

**Name :** Rajvardhan Reddy  
**Reg No :** 180905093  
**Sec :** B  
**Roll No :** 19

## **OS LAB - 5 : IPC -1 – PIPE, FIFO**

**P1)** Write a producer and consumer program in C using the FIFO queue. The producer should write a set of 4 integers into the FIFO queue and the consumer should display the 4 integers.

### **Code :**

#### **Producer :**

```
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <sys/types.h>
#include <limits.h>
#include <fcntl.h>
#include <sys/msg.h>
#include <sys/stat.h>
#include <string.h>
#define FIFO_NAME "my_fifo"
#define BUFFER_SIZE 1000
int main(int argc, char *argv[])
{
    int pipe_fd;
    int res;
    int open_mode = O_WRONLY;
    int n = 0;
    char buffer[BUFFER_SIZE + 1];
    if (access(FIFO_NAME, F_OK) == -1)
    {
        res = mkfifo(FIFO_NAME, 0777);
        if (res != 0)
        {
            fprintf(stderr, "Could not create file%s\n",
                FIFO_NAME);
            exit(EXIT_FAILURE);
        }
    }
    printf("Process %d opening FIFO O_WRONLY\n",
        getpid());
```

```

pipe_fd = open(FIFO_NAME, open_mode);
printf("Process %d result %d\n", getpid(), pipe_fd);
if (pipe_fd != -1)
{
printf("Enter 4 numbers\n");
while (n < 4)
{
scanf("%s", buffer);
res = write(pipe_fd, buffer, BUFFER_SIZE);
if (res == -1)
{
fprintf(stderr, "Write Error on Pipe\n");
exit(EXIT_FAILURE);
}
n++;
}
(void)close(pipe_fd);
}
else
exit(EXIT_FAILURE);
printf("Process %d Finished\n", getpid());
exit(EXIT_SUCCESS);
}

```

### **Consumer :**

```

#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <sys/types.h>
#include <limits.h>
#include <fcntl.h>
#include <sys/msg.h>
#include <sys/stat.h>
#include <string.h>
#define FIFO_NAME "my_fifo"
#define BUFFER_SIZE 1000
int main(int argc, char *argv[])
{
int pipe_fd;
int res;
int open_mode = O_RDONLY;
int n = 0;
char buffer[BUFFER_SIZE + 1];
memset(buffer, '\0', sizeof(buffer));
printf("Process %d opening FIFO O_RDONLY\n",

```

```

getpid());
pipe_fd = open(FIFO_NAME, open_mode);
printf("Process %d result %d\n", getpid(), pipe_fd);
if (pipe_fd != -1)
{
do
{
res = read(pipe_fd, buffer, BUFFER_SIZE);
printf("%s\n", buffer);
n++;
} while (n < 4);
(void)close(pipe_fd);
}
else
exit(EXIT_FAILURE);
printf("Process %d Finished, %d bytes read\n",
getpid(), n);
exit(EXIT_SUCCESS);
}

```

## Output :

```

rajvardhan@rajvardhan-HP-Pavilion-Laptop-15-cc1xx:~/Desktop/5th_sem_LABS/OS_LAB/
lab 5$ gcc p1_producer.c -o producer
rajvardhan@rajvardhan-HP-Pavilion-Laptop-15-cc1xx:~/Desktop/5th_sem_LABS/OS_LAB/
lab 5$ ./producer
Process 77662 opening FIFO O_WRONLY
Process 77662 result 3
Enter 4 numbers
4 5 6 7
Process 77662 Finished
rajvardhan@rajvardhan-HP-Pavilion-Laptop-15-cc1xx:~/Desktop/5th_sem_LABS/OS_LAB/
lab 5$ █

```

```

rajvardhan@rajvardhan-HP-Pavilion-Laptop-15-cc1xx:~/Desktop/5th_sem_LABS/OS_LAB/
lab 5$ gcc p1_consumer.c -o consumer
rajvardhan@rajvardhan-HP-Pavilion-Laptop-15-cc1xx:~/Desktop/5th_sem_LABS/OS_LAB/
lab 5$ ./consumer
Process 77750 opening FIFO O_RDONLY
Process 77750 result 3
4
5
6
7
Process 77750 Finished, 4 bytes read
rajvardhan@rajvardhan-HP-Pavilion-Laptop-15-cc1xx:~/Desktop/5th_sem_LABS/OS_LAB/
lab 5$ █

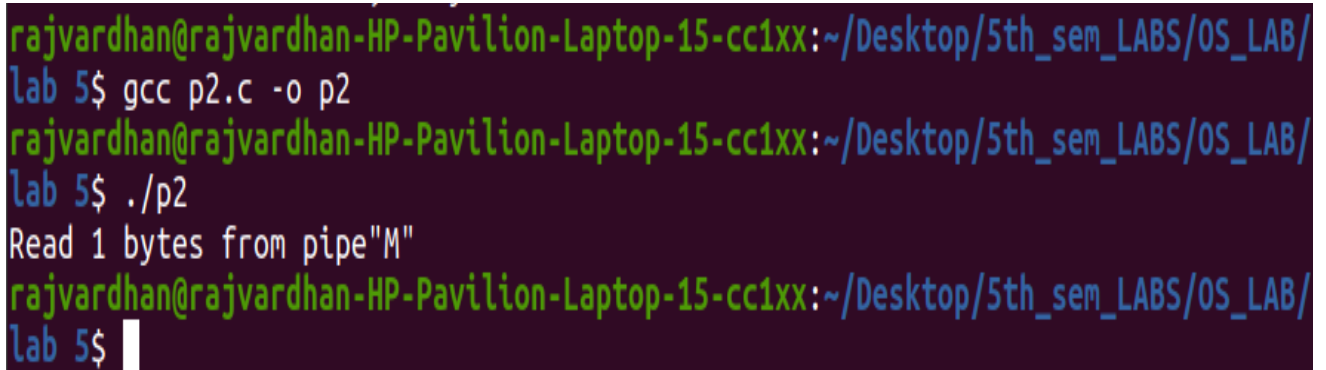
```

**P2)** Demonstrate creation, writing to, and reading from a pipe.

**Code :**

```
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <sys/types.h>
#include <sys/ipc.h>
#include <sys/msg.h>
#include <string.h>
int main(int argc, char *argv[])
{
    int n;
    int fd[2];
    char buf[1025];
    char *data = "Mid Term Lab Exams from tomorrow!";
    pipe(fd);
    write(fd[1], data, strlen(data));
    if (n = read(fd[0], buf, 1024) >= 0)
    {
        buf[n] = 0;
        printf("Read %d bytes from pipe\"%s\"\\n", n, buf);
    }
    else
        perror("Read");
    exit(0);
}
```

**Output :**



```
rajvardhan@rajvardhan-HP-Pavilion-Laptop-15-cc1xx:~/Desktop/5th_sem_LABS/OS_LAB/
lab 5$ gcc p2.c -o p2
rajvardhan@rajvardhan-HP-Pavilion-Laptop-15-cc1xx:~/Desktop/5th_sem_LABS/OS_LAB/
lab 5$ ./p2
Read 1 bytes from pipe"M"
rajvardhan@rajvardhan-HP-Pavilion-Laptop-15-cc1xx:~/Desktop/5th_sem_LABS/OS_LAB/
lab 5$
```

**P3)** Write a C program to implement one side of FIFO.

**Code :**

**User 1 :**

```
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <sys/types.h>
#include <limits.h>
#include <fcntl.h>
#include <sys/msg.h>
#include <sys/stat.h>
#include <string.h>
#define FIFO_NAME "my_fifo"
#define BUFFER_SIZE 10000
int main(int argc, char *argv[])
{
    int pipe_fd;
    int res;
    int open_mode1 = O_WRONLY;
    int open_mode2 = O_RDONLY;
    int n = 0;
    char buffer[BUFFER_SIZE + 1];
    if (access(FIFO_NAME, F_OK) == -1)
    {
        res = mkfifo(FIFO_NAME, 0777);
        if (res != 0)
        {
            fprintf(stderr, "Could not create file%s\n", FIFO_NAME);
            exit(EXIT_FAILURE);
        }
    }
    printf("You can start chatting with User 2 now\n");
    while (1)
    {
        pipe_fd = open(FIFO_NAME, open_mode2);
        printf("\nText from User 1:");
        res = read(pipe_fd, buffer, BUFFER_SIZE);
        printf("%s\n", buffer);
        close(pipe_fd);
        printf("Wait for User 1 reply\n");
        pipe_fd = open(FIFO_NAME, open_mode1);
        printf("\nEnter Text to send User 1:");
```

```

fgets(buffer, BUFFER_SIZE, stdin);
res = write(pipe_fd, buffer, BUFFER_SIZE);
close(pipe_fd);
}
(void)close(pipe_fd);
printf("Process %d Finished\n", getpid());
exit(EXIT_SUCCESS);
}

```

## **User 2 :**

```

#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <sys/types.h>
#include <limits.h>
#include <fcntl.h>
#include <sys/msg.h>
#include <sys/stat.h>
#include <string.h>
#define FIFO_NAME "my_fifo"
#define BUFFER_SIZE 10000
int main(int argc, char *argv[])
{
    int pipe_fd;
    int res;
    int open_mode1 = O_WRONLY;
    int open_mode2 = O_RDONLY;
    int n = 0;
    char buffer[BUFFER_SIZE + 1];
    if (access(FIFO_NAME, F_OK) == -1)
    {
        res = mkfifo(FIFO_NAME, 0777);
        if (res != 0)
        {
            fprintf(stderr, "Could not create file%s\n", FIFO_NAME);
            exit(EXIT_FAILURE);
        }
    }
    printf("You can start chatting with User 2 now\n");
    while (1)
    {
        pipe_fd = open(FIFO_NAME, open_mode1);
        printf("\nEnter Text to send User 2:");
        fgets(buffer, BUFFER_SIZE, stdin);
        res = write(pipe_fd, buffer, BUFFER_SIZE);
    }
}

```

```

close(pipe_fd);
printf("Wait for User 2 reply\n");
pipe_fd = open(FIFO_NAME, open_mode2);
printf("\nText from User 2:");
res = read(pipe_fd, buffer, BUFFER_SIZE);
printf("%s\n", buffer);
close(pipe_fd);
}
(void)close(pipe_fd);
printf("Process %d Finished\n", getpid());
exit(EXIT_SUCCESS);
}

```

### Output :

```

rajvardhan@rajvardhan-HP-Pavilion-Laptop-15-cc1xx:~/Desktop/5th_sem_LABS/OS_LAB/
lab 5$ gcc p3_user1.c -o user1
rajvardhan@rajvardhan-HP-Pavilion-Laptop-15-cc1xx:~/Desktop/5th_sem_LABS/OS_LAB/
lab 5$ ./user1
You can start chatting with User 2 now

Text from User 1:Good Morning

Wait for User 1 reply

Enter Text to send User 1:Good night , nice meeting you

Text from User 1:bye

Wait for User 1 reply

Enter Text to send User 1:

```

```

rajvardhan@rajvardhan-HP-Pavilion-Laptop-15-cc1xx:~/Desktop/5th_sem_LABS/OS_LAB/
lab 5$ gcc p3_user2.c -o user2
rajvardhan@rajvardhan-HP-Pavilion-Laptop-15-cc1xx:~/Desktop/5th_sem_LABS/OS_LAB/
lab 5$ ./user2
You can start chatting with User 2 now

Enter Text to send User 2:Good Morning
Wait for User 2 reply

Text from User 2:Good night , nice meeting you

Enter Text to send User 2:bye
Wait for User 2 reply

```


□

**P4)** Write a C program reading and writing a binary file in C.

**Code :**

```
#include <stdio.h>
#include <stdlib.h>
int main()
{
FILE *fptr;
int num = 0;
fptr = fopen("demo.bin", "wb+");
printf("Enter some numbers : \n");
for (int i = 0; i < 4; i++)
{
scanf("%d", &num);
fwrite(&num, sizeof(int), 1, fptr);
}
printf("Writing done!\n");
fclose(fptr);
fptr = fopen("demo.bin", "rb");
for (int i = 0; i < 4; i++)
{
fread(&num, sizeof(int), 1, fptr);
printf("%d\n", num);
}
}
```

**Output :**



```
rajvardhan@rajvardhan-HP-Pavilion-Laptop-15-cc1xx:~/Desktop/5th_sem_LABS/OS_LAB/lab
5$ gcc p4.c -o p4
rajvardhan@rajvardhan-HP-Pavilion-Laptop-15-cc1xx:~/Desktop/5th_sem_LABS/OS_LAB/lab
5$ ./p4
Enter some numbers :
45 62
72 44
Writing done!
45
62
72
44
rajvardhan@rajvardhan-HP-Pavilion-Laptop-15-cc1xx:~/Desktop/5th_sem_LABS/OS_LAB/lab
5$
```