

## OS LAB :IPC : SHARED MEMORY

---

April 6, 2020

### OBJECTIVE: TO IMPLEMENT IPC WITH THE HELP OF SHARED MEMORY CONCEPT

#### 1. Write a C program to perform the following:

One process creates a shared memory segment and writes a message ,Hello' into it. Another process opens the segment and reads the message that is ,Hello'. It will then display the message ,Hello' to the standard output device.

Hints for implementing shared memory :

**Step 1: INITIALIZE THE SHARED MEMORY (*shmget*)**

**Step 2: ATTACH (*shmat*)**

**Step 3: DETACH (*shmdt*)**

#### ➤ Program for 1<sup>st</sup> process

```
#include <sys/ipc.h>
#include <sys/shm.h>
#include <sys/types.h>
#include <stdio.h>
#include <string.h>

#define SHSIZE 100

int main()
{
    int retval, shmid;
    void *memory = NULL;
    char *p;
    key_t key= 9876;

    /*****INITIALIZATION*****/
    shmid = shmget (key, SHSIZE, IPC_CREAT | 0666);
    if(shmid <0)
    {
        printf("failure in shared memory initialization");
        perror("shmget");
    }
}
```

## OS LAB :IPC : SHARED MEMORY

---

April 6, 2020

```
        exit(1);
    }

    /*****ATTACHMENT*****/

    memory = shmat (shmid, NULL, 0);

    if (memory == NULL)
    {
        printf("failure in shared memory attachment");
        return 0;
    }

    /*****perform writting*****/

    p = (char *) memory;
    memset(p, '\0', SHMSIZE);
    memcpy (p, "HELLO", SHMSIZE );

    /*****DETACHMENT*****/

    retval = shmdt (p);

    if(retval<0)
    {
        printf("failure in shared memory detachment");
        exit(1);
    }
    return 0;
}
```

## OS LAB :IPC : SHARED MEMORY

---

April 6, 2020

- Program for 2<sup>nd</sup> process

```
#include <sys/ipc.h>
#include <sys/shm.h>
#include <sys/types.h>
#include <stdio.h>
#include <string.h>

#define SHSIZE 100

int main()
{
    int retval, shmid;
    void *memory = NULL;
    char *p;
    key_t key= 9876;

    /*****INITIALIZATION*****/
    shmid = shmget (key, SHSIZE, IPC_CREAT | 0666);
    if(shmid <0)
    {
        printf("failure in shared memory initialization");
        perror("shmget");
        exit(1);
    }

    /*****ATTACHMENT*****/

    memory = shmat (shmid, NULL, 0);

    if (memory == NULL)
    {
        printf("failure in shared memory attachment");
        exit(1);
    }
}
```

## OS LAB :IPC : SHARED MEMORY

---

April 6, 2020

```
    }

    /*****perform printing*****/

    p = (char *) memory;

    printf("the MESSAGE is %s \n", p);

    /*****DETACHMENT*****/

    retval = shmdt (p);

    if(retval<0)
    {
        printf("failure in shared memory detachment");
        exit(1);
    }

    /*****Deallocation of shared memory*****/

    retval = shmctl ( shmid, IPC_RMID, NULL);

    return 0;

}
```

[N.B: Like the previous lab assignments of named pipe I advise to use two separate terminals for execution.]