Linux-IPC-Message-Queues

Linux IPC-Message Queues

AIM:

To write a C program that receives a message from message queue and display them

DESIGN STEPS:

Step 1:

Navigate to any Linux environment installed on the system or installed inside a virtual environment like virtual box/vmware or online linux JSLinux (https://bellard.org/jslinux/vm.html?url=alpine-x86.cfg&mem=192) or docker.

Step 2:

Write the C Program using Linux message queues API

Step 3:

Execute the C Program for the desired output.

PROGRAM:

C program that receives a message from message queue and display them

writer.c

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

// structure for message queue
struct mesg_buffer {
        long mesg_type;
        char mesg_text[100];
} message;
int main()
{       key_t key;
        int msgid;
    // ftok to generate unique key
        key = ftok("progfile", 65);
        // msgget creates a message queue
        // and returns identifier
        msgid = msgget(key, 0666 | IPC_CREAT);
        message.mesg_type = 1;
        printf("Write Data : ");
        gets(message.mesg_text);
        // msgsnd to send message
        msgsnd(msgid, &message, sizeof(message), 0);
        // display the message
        printf("Data send is : %s \n", message.mesg_text);
        return 0;
}
```

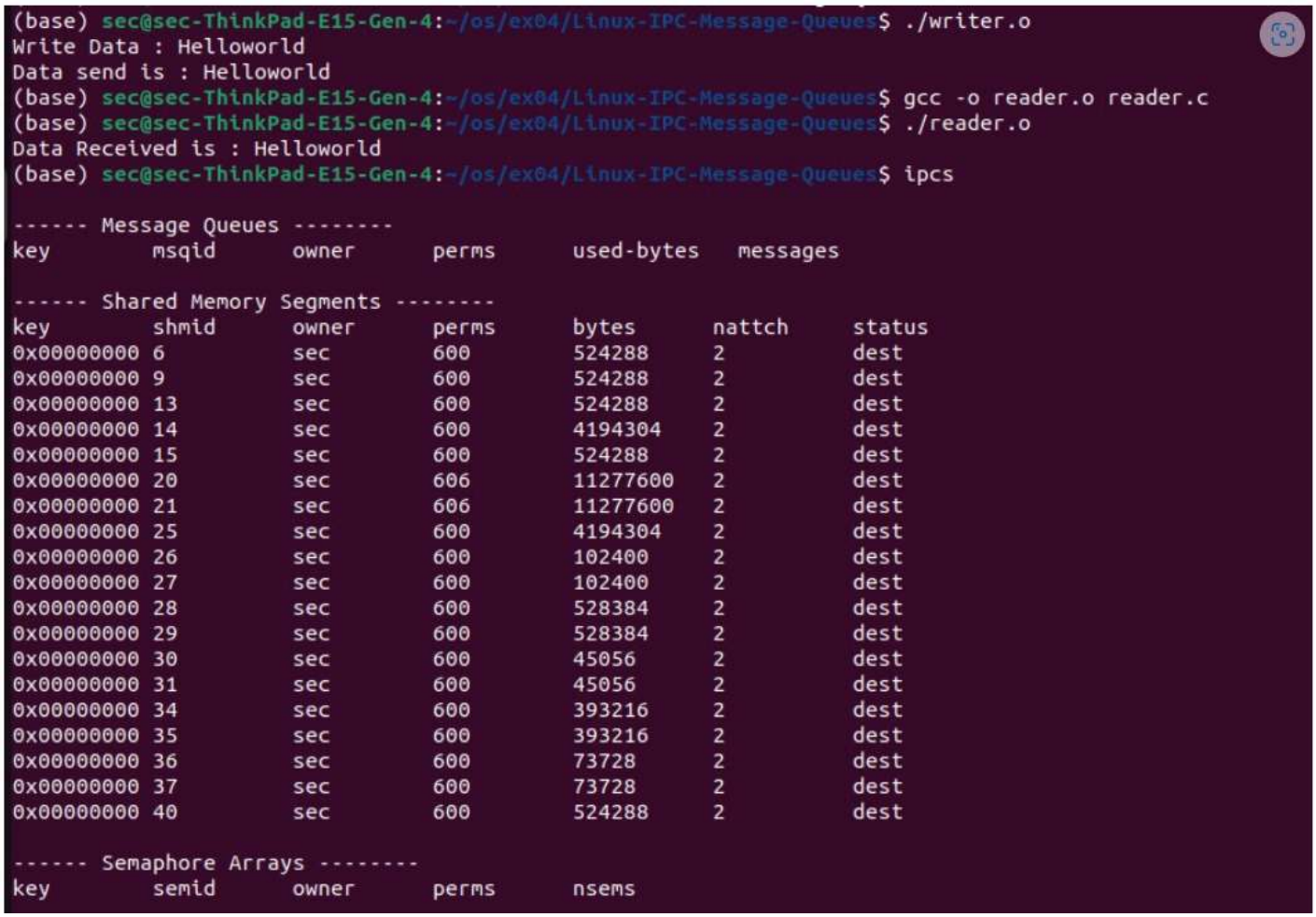
reader.c

```
// C Program for Message Queue (Reader Process)
#include <stdio.h>
#include <sys/ipc.h>
#include <sys/msg.h>

// structure for message queue
struct mesg_buffer {
        long mesg_type;
        char mesg_text[100];
} message;
int main()
{
        key_t key;
        int msgid;
        // ftok to generate unique key
        key = ftok("progfile", 65);
        // msgget creates a message queue
        // and returns identifier
        msgid = msgget(key, 0666 | IPC_CREAT);
        // msgrcv to receive message
        msgrcv(msgid, &message, sizeof(message), 1, 0);
        // display the message
        printf("Data Received is : %s \n",
                        message.mesg_text);

        // to destroy the message queue
        msgctl(msgid, IPC_RMID, NULL);
        return 0;
}
```

OUTPUT



```
(base) sec@sec-ThinkPad-E15-Gen-4:~/os/ex04/Linux-IPC-Message-Queues$ ./writer.o
Write Data : Helloworld
Data send is : Helloworld
(base) sec@sec-ThinkPad-E15-Gen-4:~/os/ex04/Linux-IPC-Message-Queues$ gcc -o reader.o reader.c
(base) sec@sec-ThinkPad-E15-Gen-4:~/os/ex04/Linux-IPC-Message-Queues$ ./reader.o
Data Received is : Helloworld
(base) sec@sec-ThinkPad-E15-Gen-4:~/os/ex04/Linux-IPC-Message-Queues$ ipcs

------ Message Queues --------
key        msqid      owner      perms      used-bytes   messages

------ Shared Memory Segments --------
key        shmid      owner      perms      bytes      nattch     status
0x00000000 6          sec        600        524288     2          dest
0x00000000 9          sec        600        524288     2          dest
0x00000000 13         sec        600        524288     2          dest
0x00000000 14         sec        600        4194304    2          dest
0x00000000 15         sec        600        524288     2          dest
0x00000000 20         sec        606        11277600   2          dest
0x00000000 21         sec        606        11277600   2          dest
0x00000000 25         sec        600        4194304    2          dest
0x00000000 26         sec        600        102400     2          dest
0x00000000 27         sec        600        102400     2          dest
0x00000000 28         sec        600        528384     2          dest
0x00000000 29         sec        600        528384     2          dest
0x00000000 30         sec        600        45056      2          dest
0x00000000 31         sec        600        45056      2          dest
0x00000000 34         sec        600        393216     2          dest
0x00000000 35         sec        600        393216     2          dest
0x00000000 36         sec        600        73728      2          dest
0x00000000 37         sec        600        73728      2          dest
0x00000000 40         sec        600        524288     2          dest

------ Semaphore Arrays --------
key        semid      owner      perms      nsems
```

RESULT:

The programs are executed successfully.