Output

PIPE

Hello, World #1
Hello, World #2
Hello, World #3

o/p Verified

Lab : 2

21/1/2020

INTERPROCESS COMMUNICATION USING PIPE & FIFO

Aim

To implement programs for Inter Process Communication using PIPE, FIFO.

Program

PIPE

```
#include <stdio.h>
#include <unistd.h>
#include <stdlib.h>
#define MSGSIZE 16
char * msg1 = "hello, world #1";
char * msg2 = "hello, world #2";
char * msg3 = "hello, world #3";
int main()
{
    char inbuf[MSGSIZE];
    int p[2], i;
    if (pipe(p) < 0)
        exit(1);
    write(p[1], msg1, MSGSIZE);
    write(p[1], msg2, MSGSIZE);
    write(p[1], msg3, MSGSIZE);
    for(i=0; i<3; i++)
    {
        read(p[0], inbuf, MSGSIZE);
        printf("%s\n", inbuf);
    }
    return 0;
}
```

FIFO

Fifo 1	Fifo 2
Hello	User1: Hello
User2: Hey	Hey
What's up?	User1: What's up?
User2: Good	Good

o/p Verified

FIFO

Fifo 1

```
#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("User 2: %s\n", arr1);
        close(fd);
    }
    return 0;
}
```

Fifo 2

```
#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";
    char str1[80], str2[80];
    while (1)
    {
        fd1= open(myfifo, O_RDONLY);
        read (fd1, str1,80);
        printf("User 1: %s\n",str1);
        close (fd1);
        fd1= open(myfifo, O_WRONLY);
        fgets (str2,80, stdin);
        write (fd1, str2, strlen(str2)+1);
        close (fd1);
    }
    return 0;
}
```

Result

Implemented programs for InterProcess Communication using PIPE, FIFO.

80
23/1/2020