# CS343 Operating Systems

## Lecture 2

# Introduction to Process States


**John Jose**

**Associate Professor**

**Department of Computer Science & Engineering**

**Indian Institute of Technology Guwahati**

# Operating System Services

- The OS structure is divided into many sub-components.
  - Process Execution
  - Process Management
  - Memory Management
  - File Management
  - Storage Management
  - I/O Sub-system Management
  - Protection and Security
  - User Interface

# Process Management

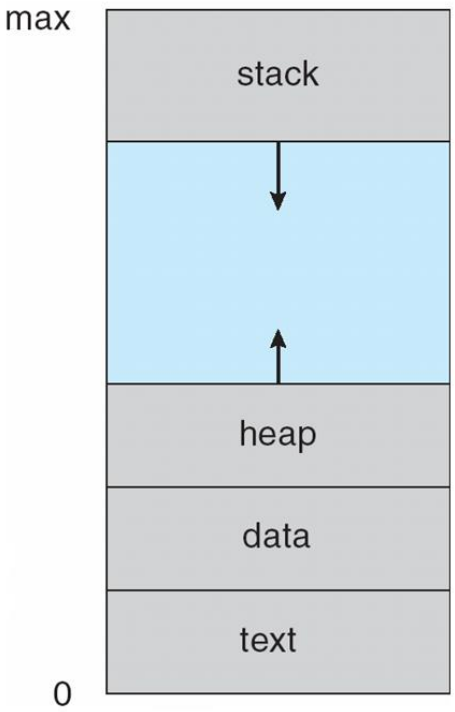
- Creating and deleting both user and system processes
- Suspending and resuming processes
- Providing mechanisms for process synchronization
- Providing mechanisms for process communication
- Providing mechanisms for deadlock handling

# Process Concept

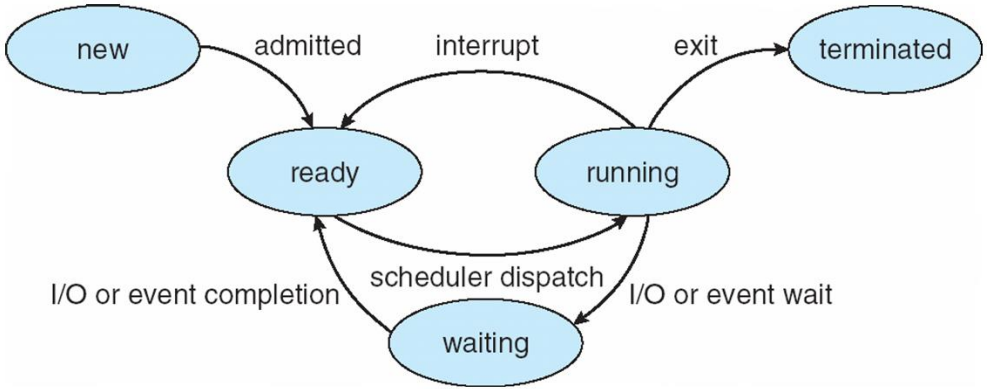
- A process is a program in execution
- It is a unit of work within the system
- Program is a passive entity, process is an active entity
- Process needs resources to accomplish its task
- These resources include CPU, memory, I/O, files, etc.
- Program becomes process when executable file loaded into memory
- Program execution is initiated by GUI mouse clicks / command line entry

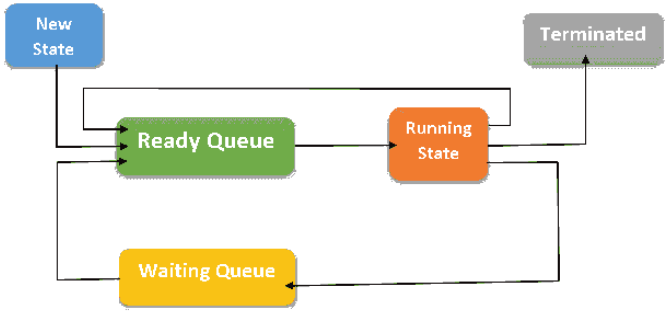
# Process Concept

- One program can have several processes
- **Process** has multiple parts
  - The program code, also called **text section**
  - Current activity - **program counter**, registers
  - **Stack** containing temporary data like function parameters, return addresses, local variables
  - **Data section** containing global variables
  - **Heap** -dynamically allocated memory during run time



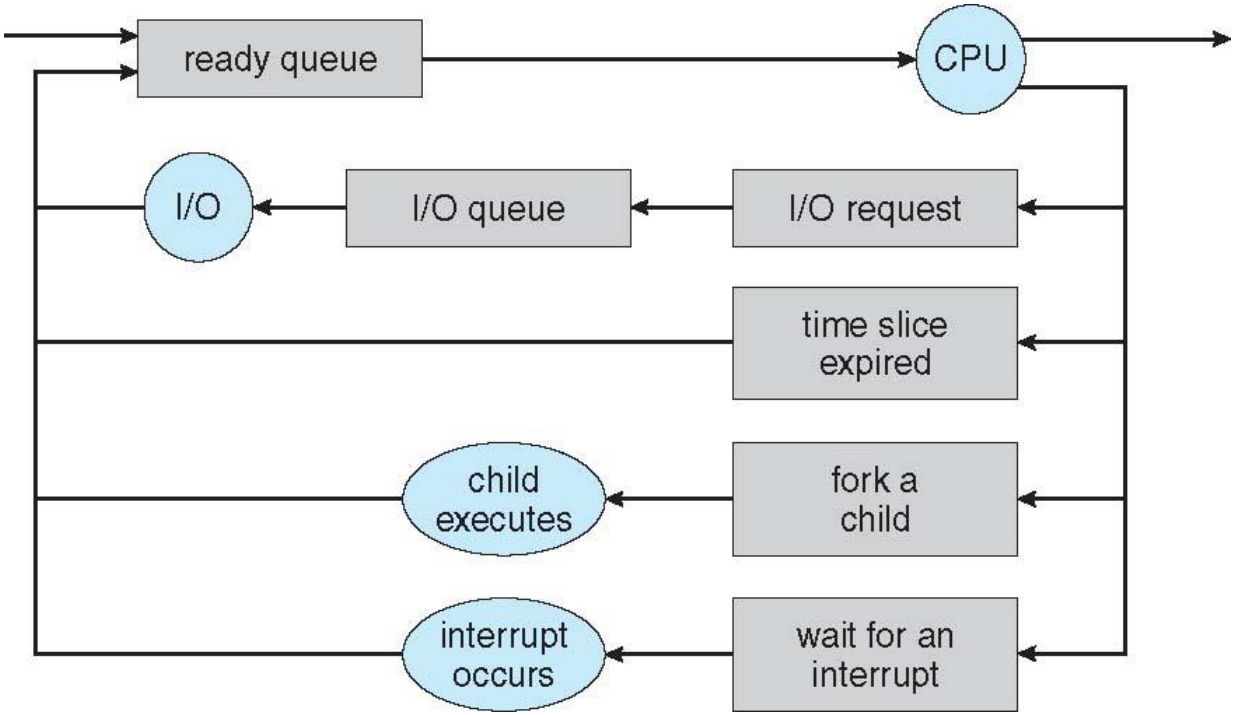
# Process State Diagram



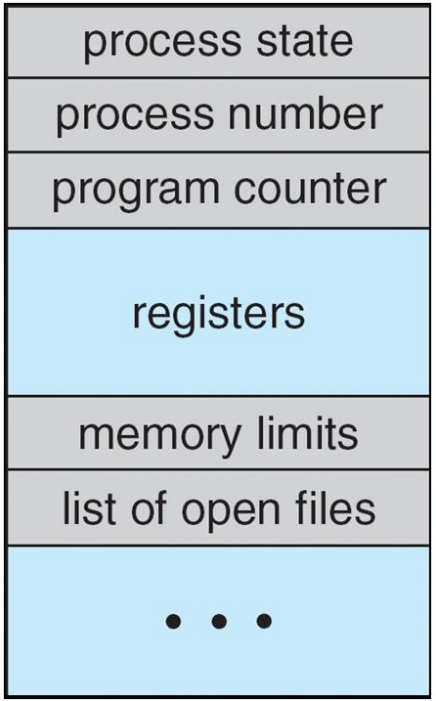

- **new**: The process is being created
- **running**: Instructions are being executed
- **waiting**: The process is waiting for some event to occur
- **ready**: The process is waiting to processor assignment.
- **terminated**: The process has finished execution

# Representation of Process Scheduling



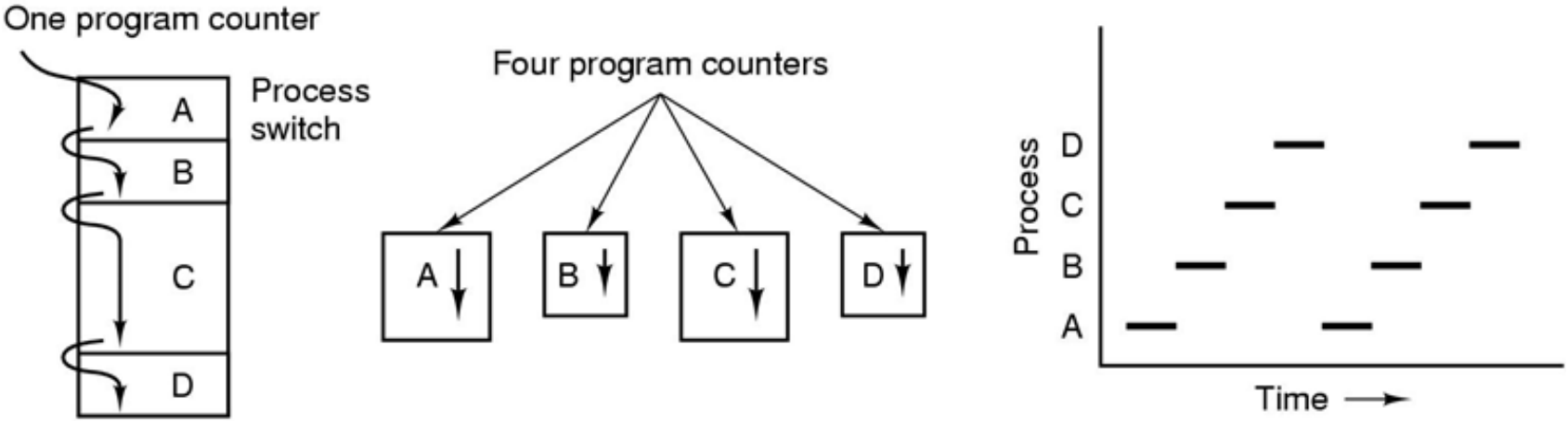
# Process Control Block (PCB)

- Process state – running, waiting, etc
- Program counter – location of instruction to next execute
- CPU registers – contents of all process-centric registers
- CPU scheduling information- priorities, scheduling queue pointers
- Memory-management information – memory allocated to the process
- Accounting information – CPU used, clock time elapsed since start, time limits
- I/O status information – I/O devices allocated to process, list of open files

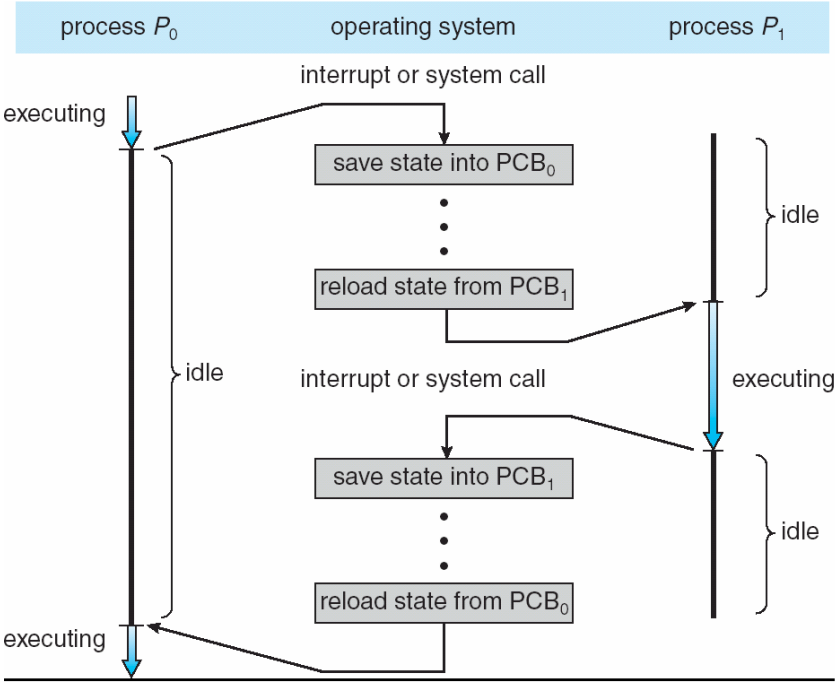


# Multiprogramming

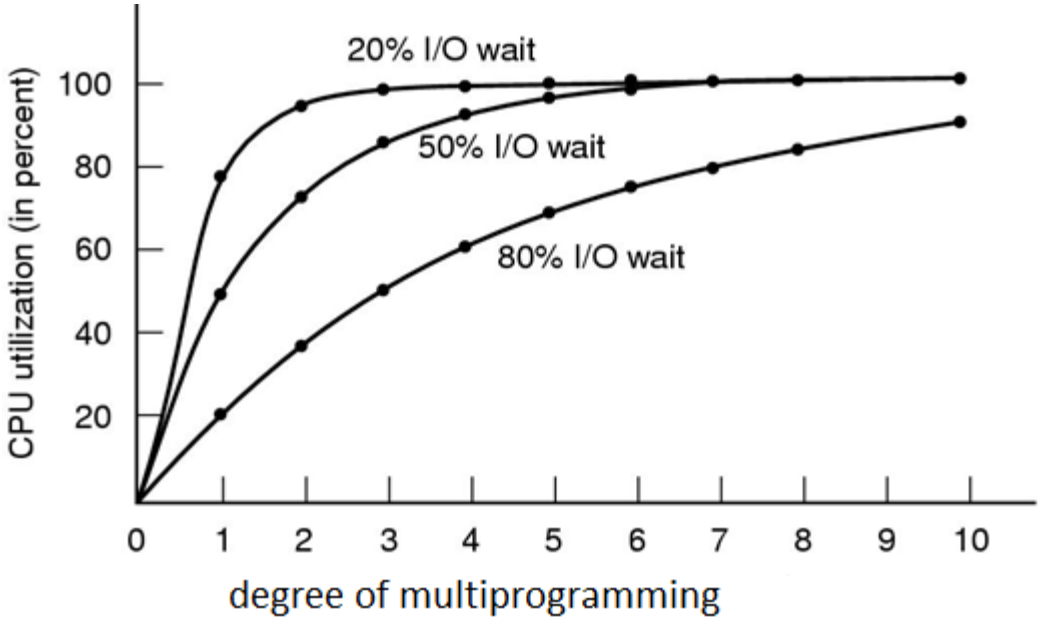
- In a uni-processor system, CPU is running only one process.
- In a multiprogramming system, CPU switches from processes quickly.



# Context Switch From One Process to Another

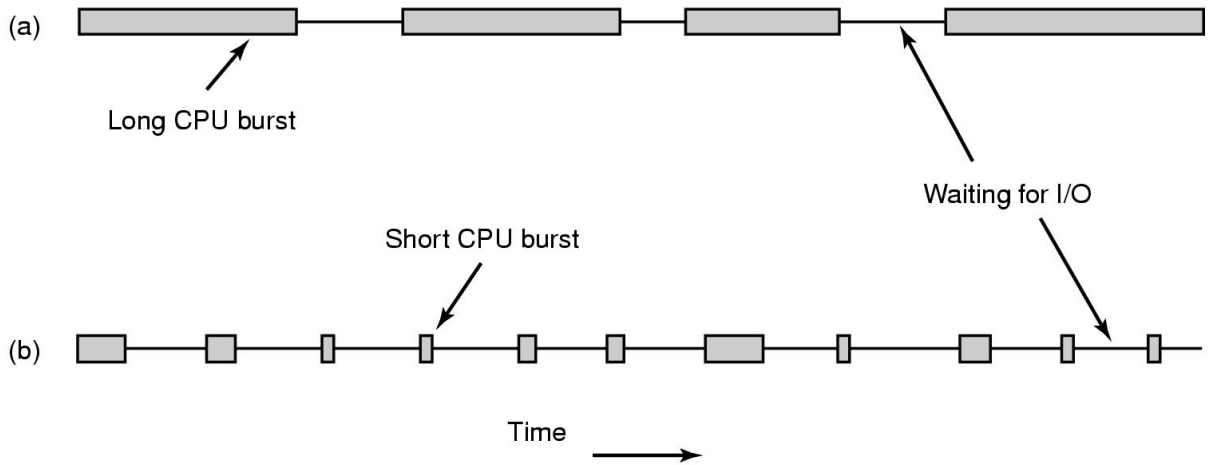


# Modeling Multiprogramming




CPU utilization as a function of the number of processes in memory.

# CPU vs I/O Bound Processes



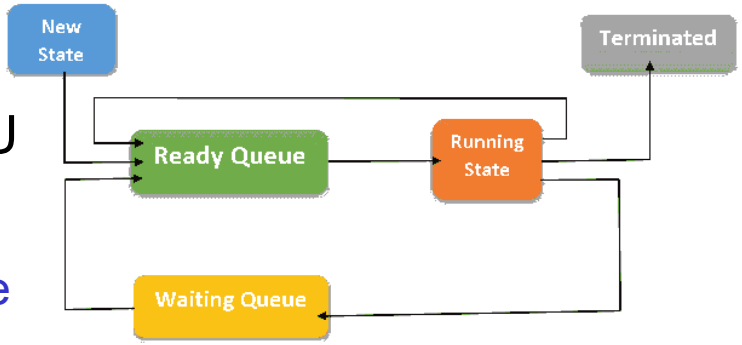

Bursts of CPU usage alternate with periods of waiting for I/O.
(a) A CPU-bound process. (b) An I/O-bound process.

# Process Scheduling

- Maximize CPU use, quickly switch processes onto CPU for time sharing
- **Process scheduler** selects among available processes for next execution on CPU
- Maintains **scheduling queues** of processes
  - **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

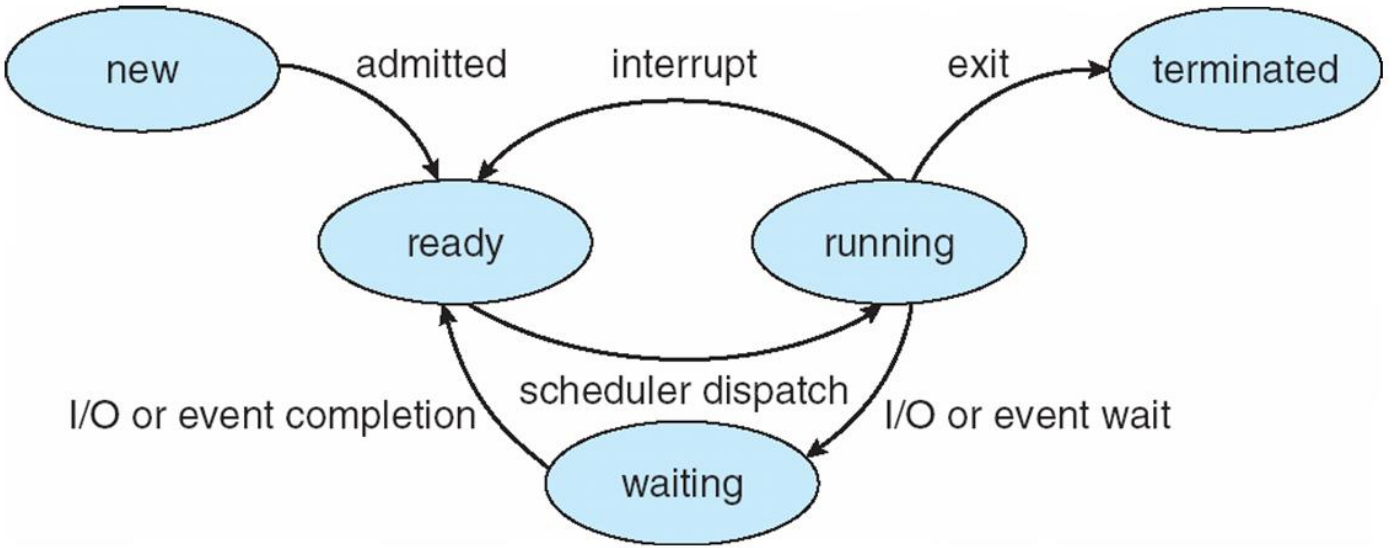
# CPU vs Job Schedulers

- **Short-term scheduler (CPU scheduler)** selects from among the processes that are ready to execute and allocates the CPU to one of them.
  - Selection from Ready state to Running state
  - Short-term scheduler is invoked frequently




- **Long-term scheduler (Job scheduler)** – selects which processes should be brought into the ready queue from the job queue
  - Long-term scheduler is invoked less frequently
  - It controls the **degree of multiprogramming**

# Process State Diagram



# CPU Scheduler

- Whenever the CPU becomes idle, the OS must select one of the processes in the ready queue to be executed.
- The selection process is carried out by the **CPU scheduler** of the OS.
- The scheduler selects a process that is in ready state and allocates the CPU to that process.
- Ready queue can be implemented as
  - FIFO queue
  - Priority queue
  - Tree
  - Unordered linked list

# Dispatcher

- The **dispatcher** is the module that gives control of the CPU to the process selected by the CPU scheduler.
- Function of dispatcher involves the following:
  - Switching context
  - Switching to user mode
  - Initiate restarting/ resumption of selected user program
- Dispatcher should be as fast as possible, since it is invoked during every process switch.
- The time it takes for the dispatcher to stop one process and start another running is known as the dispatch latency.


**johnjose@iitg.ac.in**
**http://www.iitg.ac.in/johnjose/**