190905514 **MOHAMMAD TOFIK** **BATCH : C3**
ROLLNO : 62 **SECTION : C** **SEM : 5th**

-: OS-LAB-5 :-

LAB EXCERCISE :

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.

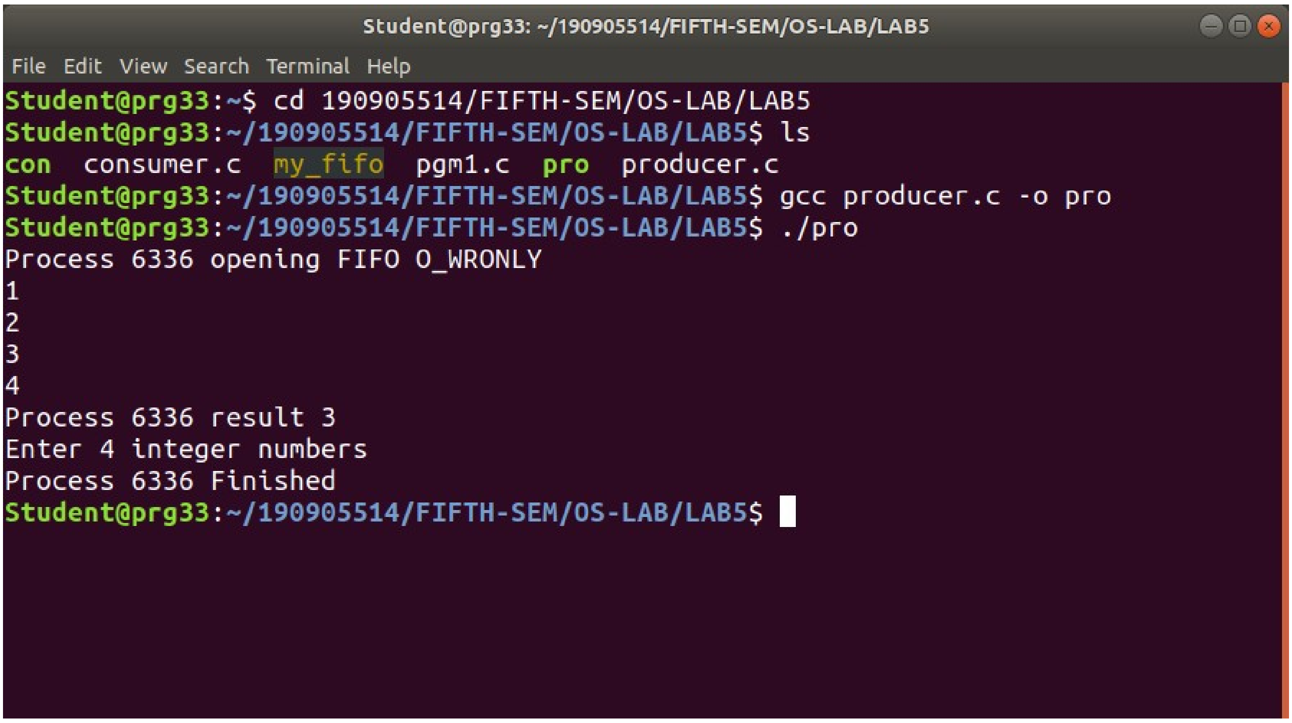
producer.c

```
#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 integer 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

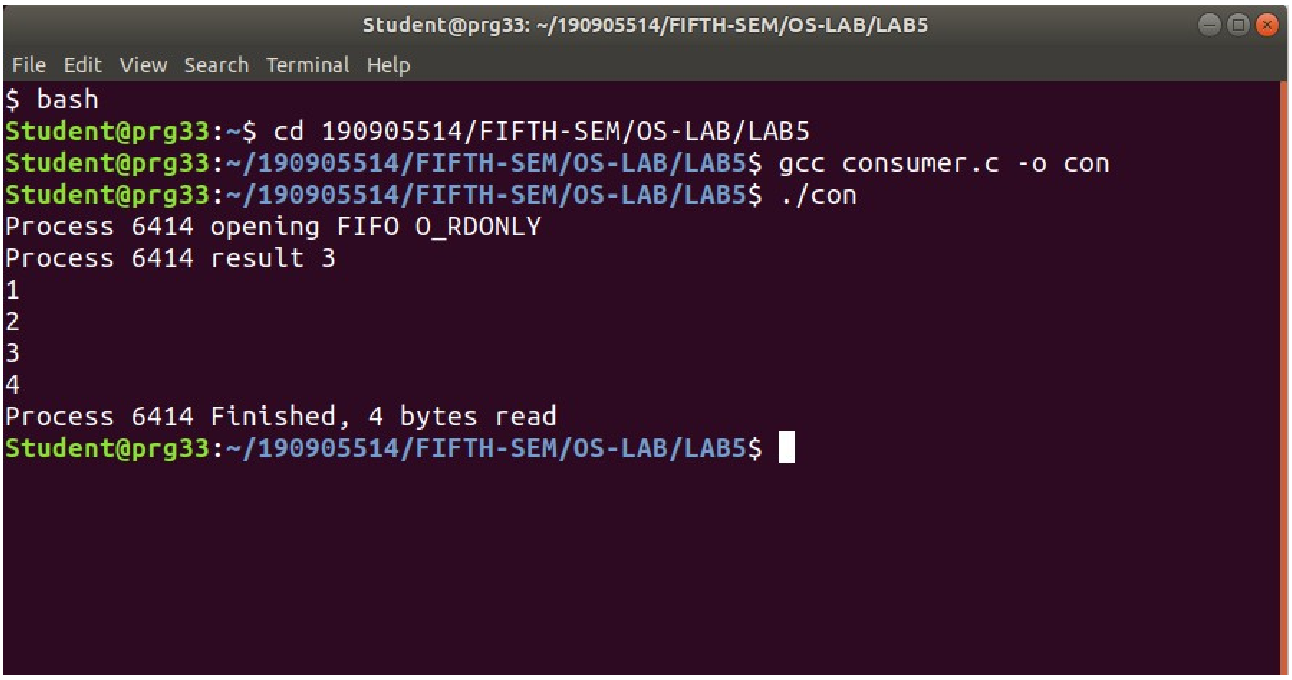
```
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <sys/types.h>
#include <limits.h>
#include <fcntl.h>
#include <sys/msg.h>
#include <sys/stat.h>
#include <string.h>

#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);
}
```

OUTPUT



Student@prg33: ~/190905514/FIFTH-SEM/OS-LAB/LAB5

```
Student@prg33:~$ cd 190905514/FIFTH-SEM/OS-LAB/LAB5
Student@prg33:~/190905514/FIFTH-SEM/OS-LAB/LAB5$ ls
con  consumer.c  my_fifo  pgm1.c  pro  producer.c
Student@prg33:~/190905514/FIFTH-SEM/OS-LAB/LAB5$ gcc producer.c -o pro
Student@prg33:~/190905514/FIFTH-SEM/OS-LAB/LAB5$ ./pro
Process 6336 opening FIFO O_WRONLY
1
2
3
4
Process 6336 result 3
Enter 4 integer numbers
Process 6336 Finished
Student@prg33:~/190905514/FIFTH-SEM/OS-LAB/LAB5$ 
```



Student@prg33: ~/190905514/FIFTH-SEM/OS-LAB/LAB5

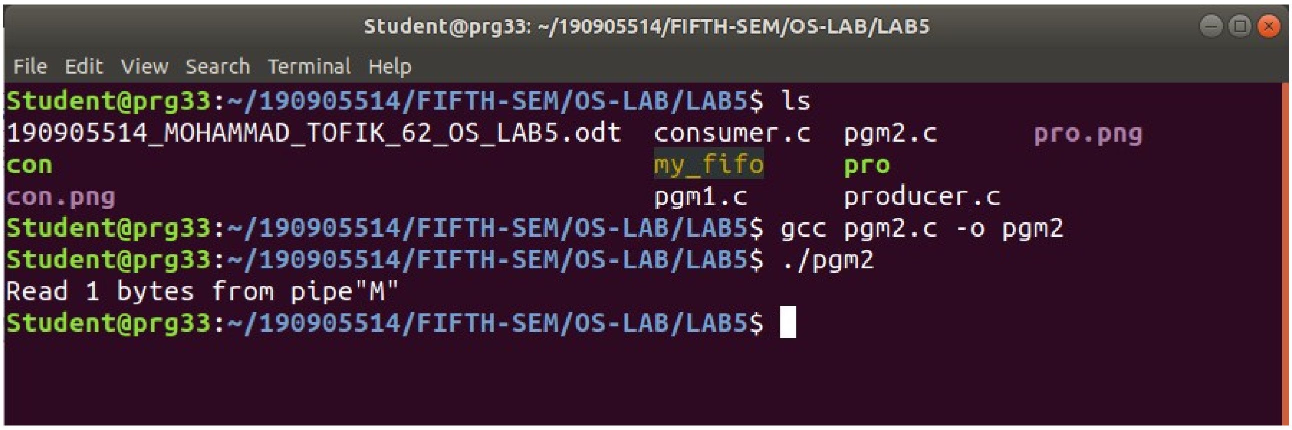
```
$ bash
Student@prg33:~$ cd 190905514/FIFTH-SEM/OS-LAB/LAB5
Student@prg33:~/190905514/FIFTH-SEM/OS-LAB/LAB5$ gcc consumer.c -o con
Student@prg33:~/190905514/FIFTH-SEM/OS-LAB/LAB5$ ./con
Process 6414 opening FIFO O_RDONLY
Process 6414 result 3
1
2
3
4
Process 6414 Finished, 4 bytes read
Student@prg33:~/190905514/FIFTH-SEM/OS-LAB/LAB5$ 
```

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

pgm2.c

```c
#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 = "MOHAMMAD TOFIK MANIPAL MIT";
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


Student@prg33: ~/190905514/FIFTH-SEM/OS-LAB/LAB5
File Edit View Search Terminal Help

```
Student@prg33:~/190905514/FIFTH-SEM/OS-LAB/LAB5$ ls
190905514_MOHAMMAD_TOFIK_62_OS_LAB5.odt  consumer.c  pgm2.c      pro.png
con                                      my_fifo     pro
con.png                                  pgm1.c      producer.c
Student@prg33:~/190905514/FIFTH-SEM/OS-LAB/LAB5$ gcc pgm2.c -o pgm2
Student@prg33:~/190905514/FIFTH-SEM/OS-LAB/LAB5$ ./pgm2
Read 1 bytes from pipe"M"
Student@prg33:~/190905514/FIFTH-SEM/OS-LAB/LAB5$ 
```

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

user1.c

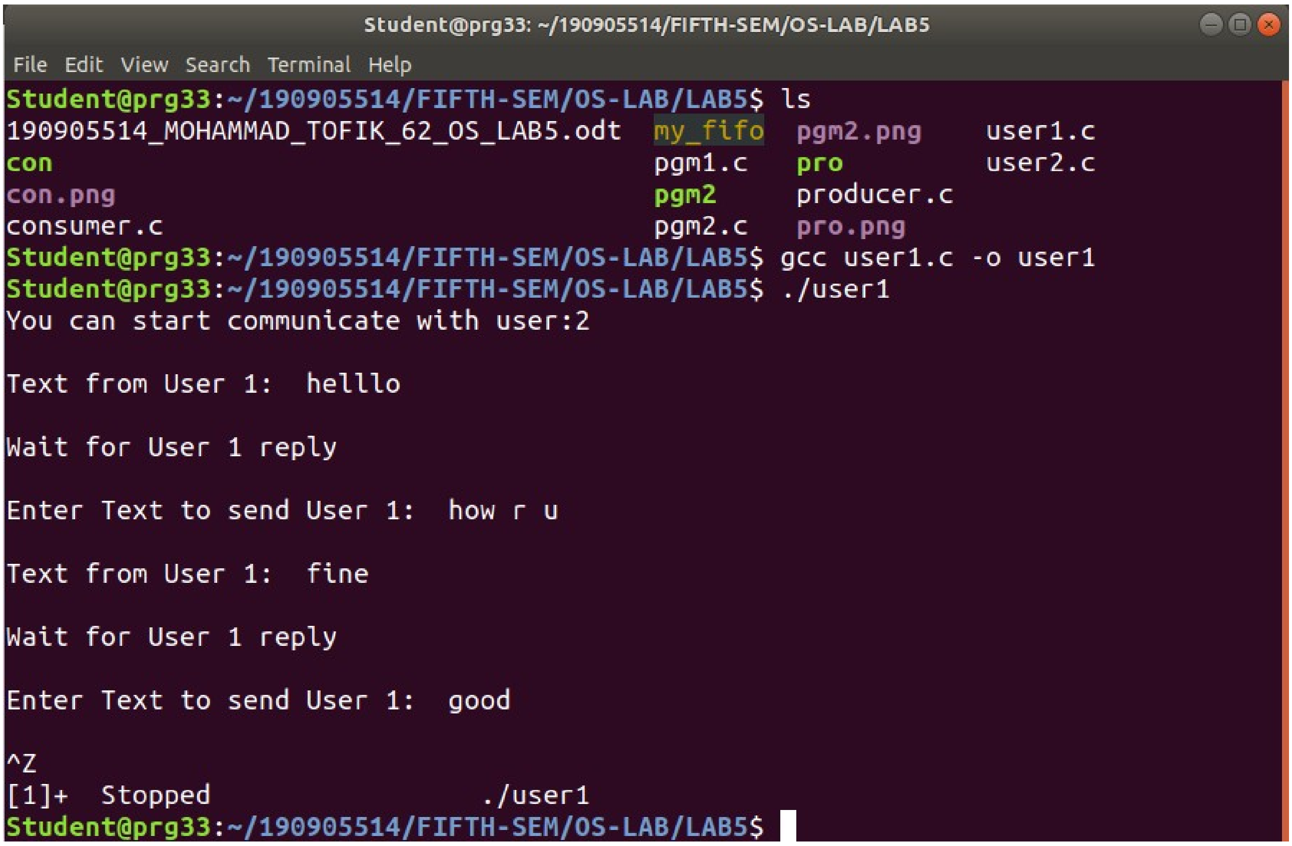
```
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <sys/types.h>
#include <limits.h>
#include <fcntl.h>
#include <sys/msg.h>
#include <sys/stat.h>
#include <string.h>

#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 communicate with user:2\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);
}
```

user2.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
```

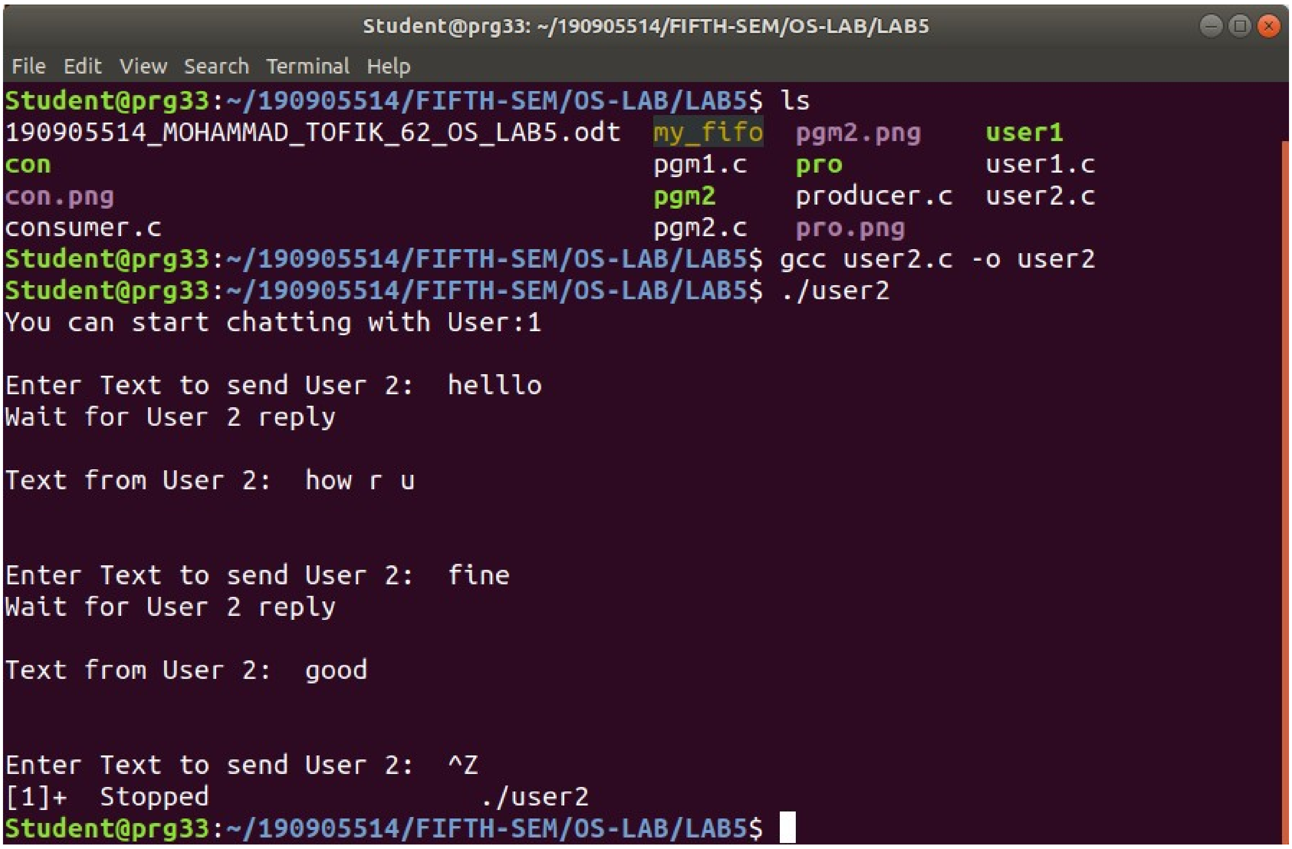
```
int main(int argc, char *argv[])
{
int pipe_fd;
int res;
int open_mode1 = O_WRONLY;
int open_mode2 = O_RDONLY;
int n = 0;
char buffer[BUFFER_SIZE + 1];

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 User:1\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



Student@prg33: ~/190905514/FIFTH-SEM/OS-LAB/LAB5
File Edit View Search Terminal Help

```
Student@prg33:~/190905514/FIFTH-SEM/OS-LAB/LAB5$ ls
190905514_MOHAMMAD_TOFIK_62_OS_LAB5.odt  my_fifo  pgm2.png    user1.c
con                                      pgm1.c   pro         user2.c
con.png                                  pgm2     producer.c
consumer.c                               pgm2.c   pro.png
Student@prg33:~/190905514/FIFTH-SEM/OS-LAB/LAB5$ gcc user1.c -o user1
Student@prg33:~/190905514/FIFTH-SEM/OS-LAB/LAB5$ ./user1
You can start communicate with user:2

Text from User 1:  helllo

Wait for User 1 reply

Enter Text to send User 1:  how r u

Text from User 1:  fine

Wait for User 1 reply

Enter Text to send User 1:  good

^Z
[1]+  Stopped                 ./user1
Student@prg33:~/190905514/FIFTH-SEM/OS-LAB/LAB5$ 
```



Student@prg33: ~/190905514/FIFTH-SEM/OS-LAB/LAB5
File Edit View Search Terminal Help

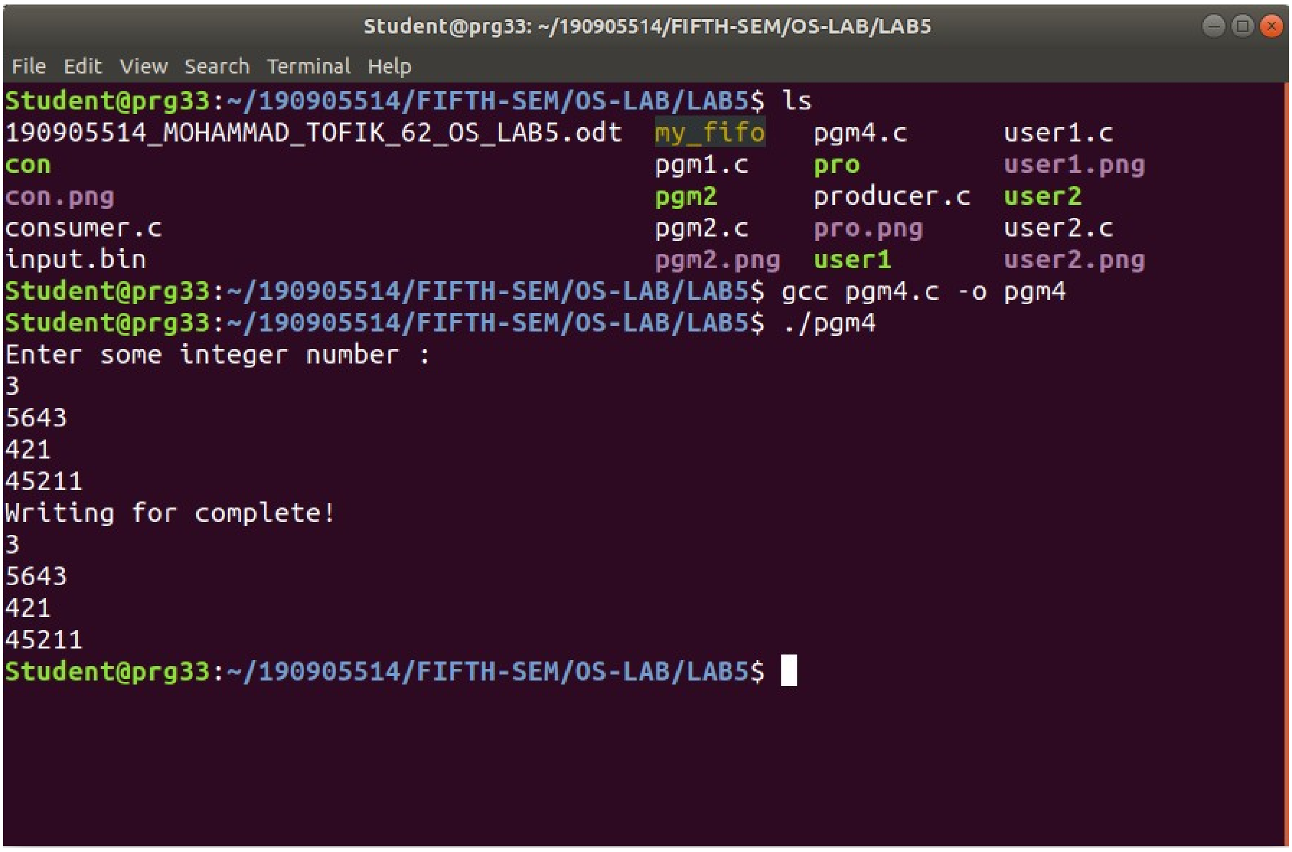
```
Student@prg33:~/190905514/FIFTH-SEM/OS-LAB/LAB5$ ls
190905514_MOHAMMAD_TOFIK_62_OS_LAB5.odt  my_fifo  pgm2.png    user1
con                                      pgm1.c   pro         user1.c
con.png                                  pgm2     producer.c  user2.c
consumer.c                               pgm2.c   pro.png
Student@prg33:~/190905514/FIFTH-SEM/OS-LAB/LAB5$ gcc user2.c -o user2
Student@prg33:~/190905514/FIFTH-SEM/OS-LAB/LAB5$ ./user2
You can start chatting with User:1

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

Text from User 2:  how r u


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

Text from User 2:  good


Enter Text to send User 2:  ^Z
[1]+  Stopped                 ./user2
Student@prg33:~/190905514/FIFTH-SEM/OS-LAB/LAB5$ 
```

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

pgm4.c

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

int main()
{

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

printf("Enter some integer number : \n");

for (int i = 0; i < 4; i++)
{

scanf("%d", &num);
fwrite(&num, sizeof(int), 1, fptr);
}

printf("Writing for complete!\n");
fclose(fptr);

fptr = fopen("input.bin", "rb");

for (int i = 0; i < 4; i++)
{

fread(&num, sizeof(int), 1, fptr);
printf("%d\n", num);
}
}
```

OUTPUT



Student@prg33: ~/190905514/FIFTH-SEM/OS-LAB/LAB5

```
Student@prg33:~/190905514/FIFTH-SEM/OS-LAB/LAB5$ ls
190905514_MOHAMMAD_TOFIK_62_OS_LAB5.odt  my_fifo    pgm4.c      user1.c
con                                      pgm1.c     pro         user1.png
con.png                                  pgm2       producer.c  user2
consumer.c                               pgm2.c     pro.png     user2.c
input.bin                                pgm2.png   user1       user2.png
Student@prg33:~/190905514/FIFTH-SEM/OS-LAB/LAB5$ gcc pgm4.c -o pgm4
Student@prg33:~/190905514/FIFTH-SEM/OS-LAB/LAB5$ ./pgm4
Enter some integer number :
3
5643
421
45211
Writing for complete!
3
5643
421
45211
Student@prg33:~/190905514/FIFTH-SEM/OS-LAB/LAB5$ 
```