Process

Organized By: Vinay Arora
Assistant Professor
CSED, TU

Disclaimer

This is NOT A COPYRIGHT MATERIAL

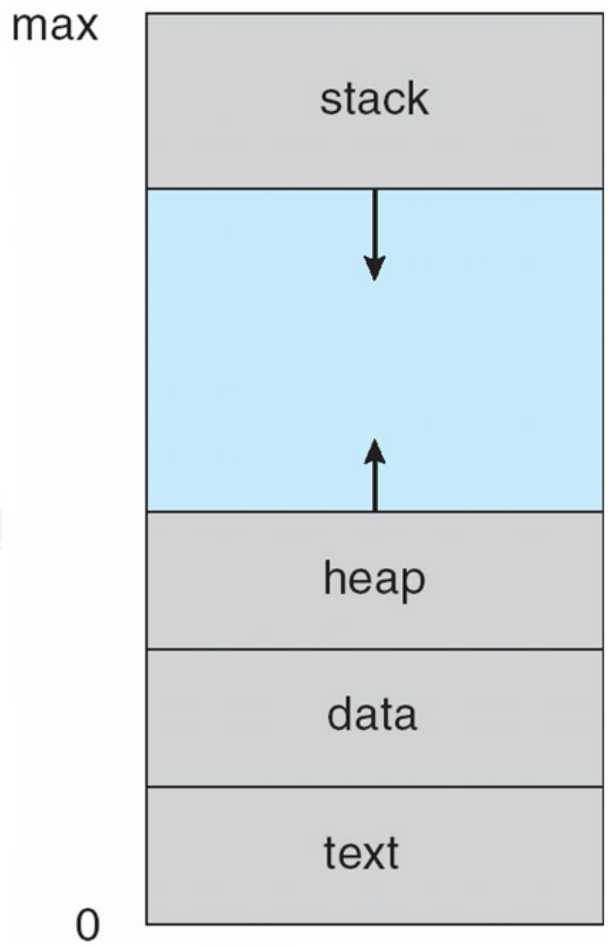
***Content has been taken mainly from the following books**:*

Operating Systems Concepts By Silberschatz & Galvin,
Operating Systems: Internals and Design Principles By William Stallings
www.os-book.com
www.cs.jhu.edu/~yairamir/cs418/os2/sld001.htm
www.personal.kent.edu/~rmuhamma/OpSystems/os.html
http://msdn.microsoft.com/en-us/library/ms685096(VS.85).aspx
http://www.computer.howsttuffworks.com/operating-system6.htm
http://williamstallings.com/OS/Animations.html
Etc…

Process

- A Program in Execution is called as a *PROCESS*.
- An operating system executes a variety of programs:
  - Batch System – Jobs
  - Time-Shared Systems – User Programs or Tasks
- A Process Includes:
  - Program Counter
  - Stack
  - Data Section

Process in Memory



Process

- A *Program* is a Passive Entity and a *Process* is an Active Entity.
- Process associates PC and a set of other resources with it.
- A Program becomes a Process when an executable file is loaded into memory.
- Two Techniques are present for loading Executable Files.
  (Double Clicking EXE, Running EXE through Command Prompt)

Set of Values held by a PROCESS

Current value of Program Counter (PC)

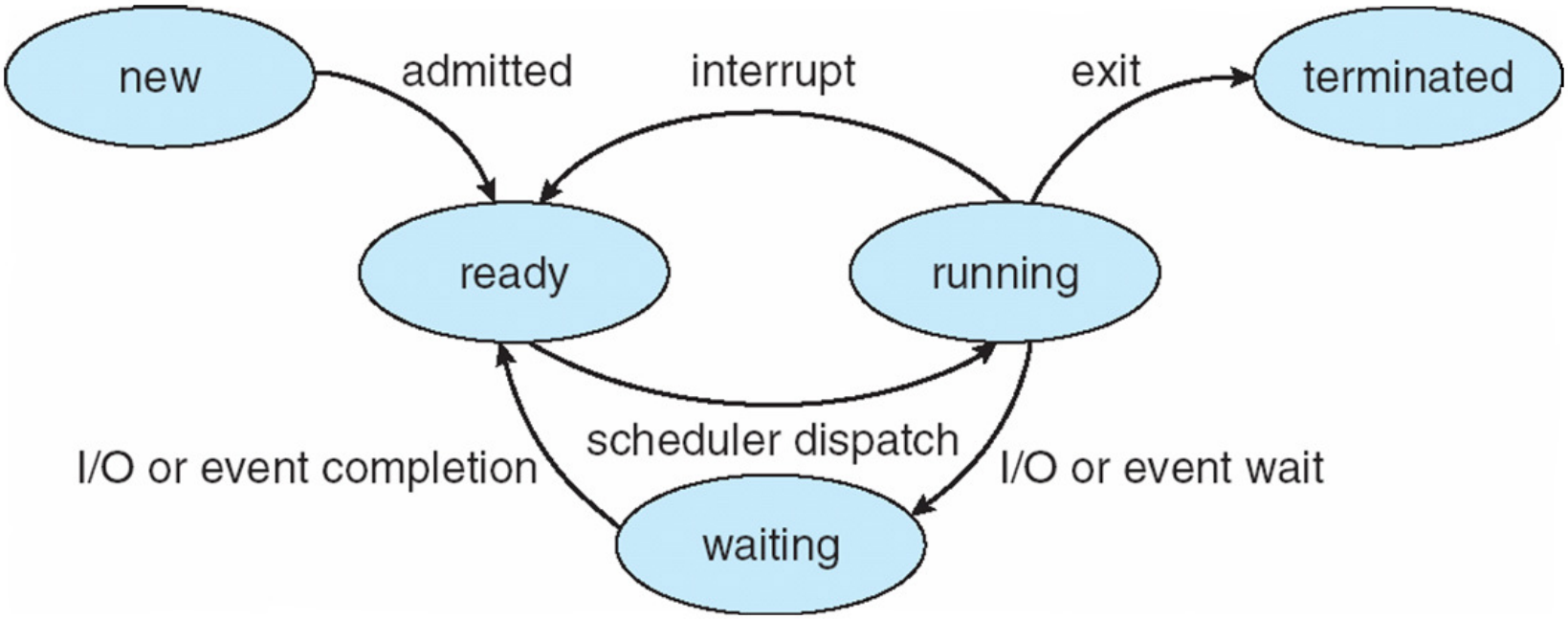
Contents of the Processors Registers

Value of the Variables

The *Process Stack* (SP) which typically contains temporary data such as subroutine parameter, return address, and temporary variables.

A Data Section that contains Global Variables.

Process States



Process States

New State: The Process being created.

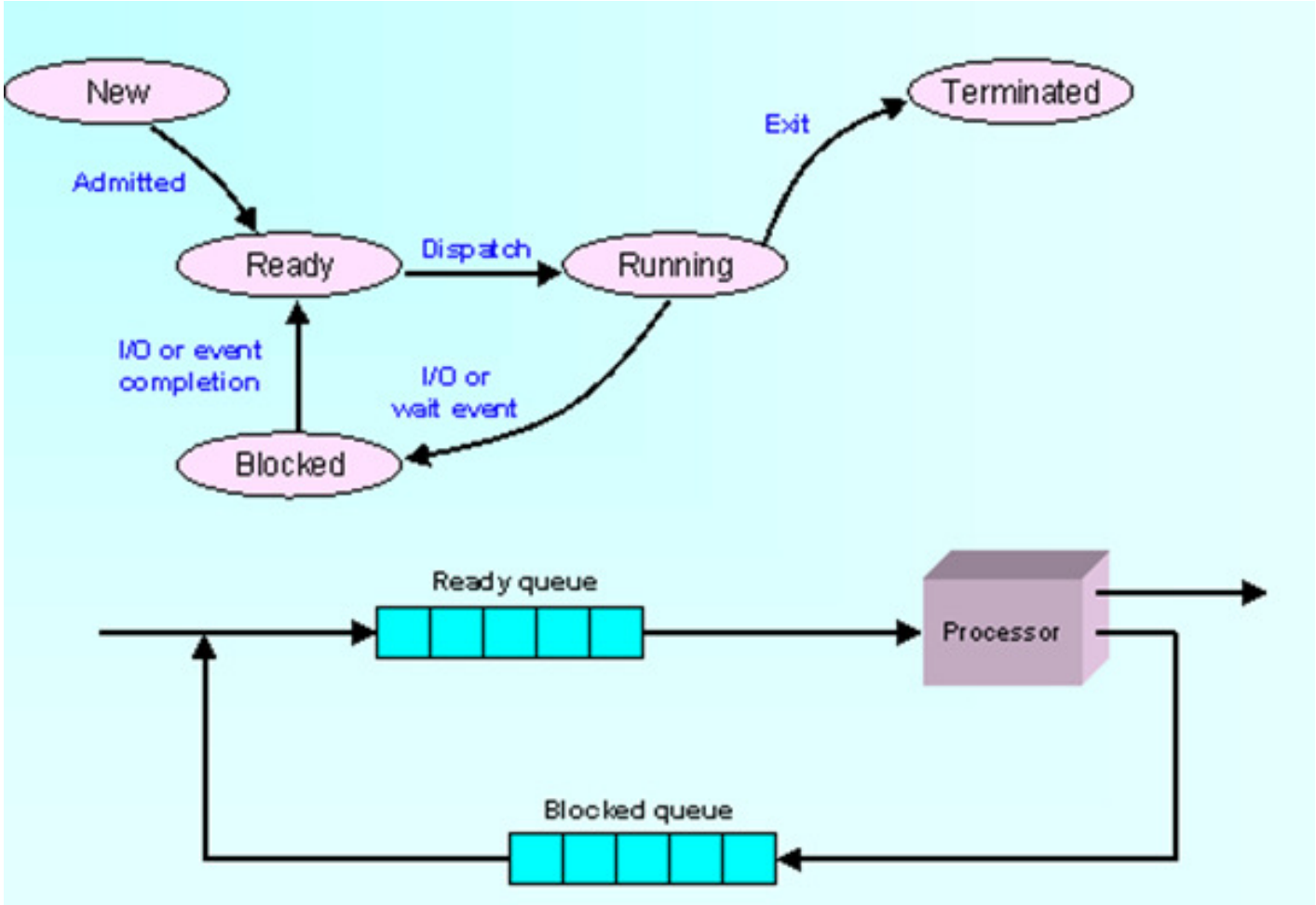
Ready State: The Process is waiting to be assigned to a Processor.

Running State: A Process is said to be running if it has the CPU.

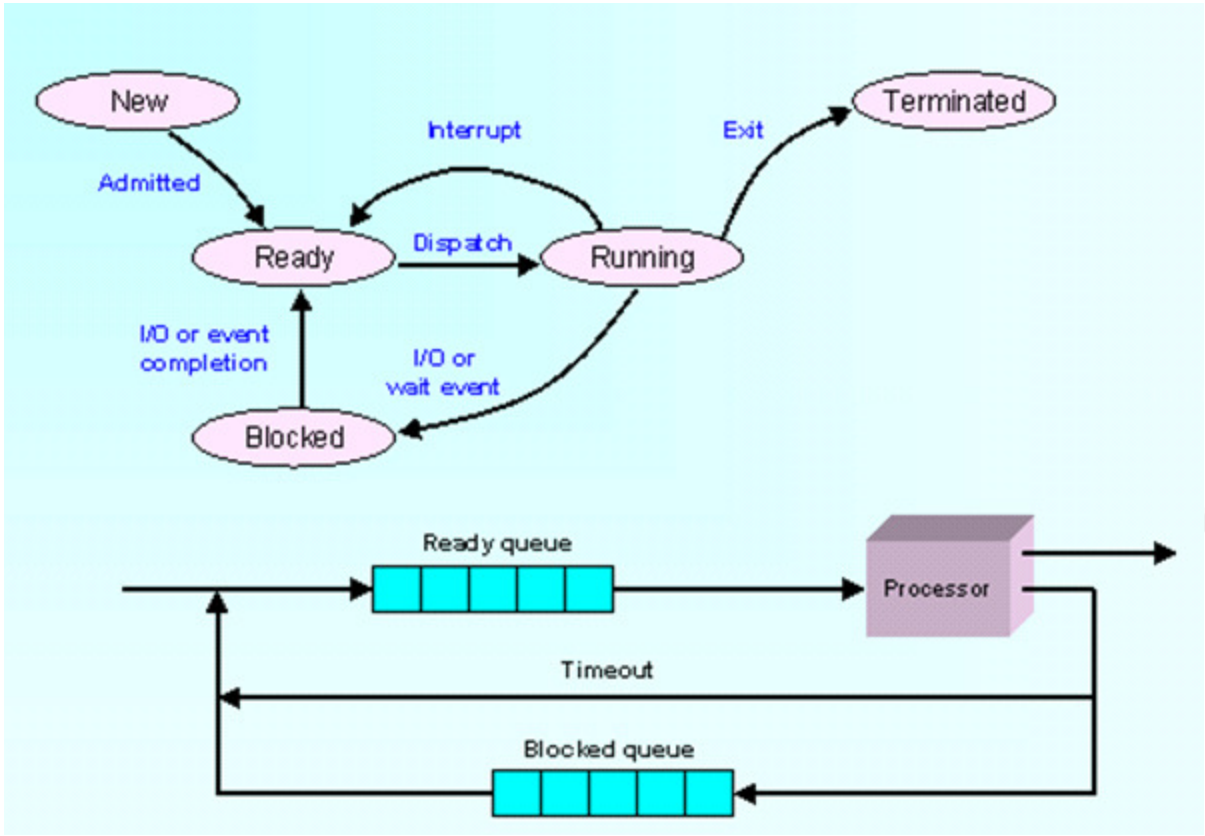
Blocked (or Waiting) State: A Process is said to be blocked if it is waiting for some event to happen.

Terminated State: The Process has finished *Execution*.

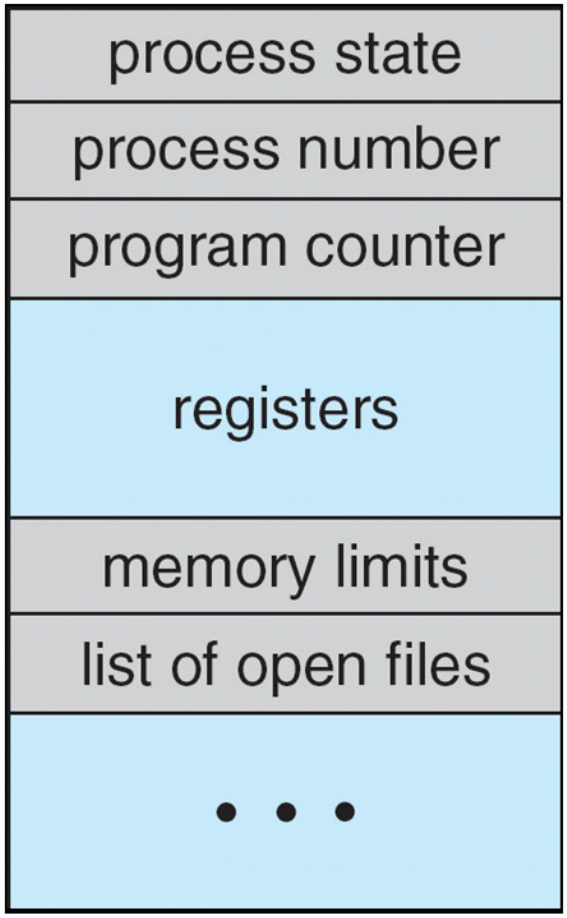
State Diagram



State Diagram



Process Control Block



Process Control Block

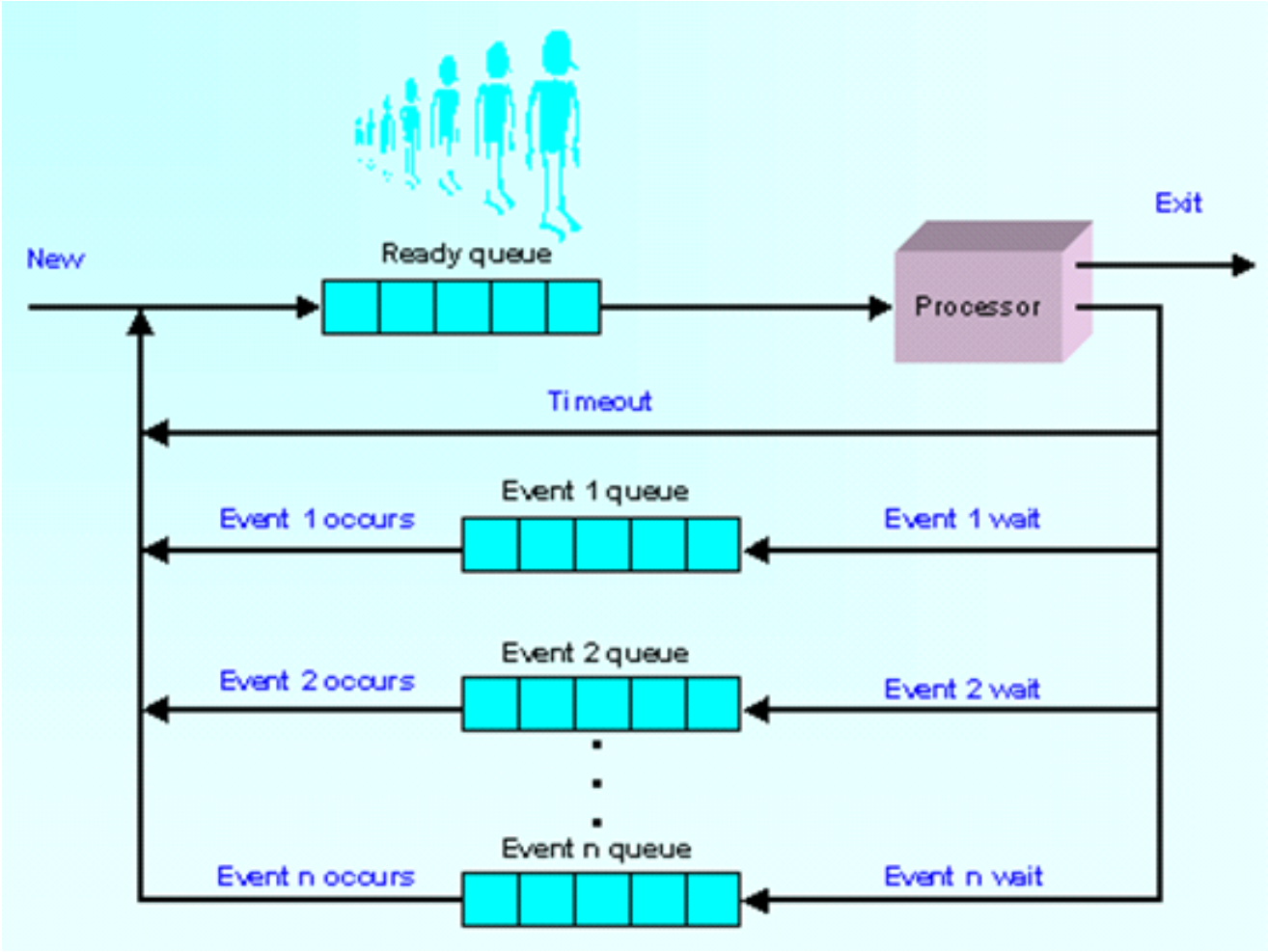
Information associated with each Process

- Process state
- Program counter
- CPU registers
- CPU scheduling information
- Memory-management information
- Accounting information
- I/O status information

Various Queues

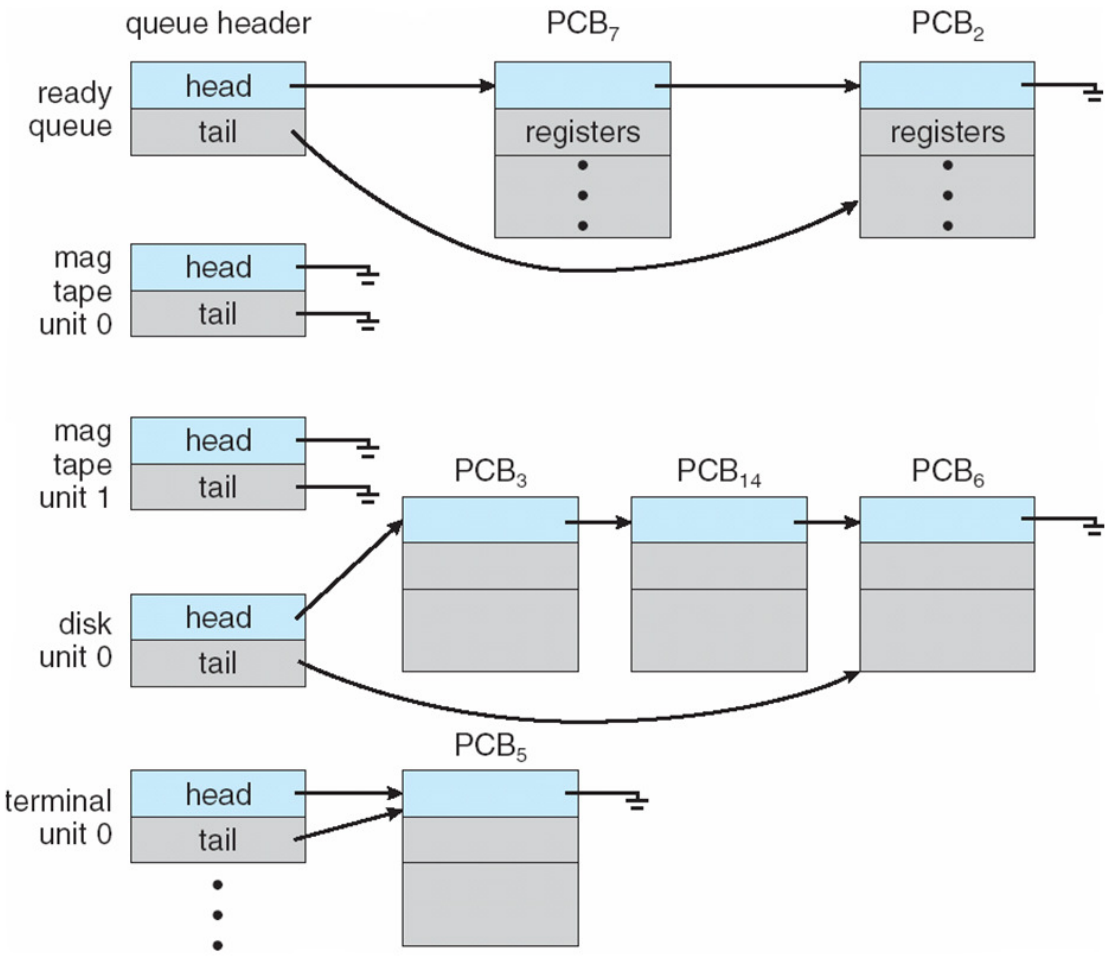
- Job Queue – Set of all processes in the system
- Ready Queue – Set of all processes residing in main memory, ready and waiting to execute
- Device Queues – Set of processes waiting for an I/O device
- Processes migrate among the various queues

Multiple Blocked Queues



Vinay Arora
CSED,TU

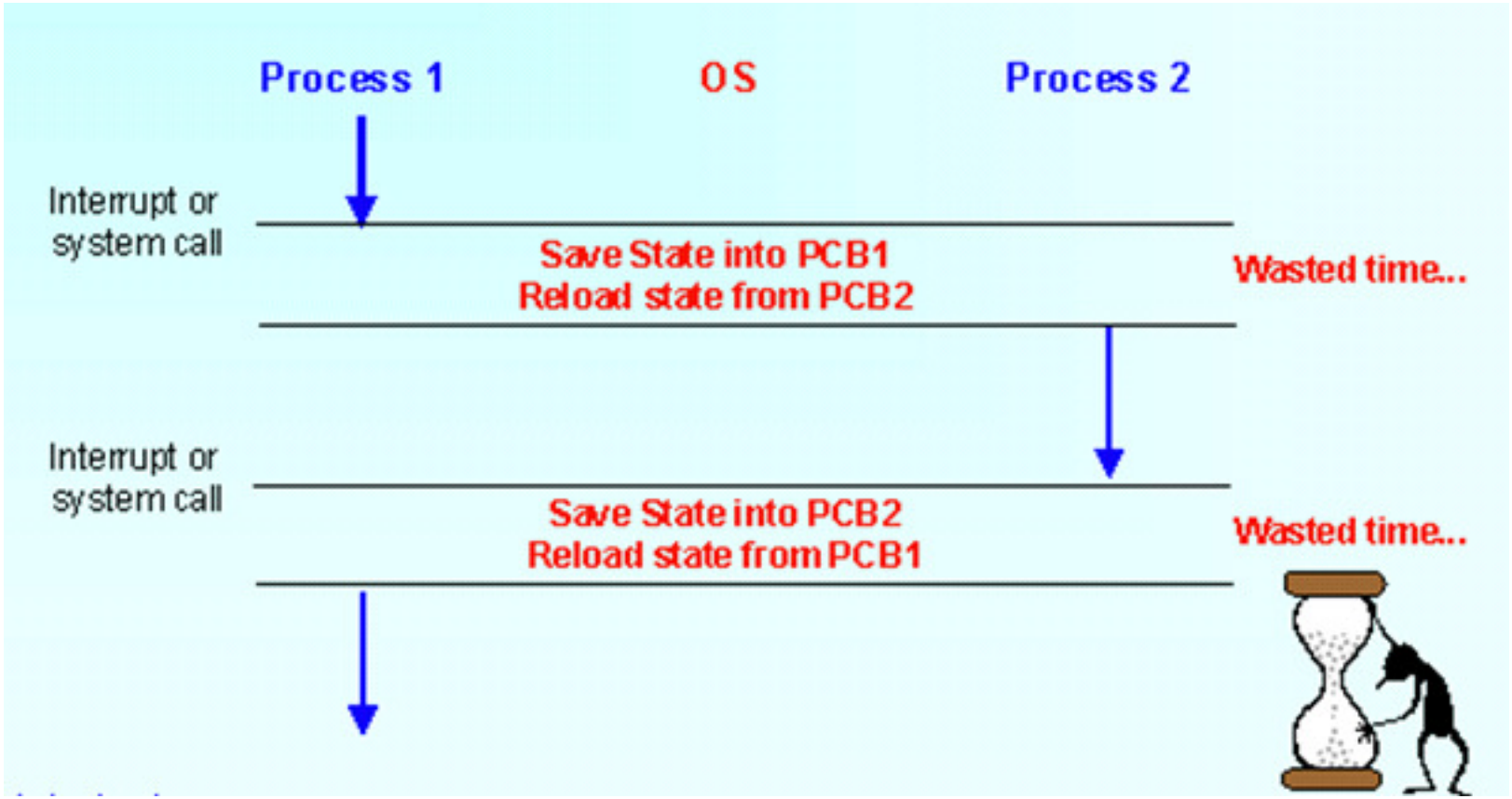
Ready Queue

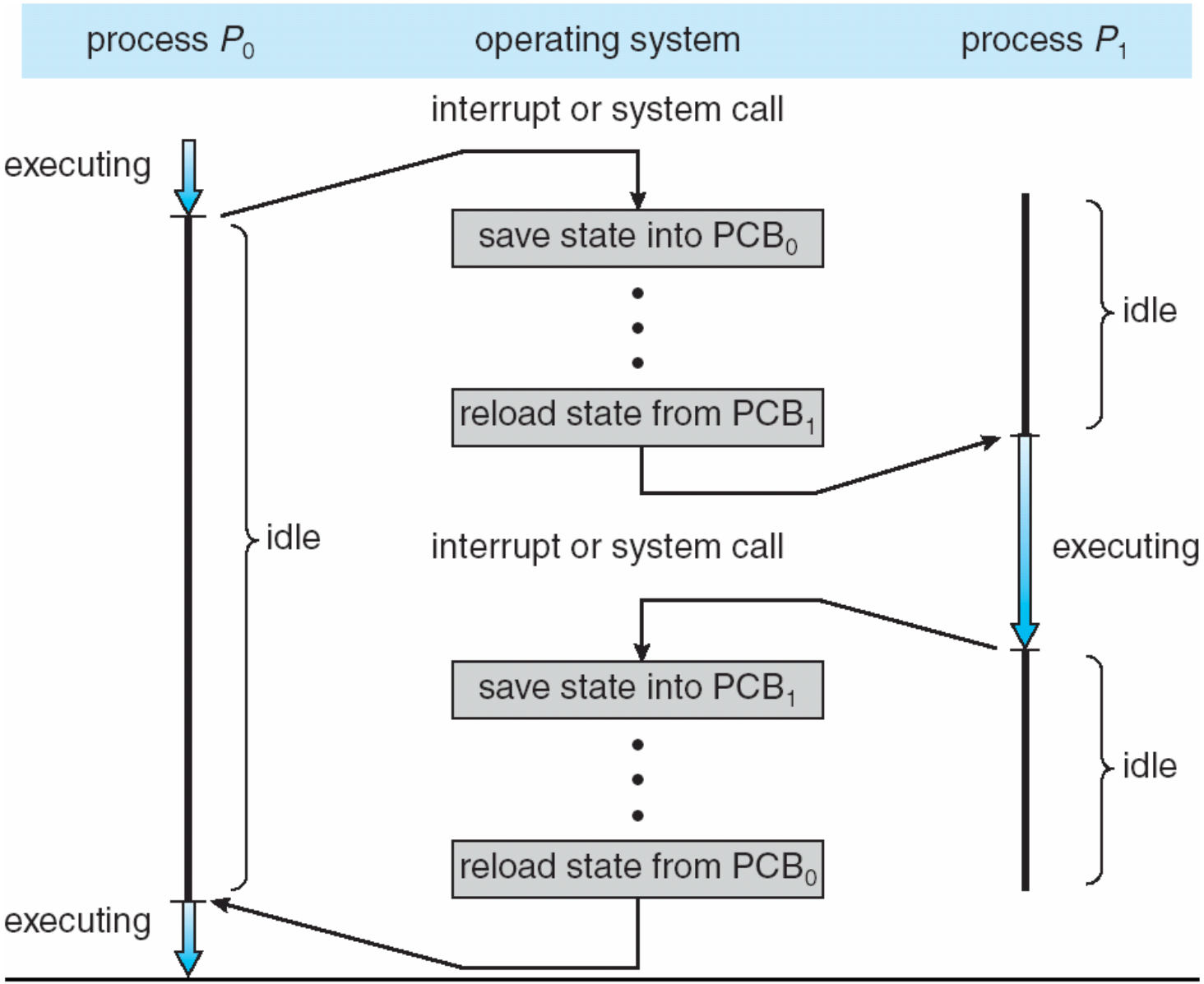

queue header
PCB7
PCB2
ready queue
head
tail
registers
registers
mag tape unit 0
head
tail
mag tape unit 1
head
tail
PCB3
PCB14
PCB6
disk unit 0
head
tail
PCB5
terminal unit 0
head
tail

Context Switch

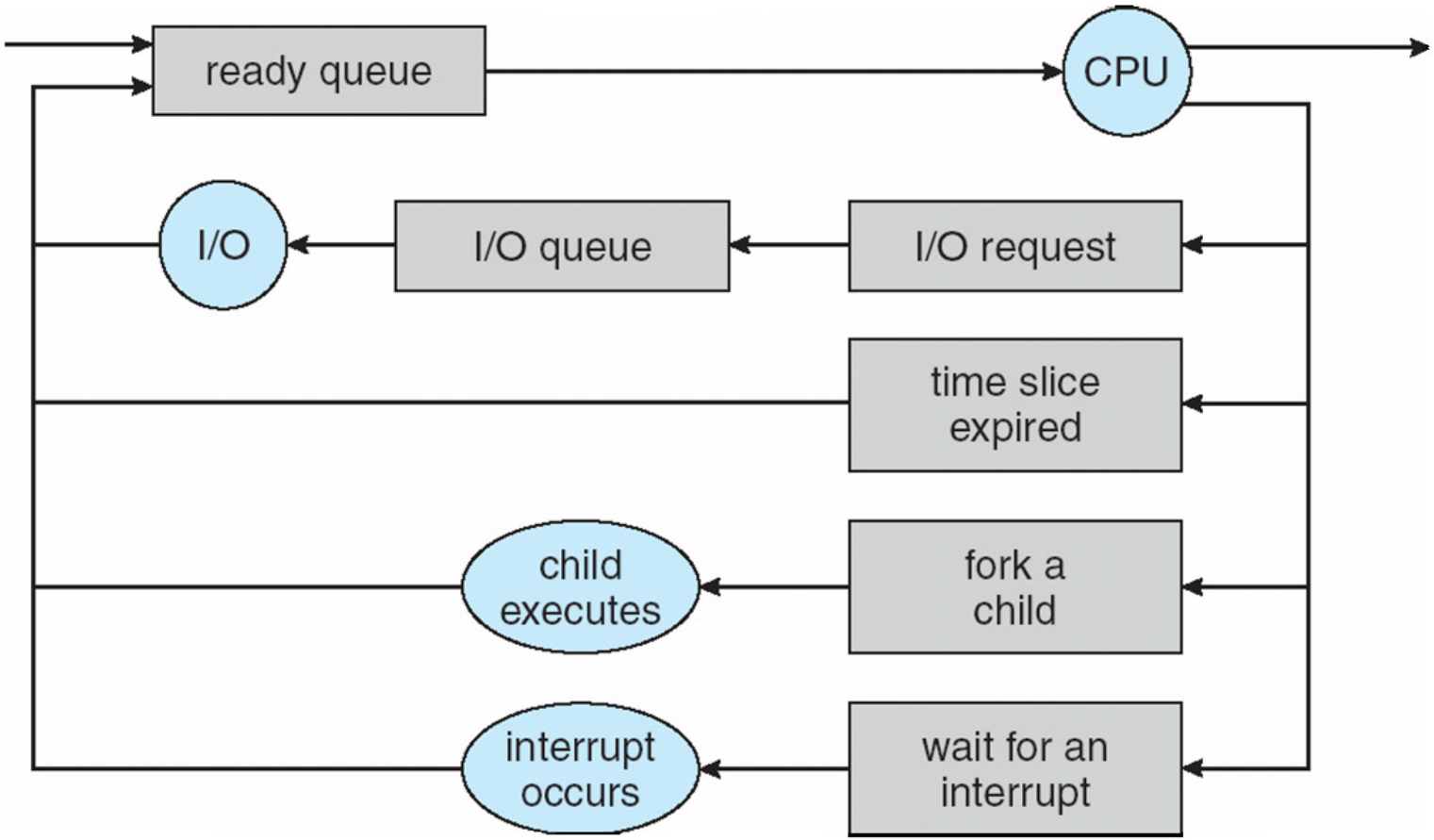
- When CPU switches to another process, the system must save the state of the old process and load the saved state for the new process via a *Context Switch.*
- *Context* of a process represented in the PCB
- Context-switch time is overhead.
- Time dependent on hardware support

Context Switching




process P_0
operating system
process P_1
interrupt or system call
executing
save state into PCB_0
reload state from PCB_1
idle
idle
interrupt or system call
executing
save state into PCB_1
reload state from PCB_0
idle
executing

Process Scheduling Diagram



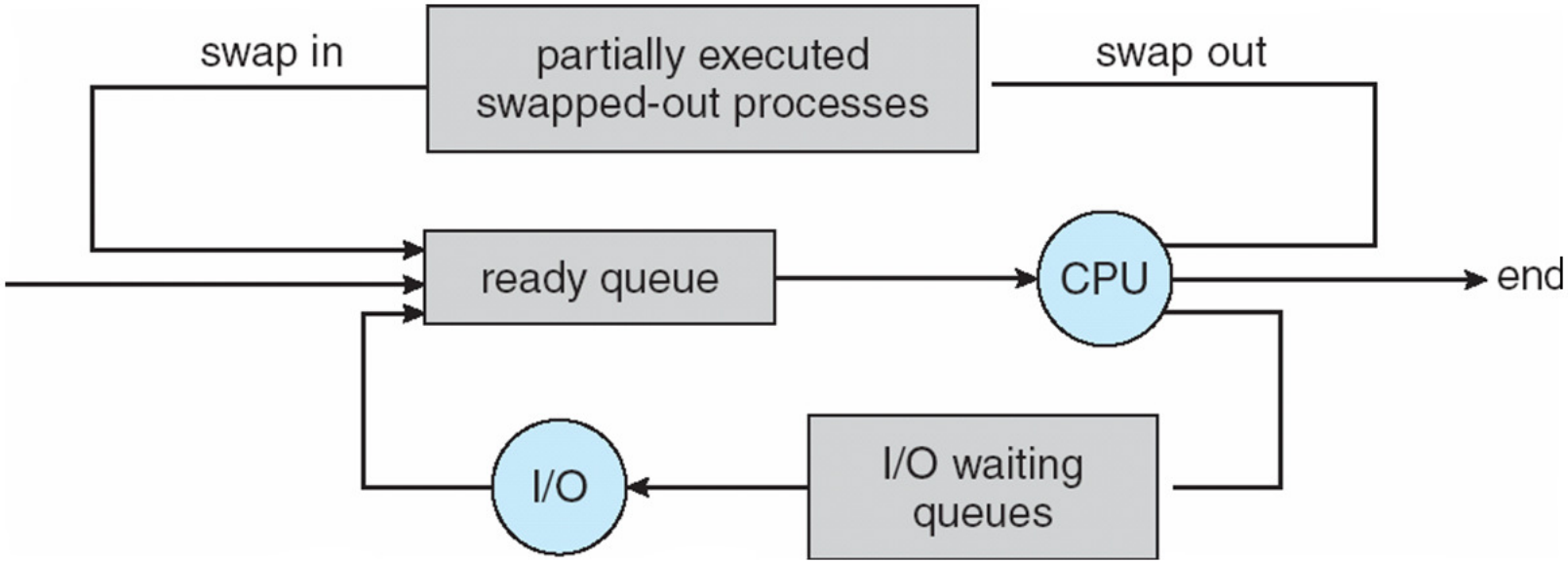
LSM Term Scheduler

- Long-Term Scheduler (or Job scheduler) – Selects which Processes should be brought into the Ready queue

- Short-Term Scheduler (or CPU scheduler) – Selects which Process should be executed next and allocates CPU

- Medium-Term Scheduler – Intermediate Level of Scheduling. Limit the Multiprogramming and executes Swapping.

Medium Term Scheduler



Parent & Child Process

- **Parent** Process create **Children** Processes, which, in turn create other Processes, forming a tree of Processes
- Generally, Process identified and managed via a **Process Identifier** (PId)
- Resource Sharing
  - Parent and Children share all Resources
  - Children share subset of parent's Resources
  - Parent and child share no Resources
- Execution
  - Parent and Children execute Concurrently
  - Parent waits until children Terminate

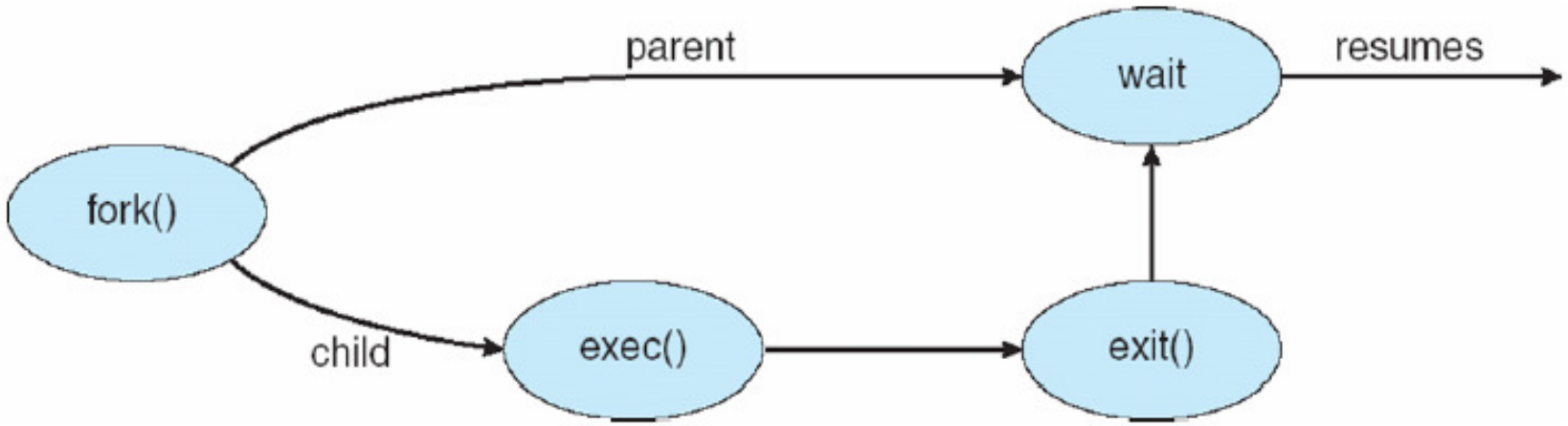
Process Creation (Unix)

- UNIX *Examples*

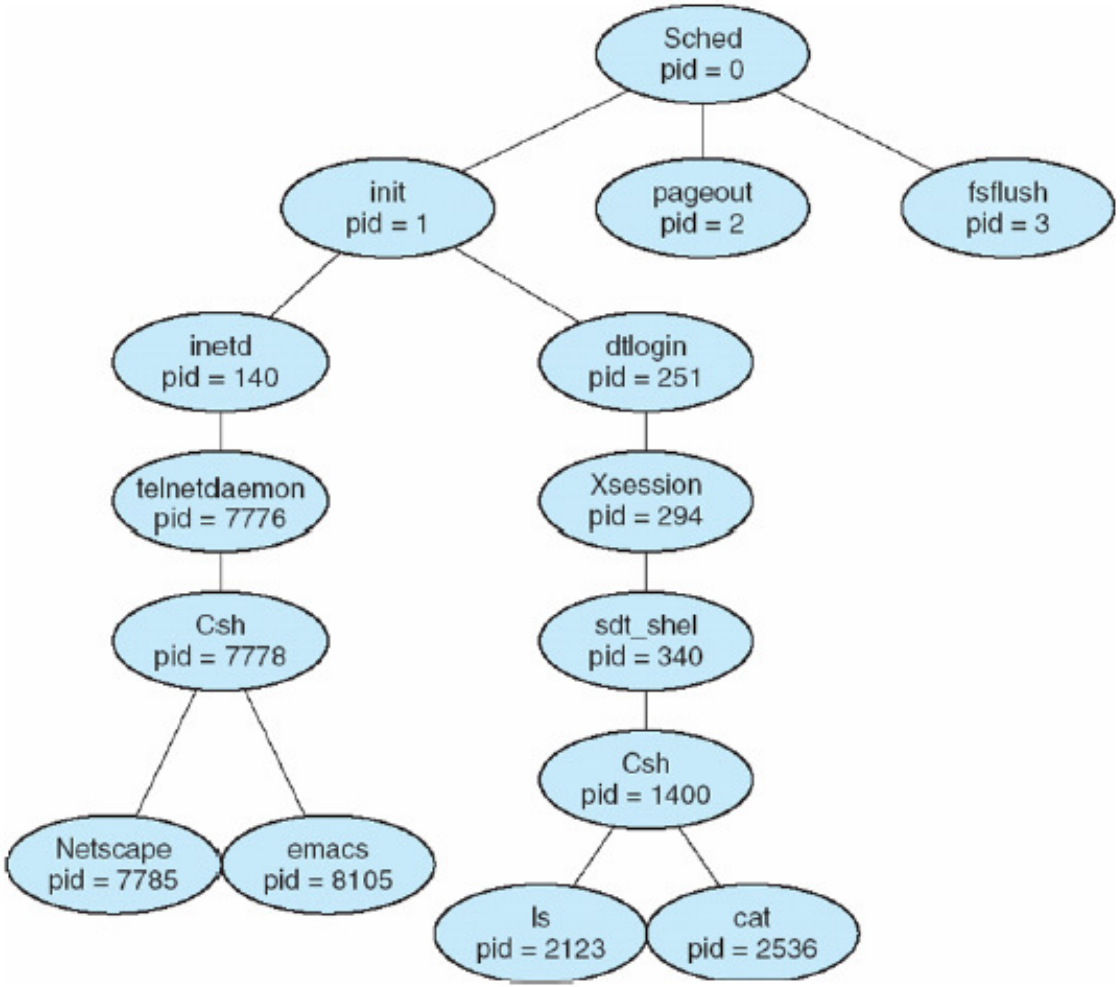
- ***Fork*** System Call creates new Process.

- ***Exec*** System Call used after a fork to replace the process' memory space with a new Program.

Process Creation (Unix)



Tree of Processes on Solaris



Vinay Arora
CSED,TU

Process Termination

- Process executes last statement and asks the operating system to delete it (exit)
  - Output data from child to parent (via wait)
  - Process' resources are deallocated by operating system
- Parent may terminate execution of children processes (abort)
  - Child has exceeded allocated resources
  - Task assigned to child is no longer required
  - If parent is exiting
    - Some operating system do not allow child to continue if its parent terminates
- All children terminated - *Cascading Termination*

InterProcess Communication

- Processes within a system may be *Independent* or *Cooperating*
- Cooperating process can affect or be affected by other processes, including sharing data
- Reasons for cooperating processes:
  - Information sharing
  - Computation speedup
  - Modularity
  - Convenience
- Cooperating processes need Inter Process Communication (IPC)
- Two models of IPC - Shared memory & Message passing

Direct Communication

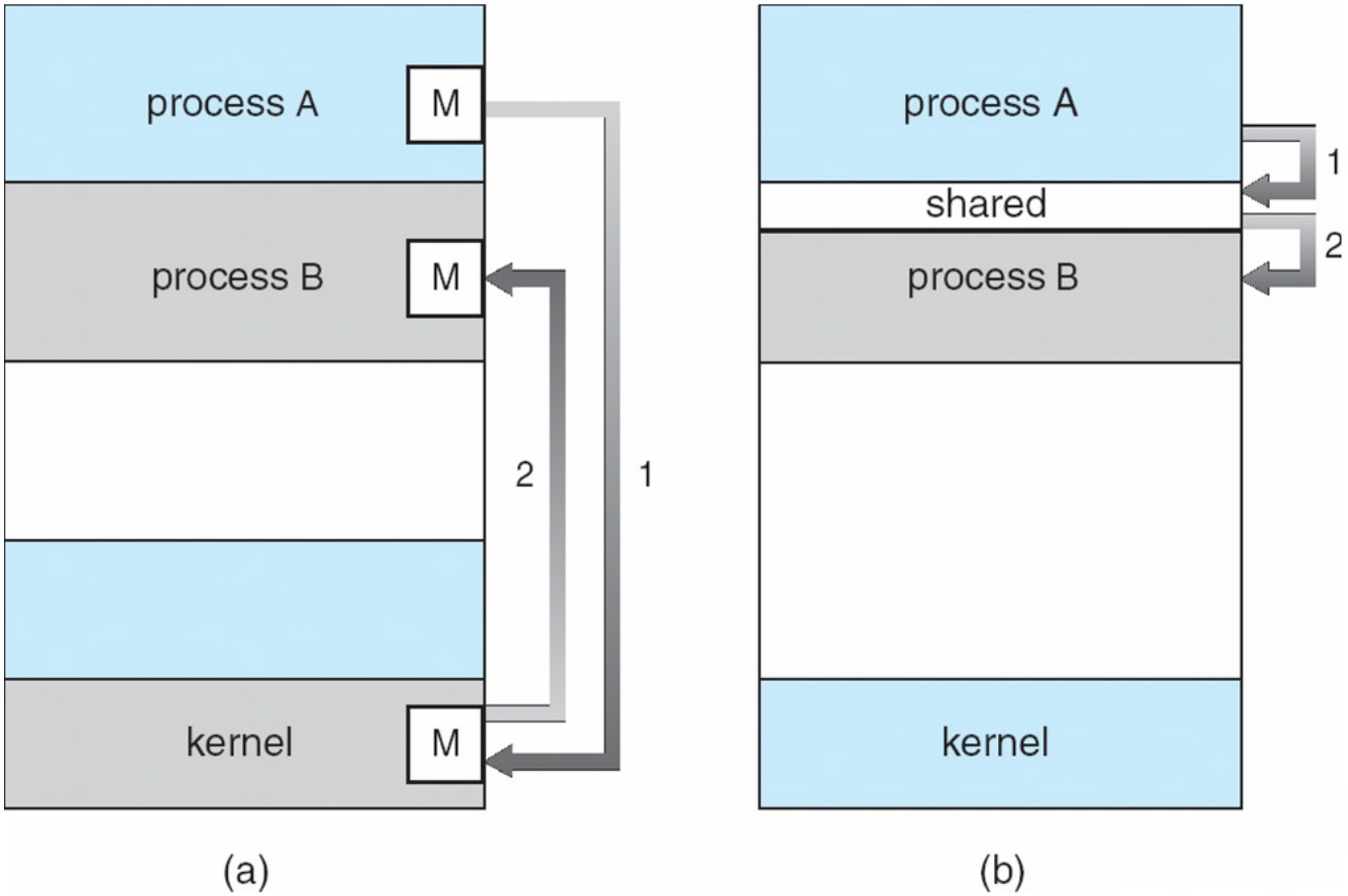
- Processes must name each other explicitly:
  - ***send (P, message)*** –send a message to process P
  - ***receive (Q, message)*** –receive a message from process Q

- Properties of Communication link
  - Links are established automatically
  - A link is associated with exactly one pair of communicating processes
  - Between each pair there exists exactly one link
  - The link may be unidirectional, but is usually bi-directional

Interaction b/w Processes

- Data Sharing – Common Variable/Data.
- Message Passing – Communication b/w 2 processes.
- Synchronization – Coordination in Activities.
- Signals – To Convey a situation to a Process so that it can perform some special actions to handle the situation.
- Concurrency & Parallelism.
- Computational Speed Up.
- Scheduling & Dispatching.

Message Passing & Shared Memory



Vinay Arora
CSED,TU

Thnx…