[100]="";
    int fd,fd1,fd2;
    mkfifo("fifo1",0600);
    mkfifo("fifo2",0600);
    fd=open("fifo1",O_RDONLY);
    fd1=open("fifo2",O_WRONLY);
    read=(fd,fname,25);
    fd2=open(fname,O_RDONLY);
    while(read(fd2,fcontent,100)!=0)
    {
        printf("%s\n",fcontent);
        if(fd<0)
            write(fd1,"file notexit",14);
        else
```

```

        write(fd1,fcontent,strlen(fcontent));
    }
    close(fd);
    close(fd1);
    close(fd2);
}

```

Client:

```

#include<stdio.h>
#include<stdlib.h>
#include<errno.h>
#include<fcntl.h>
#include<sys/types.h>
#include<unistd.h>
#include<sys/stat.h>
int main()
{
    char s[100]=" ";
    char s1[1000]=" ";
    int fd,fd1;
    fd=open("fifo1",O_WRONLY);
    fd1=open("fifo2",O_RDONLY);
    printf("\nenter file name:");
    scanf("%s",s);
    write(fd,s1,strlen(s));
    while(read(fd1,s1,1000)!=0)
    {
        printf("file content :%s",s1);
    }
}

```

```
}  
}
```

OUTPUT:

Step1 :run server

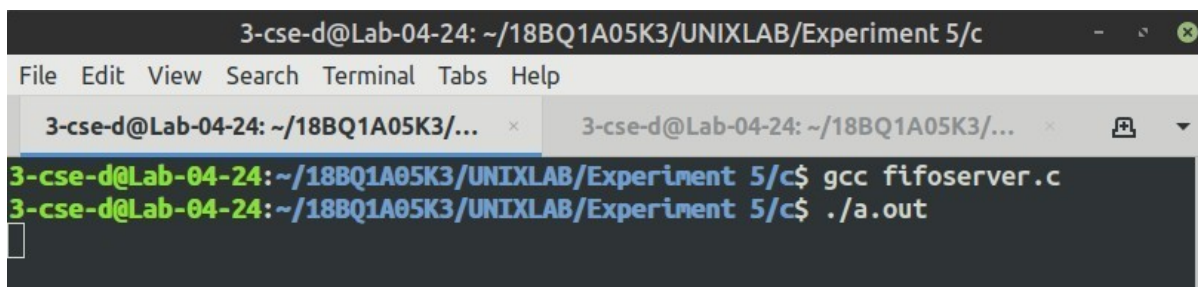
Step 2:Run client

Step 3: give filename in client

Step 4: server opens that file for us and we can see the content of the file given in client.

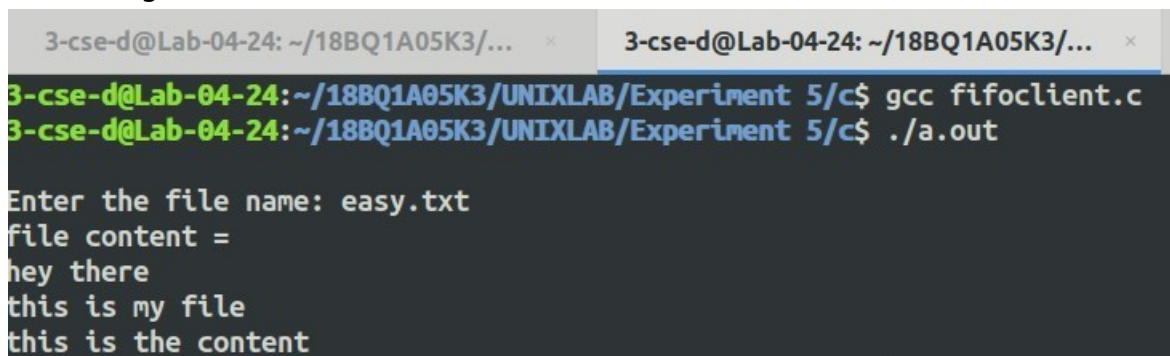
OUTPUT SCREEN SHOTS:

Executing server



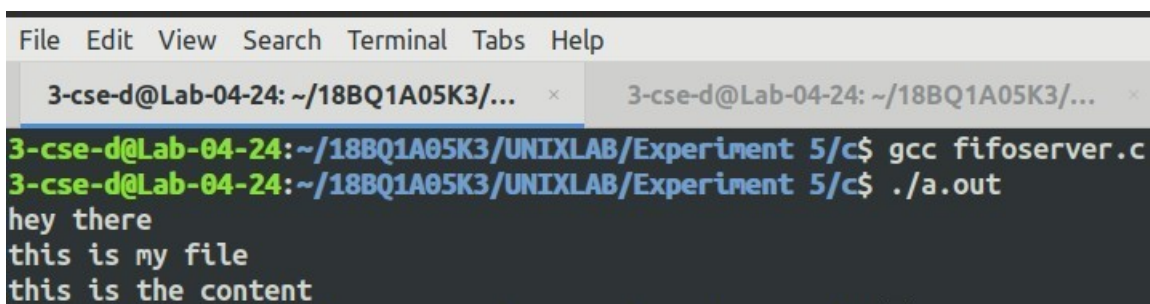
```
3-cse-d@Lab-04-24: ~/18BQ1A05K3/UNIXLAB/Experiment 5/c  
File Edit View Search Terminal Tabs Help  
3-cse-d@Lab-04-24: ~/18BQ1A05K3/... x 3-cse-d@Lab-04-24: ~/18BQ1A05K3/... x  
3-cse-d@Lab-04-24:~/18BQ1A05K3/UNIXLAB/Experiment 5/c$ gcc fifoserver.c  
3-cse-d@Lab-04-24:~/18BQ1A05K3/UNIXLAB/Experiment 5/c$ ./a.out
```

Executing client



```
3-cse-d@Lab-04-24: ~/18BQ1A05K3/... x 3-cse-d@Lab-04-24: ~/18BQ1A05K3/... x  
3-cse-d@Lab-04-24:~/18BQ1A05K3/UNIXLAB/Experiment 5/c$ gcc fifoclient.c  
3-cse-d@Lab-04-24:~/18BQ1A05K3/UNIXLAB/Experiment 5/c$ ./a.out  
  
Enter the file name: easy.txt  
file content =  
hey there  
this is my file  
this is the content
```

Server showing the file which we gave input to the client.



```
File Edit View Search Terminal Tabs Help  
3-cse-d@Lab-04-24: ~/18BQ1A05K3/... x 3-cse-d@Lab-04-24: ~/18BQ1A05K3/... x  
3-cse-d@Lab-04-24:~/18BQ1A05K3/UNIXLAB/Experiment 5/c$ gcc fifoserver.c  
3-cse-d@Lab-04-24:~/18BQ1A05K3/UNIXLAB/Experiment 5/c$ ./a.out  
hey there  
this is my file  
this is the content
```