P1.c

1 #include<stdio.h>

2 #include<sys/msg.h>

3 #include<string.h>

4 struct msgbuf

5 {

6 long mtype; //message type

7 char mtext[20]; //message text

8 };

9

10 main(int argc,char \*argv[])

11 {

12 int id,ret;

13 struct msgbuf v;

14 id=msgget(5,IPC\_CREAT|0664);

15 if(id<0)

16 {

17 perror("msgget");

18 return;

19 }

20

21 if(argc!=3)

22 {

23 printf("./a.out mtype msg\n");

24 return;

25 }

26 v.mtype=atoi(argv[1]);

27 strcpy(v.mtext,argv[2]);

28

29 ret=msgsnd(id,&v,strlen(v.mtext)+1,0);

30 if(ret==-1)

31 {

32 perror("msgsnd");

33 return;

34 }

35 printf("message send successfully...\n");

36 }

P2.c

1 #include<stdio.h>

2 #include<sys/msg.h>

3 #include<string.h>

4 //#define MSG\_EXCEPT 020000

5 struct msgbuf

6 {

7 long mtype; //message type

8 //char mtext[20]; //message text

9 char mtext[5];

10 };

11

12 main(int argc,char \*argv[])

13 {

14 int id,ret;

15 struct msgbuf v;

16 id=msgget(5,IPC\_CREAT|0664);

17 if(id<0)

18 {

19 perror("msgget");

20 return;

21 }

22

23 if(argc!=2)

24 {

25 printf("./a.out mtype\n");

26 return;

27 }

28 //ret=msgrcv(id,&v,sizeof(v.mtext),atoi(argv[1]),0);

29 //ret=msgrcv(id,&v,sizeof(v.mtext),atoi(argv[1]),IPC\_NOWAIT);

30 //ret=msgrcv(id,&v,sizeof(v.mtext),atoi(argv[1]),MSG\_EXCEPT);

31 ret=msgrcv(id,&v,sizeof(v.mtext),atoi(argv[1]),MSG\_NOERROR);

32 if(ret==-1)

33 {

34 perror("msgrcv");

35 return;

36 }

37

38 printf("msg received from queue is:mtype:%d message:%s\n",v.mtype,v.mtext);

39 }

P3.c

1 //wap to find the size of the message queue?

2 #include<stdio.h>

3 #include<sys/msg.h>

4 main()

5 {

6 int id;

7 struct msqid\_ds v;

8 id=msgget(5,IPC\_CREAT|0664);

9 if(id<0)

10 {

11 perror("msgget");

12 return;

13 }

14

15 msgctl(id,IPC\_STAT,&v);

16 printf("size of Msgq:%d\n",v.msg\_qbytes);

17

18 }