IT253 - OPERATING SYSTEMS

ASSIGNMENT 4

Name: **Sachin Prasanna**
Roll No.: **211IT058**

1)

Outputs:

Server Side:

```
student@itadmin-HP-ProDesk-600-G5-MT:~/Desktop/OS Labs/oslab_4$ gcc msgwrite.c -o ser
student@itadmin-HP-ProDesk-600-G5-MT:~/Desktop/OS Labs/oslab_4$ ./ser
Write Data: This is OS Lab 4
Data sent is: This is OS Lab 4

student@itadmin-HP-ProDesk-600-G5-MT:~/Desktop/OS Labs/oslab_4$
```

Client Side:

```
student@itadmin-HP-ProDesk-600-G5-MT:~/Desktop/OS Labs/oslab_4$ gcc msgread.c -o cli
student@itadmin-HP-ProDesk-600-G5-MT:~/Desktop/OS Labs/oslab_4$ ./cli
Data Recieved is: This is OS Lab 4

student@itadmin-HP-ProDesk-600-G5-MT:~/Desktop/OS Labs/oslab_4$
```

Observations:

This is a simple example of inter-process communication using message queues in C. The message to be sent has a type and the message content. The msgget function is used to create the message queue with the unique key. Then, data is inputted in by

the user and placed in the message queue by the server, which includes both message content and type.

At the client side, the queue is accessed and the particular message addressed for it is opened using the **message type**. Thereafter, the message is read and outputted on the client's side. Finally, the message queue is destroyed by msgctl().

2)

Code Written:

Producer (Server):

```
#include <stdio.h>
#include <sys/ipc.h>
#include <sys/msg.h>

#define MAX 20

struct mesg_buffer
{
   long mesg_type;
   char mesg_text[100];
} messageToConsumer1, messageToConsumer2;

int main()
{

   key_t key;
   int msgid;

   key = ftok("progfile", 65);
```

```
   msgid = msgget(key, 0666 | IPC_CREAT);

   // Message Consumer 1
   messageToConsumer1.mesg_type = 1;

   printf("Write Data to Consumer 1: ");
   fgets(messageToConsumer1.mesg_text, MAX, stdin);

   msgsnd(msgid, &messageToConsumer1, sizeof(messageToConsumer1), 0);

   printf("Data sent to Consumer 1 is: %s \n",
messageToConsumer1.mesg_text);

   // Message Consumer 2

   messageToConsumer2.mesg_type = 2;

   printf("Write Data to Consumer 2: ");
   fgets(messageToConsumer2.mesg_text, MAX, stdin);

   msgsnd(msgid, &messageToConsumer2, sizeof(messageToConsumer2), 0);

   printf("Data sent to Consumer 2 is: %s \n",
messageToConsumer2.mesg_text);

      msgctl(msgid, IPC_RMID, NULL);


   return 0;
}
```

Consumer 1 (Client 1):

```
#include <stdio.h>
#include <sys/ipc.h>
#include <sys/msg.h>

struct mesg_buffer {
   long mesg_type;
```

```
   char mesg_text[100];
} message;

int main(){

   key_t key;
   int msgid;

   key = ftok("progfile", 65);

   msgid = msgget(key, 0666 | IPC_CREAT);

   msgrcv(msgid, &message, sizeof(message), 1, 0);

   printf("Data Recieved is: %s \n", message.mesg_text);

   //msgctl(msgid, IPC_RMID, NULL);

   return 0;
}
```

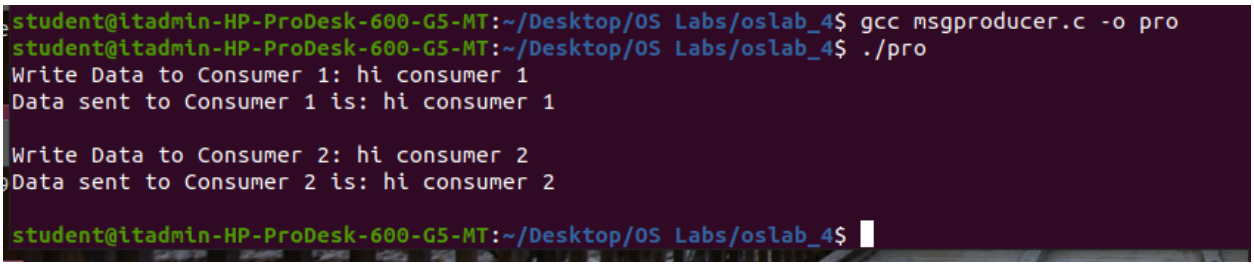
Consumer 2 (Client 2):

```
#include <stdio.h>
#include <sys/ipc.h>
#include <sys/msg.h>

struct mesg_buffer {
   long mesg_type;
   char mesg_text[100];
} message;

int main(){

   key_t key;
   int msgid;

   key = ftok("progfile", 65);

   msgid = msgget(key, 0666 | IPC_CREAT);
```

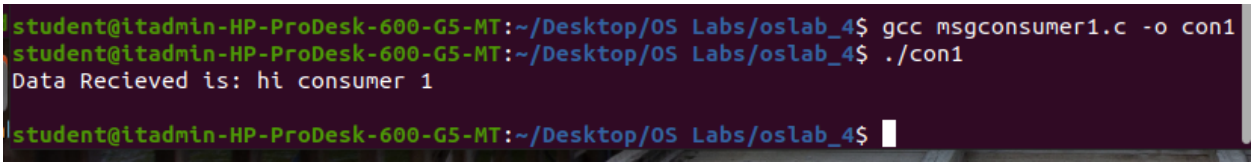
```
    msgrcv(msgid, &message, sizeof(message), 2, 0);

    printf("Data Recieved is: %s \n", message.mesg_text);

    //msgctl(msgid, IPC_RMID, NULL);

    return 0;
}
```

Outputs:

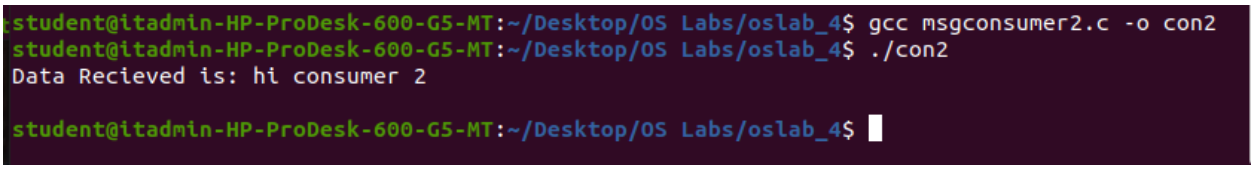
Producer (Server):



Consumer 1 (Client 1):



Consumer 2 (Client 2):


student@itadmin-HP-ProDesk-600-G5-MT:~/Desktop/OS Labs/oslab_4$ gcc msgconsumer2.c -o con2
student@itadmin-HP-ProDesk-600-G5-MT:~/Desktop/OS Labs/oslab_4$ ./con2
Data Recieved is: hi consumer 2
student@itadmin-HP-ProDesk-600-G5-MT:~/Desktop/OS Labs/oslab_4$

****