

---

---

<b>EX NO: 6B</b>	<b>IPC – CHAT MESSAGING</b>
<b>DATE:</b>	

---

---

## AIM

To exchange message between server and client using message queue.

## ALGORITHM

### Server

1. Declare a structure *mesgq* with *type* and *text* fields.
2. Initialize *key* to 2013 (some random value).
3. Create a message queue using *msgget* with *key* & *IPC\_CREAT* as parameter.
  - a. If message queue cannot be created then stop.
4. Initialize the message *type* member of *mesgq* to 1.
5. Do the following until user types Ctrl+D
  - a. Get message from the user and store it in *text* member.
  - b. Delete the new line character in *text* member.
  - c. Place message on the queue using *msgsend* for the client to read.
  - d. Retrieve the response message from the client using *msgrcv* function e. Display the *text* contents.
6. Remove message queue from the system using *msgctl* with *IPC\_RMID* as parameter.
7. Stop

### Client

1. Declare a structure *mesgq* with *type* and *text* fields.
  2. Initialize *key* to 2013(same value as in server).
  3. Open the message queue using *msgget* with *key* as parameter.
    - a. If message queue cannot be opened then stop.
  4. Do while the message queue exists
    - a. Retrieve the response message from the server using *msgrcv* function b. Display the *text* contents.
    - c. Get message from the user and store it in *text* member.
    - d. Delete the new line character in *text* member.
    - e. Place message on the queue using *msgsend* for the server to read.
  5. Print "Server Disconnected".
  6. Stop.
- 
-

---

---

62

---

## PROGRAM

### Server

```
/*Serverchatprocess-srvmsg.c*/
#include<stdio.h>
#include<stdlib.h>
#include<string.h>
#include<sys/types.h>
#include<sys/ipc.h>
#include<sys/msg.h>
structmesgq
{ longtype;
  chartext[200];
}mq;

main()
{ int msqid,len;
  key_tkey=2013;

  if((msqid=msgget(key,0644|IPC_CREAT))== -1)
  { perror("msgget");
    exit(1);
  }

  printf("Enter text to quit:\n"); mq.type=1;

  while(fgets(mq.text,sizeof(mq.text),stdin)!=NULL)
  { len=strlen(mq.text);
    if(mq.text[len-1]=='\n') mq.text[len-1]='\0'; msgsnd(msqid,&mq,len+1,0);

    msgrcv(msqid,&mq,sizeof(mq.text),0,0); printf("FromClient: \"%s\"\n",mq.text);
  }
  msgctl(msqid,IPC_RMID,NULL);
}
```

### Client

```
/*Clientchatprocess-climsg.c*/

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

---

```
#include<string.h>
#include<sys/types.h>
#include<sys/ipc.h>
#include<sys/msg.h>
```

```

struct msgq
{
    long type;
    char text[200];
} mq;

main()
{
    int msqid, len;
    key_t key = 2013;

    if((msqid = msgget(key, 0644)) == -1)
    {
        printf("Server not active\n");
        exit(1);
    }

    printf("Client ready:\n");
    while(msgrcv(msqid, &mq, sizeof(mq.text), 0, 0) != -1)
    {
        printf("From Server: \"%s\"\n", mq.text);

        fgets(mq.text, sizeof(mq.text), stdin);
        len = strlen(mq.text);
        if(mq.text[len-1] == '\n') mq.text[len-1] = '\0';
        msgsnd(msqid, &mq, len+1, 0);
    }
    printf("Server disconnected\n");
}

```

## OUTPUT:

### //CHAT SERVER

```
mohamedinam@Mohamed-Inam-PC: ~  
mohamedinam@Mohamed-Inam-PC:~$ gcc chatserver.c -o chats  
mohamedinam@Mohamed-Inam-PC:~$ ./chats  
Enter text, ^D to quit:  
hi  
how  
From Client: "hello"  
how r u  
From Client: "how r u"  
fyn  
From Client: "fyn"  
█
```

### //CHAT CLIENT

```
mohamedinam@Mohamed-Inam-PC: ~  
mohamedinam@Mohamed-Inam-PC:~$ gcc chatclient.c -o chatc  
mohamedinam@Mohamed-Inam-PC:~$ ./chatc  
Client ready :  
From Server: "hi"  
hello  
From Server: "how"  
how r u  
From Server: "how r u"  
fyn  
From Server: "fyn"  
█
```

---

## **RESULT**

Thus chat session between client and server was done using message queue.