**Develop a C program which demonstrates interprocess communication between a reader process and a writer process. Use mkfifo, open, read, write and close APIs in your program.**

In computing, a named pipe (also known as a FIFO) is one of the methods for inter-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.
* 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. Creating a FIFO file: In order to create a FIFO file, a function calls i.e. mkfifo is used.

**int mkfifo(const char \*pathname, mode\_t mode);**

mkfifo() makes a FIFO special file with name **pathname**. Here **mode** specifies the FIFO’s permissions. It is modified by the processes unmask in the usual way: the permissions of the created file are (mode & ~unmask).

**Using FIFO:** As named pipe(FIFO) is a kind of file, we can use all the system calls associated with it i.e. open, read, write, close.

**Example Programs to illustrate the named pipe:** There are two programs that use the same FIFO. Program 1 writes first, then reads. The program 2 reads first, then write. They both keep doing it until terminated.

**P1: Write First**

#include <stdio.h>

#include <string.h>

#include <fcntl.h>

#include <sys/stat.h>

#include <sys/types.h>

#include <unistd.h>

int main()

{

int fd;

char \* myfifo = "/tmp/myfifo";

mkfifo(myfifo, 0666);

char arr1[80], arr2[80];

while (1)

{

fd = open(myfifo, O\_WRONLY);

fgets(arr2, 80, stdin);

write(fd, arr2, strlen(arr2)+1);

close(fd);

fd = open(myfifo, O\_RDONLY);

read(fd, arr1, sizeof(arr1));

printf("User2: %s\n", arr1);

close(fd);

}

return 0;

}

**P2: Read first**

#include <stdio.h>

#include <string.h>

#include <fcntl.h>

#include <sys/stat.h>

#include <sys/types.h>

#include <unistd.h>

int main()

{

int fd1;

char \* myfifo = "/tmp/myfifo";

mkfifo(myfifo, 0666);

char str1[80], str2[80];

while (1)

{

fd1 = open(myfifo,O\_RDONLY);

read(fd1, str1, 80);

printf("User1: %s\n", str1);

close(fd1);

fd1 = open(myfifo,O\_WRONLY);

fgets(str2, 80, stdin);

write(fd1, str2, strlen(str2)+1);

close(fd1);

}

return 0;

}

