T21287
Jayesh Patil TE IT

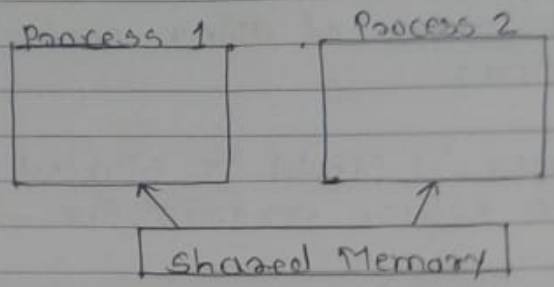
Assignment no - 7 (b)

Aim :- Inter-Process Communication using Shared memory using system v Application to demonstrate client and serve programs in which server process creates shared memory segment and writes the message to shared memory segment client process reads the message from the shared memory segment and display it to screen.

Objectives :- To study
Linux Kernel architecture
Shared memory

Theory :-

Inter-Process Communication using shared memory using system v Application to demonstrate client and server programs in which server process creates a shared memory segment and writes the message to shared memory segment client process reads the message from shared memory segment and displays it to the screen.



Shared memory is a feature supported by UNIX system V, including linux, sunos and solaris one process must explicitly ask for an area using key to be shared by other process this process will be called the server. All other processes, the clients that know the shared area can access it To protect a shared memory from being accessed at the same time by several process a synchronization protocol must be setup

A shared memory segment is identified by a unique integer, the shared memory ID the shared memory itself is described by structure of type shmid_ds in kernode file system. To use this file, file sys/types.h and sys/ipc.h must be included therefore, your program should start with the following lines.

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

A general scheme of using shared memory is the following:

for a sever; it should be started before any client. The sever perform the following task.

1. Ask for shared memory with a memory key and memorize the returned shared memory I.D This is performed by system call shmget()
2. Attach this shared memory to severs add space with system call shmat()
3. Initialize the shared memory if necessary
4. Do something and wait for all client's completion.
5. Detach the shared memory with system shmdt()

for the client part, the procedure is almost the same

1. Ask for shared memory with same memory key and memorize the returned shared memory ID
2. Attach this shared memory to client's add space
3. use the memory
4. Detach all shared memory segments if necessary
5. Exit.