SSN COLLEGE OF ENGINEERING, KALAVAKKAM
(An Autonomous Institution, Affiliated to Anna University, Chennai)
SSN College of Engineering

Department of Computer Science and Engineering

UCS1411 – Operating Systems Laboratory

II Year CSE - A Section (IV Semester)

Exercise – 5- Inter process communication

Lab Exercise 5 InterProcess Communications using Shared Memory

<u>Study the following system calls.</u>
Shared memory - shmget, shmat, shmdt, shmctl

<u>Additional optional system calls</u>
Pipe - pipe, mkfifo, mknod, open, read, write, close
Message Queue – msgget, msgsnd, msgrcv, msgctl

Aim:
Develop the following applications that uses interprocess communication concepts using shared memory.

1.Develop an application for getting a name in parent and convert it into uppercase in child using shared memory.
2.Develop an client / server application for file transfer using shared memory.
3. Develop an client/server chat application using shared memory.

<u>Comment:</u>

This program should be run as a single program in which parent is one process and child is another process.

<u>Client / server:</u>

Keep common code and parent code in one program as server.c

Keep common code and child code in another program as client.c

Run server.c in one terminal and client.c in another terminal. Communicate between the two processes.

Example :

Shared memory :

```
#include <sys/ipc.h>
# define NULL 0
#include <sys/shm.h>
#include <sys/types.h>
# include<unistd.h>
# include<stdio.h>
# include<stdlib.h>
# include<string.h>
#include <sys/wait.h>
#include <stdio_ext.h>
// parent writing a char in shared memory and child reads it and prints it.
int main()
{
        int pid;
        char *a,*b,c;
        int id,n,i;
        // you can create a shared memory between parent and child here or you can //create
        inside them separately.
        id=shmget(111,50,IPC_CREAT | 00666);
        pid=fork();
        if(pid>0) //parent
                {
                        // id=shmget(111,50,IPC_CREAT | 00666);
                        a=shmat(id,NULL,0);
                        a[0]='d';
                        wait(NULL);
                        shmdt(a);
                }
        else //child
                {
                        sleep(3);
                        //id=shmget(111,50,0);
                        b=shmat(id,NULL,0);
                        printf("\n child %c\n",b[0]);
                        shmdt(b);
                }
        shmctl(id, IPC_RMID,NULL);
}
```