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.

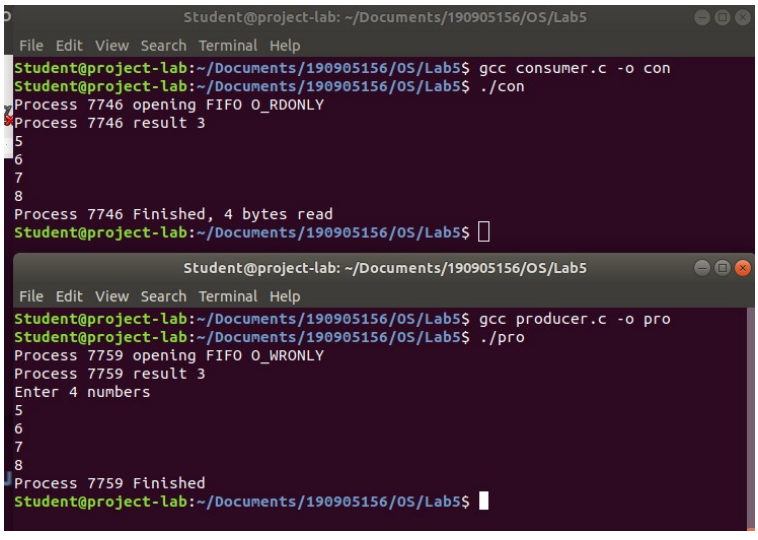
   producer.c

```
#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);
}
```

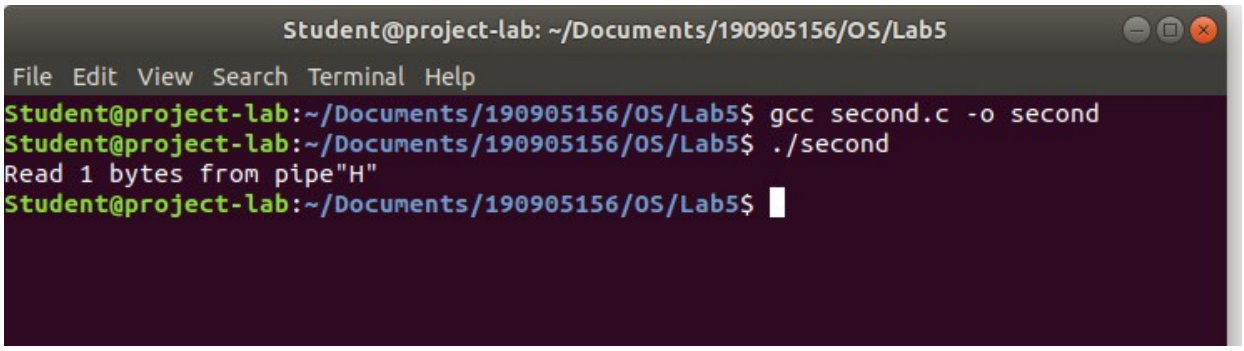
consumer.c

```
#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);
}
```

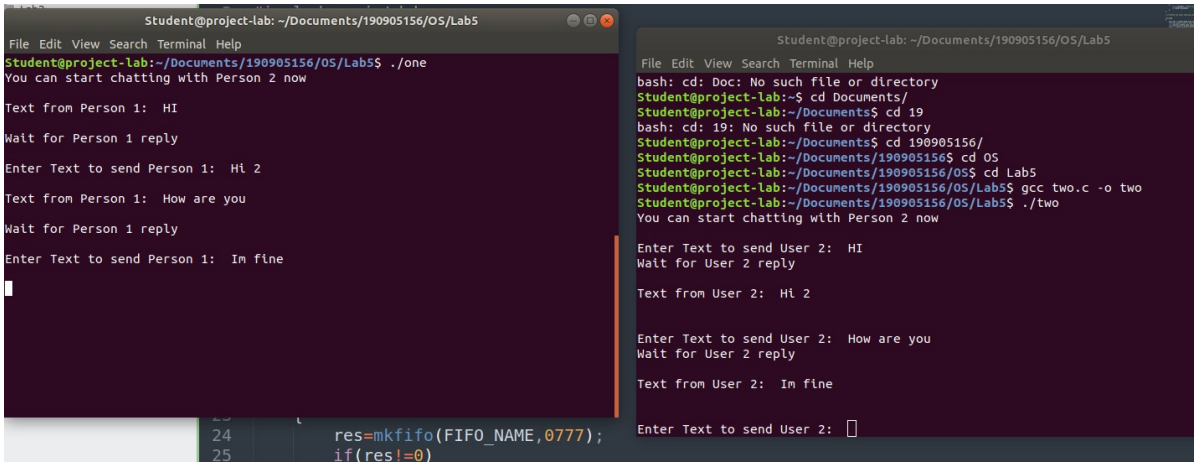


2. Demonstrate creation, writing to, and reading from a pipe.

```
#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);
}
```



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



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

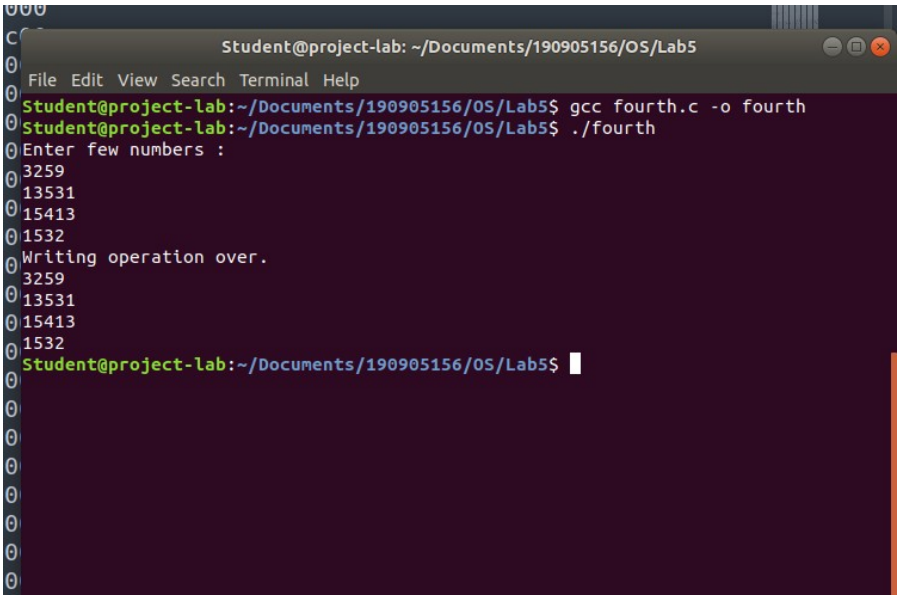
```
#include<stdio.h>
#include<stdlib.h>
#include<unistd.h>
#include<sys/types.h>
#include<limits.h>
#include<fcntl.h>
#include<sys/msg.h>
#include<sys/stat.h>
#include<string.h>
#define FIFO_NAME "my_fifo"
#define BUFFER_SIZE 10000

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=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_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);
    }
```

```
}
```


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