

Operating Systems Lab 5

Dipesh Singh - 190905520

Question 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 integers.

Producer.c :

```
#include <unistd.h>
#include <stdio.h>
#include <stdlib.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_queue"
#define BUFFER_SIZE 1000

int main(int argn, char* args[]){
    int pipe_fd, res, open_mode = O_WRONLY, 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);
    if(pipe_fd != -1){
        printf("Enter 4 numbers : \n");
        while(n<4){
            printf("%d : ", n+1);
            scanf("%s", buffer);
            res = write(pipe_fd, buffer, BUFFER_SIZE);
            if(res == -1){
                fprintf(stderr, "Write error on PIPE\n");
                exit(EXIT_FAILURE);
            }
            n++;
        }
        close(pipe_fd);
    }
    else{
        exit(EXIT_FAILURE);
    }
}
```

```

        printf("Process %d finished\n", getpid());
        exit(EXIT_SUCCESS);
}

```

Consumer.c :

```

#include <unistd.h>
#include <stdio.h>
#include <stdlib.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_queue"
#define BUFFER_SIZE 1000

int main(int argc, char *argv[])
{
    int pipe_fd, res, open_mode=O_RDONLY, 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);
    int bytes_read = 0;
    if (pipe_fd != -1)
    {
        do
        {
            printf("%d : ", n+1);
            res = read(pipe_fd, buffer, BUFFER_SIZE);
            printf("%s\n", buffer);
            bytes_read += BUFFER_SIZE;
            n++;
        }while(n<4);
        close(pipe_fd);
    }
    else
        exit(EXIT_FAILURE);
    printf("Process %d Finished, %d bytes read\n", getpid(),
bytes_read);
    exit(EXIT_SUCCESS);
}

```

Output :

```
ugcse@prg28:~/Desktop/190905520/OS/lab5$ ./prod
Process 4135 opening FIFO O_WRONLY
Enter 4 numbers :
1 : 1
2 : 6
3 : 2
4 : 8
Process 4135 finished
ugcse@prg28:~/Desktop/190905520/OS/lab5$ gcc q1_producer.c -o prod
ugcse@prg28:~/Desktop/190905520/OS/lab5$ ./prod
Process 4243 opening FIFO O_WRONLY
Enter 4 numbers :
1 : 123456
2 : 0987654
3 : 543657
4 : 12354
Process 4243 finished
ugcse@prg28:~/Desktop/190905520/OS/lab5$

ugcse@prg28:~/Desktop/190905520/OS/lab5$ ./cons
Process 4145 opening FIFO O_RDONLY
Process 4145 result 3
1 : 1
2 : 6
3 : 2
4 : 8
Process 4145 Finished, 4000 bytes read
ugcse@prg28:~/Desktop/190905520/OS/lab5$ gcc q1_consumer.c -o cons
ugcse@prg28:~/Desktop/190905520/OS/lab5$ ./cons
Process 4242 opening FIFO O_RDONLY
1 : 123456
2 : 0987654
3 : 543657
4 : 12354
Process 4242 Finished, 4000 bytes read
ugcse@prg28:~/Desktop/190905520/OS/lab5$
```

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

Source Code :

```
#include <unistd.h>
#include <stdio.h>
#include <stdlib.h>
#include <sys/types.h>
#include <sys/ipc.h>
#include <sys/msg.h>
#include <string.h>

int main(int argc, char *argv[])
{
    int n, fd[2];
    char buf[1025], *data="Sample Data for Operating Systems Lab - 190905520";
    pipe(fd);
    write(fd[1], data, strlen(data));
    n=read(fd[0], buf, 1024);
    if(n>=0)
    {
        buf[n]=0;
        printf("Read %d bytes from pipe\n\n%s\n\n", n, buf);
    }
    else
        perror("Read");
    exit(0);
}
```

Output :

```
ugcse@prg28:~/Desktop/190905520/OS/lab5$ ./pipe
Read 49 bytes from pipe
"Sample Data for Operating Systems Lab - 190905520"
```

Question 3 : Write a program to implement one side of FIFO.

Program_1.c

```
#include <unistd.h>
#include <stdio.h>
#include <stdlib.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_queue"
#define BUFFER_SIZE 10000

int main(int argc, char *argv[])
{
    int pipe_fd, res, open_mode1=O_WRONLY, open_mode2=O_RDONLY,
    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("Creating a program to communicate with another program\n");
    printf("In program 1 right now and will start communicating\n");

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

        printf("\nText from program 2: ");
        res=read(pipe_fd,buffer,BUFFER_SIZE);
        if(strstr(buffer, "exit") != NULL) {
            break;
        }
        printf("%s\n",buffer );
        close(pipe_fd);
    }
}
```

```

        pipe_fd=open(FIFO_NAME,open_mode1);

        printf("\nEnter Text to send to program 2:  ");
        fgets(buffer,BUFFER_SIZE,stdin);
        res=write(pipe_fd,buffer,BUFFER_SIZE);

        close(pipe_fd);

    }

    close(pipe_fd);

    printf("Process %d Finished\n",getpid());
    exit(EXIT_SUCCESS);
}

```

Program_2.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_queue"
#define BUFFER_SIZE 10000

int main(int argc, char *argv[])
{
    int pipe_fd, res, open_mode1=O_WRONLY, open_mode2=O_RDONLY,
    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("In program 2 right now and will start communicating with program 1\n");
}

```

```

while(1)
{
    pipe_fd=open(FIFO_NAME,open_mode1);
    printf("\nEnter Text to send program 1:  ");
    fgets(buffer,BUFFER_SIZE,stdin);
    res=write(pipe_fd,buffer,BUFFER_SIZE);

    close(pipe_fd);

    pipe_fd=open(FIFO_NAME,open_mode2);

    printf("\nText from program 1:  ");
    res=read(pipe_fd,buffer,BUFFER_SIZE);
    if(strstr(buffer, "exit") != NULL) {
        break;
    }
    printf("%s\n",buffer );
    close(pipe_fd);

}
close(pipe_fd);

printf("Process %d Finished\n",getpid() );
exit(EXIT_SUCCESS);
}

```

Output :

<pre> ugcse@prg28:~/Desktop/190905520/05/Lab5\$./two In program 2 right now and will start communicating with program 1 Enter Text to send program 1: Hello Program 1 Text from program 1: Hello Program 2 Enter Text to send program 1: what's your pid Text from program 1: i have no idea, it will show after i finish Enter Text to send program 1: how do we end this? Text from program 1: just type E X I T Enter Text to send program 1: ok bye, exit ^C ugcse@prg28:~/Desktop/190905520/05/Lab5\$ █ </pre>	<pre> ugcse@prg28:~/Desktop/190905520/05/Lab5\$./one Creating a program to communicate with another program through fifo In program 1 right now and will start communicating with program 2 Text from program 2: Hello Program 1 Enter Text to send to program 2: Hello Program 2 Text from program 2: what's your pid Enter Text to send to program 2: i have no idea, it will show after i finish Text from program 2: how do we end this? Enter Text to send to program 2: just type E X I T Text from program 2: Process 6779 Finished ugcse@prg28:~/Desktop/190905520/05/Lab5\$ █ </pre>
---	---

Question 4 : Write a program reading and writing binary files in C.

Source Code :

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

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

    printf("Enter 4 numbers of your choice: \n");

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

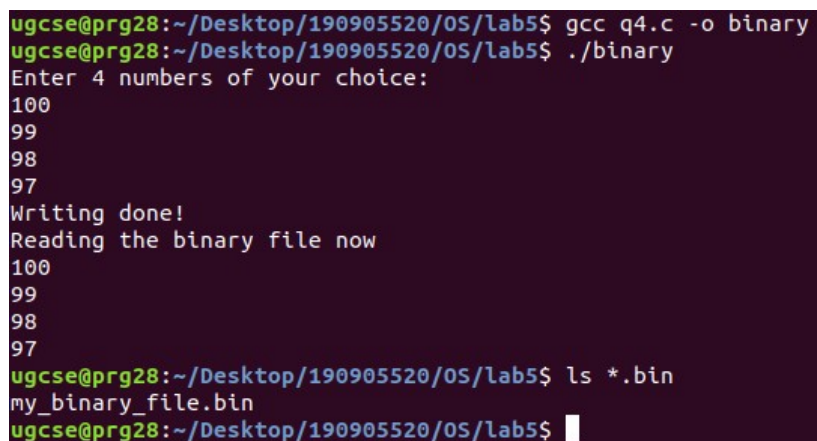
    printf("Writing done!\n");
    fclose(fout);

    printf("Reading the binary file now\n");

    fout=fopen("my_binary_file.bin","rb");

    for(int i=0;i<4;i++)
    {
        fread(&num,sizeof(int),1,fout);
        printf("%d\n",num);
    }
    fclose(fout);
}
```

Output :



```
ugcse@prg28:~/Desktop/190905520/OS/lab5$ gcc q4.c -o binary
ugcse@prg28:~/Desktop/190905520/OS/lab5$ ./binary
Enter 4 numbers of your choice:
100
99
98
97
Writing done!
Reading the binary file now
100
99
98
97
ugcse@prg28:~/Desktop/190905520/OS/lab5$ ls *.bin
my_binary_file.bin
ugcse@prg28:~/Desktop/190905520/OS/lab5$
```