**Shared Memory:**

#include <sys/types.h>

#include <sys/ipc.h>

#include <sys/shm.h>

#include <stdio.h>

#include <stdlib.h>

#define MAXSIZE 27

void die(char \*s)

{

perror(s);

exit(1);

}

int main()

{

char c;

int shmid;

key\_t key;

char \*shm, \*s;

key = 5678;

if ((shmid = shmget(key, MAXSIZE, IPC\_CREAT | 0666)) < 0)

die("shmget");

if ((shm = shmat(shmid, NULL, 0)) == (char \*) -1)

die("shmat");

/\*

\* \* Put some things into the memory for the

\* other process to read.

\* \*/

s = shm;

for (c = 'a'; c <= 'z'; c++)

\*s++ = c;

/\*

\* Wait until the other process

\* changes the first character of our memory

\* to '\*', indicating that it has read what

\* we put there.

\*/

while (\*shm != '\*')

sleep(1);

exit(0);

}

//SHMClient.C

#include <sys/types.h>

#include <sys/ipc.h>

#include <sys/shm.h>

#include <stdio.h>

#include <stdlib.h>

#define MAXSIZE 27

void die(char \*s)

{

perror(s);

exit(1);

}

int main()

{

int shmid;

key\_t key;

char \*shm, \*s;

key = 5678;

if ((shmid = shmget(key, MAXSIZE, 0666)) < 0)

die("shmget");

if ((shm = shmat(shmid, NULL, 0)) == (char \*) -1)

die("shmat");

//Now read what the server put in the memory.

for (s = shm; \*s != '\0'; s++)

putchar(\*s);

putchar('\n');

/\*

\*Change the first character of the

\*segment to '\*', indicating we have read

\*the segment.

\*/

\*shm = '\*';

exit(0);

}

**Message Passing:**

#include <sys/types.h>

#include <sys/ipc.h>

#include <sys/msg.h>

#include <stdio.h>

#include <string.h>

#include <stdlib.h>

#define MAXSIZE 128

void die(char \*s)

{

perror(s);

exit(1);

}

typedef struct msgbuf

{

long mtype;

char mtext[MAXSIZE];

};

main()

{

int msqid;

int msgflg = IPC\_CREAT | 0666;

key\_t key;

struct msgbuf sbuf;

size\_t buflen;

key = 1234;

if ((msqid = msgget(key, msgflg )) < 0) //Get the message queue ID for the given key

die("msgget");

//Message Type

sbuf.mtype = 1;

printf("Enter a message to add to message queue : ");

scanf("%[^\n]",sbuf.mtext);

getchar();

buflen = strlen(sbuf.mtext) + 1 ;

if (msgsnd(msqid, &sbuf, buflen, IPC\_NOWAIT) < 0)

{

printf ("%d, %d, %s, %d\n", msqid, sbuf.mtype, sbuf.mtext, buflen);

die("msgsnd");

}

else

printf("Message Sent\n");

exit(0);

}

//IPC\_msgq\_rcv.c

#include <sys/types.h>

#include <sys/ipc.h>

#include <sys/msg.h>

#include <stdio.h>

#include <stdlib.h>

#define MAXSIZE 128

void die(char \*s)

{

perror(s);

exit(1);

}

typedef struct msgbuf

{

long mtype;

char mtext[MAXSIZE];

} ;

main()

{

int msqid;

key\_t key;

struct msgbuf rcvbuffer;

key = 1234;

if ((msqid = msgget(key, 0666)) < 0)

die("msgget()");

//Receive an answer of message type 1.

if (msgrcv(msqid, &rcvbuffer, MAXSIZE, 1, 0) < 0)

die("msgrcv");

printf("%s\n", rcvbuffer.mtext);

exit(0);

}