

## Task 6

**Task:** To write a C program to implement concept of Inter Process Communication

### //Program 1 (Writes first, then reads)

// C program to implement one side of FIFO (Writes first, then reads)

```
#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"; // FIFO file path
```

```
    // Create the named FIFO file with read-write permissions
```

```
    mkfifo(myfifo, 0666);
```

```
    char writeMsg[80], readMsg[80];
```

```
    while (1) {
```

```
        // Open FIFO for write only
```

```
        fd = open(myfifo, O_WRONLY);
```

```
        printf("User1: Enter message: ");
```

```

fgets(writeMsg, 80, stdin); // Take input from the user

// Write input string to FIFO and close it
write(fd, writeMsg, strlen(writeMsg) + 1);

close(fd);

// Open FIFO for Read only
fd = open(myfifo, O_RDONLY);

// Read from FIFO
read(fd, readMsg, sizeof(readMsg));

// Print the received message
printf("User2: %s\n", readMsg);

close(fd);
}

return 0;
}

```

## **//Program 2 (Reads first, then writes)**

```

// C program to implement one side of FIFO (Reads first, then writes)

#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"; // FIFO file path


    // Create the named FIFO file with read-write permissions
    mkfifo(myfifo, 0666);

    char readMsg[80], writeMsg[80];

    while (1) {

        // Open FIFO for Read only
        fd = open(myfifo, O_RDONLY);

        // Read message from FIFO
        read(fd, readMsg, sizeof(readMsg));

        // Print the received message
        printf("User1: %s\n", readMsg);

        close(fd);

        // Open FIFO for write only
        fd = open(myfifo, O_WRONLY);

        printf("User2: Enter message: ");
```

```
fgets(writeMsg, 80, stdin); // Take input from the user

// Write input string to FIFO and close it
write(fd, writeMsg, strlen(writeMsg) + 1);

close(fd);

}

return 0;

}
```

**FIFO (Named Pipe)** is an **Inter-Process Communication (IPC)** mechanism that allows processes to exchange data using a special file in the filesystem. It follows the **First In, First Out (FIFO)** principle, meaning data written first is read first.

### **Named Pipe Creation**

- The `mkfifo()` function is used to create a named pipe `/tmp/myfifo`, which serves as the communication channel between two processes.

### **Process Communication**

- **Program 1 (User1)** writes a message first, then waits to receive a reply.
- **Program 2 (User2)** waits to receive a message first, then writes a reply.

### **Alternating Read/Write**

- Each process reads and writes alternately, enabling a basic chat-like interaction.

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

declares a **pointer to a character array (string)** that holds the path of the **named pipe (FIFO)** in the filesystem.

**char \*myfifo** → Declares a pointer to a character string.

**"/tmp/myfifo"** → This is the actual path where the **FIFO file** will be created.

**/tmp/ directory** is commonly used for temporary files and inter-process communication.

The **FIFO file remains in the filesystem** until deleted.