

Assignment 8

→ Aim : Inter Process Communication using Shared Memory using System V

→ Objective :

Application to demonstrate Client-Server Programs in which server process creates a shared memory segment & write the message to ~~send~~ shared memory segment. Client process reads the message from the shared memory segment & displays it to the screen.

→ Theory :

A] Shared Memory :

Shared memory allow 2 processes (unrelated) to access the same memory. Shared memory is a very efficient way of transferring data between 2 running processes. Shared memory is a special range of addresses that is created by IPC for one process and appears in the address space of that process.

All process can access the memory locations just as if the memory had been allocated by malloc. If one process writes to the shared memory, the changes

immediately become visible to any other process that has access to the same shared memory.

By itself, shared memory does not provide any synchronization facilities.

1. int shmget (key_t key, size_t size, int shmflag)

- Used to create a shared memory.
- The program provides a key, which effectively names the shared memory segment and the function returns a memory identifier.
- The second parameter size specifies the size/amount of memory required in bytes.
- The third flag ~~contains~~ consists of 9 different flags that are used to set mode for creating files.

2. void *shmat (int shm_id, const void *shm_addr, int shm_flag)

- This function is used to attach the shared memory to the address space of the process.
- The first parameter is the shm_id identifier returned by shmget().
- The second parameter is the address at which the shared memory is to

- be attached to the current process.
- The ~~third~~ parameter is a set of bitwise flags.

3. shmctl (int shm_id, int command, struct shmid_ds *buf);

- Used for controlling functions for shared memory.
- The 1st parameter is the identifier returned by shmget ().
- The 2nd parameter is the command (IPC_STAT, IPC_SET & IPC_RMID).

→ Conclusion:

Topics Covered :

1. Shared Memory
 2. Synchronizing operations of Shared Memory
-