****

**LAB EXERCISE 5**

**Inter Process Communication**

Name: Jayannthan P T

Dept: CSE ‘A’

Roll No.: 205001049

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.

**Code:**

*/\*Develop an application for getting a name in parent and convert it into uppercase in child using shared memory.\*/*

*/\*Server Code\*/*

#include <sys/ipc.h>

# define NULL 0

# include <sys/shm.h>

# include <sys/types.h>

# include<unistd.h>

# include<stdio.h>

# include<string.h>

# include<stdlib.h>

# include<string.h>

# include <sys/wait.h>

# include <stdio\_ext.h>

**int** main()

{

*// ftok to generate unique key*

**int** key = ftok("abc",67);

*// shmget returns an identifier in shmid*

**int** shmid = shmget(key,1024,0666|IPC\_CREAT);

*// shmat to attach to shared memory*

**char** \*str = (**char**\*) shmat(shmid,(**void**\*)0,0);

    printf("Write Data : ");

    scanf("%s",str);

    printf("Data written in memory: %s\n",str);

*//detach from shared memory*

    shmdt(str);

    return 0;

}

*/\*Develop an application for getting a name in parent and convert it into uppercase in child using shared memory.\*/*

*/\*Client Code\*/*

#include <sys/ipc.h>

# define NULL 0

#include <sys/shm.h>

#include <sys/types.h>

# include<unistd.h>

# include<stdio.h>

# include<string.h>

# include<stdlib.h>

# include<string.h>

#include <sys/wait.h>

#include <stdio\_ext.h>

**int** main()

{

*// ftok to generate unique key*

**int** key = ftok("abc",67);

*// shmget returns an identifier in shmid*

**int** shmid = shmget(key,1024,0666|IPC\_CREAT);

*// shmat to attach to shared memory*

**char** \*str = (**char**\*) shmat(shmid,(**void**\*)0,0);

    printf("Data read from memory: %s\n",(str));

    for (**int** i = 0; str[i]!='\0'; i++)

    {

        if(str[i] >= 'a' && str[i] <= 'z')

        {

            str[i] = str[i] -32;

        }

    }

    printf("Data converted to uppercase: %s\n",(str));

*//detach from shared memory*

    shmdt(str);

*// destroy the shared memory*

    shmctl(shmid,IPC\_RMID,NULL);

    return 0;

}

**Output:**

2.Develop a client / server application for file transfer using shared memory.

**Code:**

*/\*Server Code\*/*

#include <stdio.h>

#include <sys/stat.h>

#include <sys/types.h>

#include <fcntl.h>

#include <unistd.h>

#include <string.h>

#include<math.h>

#define FIFO\_FILE "/tmp/filesharingspace"

**void** delay()

{

    for (**int** c = 1; c <= 32767; c++)

            for (**int** d = 1; d <= 32767; d++)

                {}

}

**int** main()

{

**int** fd;

**char** readbuf[1024];

**char** end[10];

**int** to\_end;

**int** read\_bytes;

**char** str[30];

    mkfifo(FIFO\_FILE, S\_IFIFO|0640);

    strcpy(end, "end");

    fd = open(FIFO\_FILE, O\_RDWR);

    while(1)

    {

            read\_bytes = read(fd, readbuf, sizeof(readbuf));

            readbuf[read\_bytes] = '\0';

            printf("\nFile request received: %s\nContents of file sent...\n",readbuf);

            to\_end = strcmp(readbuf, end);

            if (to\_end == 0)

            {

                close(fd);

                break;

            }

**int** file=open(readbuf,O\_RDONLY);

            if(file==-1)

            {

                strcpy(str,"File Not Found!!!!\n");

                sprintf(readbuf,"File Not Found!!!");

            }

            else

            {

                strcpy(str,"Sent!\n");

**char** contents[1024];

                read(file,contents,1024);

                close(file);

                strcpy(readbuf,contents);

            }

            delay();

            printf("%s",str);

            write(fd, readbuf, strlen(readbuf));

    }

    return 0;

}

*/\*Client Code\*/*

#include <stdio.h>

#include <sys/stat.h>

#include <sys/types.h>

#include <fcntl.h>

#include <unistd.h>

#include <string.h>

#include <stdlib.h>

#define FIFO\_FILE "/tmp/filesharingspace"

**int** main()

{

**int** fd;

**int** end\_process;

**int** stringlen;

**int** read\_bytes;

**char** readbuf[1024];

**char** end\_str[5];

    printf("FIFO\_CLIENT: Request files from Server ( to end enter \"end\")\n");

    fd = open(FIFO\_FILE, O\_CREAT|O\_RDWR);

    strcpy(end\_str, "end");

    while (1)

    {

        printf("Enter filename: ");

        fgets(readbuf, sizeof(readbuf), stdin);

        stringlen = strlen(readbuf);

        readbuf[stringlen - 1] = '\0';

        if (strcmp(readbuf, end\_str)!= 0)

        {

            write(fd, readbuf, strlen(readbuf));

            printf("\nFile: %s\n",readbuf);

            printf("File Request Sent...\n");

            read\_bytes = read(fd, readbuf, sizeof(readbuf));

            readbuf[read\_bytes] = '\0';

            printf("Contents of File: \n%s\n\n",readbuf);

        }

        else

        {

            write(fd, readbuf, strlen(readbuf));

            close(fd);

            break;

        }

    }

   return 0;

}

**Output:**

1. Develop an client/server chat application using shared memory.

**Code:**

*/\*Server Code\*/*

#include <stdlib.h>

#include <string.h>

#include <sys/types.h>

#include <sys/ipc.h>

#include <sys/msg.h>

#include <stdio.h>

**struct** mesgq

{

**long** type;

**char** text[200];

} mq;

**void** main()

{

**int** len, msqid1;

**key\_t** key1 = 1000;

    if ((msqid1 = msgget(key1, 0644 | IPC\_CREAT)) == -1)

    {

        perror("msgget error client 1");

        exit(1);

    }

    printf("Server ready :\n");

    printf("PRESS CTRL+D to quit:\n\n");

    mq.type = 1;

    printf("You: ");

    while (fgets(mq.text, sizeof(mq.text), stdin) != NULL)

    {

        len = strlen(mq.text);

        if (mq.text[len - 1] == '\n')

            mq.text[len - 1] = '\0';

        msgsnd(msqid1, &mq, len + 1, 0);

            if (msgrcv(msqid1, &mq, sizeof(mq.text), 1, 0));

            printf("Client: ");

            printf("%s\n\n", mq.text);

            printf("You: ");

    }

    msgctl(msqid1, IPC\_RMID, NULL);

}

*/\*Client Code\*/*

#include<sys/msg.h>

#include<sys/ipc.h>

#include<sys/types.h>

#include<string.h>

#include<stdlib.h>

#include<stdio.h>

**struct** mesgq{

**long** type;

**char** text[200];

}mq;

**void** main(){

**int** msqid1, len;

**key\_t** key1 = 1000;

    if ((msqid1 = msgget(key1, 0644)) == -1){

        printf("Server not active\n");

        exit(1);

    }

    printf("Client 1 Ready!\n");

    while (msgrcv(msqid1, &mq, sizeof(mq.text), 1, 0) != -1)

    {

        printf("\nServer: ");

        printf("%s\n", mq.text);

        printf("You: ");

        fgets(mq.text, sizeof(mq.text),stdin);

        len = strlen(mq.text);

        if (mq.text[len-1] == '\n')

        mq.text[len-1] = '\0';

        msgsnd(msqid1, &mq, len+1, 0);

    }

    printf("Server Disconnected\n");

}