**Implement two processes – a client process and a server process,**

**which reside on the same machine. The server process sets up a**

**message queue for communication with the client process. The client**

**process sends the server process some commands (standard linux**

**commands like ls, ps-eaf) to be executed by putting an appropriate**

**message in the message queue. The server executes the command and**

**returns the result to the client process, which it outputs to a file**

**BY USING MESSAGE QUEUE**

**Writers code:-**

#include <stdio.h>

#include <sys/ipe.h>

#include <sys/msg h>

#include <stdlib.h>

#include <string.h>

#include "queue.h"

void report\_and\_exit(const char\* msg) {

perror(msg);

exit(-1);

{

int main()

key\_t key = ftok/PathName ProjectId).

if (key<0) report\_and\_exit("couldn't get key...");

int qid = msgget(key, 0666 IPC\_CREAT)

if (qid <0) report\_and\_exit("couldn't get queue id...");

char payloads[] = ["msg1", "msg2", "msg3", "msg4", "msg5", "msg6"} int types[] (1, 1, 2, 2, 3.3%) ;

for (i=0; i<MsgCount; i++)

queuedMessage msg.

msg.type = types[i].

strepy(msg.payload, payloads[i]):

msgsnd(qid, &msg, sizeof(msg), IPC\_NOWAIT)

printf("%s sent as type "in", msg payload, (int) msg.type).

return 0;

}

**READERS CODE:-**

#include <stdio.h>

#include <sys/ipc.h>

#include <sys/msg.h>

#include <stdlib.h>

#include "queue.h"

void report\_and\_exit(const char\* msg) {

perror(msg).

exit(-1)

{

int main()

key\_t key=ftok(PathName ProjectId);

if (key<0) report\_and\_exit("key not gotten..."); int qid = msgget(key, 0666 IPC\_CREAT);

if (qid <0) report\_and\_exit("no access to queue\_").

int types[]=(3,1,2, 1, 3.2), int i;

for (i=0; i<MsgCount; i++) queuedMessage msg.defined in queue.h

if (msgrev(qid, &msg, sizeof(msg), types[1], MSG\_NOERROR IPC\_NOWAIT)<0) puts("msgrev trouble..."). printf("%s received as type "in" msg payload, (int) msg type),

remove the queue

if (msget(qid IPC\_RMID, NULL)<0) / NULL - 'no flags" \*/

report\_and\_exit("trouble removing queue").

return 0;

}

***OUTPUT:-***

