**Name**: Akshat Rajesh Andhale **Class**: TEIT

**Roll** **no**.: 03

**Assignment no.: 07(B):** Inter-process Communication using Shared Memory using System V

**CODE**:

**1) CLIENT:**

#include<sys/shm.h>

#include<stdio.h>

#include<stdlib.h>

#include<string.h>

#include<unistd.h>

#define data\_not\_filled -1

#define data\_filled 0

#define data\_read\_client 1

typedef struct data

{

int status;

char buff[100];

}data1;

int main()

{

int shm\_id, ret\_val;

key\_t key;

data1 \*shm\_ptr;

data1 \*s;

key=ftok(".",'A');

shm\_id=shmget (key, 0, 0666);

if (shm\_id<0)

{

printf ("\n shmget error");

exit(-1);

}

else

{

printf ("\n shared memory id is=%d", shm\_id);

}

shm\_ptr=(data1 \*) shmat (shm\_id,0,0);

system("ipcs -m");

if ((int) shm\_ptr==-1)

{

printf ("\n Error for shmat");

exit(-1);

}

while (shm\_ptr->status==data\_not\_filled)

{

printf("\n Client waiting for message");

sleep (3);

}

printf ("\n Contents recieved by client is::\n");

printf ("%s", shm\_ptr->buff);

shm\_ptr->status=data\_read\_client;

ret\_val=shmdt((void \*) shm\_ptr);

if (ret\_val==0)

{

printf ("\n shared memory detached");

}

else

{

printf ("\n shmdt error");

}

return 0;

}

**OUTPUT:**

shared memory id is=1234567

------ Shared Memory Segments --------

key shmid owner perms bytes nattch status

0x00000000 1234567 user 666 100 1

Client waiting for message

Client waiting for message

Client waiting for message

Contents received by client is::

hello world

shared memory detached

**2) SERVER:**

#include<sys/shm.h>

#include<stdio.h>

#include<stdlib.h>

#include<string.h>

#include<unistd.h>

#define data\_not\_filled -1

#define data\_filled 0

#define data\_read\_client 1

typedef struct data {

    int status;

    char buff [100];

}data1;

int main ()

{

    int shm\_id, ret\_val, i=0;

    key\_t key;

    char ch;

    data1 \*shm\_ptr;

    data1 \*s;

    key=ftok(".",'A');

    shm\_id=shmget (key, sizeof (data1), IPC\_CREAT | 0666) ;

    if (shm\_id<0)

    {

        printf("\n shmget error");

        exit(-1);

    }

    else

    {

        printf("\n Shared memory created");

        printf("\n Shared memory id is =%d", shm\_id);

    }

    system("ipcs -m");

    shm\_ptr=(data1 \*) shmat (shm\_id,0,0);

    system("ipcs -m");

    if ((int) shm\_ptr==-1)

    {

        printf("\n Error for shmat");

        exit(-1);

    }

    shm\_ptr->status=data\_not\_filled;

    printf("\n Enter data\n");

    ch=getchar();

    while (ch!='#')

    {

        shm\_ptr->buff[i]=ch;

        i++;

        ch=getchar();

    }

    shm\_ptr->buff[i]='\0';

    shm\_ptr->status=data\_filled;

    while (shm\_ptr->status != data\_read\_client)

    {

        printf("\n Server waiting for termination");

        sleep(1);

    }

    ret\_val=shmdt( (void \*) shm\_ptr);

    if (ret\_val==0)

    {

        printf("\n shared memory detached");

    }

    else

    {

        printf("\n shmdt error");

    }

    if (shmctl (shm\_id, IPC\_RMID, 0) < 0)

    {

        printf("shmctl error");

        exit(0);

    }

    return 0;

}

**OUTPUT:**

Shared memory created

Shared memory id is =1234567

------ Shared Memory Segments --------

key shmid owner perms bytes nattch status

0x00000000 1234567 user 666 100 0

Enter data:

hello world#

------ Shared Memory Segments --------

key shmid owner perms bytes nattch status

0x00000000 1234567 user 666 100 1

Server waiting for termination

Server waiting for termination

Server waiting for termination

(shared memory will detach after client reads the data)

Shared memory detached