

Date: 22-11-21

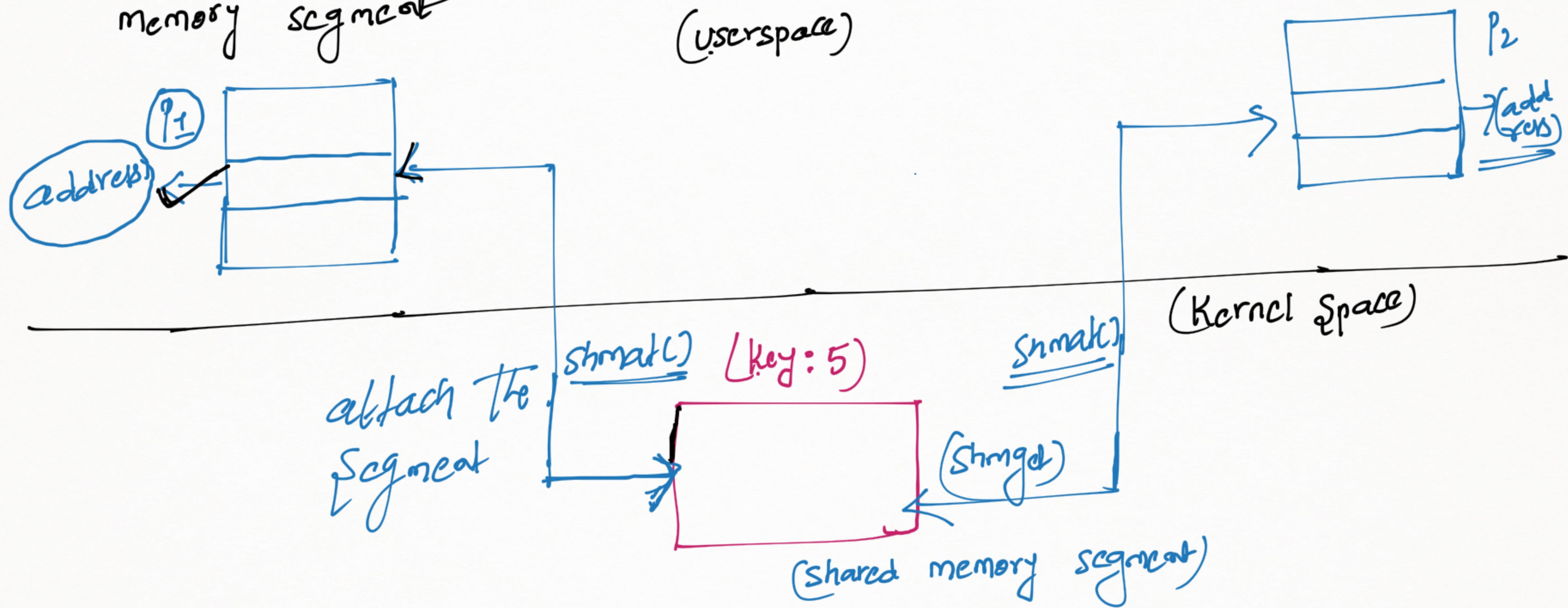
SHARED MEMORY

- Among all ipc's shared memory is the fastest ipc mechanism.
- Shared memory is Asynchronous ipc.
- There is no concept of empty shared memory (or) full shared memory.

The system calls required for shared memory are

- ① shmgel () → create shared memory (or) open existing shared memory
- ② shmat () → useful for attaching the shared memory segment
- ③ shmdet () → useful for detaching the shared memory segment
- ④ shmctl () → shared memory control operations-

→ \$ ipcs -m command will display information about shared memory segment



→ shmctl()

When a shared memory segment created, Kernel creates a data structure to maintain the information about shared memory.

1) IPC_STAT: copy information from kernel data structure
To Third argument of shmctl()
struct shmid_ds v; ↓
↓ structure pointer
shmctl(fd, IPC_STAT, &v)

2) IPC_SET:

Using this command set values into kernel

data structure

Shmctl (fd, IPC_SET, &v);

↳ information copied from

this structure variable to kernel data structure

3) IPC_RMID

Using this command it is possible to
delete the shared memory
segment

II shmctl(fd, IPC_RMID, 0)

↳ will remove the shared
memory (when all processes detach done then removed)

\$ ipcrm -m fd }
\$ ipcrm -M key }

Commands

Shmat(fd, 0, 0)

Shmat 2nd and 3rd arguments

Shmat(fd, 0x1000, 0)

p = Shmat(fd, 0, 0)

Kernel will attach to
segment where ever the
address free

→ If this address already in use then
Shmat fails

→ portability issue

↳ An address valid in
one system may invalid in other system

P = Shmat(fd, 0, SHM-RD ONLY)
↳ If this flag is provided
Then the shared memory segment attached to the
process only for reading
Shmat(fd, 0, 0)
↳ If third argument is zero
Then the process can read/write the data from the
segment