

Lab5 – OS Lab

Name: Rhea Adhikari

Reg No: 190905156

Roll No: 23

1. 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.

producer.c

```
#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.c
#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);
}

```

```

Student@project-lab: ~/Documents/190905156/OS/Lab5
File Edit View Search Terminal Help
Student@project-lab:~/Documents/190905156/OS/Lab5$ gcc consumer.c -o con
Student@project-lab:~/Documents/190905156/OS/Lab5$ ./con
Process 7746 opening FIFO O_RDONLY
Process 7746 result 3
5
6
7
8
Process 7746 Finished, 4 bytes read
Student@project-lab:~/Documents/190905156/OS/Lab5$

Student@project-lab: ~/Documents/190905156/OS/Lab5
File Edit View Search Terminal Help
Student@project-lab:~/Documents/190905156/OS/Lab5$ gcc producer.c -o pro
Student@project-lab:~/Documents/190905156/OS/Lab5$ ./pro
Process 7759 opening FIFO O_WRONLY
Process 7759 result 3
Enter 4 numbers
5
6
7
8
Process 7759 Finished
Student@project-lab:~/Documents/190905156/OS/Lab5$

```

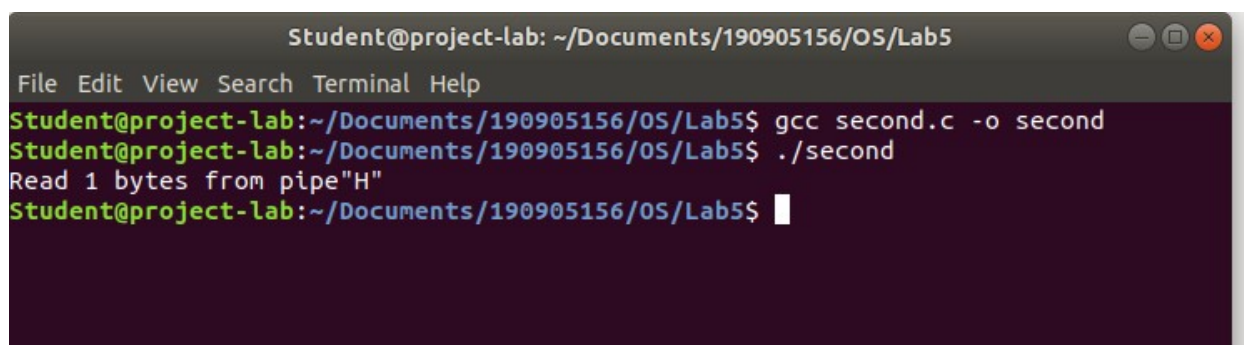
2. Demonstrate creation, writing to, and reading from a pipe.

```
#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="Hello there this is Rhea";
    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);
}
```



The screenshot shows a terminal window titled "Student@project-lab: ~/Documents/190905156/OS/Lab5". The terminal contains the following commands and output:

```
Student@project-lab:~/Documents/190905156/OS/Lab5$ gcc second.c -o second
Student@project-lab:~/Documents/190905156/OS/Lab5$ ./second
Read 1 bytes from pipe"H"
Student@project-lab:~/Documents/190905156/OS/Lab5$
```

3. Write a C program to implement one side of FIFO.

```
Student@project-lab: ~/Documents/190905156/OS/Lab5
File Edit View Search Terminal Help
Student@project-lab:~/Documents/190905156/OS/Lab5$ ./one
You can start chatting with Person 2 now

Text from Person 1: HI
Wait for Person 1 reply

Enter Text to send Person 1: HI 2
Text from Person 1: How are you
Wait for Person 1 reply

Enter Text to send Person 1: Im fine
24 res=mkfifo(FIFO_NAME,0777);
25 if(res!=0)
```

```
Student@project-lab: ~/Documents/190905156/OS/Lab5
File Edit View Search Terminal Help
bash: cd: Doc: No such file or directory
Student@project-lab:~$ cd Documents/
Student@project-lab:~/Documents$ cd 19
bash: cd: 19: No such file or directory
Student@project-lab:~/Documents$ cd 190905156/
Student@project-lab:~/Documents/190905156$ cd OS
Student@project-lab:~/Documents/190905156/OS$ cd Lab5
Student@project-lab:~/Documents/190905156/OS/Lab5$ gcc two.c -o two
Student@project-lab:~/Documents/190905156/OS/Lab5$ ./two
You can start chatting with Person 2 now

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

Text from User 2: HI 2

Enter Text to send User 2: How are you
Wait for User 2 reply

Text from User 2: Im fine

Enter Text to send User 2: 
```

one.c

```
#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 Person 2 now\n");

    while(1)
    {
        pipe_fd=open(FIFO_NAME,open_mode2);

        printf("\nText from Person 1: ");
```

```

        res=read(pipe_fd,buffer,BUFFER_SIZE);
        printf("%s\n",buffer );
        close(pipe_fd);

        printf("Wait for Person 1 reply\n");
        pipe_fd=open(FIFO_NAME,open_mode1);

        printf("\nEnter Text to send Person 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);
}

```

two.c

```

#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=mknod(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 Person 2 now\n");

    while(1)

```

```

    {
        pipe_fd=open(FIFO_NAME,open_mode1);
        printf("\nEnter Text to send Person 2: ");
        fgets(buffer,BUFFER_SIZE,stdin);
        res=write(pipe_fd,buffer,BUFFER_SIZE);

        close(pipe_fd);

        printf("Wait for Person 2 reply\n");
        pipe_fd=open(FIFO_NAME,open_mode2);

        printf("\nText from Person 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);
}

```

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

```

#include<stdio.h>
#include<stdlib.h>

int main()
{
    FILE* fptr;
    int num=0;
    fptr=fopen("demo.bin","wb+");

    printf("Enter few numbers : \n");

    for(int i=0;i<4;i++)
    {
        scanf("%d",&num);
        fwrite(&num,sizeof(int),1,fptr);
    }

    printf("Writing operation over.\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);
    }
}

```

}

```
000
0
0 Student@project-lab: ~/Documents/190905156/OS/Lab5
0 File Edit View Search Terminal Help
0 Student@project-lab:~/Documents/190905156/OS/Lab5$ gcc fourth.c -o fourth
0 Student@project-lab:~/Documents/190905156/OS/Lab5$ ./fourth
0 Enter few numbers :
0 3259
0 13531
0 15413
0 1532
0 Writing operation over.
0 3259
0 13531
0 15413
0 1532
0 Student@project-lab:~/Documents/190905156/OS/Lab5$
0
0
0
0
0
0
0
0
```