

---

---

<b>EX NO: 6A</b>	<b>SHARED MEMORY</b>
<b>DATE:</b>	

---

## AIM

To demonstrate communication between process using shared memory.

## ALGORITHM:

### Server

1. Initialize size of sharedmemory *shmsize* to 27.
2. Initialize *key* to 2013(somerandomvalue).
3. Create a sharedmemory segment using *shmget* with *key* & *IPC\_CREAT* as parameter.
  - a. If shared memory identifier *shmid* is -1, then stop.
4. Display *shmid*.
5. Attach server process to the sharedmemory using *shmat* with *shmid* as parameter.
  - a. If pointer to the sharedmemory is not obtained, then stop.
6. Clear contents of the shared region using *memset* function.
7. Write a-z onto the sharedmemory.
8. Wait till client reads the sharedmemory contents
9. Detach process from the sharedmemory using *shmdt* system call.
10. Remove sharedmemory from the system using *shmctl* with *IPC\_RMID* argument
11. Stop

### Client

1. Initialize size of sharedmemory *shmsize* to 27.
2. Initialize *key* to 2013(same value as in server).
3. Obtain access to the same sharedmemory segment using same *key*.
  - a. If obtained then display the *shmid* else print "Server not started"
4. Attach client process to the sharedmemory using *shmat* with *shmid* as parameter.
  - a. If pointer to the sharedmemory is not obtained, then stop.
5. Read contents of sharedmemory and print it.
6. After reading, modify the first character of sharedmemory to '\*'
7. Stop

---

---

58

---

**PROGRAM:****//SERVER//**

```
/*Sharedmemoryserver-shms.c*/

#include<stdio.h>
#include<stdlib.h>
#include<sys/un.h>
#include<sys/types.h>
#include<sys/ipc.h>
#include<sys/shm.h>
#define shmsize 27

main()
{ char c;
  int shm_id;
  key_t key = 2013;
  char *shm, *s;

  if((shm_id=shmget(key, shmsize, IPC_CREAT|0666))<0)
  { perror("shmget");
    exit(1);
  }
  printf("Sharedmemory id: %d\n", shm_id);

  if((shm=shmat(shm_id, NULL, 0))!=(char *)-1)
  { perror("shmat");
    exit(1);
  }

  memset(shm, 0, shmsize); s=shm;
  printf("Writing(a-z) onto shared memory\n");
  for(c='a'; c<='z'; c++) *s++=c;
  *s='\0';

  while(*shm!='*');
  printf("Client finished reading\n");

  if(shmdt(shm)!=0) fprintf(stderr, "Could not close memory segment.\n");

  shmctl(shm_id, IPC_RMID, 0);
}
```

**//CLIENT//**

---

```
#include<stdio.h>
#include<stdlib.h>
#include<sys/types.h>
```

---

```
#include<sys/ipc.h>
#include<sys/shm.h>

#define shmsize 27

main()
{
    int shmid;
    key_t key = 2013;
    char *shm, *s;

    if((shmid = shmget(key, shmsize, 0666)) < 0)
    {
        printf("Server not started\n");
        exit(1);
    }
    else
        printf("Accessing shared memory id: %d\n", shmid);

    if((shm = shmat(shmid, NULL, 0)) == (char *) -1)
    {
        perror("shmat");
        exit(1);
    }

    printf("Shared memory contents:\n");
    for(s = shm; *s != '\0'; s++)
        putchar(*s);
    putchar('\n');

    *shm = '*';
}
```

## OUTPUT:

### //SERVER

```
mohamedinam@Mohamed-Inam-PC: ~  
mohamedinam@Mohamed-Inam-PC:~$ gcc server.c -o server  
mohamedinam@Mohamed-Inam-PC:~$ ./server  
Shared memory id : 2719762  
Writing (a-z) onto shared memory  
Client finished reading  
mohamedinam@Mohamed-Inam-PC:~$ █
```

### //CLIENT

```
mohamedinam@Mohamed-Inam-PC: ~  
mohamedinam@Mohamed-Inam-PC:~$ gcc client.c -o client  
mohamedinam@Mohamed-Inam-PC:~$ ./client  
Accessing shared memory id : 2719762  
Shared memory contents:  
abcdefghijklmnopqrstuvwxyz  
mohamedinam@Mohamed-Inam-PC:~$ █
```

---

**RESULT:**

Thus contents written onto shared memory by the server process is read by the client process.