Linux Internals & Networking

System programming using Kernel interfaces

Team Emertxe

Contents

Linux Internals & Networking
Contents

EMERTXE

Inter Process Communications (IPC)

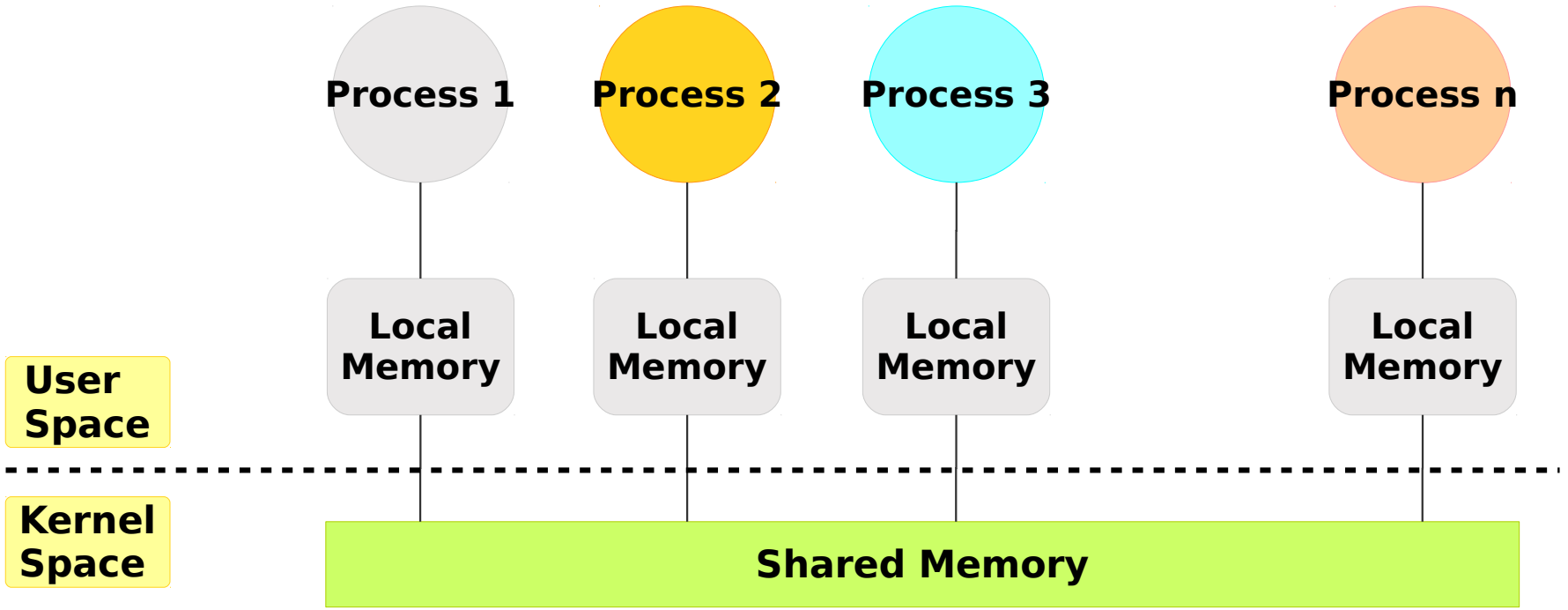
Inter Process Communications
Shared Memories - Properties

- Shared memory allows two or more processes to access the same memory
- When one process changes the memory, all the other processes see the modification
- Shared memory is the fastest form of Inter process communication because all processes share the same piece of memory
- It also avoids copying data unnecessarily

Note:
- Each shared memory segment should be explicitly de-allocated
- System has limited number of shared memory segments
- Cleaning up of IPC is system program's responsibility

Inter Process Communications

Shared vs Local Memory



Inter Process Communications
Shared Memories - Procedure

- Create
- Attach
- Read/Write
- Detach
- Remove

} **95 %**

Inter Process Communications
Shared Memories - Procedure

- To start with one process must allocate the segment
- Each process desiring to access the segment must attach to it
- Reading or Writing with shared memory can be done only after attaching into it
- After use each process detaches the segment
- At some point, one process must de-allocate the segment

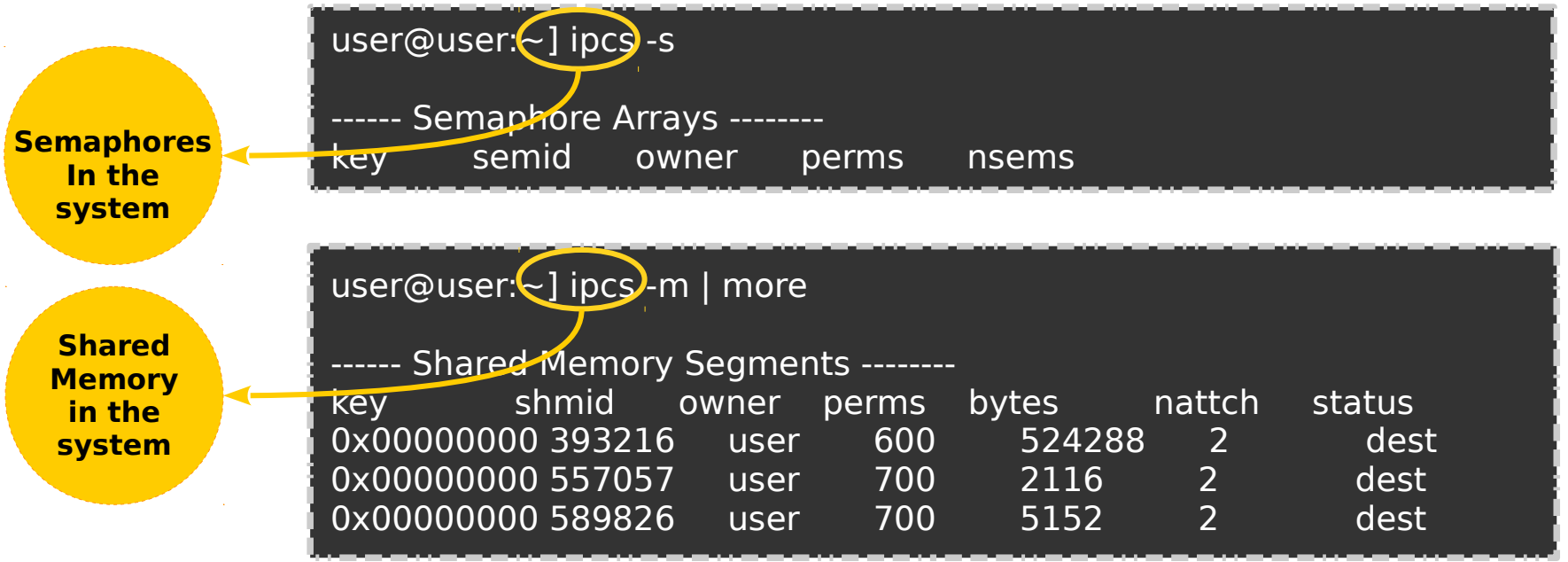
While shared memory is fastest IPC, it will create synchronization issues as more processes are accessing same piece of memory. Hence it has to be handled separately.

Inter Process Communications
Shared Memories – Function calls

Function	Meaning
int shmget(key_t key, size_t size, int shmflag)	✓ Create a shared memory segment ✓ key: Seed input ✓ size: Size of the shared memory ✓ shmflag: Permission (similar to file) ✓ RETURN: Shared memory ID / Failure
void *shmat(int shmid, void *shmaddr, int shmflag)	✓ Attach to a particular shared memory location ✓ shmid: Shared memory ID to get attached ✓ shmaddr: Exact address (if you know or leave it 0) ✓ shmflag: Leave it as 0 ✓ RETURN: Shared memory address / Failure
int shmdt(void *shmaddr)	✓ Detach from a shared memory location ✓ shmaddr: Location from where it needs to get detached ✓ RETURN: SUCCESS / FAILURE (-1)
shmctl(shmid, IPC_RMID, NULL)	✓ shmid: Shared memory ID ✓ Remove and NULL

Inter Process Communications
Synchronization - Debugging

- The ***ipcs*** command provides information on inter-process communication facilities, including shared segments.
- Use the -m flag to obtain information about shared memory.
- For example, this image illustrates that one shared memory segment, numbered 392316, is in use:



Inter Process Communications
Summary

- We have covered

Data exchange

Communication

- Pipes
- FIFO
- Shared memory
- Signals
- Sockets

Resource usage/access/control

Synchronization

- Semaphores

EMERTXE

Stay Connected

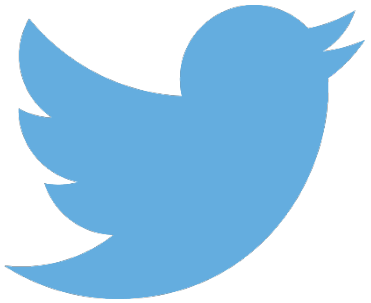
About us: Emertxe is India's one of the top IT finishing schools & self learning kits provider. Our primary focus is on Embedded with diversification focus on Java, Oracle and Android areas

Emertxe Information Technologies,

No-1, 9th Cross, 5th Main,
Jayamahal Extension,
Bangalore, Karnataka 560046

T: +91 80 6562 9666
E: training@emertxe.com

https://www.facebook.com/Emertxe



https://twitter.com/EmertxeTweet

https://www.slideshare.net/EmertxeSlides

EMERTXE

Thank You