WEEK 9

1.Write a C program to create a message queue with read and Write permissions to Write 3 messages to it with different priority numbers.

Program

```
#include <stdio.h>
#include <sys/ipc.h>
#include <fcntl.h>
#define MAX 255
      struct mesg
      {
          long type;
          char mtext[MAX];
      }  *mesg;
      char buff[MAX];
main()
{
      int mid,fd,n,count=0;;
      if((mid=msgget(1006,IPC_CREAT | 0666))<0)
      {
            printf("\n Can't create Message Q");
            exit(1);
      }
      printf("\n Queue id:%d", mid);
      mesg=(struct mesg *)malloc(sizeof(struct mesg));
      mesg ->type=6;
      fd=open("fact",O_RDONLY);
      while(read(fd,buff,25)>0)
      {
            strcpy(mesg ->mtext,buff);
            if(msgsnd(mid,mesg,strlen(mesg ->mtext),0)== -1)
                printf("\n Message Write Error");
      }

      if((mid=msgget(1006,0))<0)
      {
            printf("\n Can't create Message Q");
            exit(1);
      }
      while((n=msgrcv(mid,&mesg,MAX,6,IPC_NOWAIT))>0)
            write(1,mesg.mtext,n);
            count  ;
      if((n= = -1)&(count= =0))
            printf("\n No Message Queue on Queue:%d",mid);

}
```

Output

```
Student@ubuntu:~$ gcc msgq.c
Student@ubuntu:~$ cat > fact
Hello
Hai
Welcome
^z
Student@ubuntu:~$ ./msgq.out
Queue id :0
Mesgq created
Hello
Hai
welcome
```

2.Write a C program that receives the messages(From the above message queue as specified in (21)) and display them.

Program

```
#include <sys/types.h>
#include <sys/ipc.h>
#include <sys/msg.h>
#include <stdio.h>
#define MSGSZ 128
/* * Declare the message structure. */
typedef struct msgbuf {
  long   mtype;
  char   mtext[MSGSZ];
} message_buf;
main()
{
  int msqid;
  key_t key;
  message_buf  rbuf;
  /*   * Get the message queue id for the    * "name" 1234, which was created by    * the server.    */
  key = 1234;
  if ((msqid = msgget(key, 0666)) < 0) {
    perror("msgget");
    exit(1);
  }
  /*   * Receive an answer of message type 1.    */if (msgrcv(msqid, &rbuf, MSGSZ, 1, 0) < 0) {
    perror("msgrcv");
    exit(1);
 }
  /*   * Print the answer.    */
  printf("%s\n", rbuf.mtext);
  exit(0);
}
```

Output

```
student@gcet ~]$cc message_send.c
[student@gcet ~]$ mv a.out msgsend
[student@gcet ~]$  ./msgsend
msgget: Calling msgget(0x4d2,01666)
msgget: msgget succeeded: msqid = 0
msgget: msgget succeeded: msqid = 0
msgget: msgget succeeded: msqid = 0
Message: "Did you get this?" Sent
[student@gcet ~]$ cc message_rec.c
[student@gcet ~]$ mv a.out msgrec
[student@gcet ~]$./msgrec
```

THANK YOU