Aim: To implement ipc using message queuing with echo server.

Description: In this program echo server is used.First in client.c a message is taken from user and echoed.Later the same message is echoed and printed onto the console when the server.c is run.

Sorce code:

Client.c
```c
#include <stdio.h>
#include <stdlib.h>
#include <sys/msg.h>
#include <sys/types.h>
#include <sys/ipc.h>
#include <unistd.h>
#include <string.h>
#define MAXSIZE 80
typedef struct
{
   long  mtype;
   char mdata[MAXSIZE];
}mymsg_t;


int main()
{
    int mqid,n;
    int key=9823;
    mymsg_t mymsg;
    char buff[MAXSIZE];
     mqid=msgget(key,0666|IPC_CREAT);
    if(mqid<0)
     {
       perror("msget() error:");
       exit(-1);
     }
    printf("\nMessage queue has been created/opened and its qid=%d\n",mqid);
    printf("Enter a message,(Client):\n");
    n=read(0,&buff,MAXSIZE);
    buff[n]='\0';
    strcpy(mymsg.mdata,buff);
    printf("Message sent was:%s\n",mymsg.mdata);
    mymsg.mtype=1;
    n=msgsnd(mqid,&mymsg,MAXSIZE,0);
if(n<0)
     {
        perror("msgsnd() error:");
        exit(-1);
     }
```

```
      n=msgrcv(mqid,&mymsg,MAXSIZE,2,0);
      printf("message Echoed (Client): %s\n",mymsg.mdata);
       return 0;
   }
```

server.c

```
#include <stdio.h>
#include <stdlib.h>
#include <sys/msg.h>
#include <sys/types.h>
#include <sys/ipc.h>
#include <unistd.h>
#define MAXSIZE 80
typedef struct
{
   long  mtype;
   char mdata[MAXSIZE];
}mymsg_t;

int main()
{
    int mqid,n,i;
    int key=9823;
    mymsg_t mymsg;
    char buff[MAXSIZE];
    mqid=msgget(key,0666);
    if(mqid<0)
     {
       perror("msget() error:");
       exit(-1);
     }
     printf("Message queue is created and its id: %d\n",mqid);
      n=msgrcv(mqid,&mymsg,MAXSIZE,1,0);
      if(n<0)
       {
          perror("msgrcv() error:");
          exit(-1);
       }
      printf("Message received(Server):%s\n",mymsg.mdata);
      for(i=0;mymsg.mdata[i]!='\0';++i)
      mymsg.mdata[i]=toupper(mymsg.mdata[i]);

  mymsg.mdata[i]='\0';
      mymsg.mtype=2;
      n=msgsnd(mqid,&mymsg,MAXSIZE,0);
```

```
    if(n<0)
    {
        perror("msgsnd() error:");
        exit(-1);
    }

    return 0;

}
```
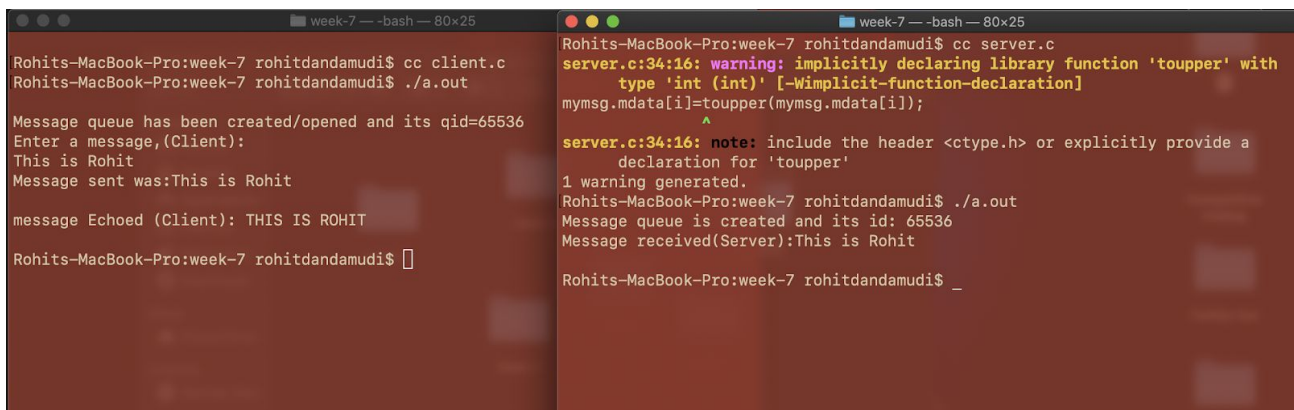
Result Analysis:

Thus the aim is achieved using message queuing with echo server

Output: