***Aim:***

To Implement Interprocess communication using shared memory.

***Description:***

1.shm\_open():

Header:

#include <sys/mman.h>

#include <sys/stat.h> /\* For mode constants \*/

#include <fcntl.h> /\* For O\_\* constants \*/

Syntax:

int shm\_open(const char \*name, int oflag, mode\_t mode);

Description:

shm\_open() creates and opens a new, or opens an existing, POSIX

shared memory object. A POSIX shared memory object is in effect

a handle which can be used by unrelated processes to mmap(2) the

same region of shared memory. The shm\_unlink() function performs

the converse operation, removing an object previously created by

shm\_open().

2.*shmget():*

Header:

#include <sys/shm.h>

Syntax:

int shmget(key\_t key, size\_t size, int shmflg);

Description:

shmget() returns the identifier of the System V shared memory

segment associated with the value of the argument key. It may be

used either to obtain the identifier of a previously created

shared memory segment (when shmflg is zero and key does not have

the value IPC\_PRIVATE), or to create a new set.

3.***shmat():***

Header:

#include <sys/shm.h>

Syntax:

void \*shmat(int shmid, const void \*shmaddr, int shmflg);

Description:

The shmat() function attaches the shared memory segment

associated with the shared memory identifier specified by shmid

to the address space of the calling process. The segment is

attached at the address specified by one of the following

criteria:

***Code:***

***InterProcSend.c***

#include <stdio.h>

#include <sys/types.h>

#include <unistd.h>

#include <string.h>

#include <fcntl.h>

#include <sys/shm.h>

#include <sys/stat.h>

#include <sys/mman.h>

#include <stdlib.h>

//typedef struct values{

// char\* something;

// int val;

//}val;

int main(){

char\* SharedMemName="Sender";

char\* message="Hi There!!\n";

val\* v1;

v1=(val\*)malloc(sizeof(val));

v1->something=message;

v1->val=2334;

int shmFD;

void\* ptr;

shmFD=shm\_open(SharedMemName,O\_CREAT|O\_RDWR,0666);

if(shmFD==-1){

write(1,"[+][shm\_open]Failed To Create Shared Memory",40);

return -1;

}

ftruncate(shmFD,4096);

ptr=mmap(0,4096,PROT\_WRITE,MAP\_SHARED,shmFD,0);

sprintf(ptr,"%s",message);

ptr+=strlen(message);

}

***InterProcRecv.c:***

#include <stdio.h>

#include <sys/mman.h>

#include <sys/types.h>

#include <sys/stat.h>

#include <fcntl.h>

#include <sys/shm.h>

#include <unistd.h>

int main(){

char buf[10];

char\* SendShm="Sender";

int shmFD;

shmFD=shm\_open(SendShm,O\_RDONLY,0666);

void\* ptr=mmap(0,4096,PROT\_READ,MAP\_SHARED,shmFD,0);

printf("%s",(char\*)ptr);

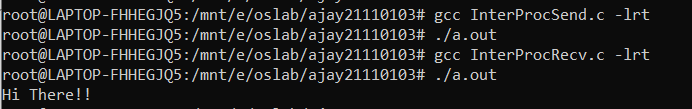
//read(shmFD,buf,10);

//write(1,buf,10);

shm\_unlink("Sender");

}

OUTPUT:



2.***IPCS1.c:***

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include <sys/types.h>

#include <sys/stat.h>

#include <unistd.h>

#include <sys/shm.h>

typedef struct test\_shm{

char\* something;

int value;

}t\_s;

int main(){

key\_t key=1234;

int id=shmget(key,1024,IPC\_CREAT|0644);

printf("Shared Memory ID:%d",id);

t\_s\*ptr=shmat(id,NULL,0);

t\_s \*entry;

entry=(t\_s\*)malloc(sizeof(t\_s));

entry->something="hi";

entry->value=29;

memcpy(ptr,entry,sizeof(t\_s));

printf("%s\n%d\n",ptr->something,ptr->value);

}

Output:



***Result***:

Thus Inter Process Communication was Established using shm.h library functions (shm\_get,shmat).