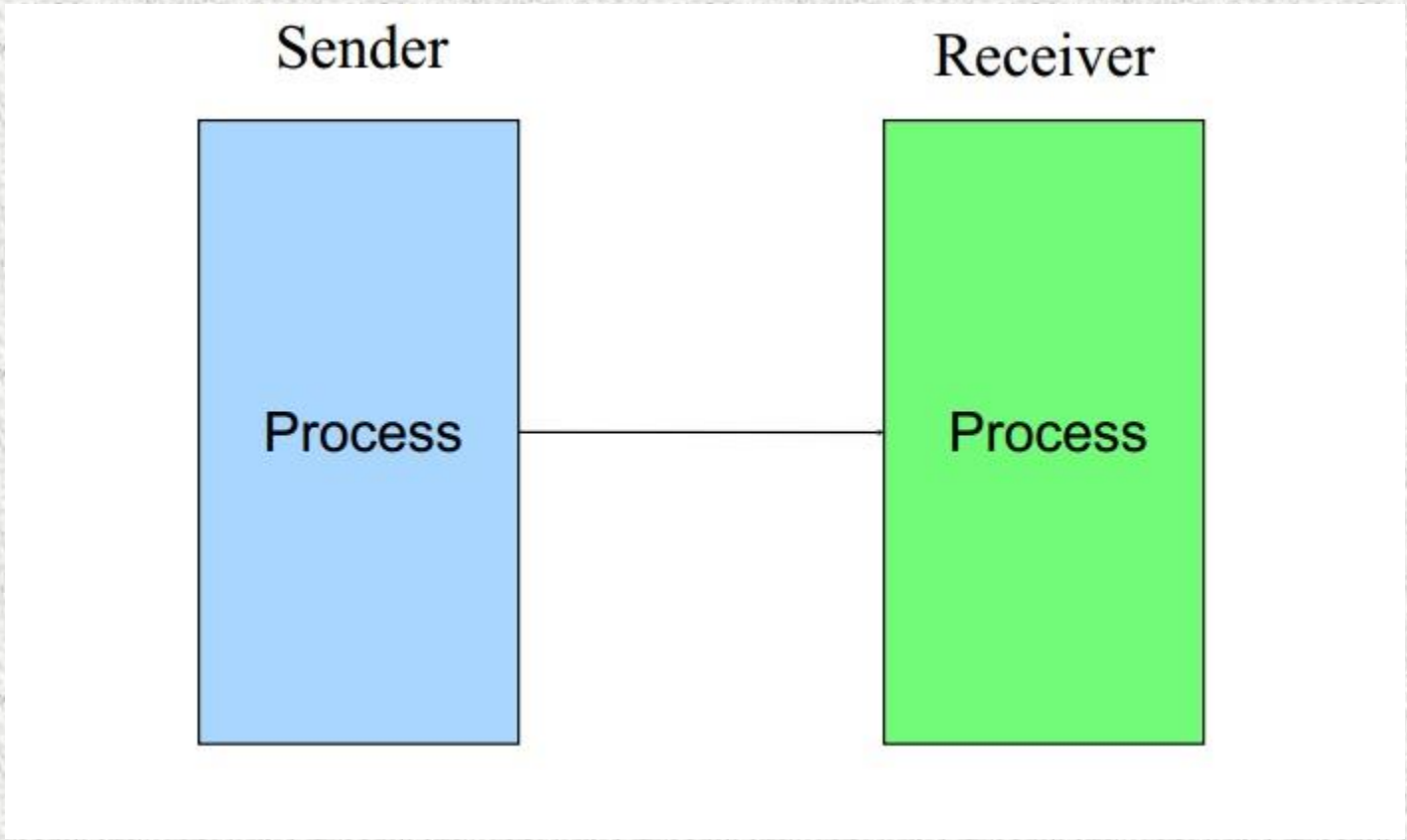
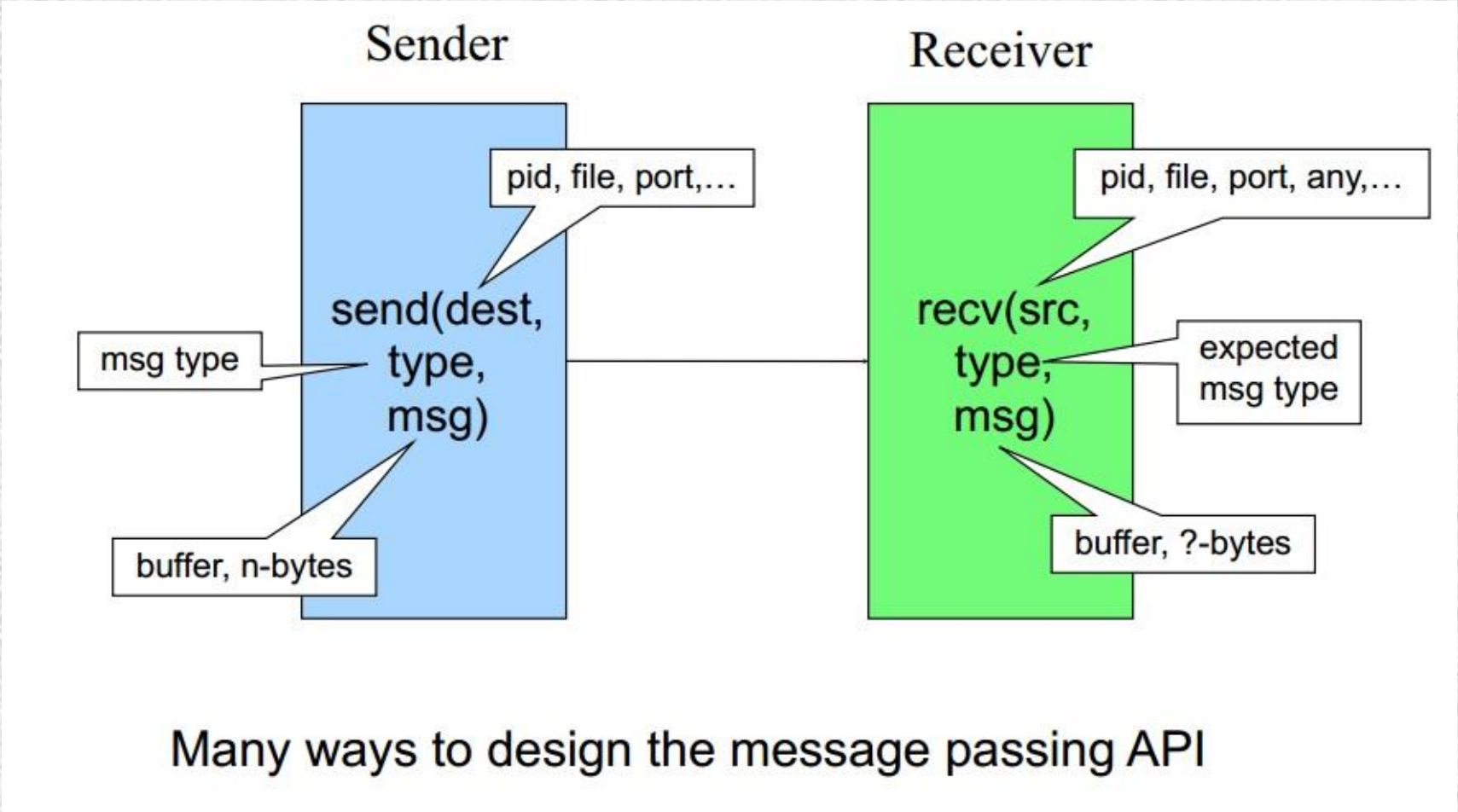
Inter Process Communication

Message Passing – Big Picture

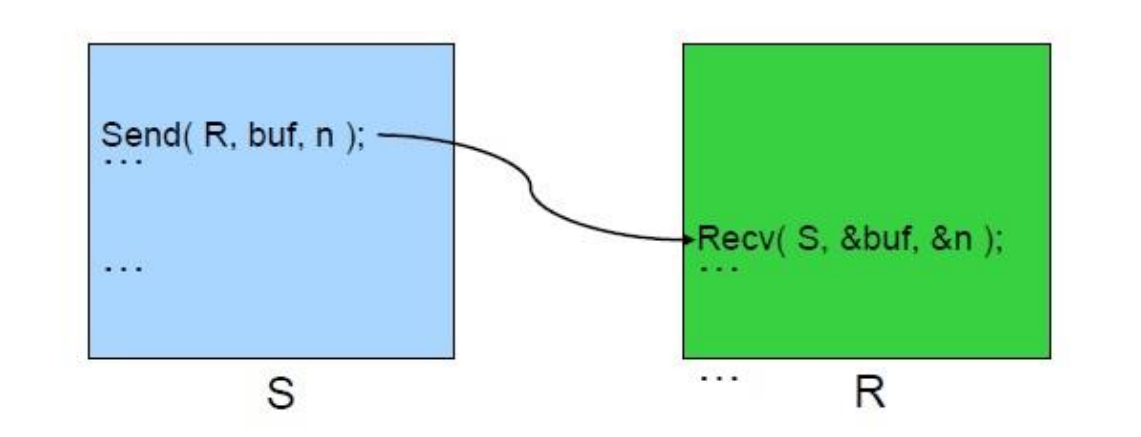


Send & Receive Primitives



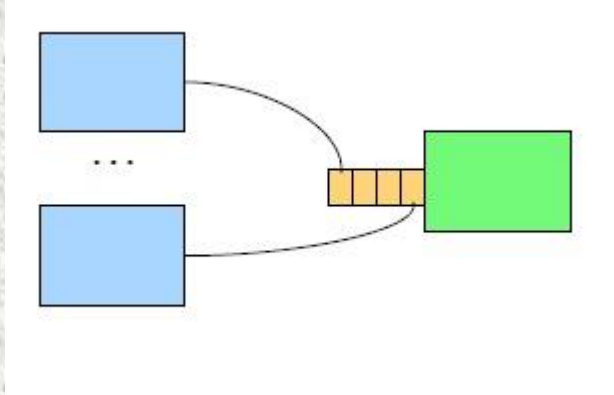
Synchronous message Passing

- Move data between processes
  - Sender: when data is ready, send it to the receiver
  - Receiver: when data has arrived, when receiver process is ready, move data to destination data structure
- Synchronization
  - Sender: signal the receiver process that a particular event happens
  - Receiver: Block until the event has happened



Naming - Direct Communication

- Single Buffer at the receiver
  - More than 1 process can send messages to the receiver
  - Search buffer to receive from particular receiver



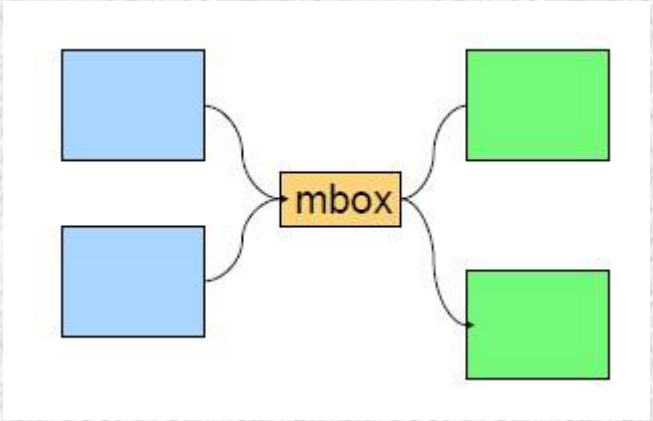
Naming - Direct Communication

- Automatic link is established
- A link is associated with 1 process
- Between each pair, there is only 1 link
- Disadvantage: Limited modularity of process definitions

Symmetric	Asymmetric
Sender & Receiver must name each other	Sender names, Receiver need not
Send(p, message) Receive(q, message)	Send(p, message) Receive(id, message)

Naming - Indirect Communication

- Use mailbox as abstraction
  - Allow many to many communication
  - Require open/close a mailbox
  - Link is established if 2 process share mailbox
  - A link may be associated with more than 2 processes
  - Many links between 2 process



Naming - Indirect Communication

- When multiple process execute a receive() which process will receive the message?
  - Allow a link 2 process max
  - 1 process to execute receive at a time
  - System will choose which process will receive

- Mailbox owned by process or OS
  - Process's address space – only process receives
  - OS allows process to create, send & receive msg and delete mailbox

Naming - Indirect Communication

- When multiple process execute a receive() which process will receive the message?
  - Allow a link 2 process max
  - 1 process to execute receive at a time
  - System will choose which process will receive

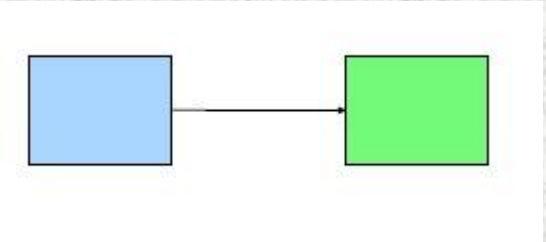
- Mailbox owned by process or OS
  - Process's address space – only process receives
  - OS allows process to create, send & receive msg and delete mailbox

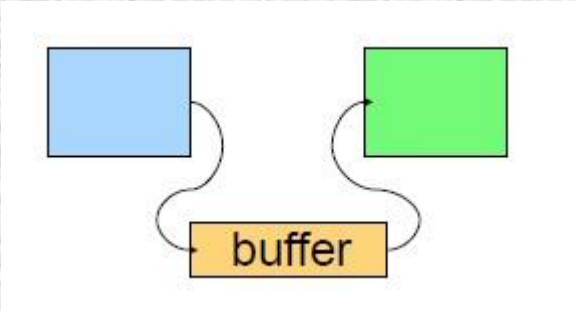
Synchronization

Synchronous Send	Asynchronous Send
Block if resource is busy	Block if resource is busy
Initiate data transfer	Initiate data transfer
Block until data is out of its source memory	and return
Rendezvous: block until receiver has done recv and sent acknowledgment	Completion Require applications to check status • Notify or signal the application

Buffering Messages

- No Buffering
  - Sender sends waits until receiver
  - receives
  - Rendezvous on each message
- Bounded Buffer
  - Finite Size
  - Sender blocks on buffer full
- Unbounded Buffer
  - Infinite Size
  - Sender Never Blocks





References

- http://www.cs.princeton.edu/courses/archive/fall08/cos318/lectures/Lec12-MessagePassing.pdf