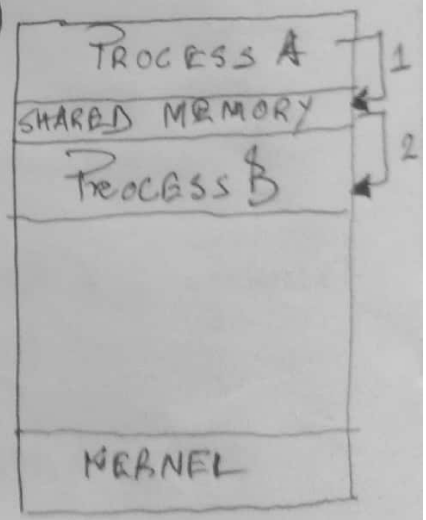
Interprocess communication using Shared Memory Concept:

1. One process will create an area in RAM which the other processes can access. — Shared Memory.
2. Both the processes can use the memory for regular works like: read, write etc.
3. Typically, a shared memory region resides in the address space of the creating Process's memory segment.
4. Other processes that wish to communicate using this shared memory segment must attach it to their address space.
5. Normally, you all know that operating System tries to prevent a process to access other processes memory. Shared memory requires to remove this restriction for all those co-operating processes.

Shared Memory in Linux: (System calls)

- int shmget(key, size, flags)
  - create a shared Memory
  - Returns ID of segment: shmid
  - key: unique identifier of the shared memory segment
  - Size: size of the shared memory rounded upto PAGE_SIZE.



- int shmat (shmid, addr, flags)
  - attach shmid shared memory to address space of the calling Process.
  - addr: pointer to the shared memory address space
- int shmdt (shmid)
  - detach the shared memory from the address space.