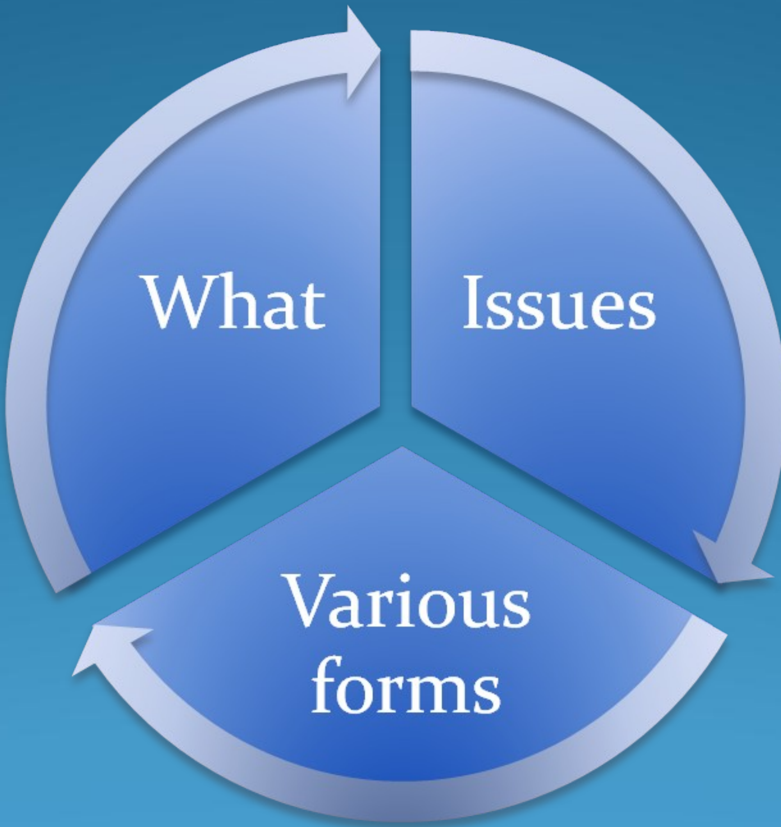
Inter Process Communication

Dr. B. Rajesh Kanna
Professor
School of Computing Science and Engineering
VIT University,

Terminologies

- Process
- Independent Process
- Co-operative Process
- Related and unrelated Process
- Process Components
  - Memory – Data, Stack, Heap, Code
  - List of opened files
  - Process state
  - Assigned resources

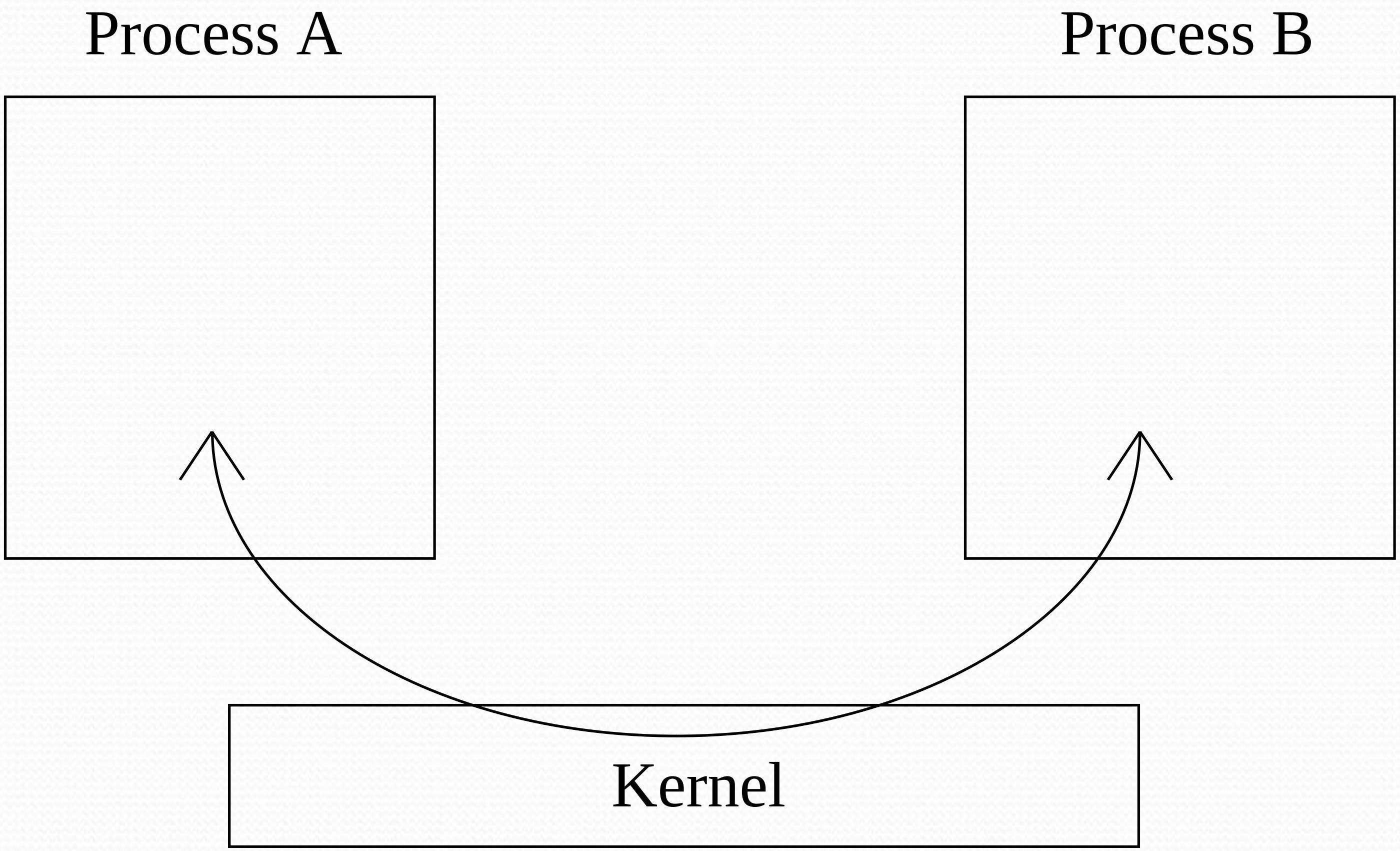
Inter Process Communication



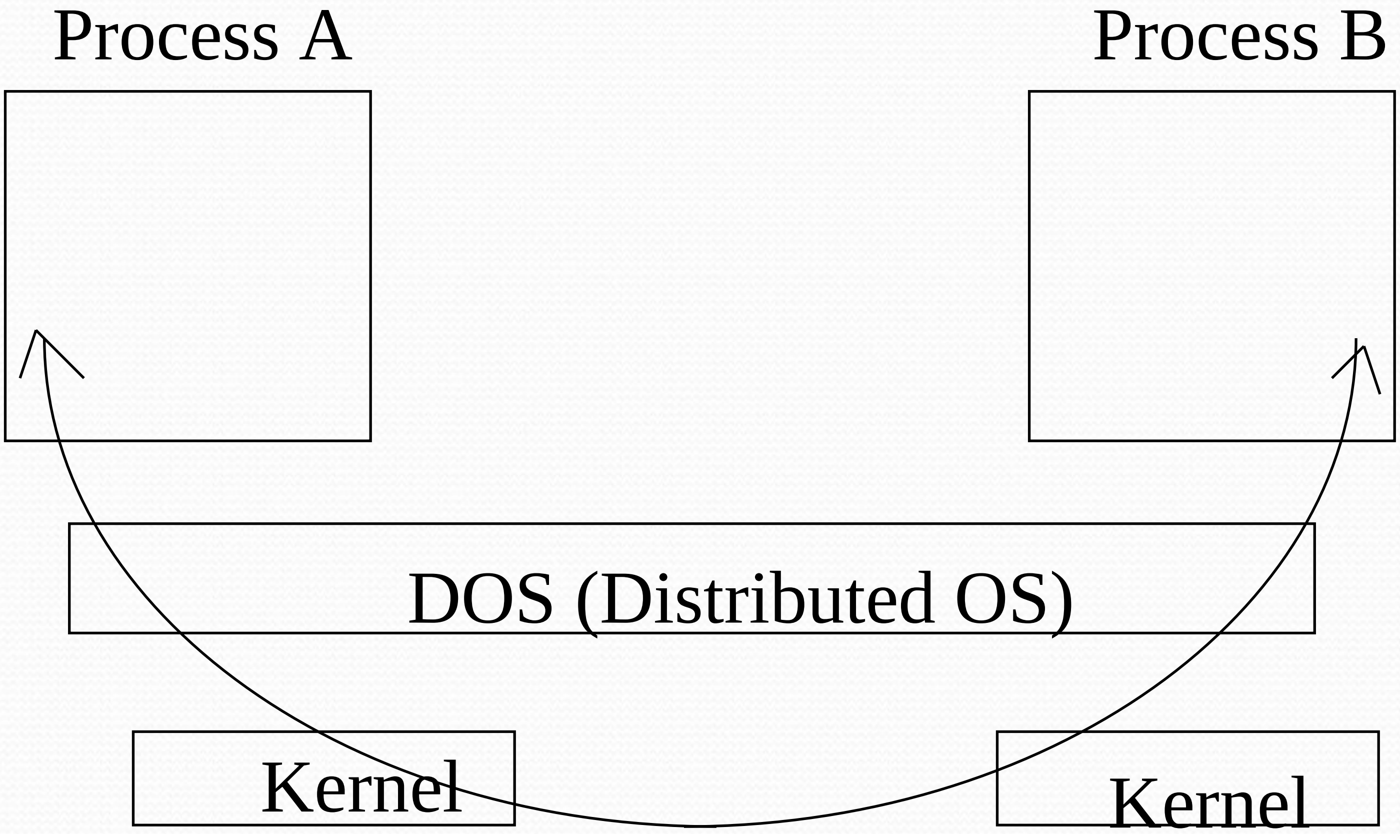
What, Uses

- Exchange of information/data between process
  - Data Transfer
  - Sharing Data
  - Event notification
  - Process Control

IPC On a Single Computer



IPC on Two Computers



Issues and solution

- Data corrupted
- Dead lock
- Atomicity

- ## Solution
  - Process synchronization

Mechanisms of IPC

- Signals (Software Interrupt)
- Pipes
- FIFOs
- Message Queues
- Semaphores
- Shared Memory
- Sockets

Characteristics of IPC

- Persistence / Life time
  - Process, Kernel
- Locality
- Simultaneous Access – one or many
- Flow of data
  - One way, Two way

- Synchronization Mechanism ?

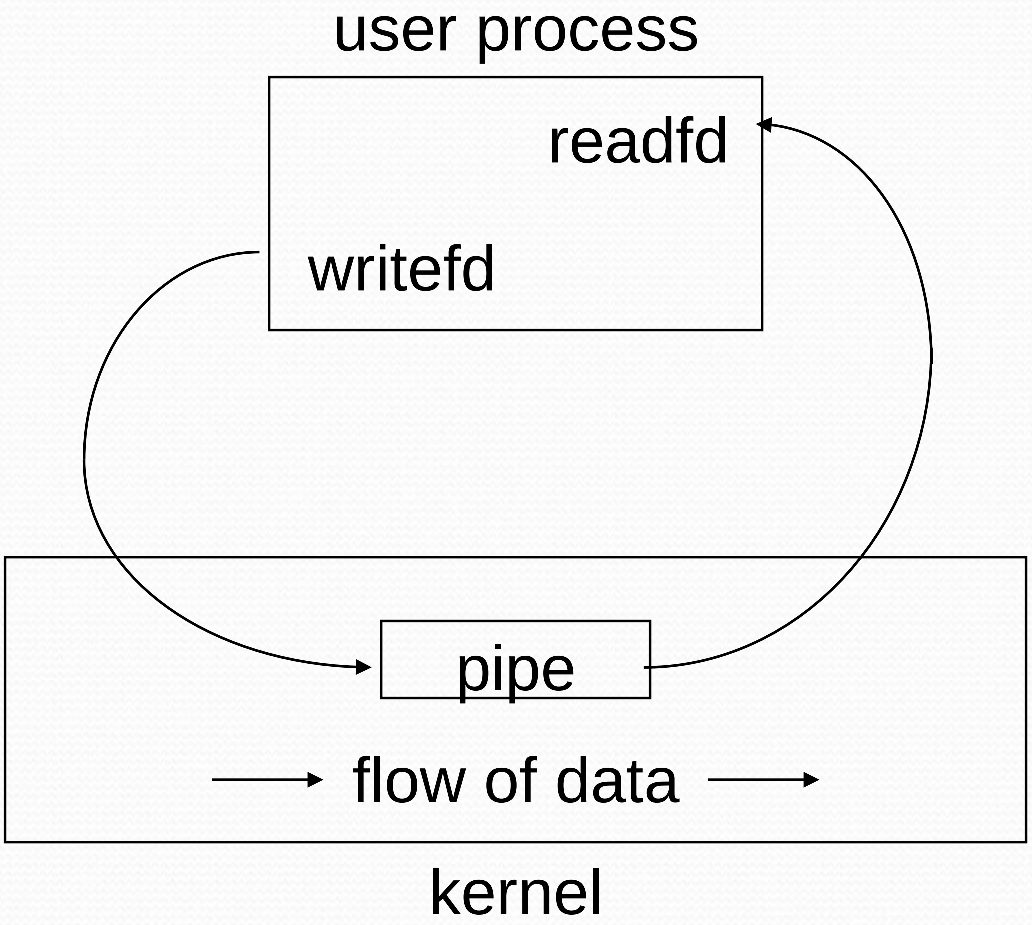
Signals

- Software interrupts
- A notification to a process that an event has occurred
  - usually the process doesn't know ahead of time exactly when a signal will occur
- Signals can be sent by:
  - a process to another process (or to itself)
  - the kernel to a process

Pipes

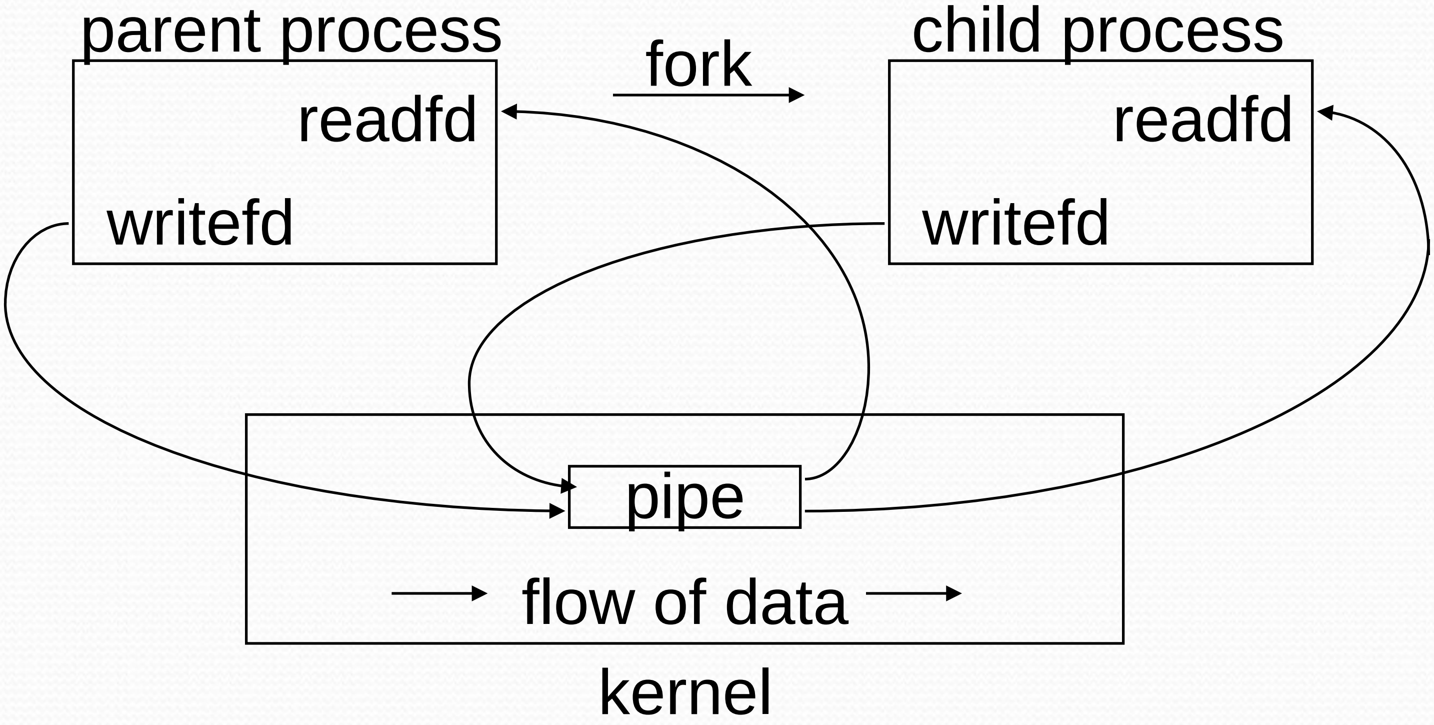
- A pipe provides a one-way flow of data
  - example: who | sort| lpr
    - output of who is input to sort
    - output of sort is input to lpr
- The difference between a file and a pipe:
  - pipe is created in the kernel space.
- Two ends are utilized
  - reading
  - writing

Pipes



Pipe Creation

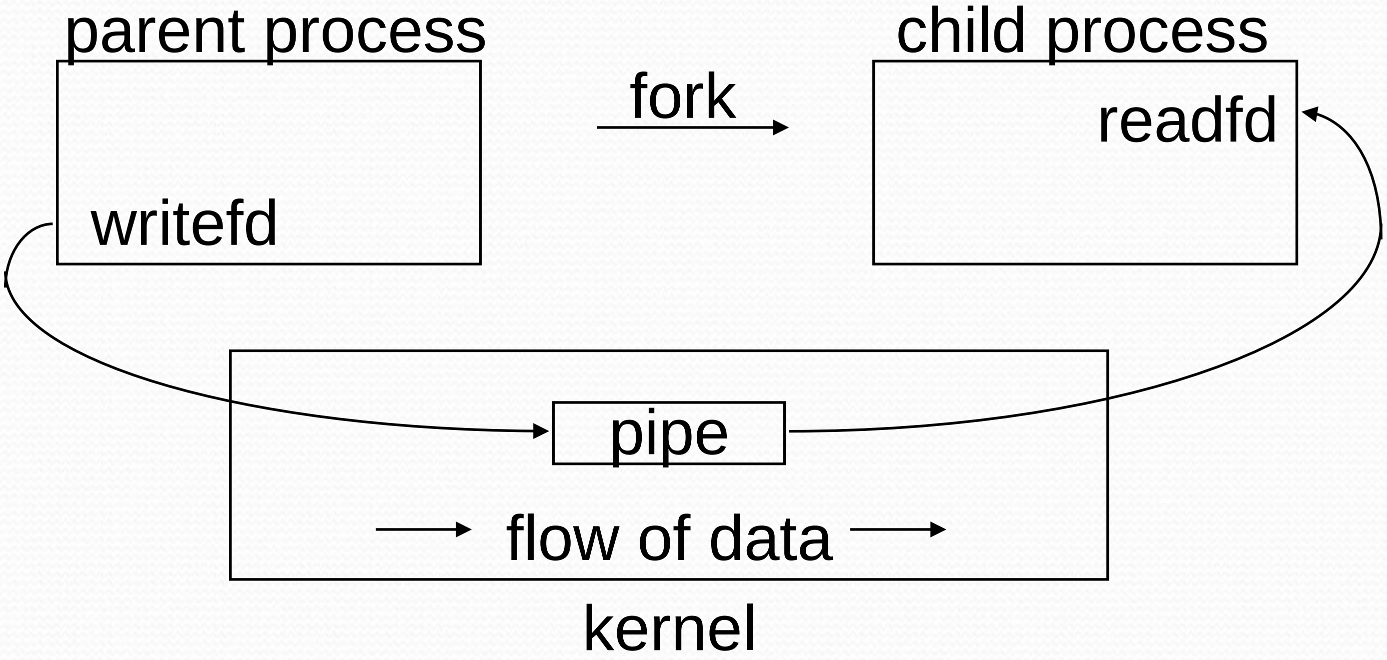
- First, a process creates a pipe, and then forks to create a copy of itself.



Pipe Examples

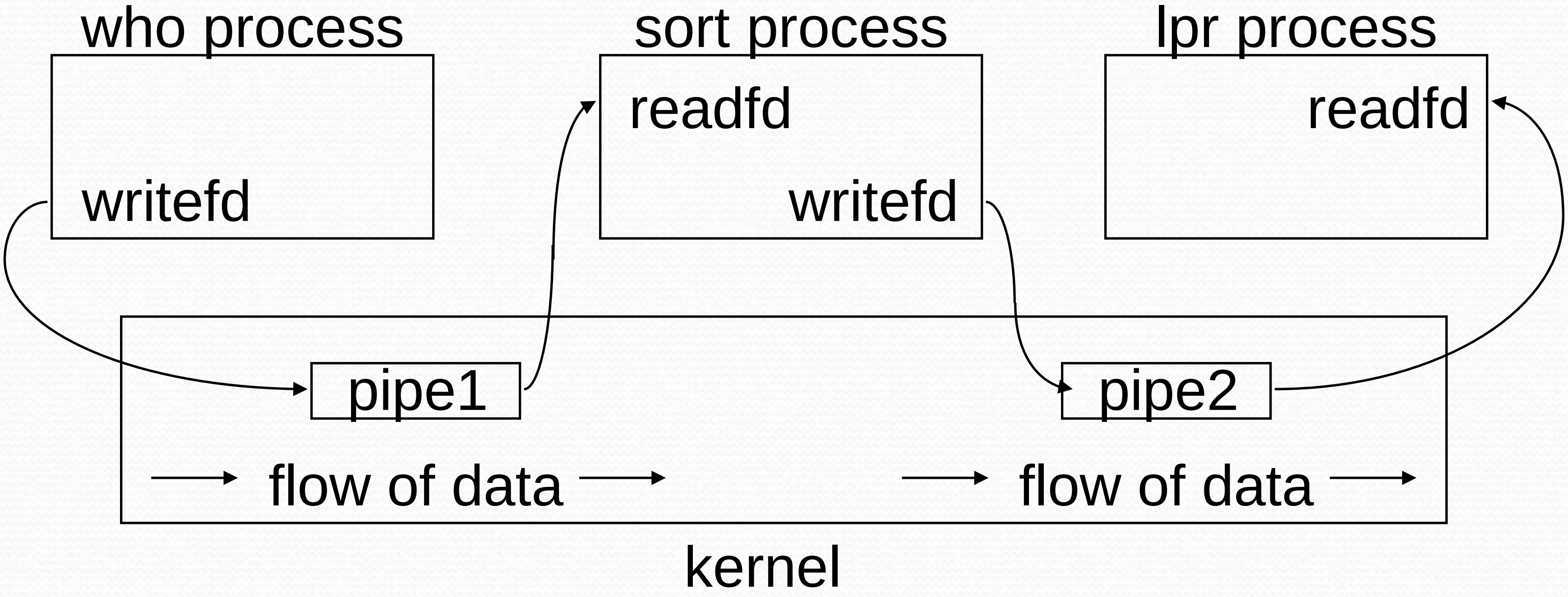
Parent opens file, child reads file

- parent closes read end of pipe
- child closes write end of pipe



who | sort | lpr

- who process writes to pipe1
- sort process reads from pipe1, writes to pipe2
- lpr process reads from pipe2



FIFO

- Named Pipe
- Unrelated Process Communication

Message Queues

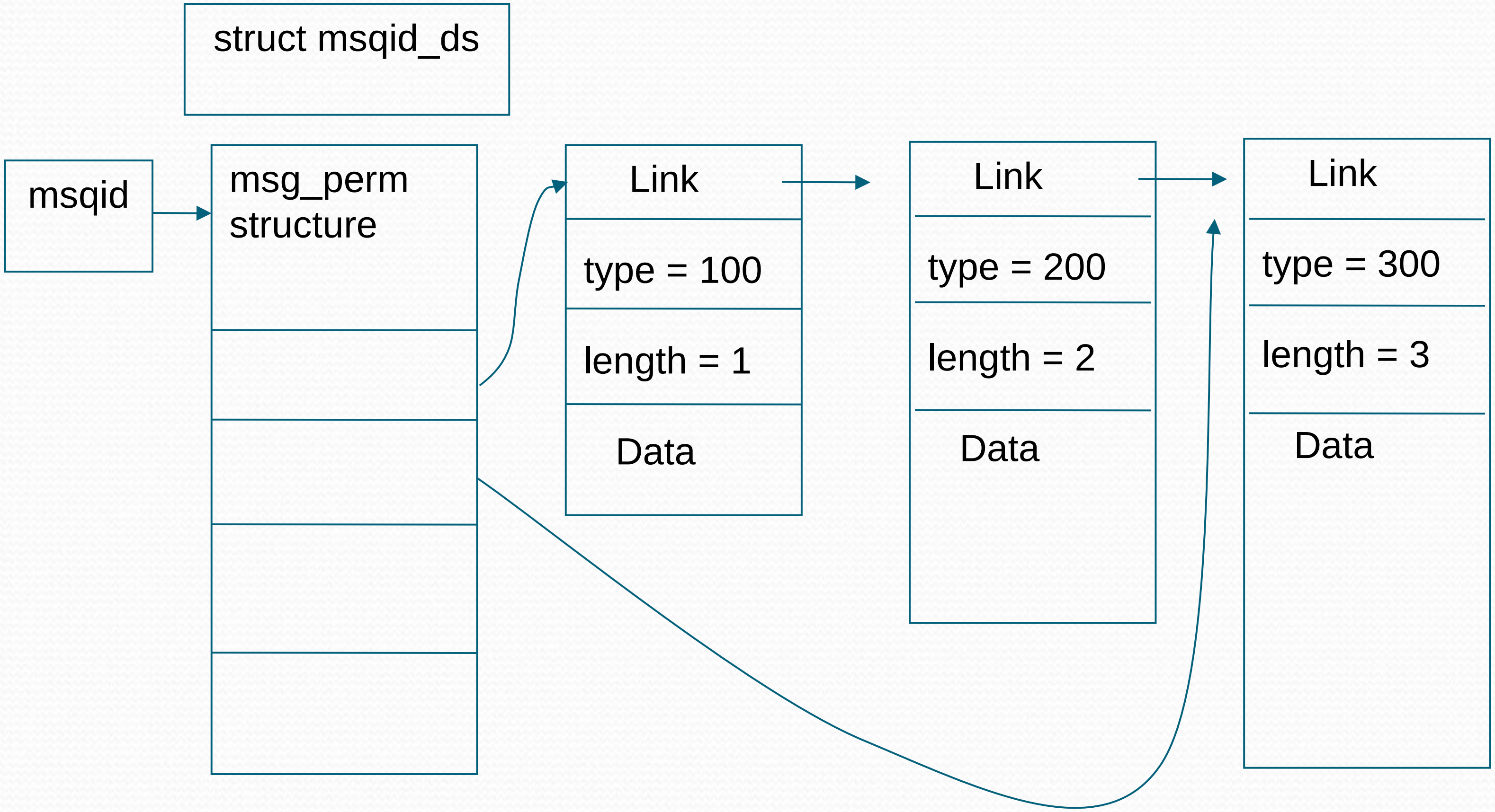
- Another form of IPC which is practically like a FIFO, but overcomes the disadvantage of it.

- Each message on the Queue has the following attributes:

  - Long int type
  - Data

- Internally Kernel maintains the message structures in the form of link lists.

Data structure for Message Queues (System V Std)



Message Queue Header

Permission

first message on queue

last message in queue

last msg send time

last msg received time

last change time

No. Of messages

max number of bytes on queue

process id of last msgsnd

last receive pid

Operations -Message Queue

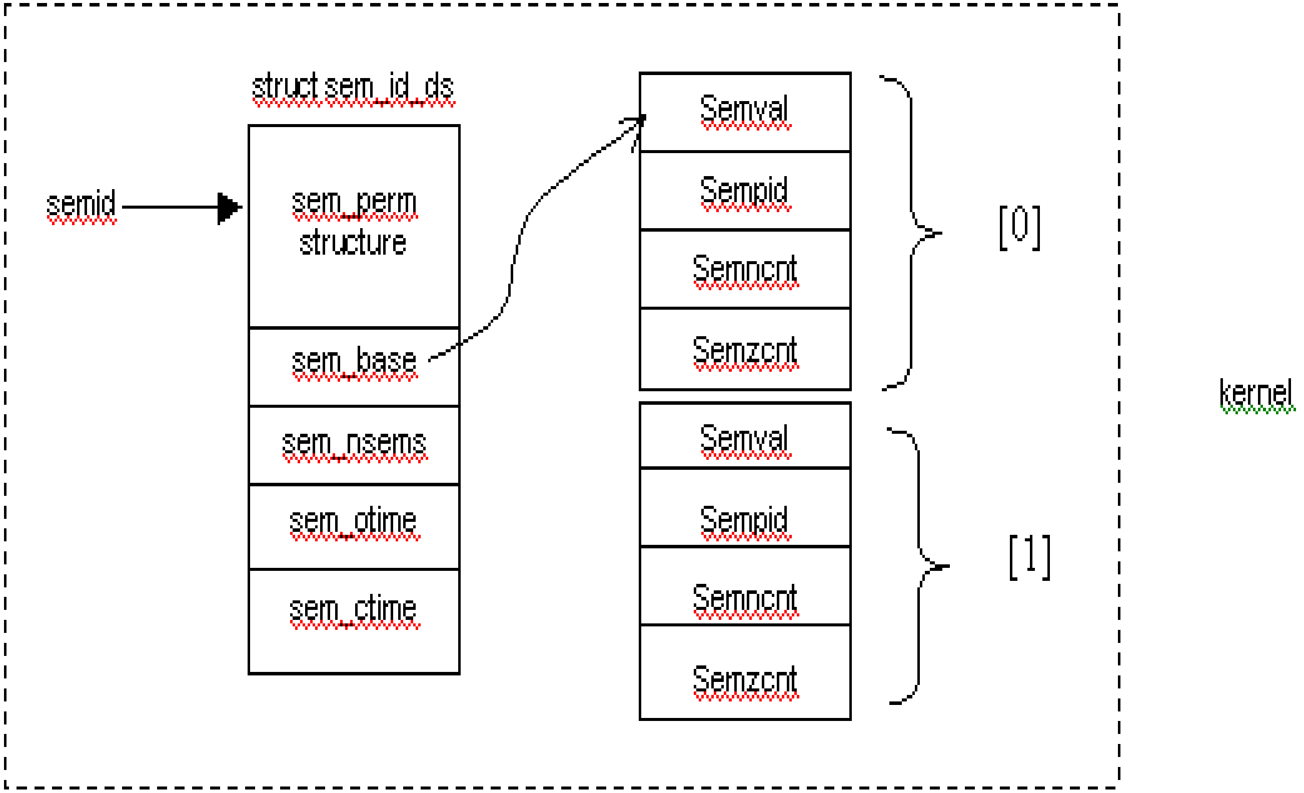
- Create / Refer
- Send
- Receive
- Get the meta information
- Set the attributes
- Delete Message queue

Semaphores

- As a form of IPC they are not used for exchange of data.

- Whenever a common resource has to be shared between more than one process, a semaphore is used to synchronize the access of the resource between the sharing processes

Data Structure for Semaphores



Operations - Semaphore

- Create / Refer – No. of semaphore sets
- Set Value – Binary or Counting
- Wait
- Signal
- Controls
- Delete Semaphore

Shared Memory

- Shared memory is the fastest & probably the easiest form of IPC.
- It has no system call overheads.
- The sender & receiver share the same memory to communicate between them

Operations –Shared Memory

- Create / Refer
- Attach
- Detach
- Get the meta information
- Set the attributes
- Delete Shared Memory

Summary

- Terminologies
- IPC
- Issues
- Various forms
  - Signal
  - Pipe
  - FIFO
  - Message Queue
  - Semaphore
  - Shared Memory