



SNS COLLEGE OF TECHNOLOGY

An Autonomous Institution

Coimbatore-35



Accredited by NBA – AICTE and Accredited by NAAC – UGC with 'A+' Grade
Approved by AICTE, New Delhi & Affiliated to Anna University, Chennai

DEPARTMENT OF ELECTRONICS & COMMUNICATION ENGINEERING

19ITT204-MICROCONTROLLER AND EMBEDDED SYSTEM

III YEAR/ V SEMESTER

UNIT 4 – PROCESS AND OPERATING SYSTEM

TOPIC – TASK COMMUNICATION



TASK

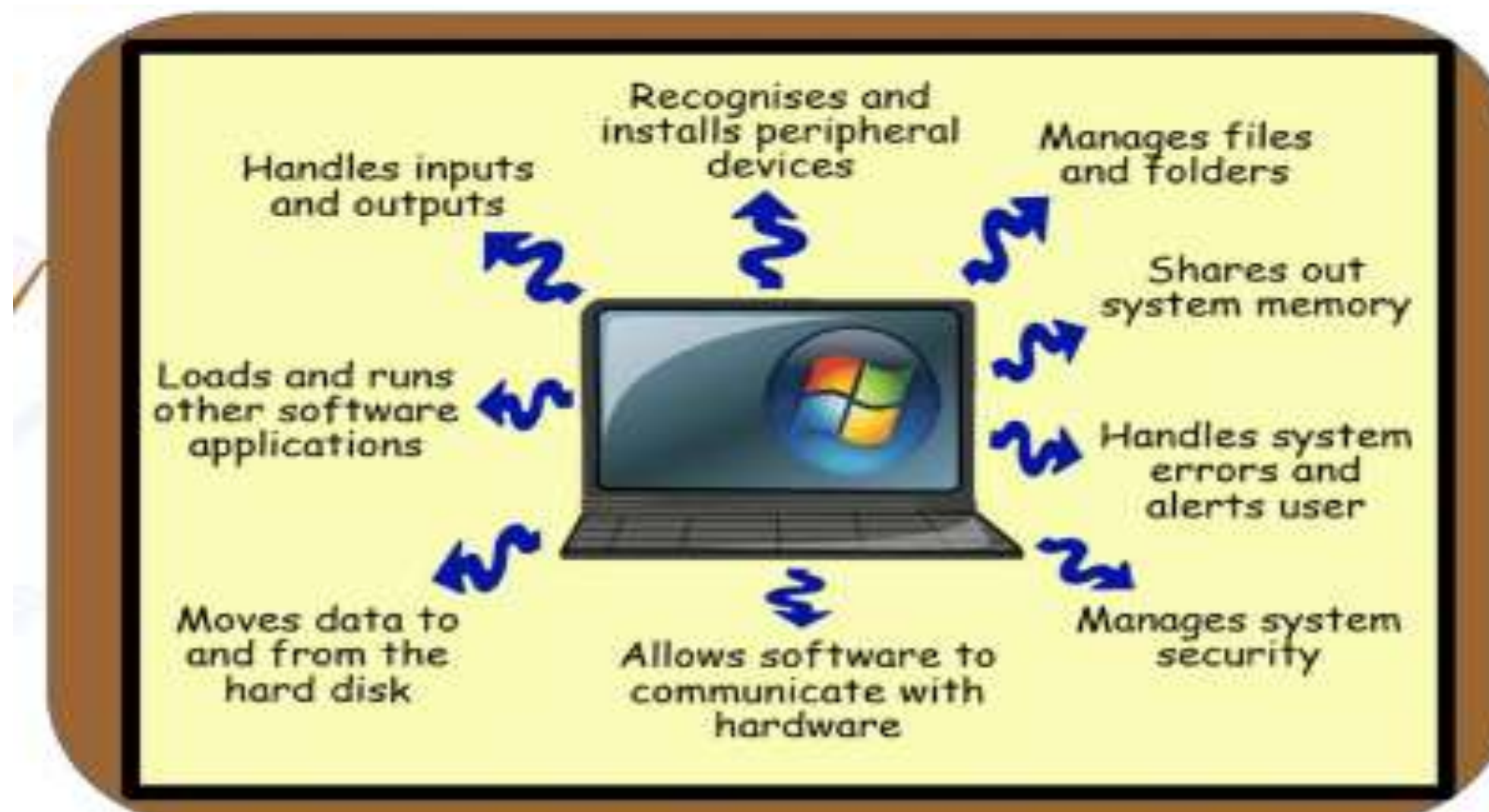


WHAT IS A TASK ?

- ✓ It is defined as a program in execution and has an order of priority , schedule or timeline for execution.
- ✓ Also called a "job" .
- ✓ A program / part of task / job in execution is called a process.



TASKS OF OPERATING SYSTEM





TASK COMMUNICATION



TASK COMMUNICATION

- ✓ *Task communication* comprises all mechanisms serving to exchange information among tasks.
- ✓ In a multitasking system multiple task/ process run concurrently (in pseudo parallelism) & each process may / may not interact .
- ✓ Based on degree of interaction , processes running on OS are classified as :
 - *Cooperating process*
 - *Competing process*



COOPERATING PROCESS



➤ ***Cooperating Process***

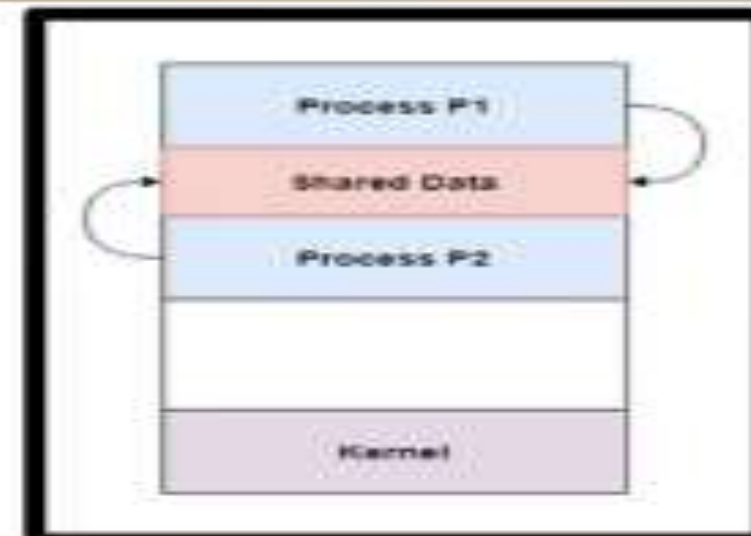
- ✓ Cooperating processes are those that can affect or are affected by other processes running on the system. Cooperating processes may share data with each other.
- ✓ Exchange of information & communication by:
 - *Cooperation by sharing*
 - *Cooperation by communication*



CO OPERATION BY SHARING



The cooperating processes can cooperate with each other using shared data such as memory, variables, files, databases etc. Critical section is used to provide data integrity and writing is mutually exclusive to prevent inconsistent data.

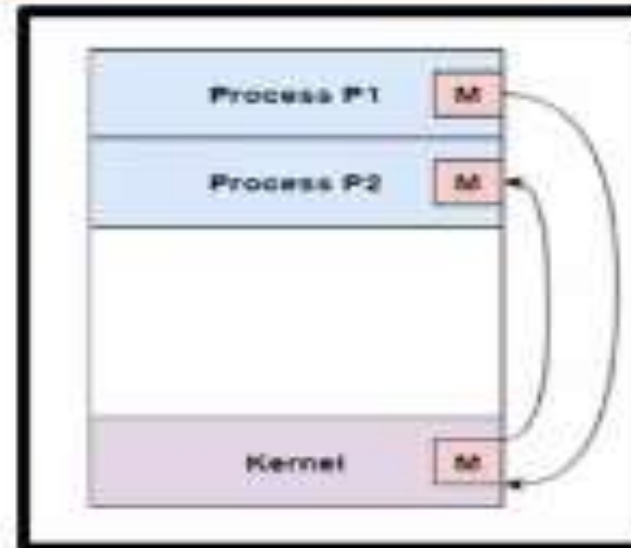




CO OPERATION BY COMMUNICATION

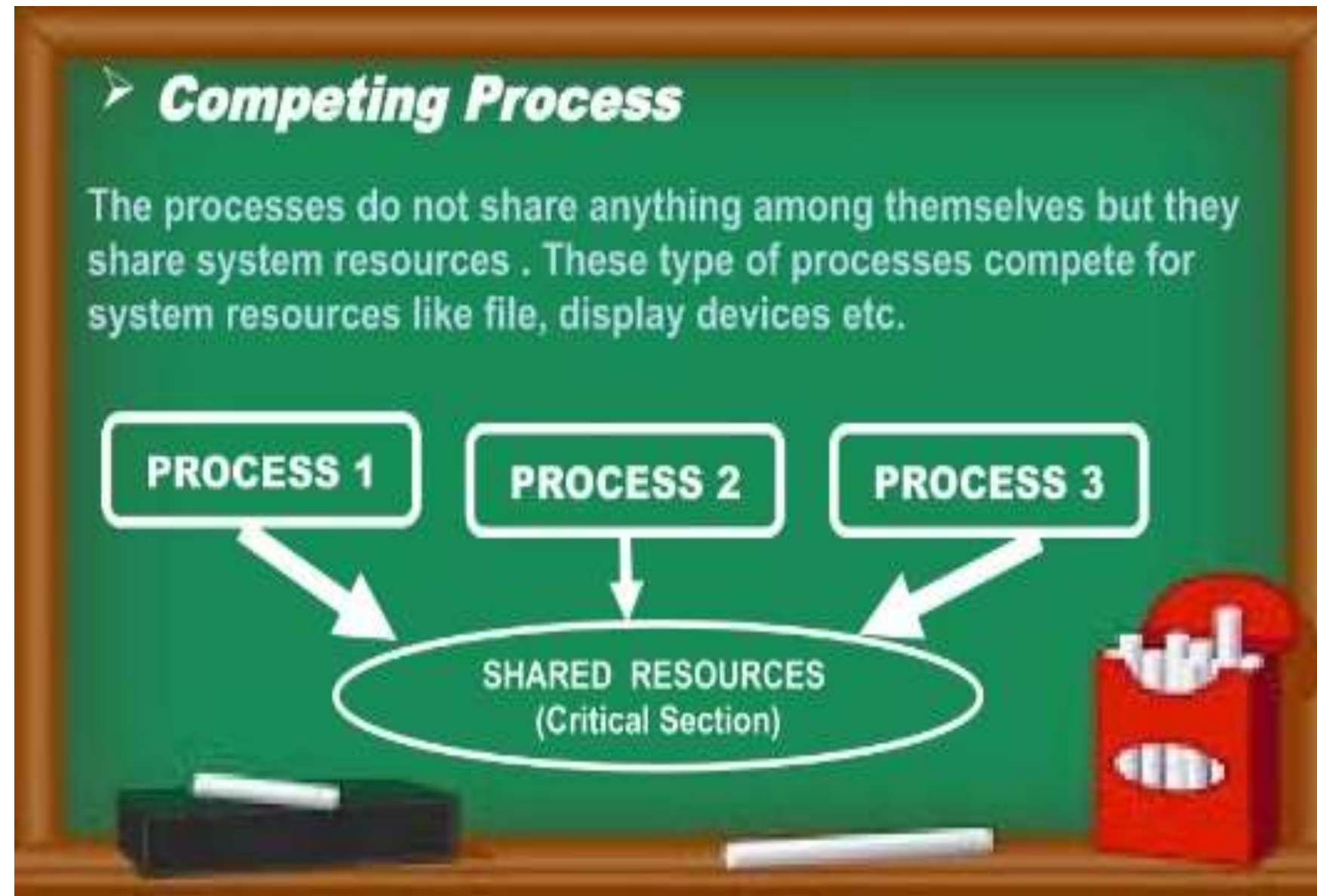


No data is shared between processes . They communicate for synchronization . The cooperating processes can cooperate with each other using messages. This may lead to deadlock if each process is waiting for a message from the other to perform a operation. Starvation is also possible if a process never receives a message.





COMPETING PROCESS





INTER PROCESS COMMUNICATION



- ✓ The mechanism by which process or task communicate.
- ✓ Essential for process coordination
- ✓ IPC Mechanisms:
 - Shared Memory
 - Message Passing
 - Remote Procedure Call & Sockets





SHARED MEMORY



Shared Memory

- ✓ Information to be communicated is written to shared memory. Process that require this information can read it from there.
- ✓ It's implementation is kernel dependant.
- ✓ Mechanisms for implementing shared memory for IPC :
 - *Pipes*
 - *Memory mapped objects*

The chalkboard is green with a wooden frame. At the bottom, there is a black eraser, a white piece of chalk, and a red chalk holder containing several white pieces of chalk.



PIPES



➤ **PIPES**

- ✓ A section of shared memory used by processes for communicating .
- ✓ Process that create a pipe – Pipe Server
- ✓ Process that connect a pipe – Pipe Client
- ✓ There are two types of pipe for IPC :
 - *Anonymous pipes*
 - *Named pipes*

PROCESS

```
graph LR; subgraph PROCESS; direction TB; W[WRITE ()]; R[READ ()]; end; P1[P[1]] --> Pipe(( )); Pipe --> P0[P[0]];
```



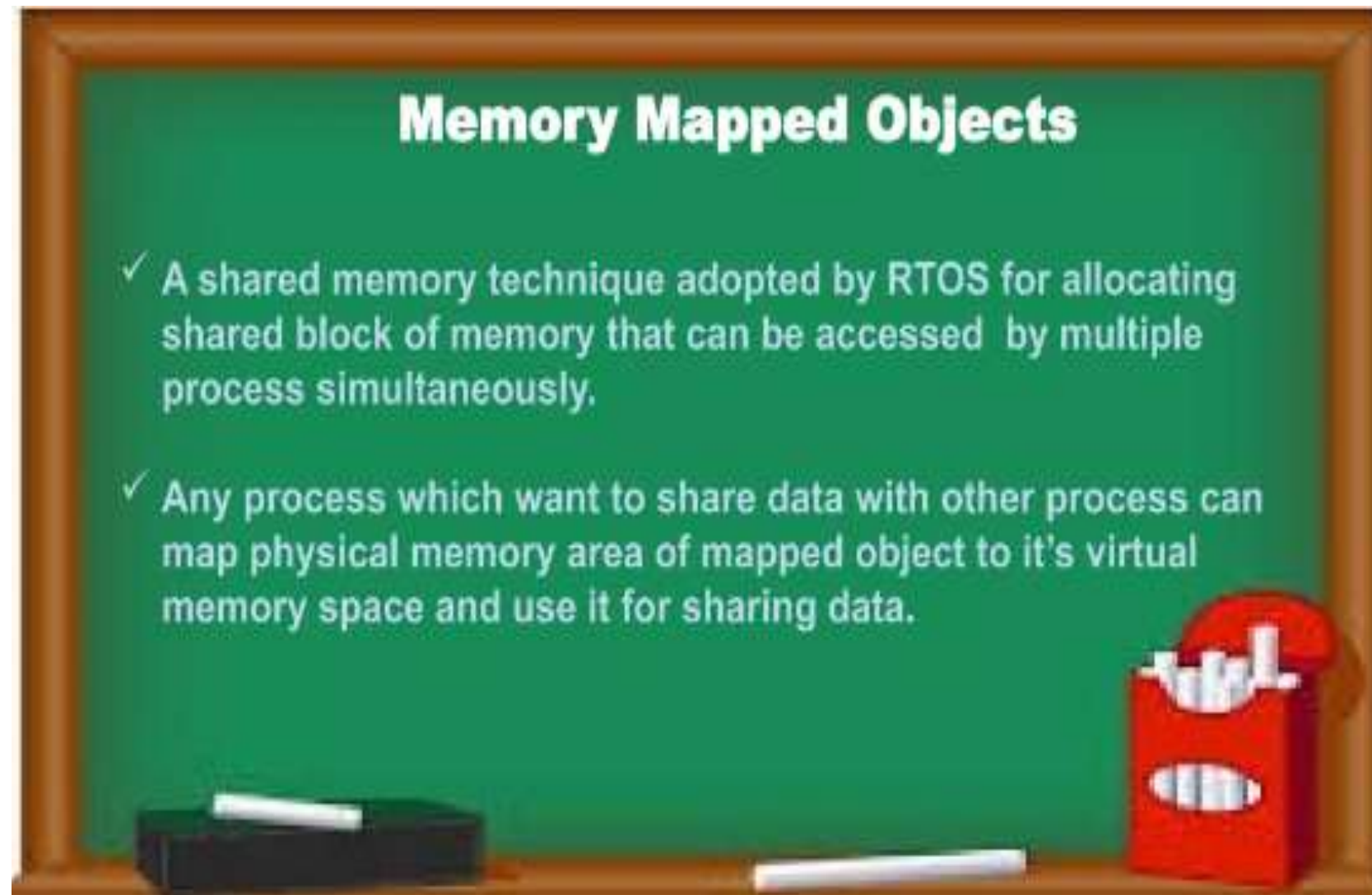
PIPES



- **Anonymous pipe:**
 - ✓ Unnamed, unidirectional for data transfer between two processes.
- **Named pipe:**
 - ✓ Named , unidirectional / bidirectional for data exchange between two processes.
 - ✓ Any process can act as both client and server allowing point to point communication.



MEMORY MAPPED OBJECTS





MESSAGE PASSING





DIRECT COMMUNICATION



➤ **Direct communication**

- ✓ Each process must explicitly name the recipient or sender of communication.
 - `send(P, message)` – send a message to process P
 - `receive(Q, message)` – receive a message from process Q
- ✓ Communication link has the following properties
 - Link is automatically established. Process must only know the identity of other process.
 - Link is associated with exactly two processes
 - B/w each pair of processes, exists exactly one link



IN DIRECT COMMUNICATION

➤ **Indirect communication**

- ✓ Messages are sent and received through mailboxes or ports.
 - Send(A, message) – send a message to mailbox
 - Receive (A, message) – receive a message from mailbox A
- ✓ Link properties
 - Link is established between a pair of processes only if both members of the pair have a shared mailbox.
 - Link may be associated with more than two processes.
 - Between each pair of communicating processes, there may be a number of different links, with each link corresponding to one mailbox.



STUBS



STUBS

- ✓ When the calling process calls a procedure, the action performed by that procedure will not be the actual code as written, but code that begins network communication.
- ✓ It has to connect to the remote machine, send all the parameters down to it, wait for replies, do the right thing to the stack and return. This is the client side *stub*.
- ✓ The server side stub has to wait for messages asking for a procedure to run.
- ✓ It has to read the parameters, and present them in a suitable form to execute the procedure locally. After execution, it has to send the results back to the calling process.



THANK YOU