**CODE MẪU**

**SHARE MEMORY – MESSAGE QUEUE**

**Share memory:**

#include <stdio.h>

#include <unistd.h>

#include <limits.h>

#include <string.h>

#include <stdlib.h>

#include <sys/types.h>

#include <sys/ipc.h>

#include <sys/shm.h>

#define SIZE 256

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

{

int \*shm, shmid, k,pid;

key\_t key;

if((key=ftok(".","a"))==-1){

perror("Key created.\n");

return 1;

}

shmid = shmget(key, SIZE, IPC\_CREAT | 0666);

if (shmid == -1) {

perror("Shared memory created.\n");

return 2;

}

shm = (int\*) shmat(shmid, 0, 0);

pid = fork();

if(pid==0) { // child

shm[0] = atoi(argv[1]);

shm[1] = atoi(argv[2]);

sleep(3);

printf("%d + %d = %d\n", shm[0], shm[1], shm[2]);

shmdt((void\*) shm);

shmctl(shmid, IPC\_RMID, (struct shmid\_ds\*) 0);

return 0;

}

else if(pid >0) { // parent

sleep(1);

shm[2] = shm[1] + shm[0];

shmdt((void\*) shm);

sleep(5);

return 0;

}

else { perror("Fork failed."); return 4; }

return 0;

}

**Message queue:**

**File: Writer.c**

// C Program for Message Queue (Writer Process)

#include <stdio.h>

#include <sys/ipc.h>

#include <sys/msg.h>

#include <string.h>

// structure for message queue

struct mesg\_buffer {

long mesg\_type;

char mesg\_text[100];

} message;

int main()

{

key\_t key;

int msgid;

// ftok to generate unique key

key = ftok("msg.txt",1);

// msgget creates a message queue

// and returns identifier

msgid = msgget(key, 0666 | IPC\_CREAT);

message.mesg\_type = 1;

printf("Write Data : ");

fgets(message.mesg\_text, sizeof(message.mesg\_text), stdin);

// msgsnd to send message

msgsnd(msgid, &message, sizeof(message),0);

// display the message

printf("Data send is : %s \n", message.mesg\_text);

return 0;

}

**File reader.c**

// C Program for Message Queue (Reader Process)

#include <stdio.h>

#include <sys/ipc.h>

#include <sys/msg.h>

#include <string.h>

// structure for message queue

struct mesg\_buffer {

long mesg\_type;

char mesg\_text[100];

} message;

int main()

{

key\_t key;

int msgid;

// ftok to generate unique key

key = ftok("msg.txt",1);

// msgget creates a message queue

// and returns identifier

msgid = msgget(key, 0666 | IPC\_CREAT);

// msgrcv to receive message

msgrcv(msgid, &message, sizeof(message),1,0);

// display the message

printf("Data Received is : %s \n", message.mesg\_text);

// to destroy the message queue

msgctl(msgid, IPC\_RMID, NULL);

return 0;

}