CSC-342: Operating Systems

Muhammad Afzal Bhatti (PhD)
Professor
Department of Computing and Technology
Iqra University, Islamabad Campus

CSC-342: Operating Systems

- Main aim of the course is to help students gain a general understanding of the principles and concepts governing the functions of operating systems and acquaint students with different choices made in Operating System design and implementation and their effects on applications, and user environments

CSC-342: Operating Systems

Course Topics

- Introduction to the Course
- Operating System: An overview
- Types of Operating Systems with historical perspective
- Process, States of a Process, Process Model
- Process Models for Different Types of Operating Systems

CSC-342: Operating Systems

Course Topics

- System and OS Structure
  - Systems Calls and Interrupts
- Process Management
  - Process and Threads
    - Process Vs Thread Scheduling
  - CPU/Process Scheduling Algorithms
- Process Coordination/Synchronization
  - Managing Critical Section Problem
    - Approaches for Critical Section Problem
      - Software, Hardware, OS, HLL Constructs
    - Classical Problems of Synchronization

CSC-342: Operating Systems

Course Topics

- Inter Process Communication (IPC)
  - Use of IPC (Messages) for Classical Synchronization Problems
- Deadlocks
  - Prevention, Avoidance, Detection, Recovery
- Memory Management
  - Contiguous Memory allocation
    - Single/Multiple Partitions
    - Fixed/Variable partitions Allocation Schemes
  - Non-Contiguous Memory allocations
    - Paging and Segmentation
    - Overlays, Virtual Memory; Demand Paging/Segmentation
      - Page Replacement Algorithmes

5

CSC-342: Operating Systems

Course Topics

- Information Management
  - I/O Management, Disk Space Management
    - Disk blocks allocation methods
  - File Organization and Access Methods
    - Disk (access) scheduling Policies
  - File Management System
    - Directory Structures
  - File System: Sharing and Protection
    - File System Case Studies
- OS Structure
- Course topics Overview

6

CSC-342: Operating Systems

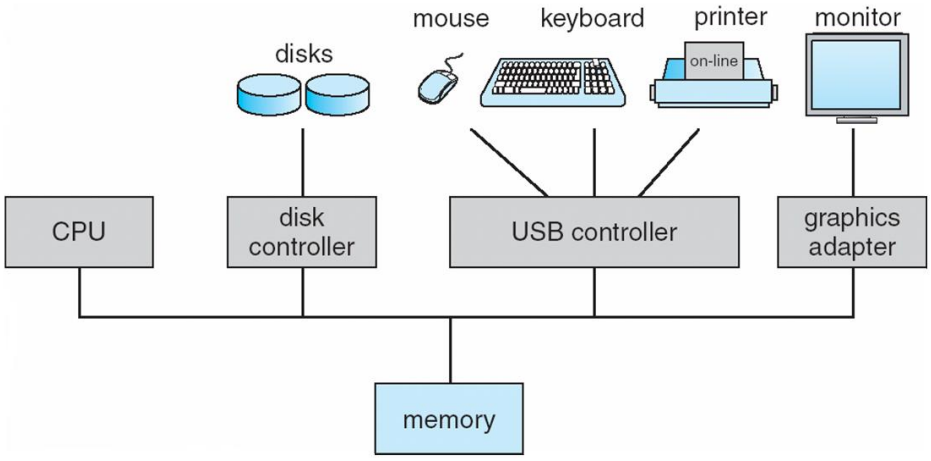
Readings/Text-Book

- Operating System Concept 9th edition, Abraham Silberschatz, et al, Wiley 2013, ISBN: 9781118063330
- Operating systems: internals and design principles, 9th edition, William Stallings, Pearson Education 2012
- https://cs-concepts-hub.github.io/Operating-Systems-Concepts/

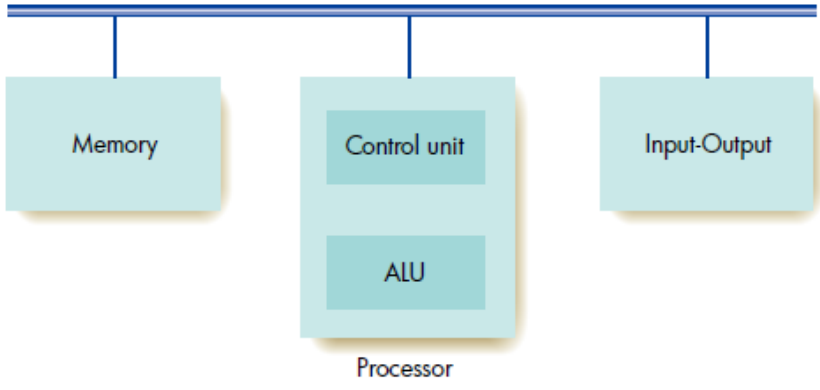
Computer/Operating System

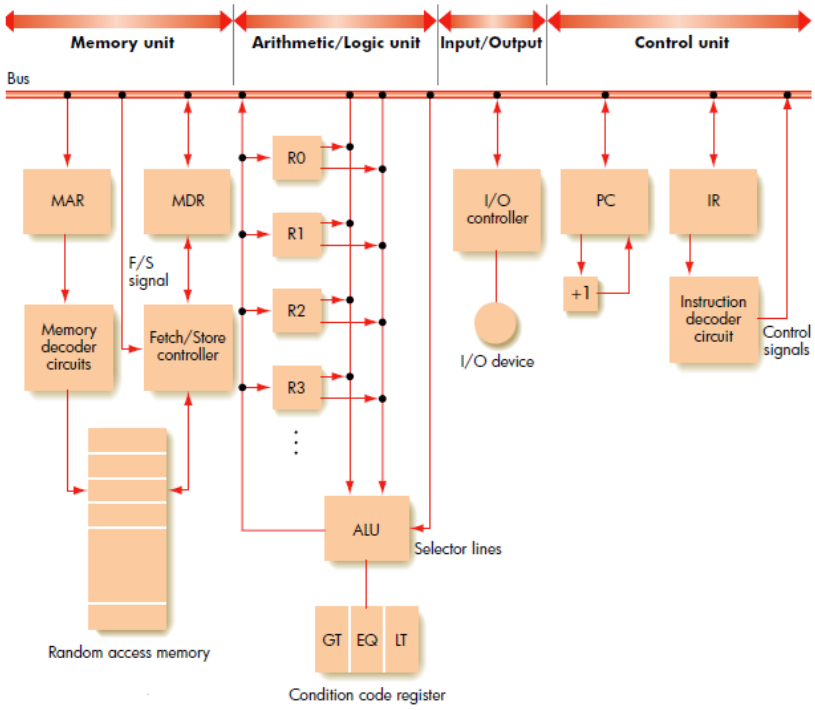
- Fundamental goal of computer systems is to execute user application programs and to make solving user problems easier
- Computer hardware is constructed towards this goal
- Hardware alone is not particularly easy to use, application programs are developed
  - require certain common operations, of controlling I/O devices, memory and CPU
- Common functions of controlling and allocating resources are then brought together into one piece of software (operating system)

Computer System

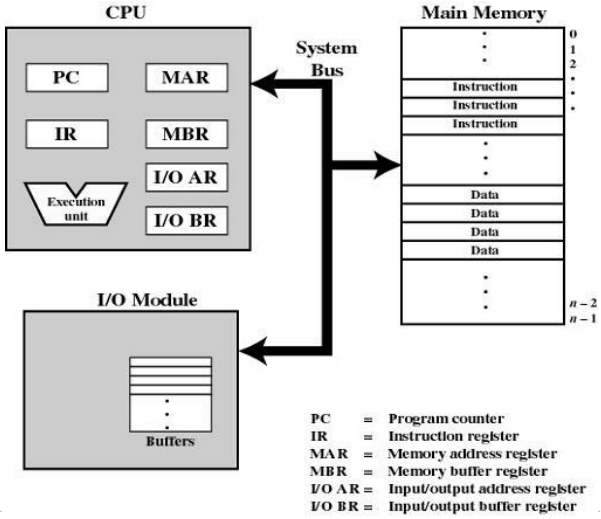


Components of Von Neumann Architecture

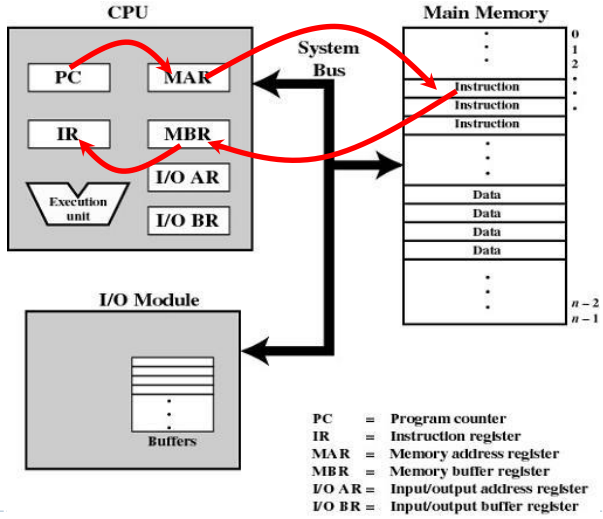




Computer System

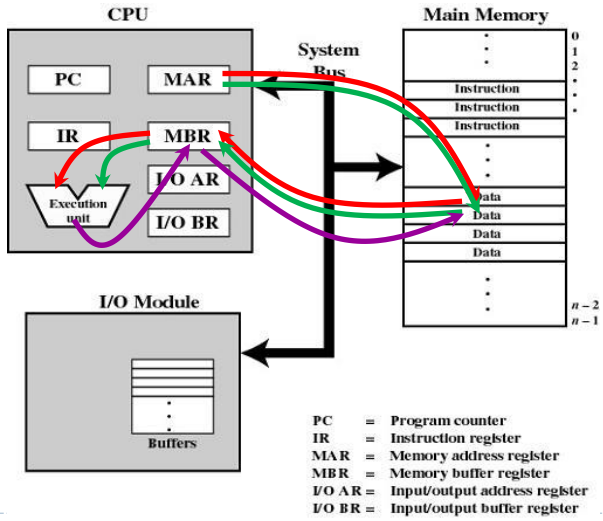


Instruction Execution



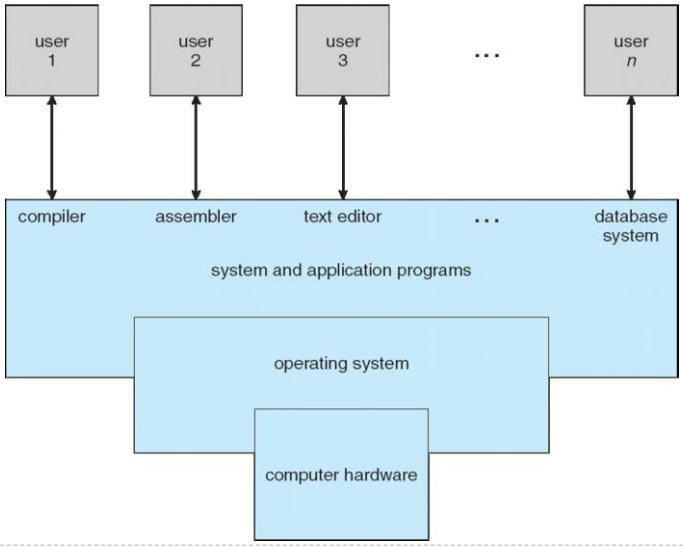
13

Instruction Execution

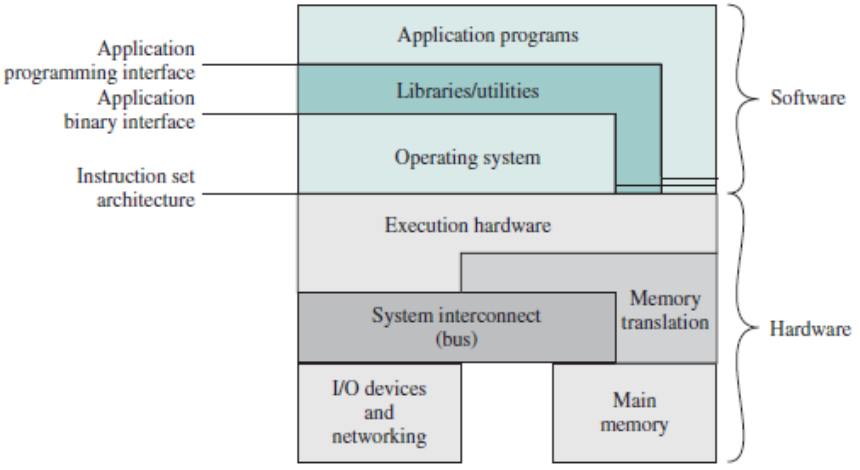


14

Computing Environment



Computer System



Operating System Objectives

- Convenience
  - An OS makes a computer more convenient to use
- Efficiency
  - An OS allows the computer system resources to be used in an efficient manner
- Accessibility
- Uniformity

Operating System

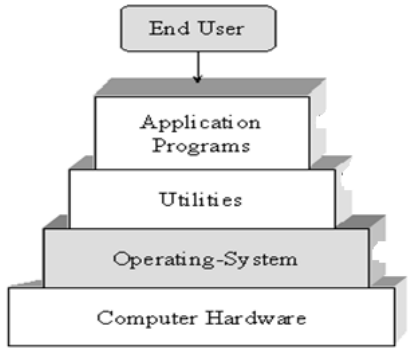
- Operating System ?
  - A control program(s) which manages computer resources and allocate these resources among competing jobs
- "An operating system is a set of manual and automatic procedures that enable a group of people to share a computer installation efficiently" [P. B. Hansen 1973]

Operating System Services

- Program Development
  - Editors, debuggers, frameworks
- Program Execution
  - Initialization, scheduling
- Access to I/O Devices
  - Uniform interface, hides details
- Controlled Access to Files
  - Authorization, sharing, caching
- System Access
  - Protection, authorization, resolve conflicts
- Error detection and response
  - Hardware/Software errors: memory error or device failure

OS: An Overview

- OS controls all the computer resources and provide the base upon which application programs are developed



- Bottom-up View
  - Resource Manager
- Top-down View
  - Base Environment

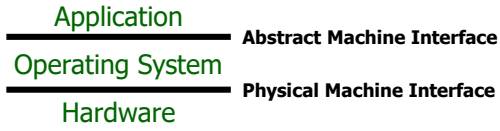
What OS Does ?

- Manage Resources
  - CPU
  - Memory
  - Storage Devices
  - I/O devices
- Manage sharing of resources among multiple applications
- How ?
  - By implementing specific algorithms
  - Maintaining Tables/Lists/Queues

What OS Does ?

- Manage Resources
  - CPU
  - Memory
  - Storage Devices
  - I/O devices
- Manage sharing of resources among multiple applications
- How ?
  - By implementing specific algorithms
  - Maintaining Tables/Lists/Queues

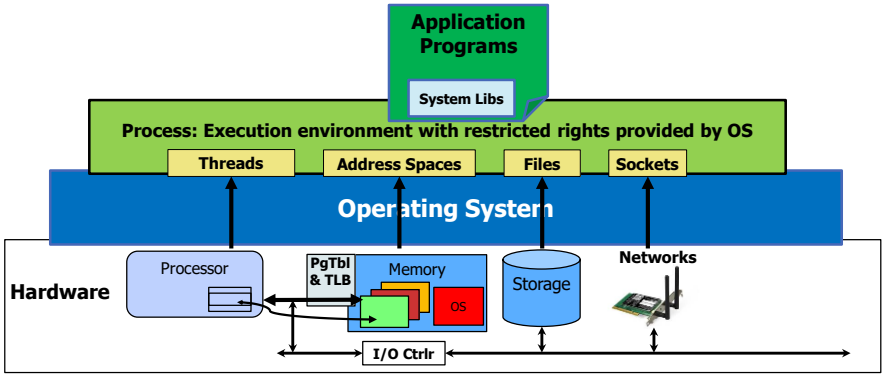
OS → Abstractions to Applications



- Provide abstractions to applications
  - File systems
  - Processes, threads
  - VM, containers
  - Naming system
  - …

Processor → Thread
Memory → Address Space
Disks, SSDs, … → Files
Networks → Sockets
Machines → Processes

OS: Abstractions of Hardware



OS: An Overview

- An Operating System is the most widely used piece of software in any computer
- Required for operations of computer
  - Kernel
  - System Programs/Utilities
- Software
  - System Vs Application
    - Functionality, Complexity, Size
  - Desired qualities in an Operating System
    - Reliable, Efficient, Provide Protection
    - Predictable, Convenient

OS Types

Designer's View

- Batch
- Resident Monitor
- Multiprogramming
- Timesharing
- Network
- Distributed

User's View

- Single User
  - Single/Multi-Tasking
- Multi User
  - Multi-Tasking
    - Foreground/Background

OS Types (History)

- Early Computers
  - Plug boards (console), Machine Level programming, Paper tape, Punched cards, Printer
  - No Operating System (User/Designer/Programmer)
  - Inconvenient
  - Time Waste (Resources Under-utilized)

OS Types

OS Types (History)

- Dedicated Computer Systems

OS Types (History)

- Dedicated Computer Systems
  - Card Readers,
  - Assemblers, lir
  - IO routines se
  - H.L.L. FORTRA
  - Two stage Cor
  - Setup time in
    - CPU Under-uti

Computer Operator

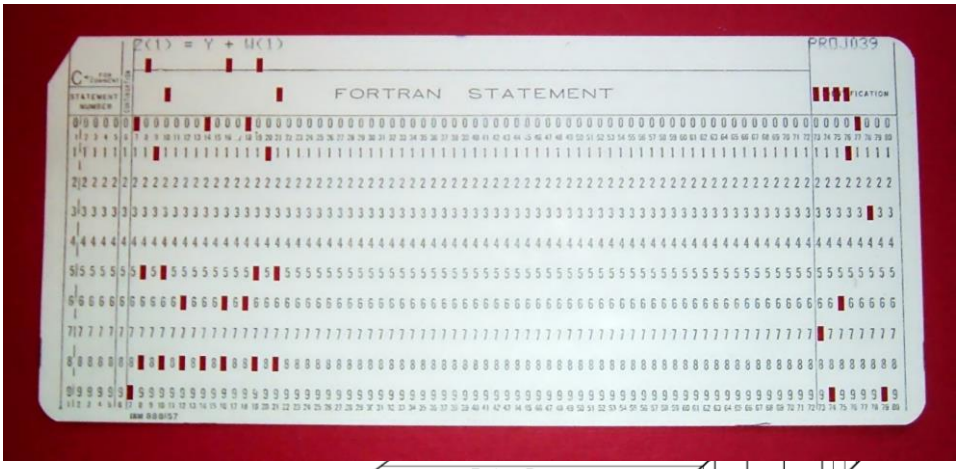
OS Types (History)

- Shared Computer System
  - User ←→ Operator ←→ System
  - Batches of jobs
  - Automatic Job Sequencing
  - Resident Monitor (First OS)
  - Control Cards (Interface for User & OS)
    - $JOB - First card of a job
    - $FTN - Execute the FORTRAN compiler
    - $ASM - Execute the assembler
    - $RUN - Execute the user program
    - $END - Final card of a job

OS Types (History)

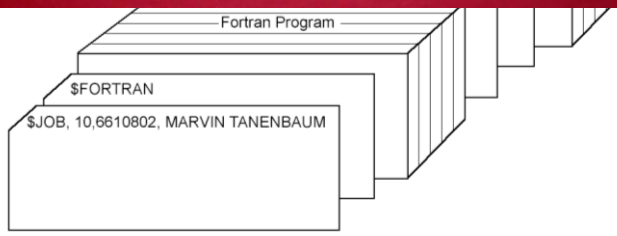
- Job Control language is the language that provides instructions to the monitor:
  - What compiler to use
  - What data to use





- Data
-

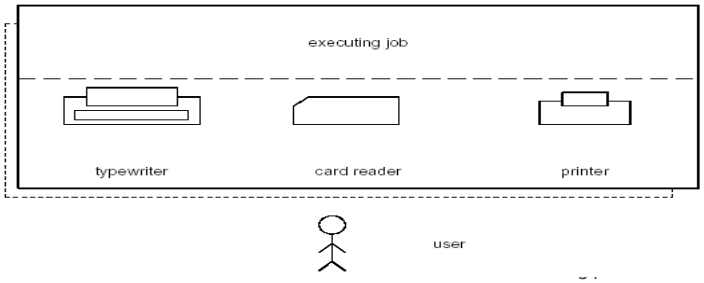
//END



OS Types (History)

Early Systems

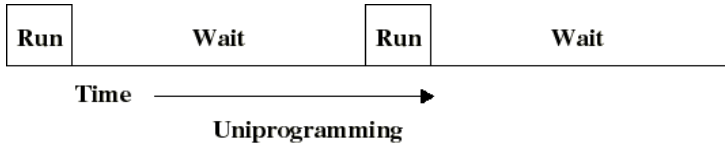
- Single user system
- Programmer/User as operator (Open Shop)
- Large machines run from console
- Paper Tape or Punched cards



OS Types (History)

Uni-programmed System

- I/O operations are exceedingly slow (compared to instruction execution)
- A program containing even a very small number of I/O operations, will spend most of its time waiting for them
- Poor CPU usage when only one program is present in memory



OS Types (History)

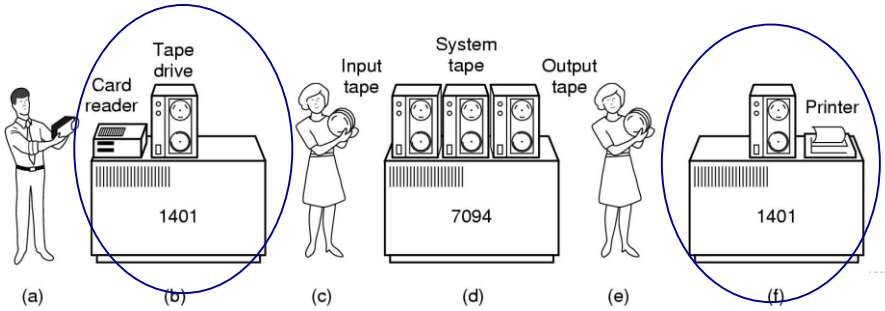
I/O and CPU speed Disparity

- IO Operations Overlapping
  - Buffering
    - Buffer (to overlap IO operation and computation)
  - On-line/Off-line Operation of IO Devices (Satellite Computing)

OS Types (History)

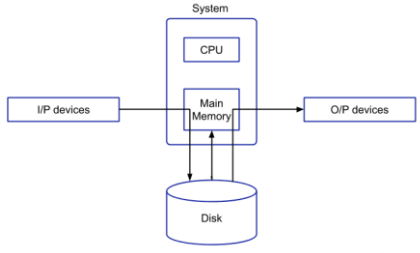
Satellite Computing

- On-line/Off-line Operation of IO Devices
  - Speed up computation by loading jobs into memory from tapes while card reading, and line printing is done off-line using smaller machines



OS Types (History)

- Disk (Direct Access Storage Device)
  - Spooling (Simultaneous Peripheral Operation On-Line)
    - Overlap I/O of one job with the computation of another job (using double buffering, DMA, etc.)



    - Remote Site Data Processing
    - Remote Printer/Card-reader

OS Types (History)

Multi-Programmed OS

- Several jobs are kept in main memory at the same time, and the CPU is multiplexed among them

OS Types (History)

Multi-Programmed OS (Required)

- Hardware support
  - I/O interrupts and DMA controllers in order to execute instructions while I/O device is busy
  - Timer interrupts for CPU to gain control
- Memory Management
  - several ready-to-run jobs must be kept in memory
- Memory Protection (data and programs)
- Software support from the OS:
  - For scheduling (which program is to be run next)
  - To manage resource contention

OS Types (History)

Multi-Programmed OS

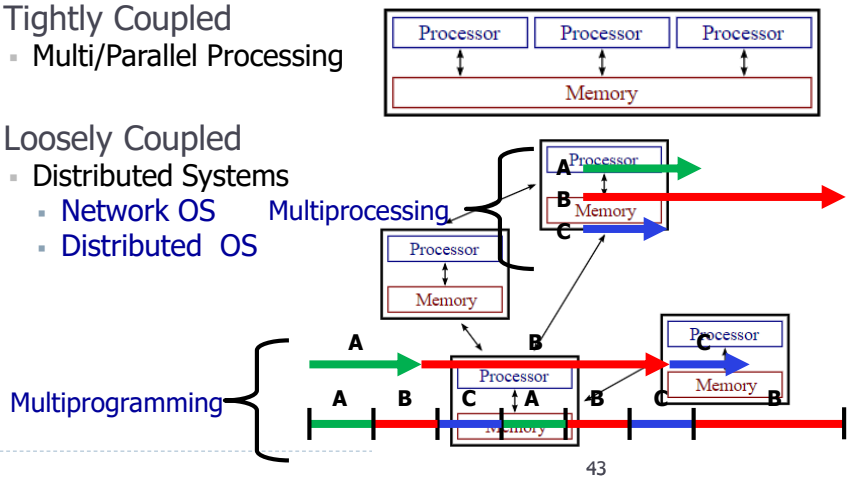
- Pure multi-programming
  - Non-Preemptive CPU Scheduling
- Multi-programming with Priorities
  - Preemptive CPU Scheduling

OS Types (History)

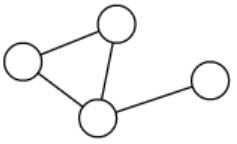
- Timesharing OS
  - Multiprogramming with Interactivity
    - Preemptive CPU Scheduling
  - Remote Procedure Call
- General/Specific Purpose OS

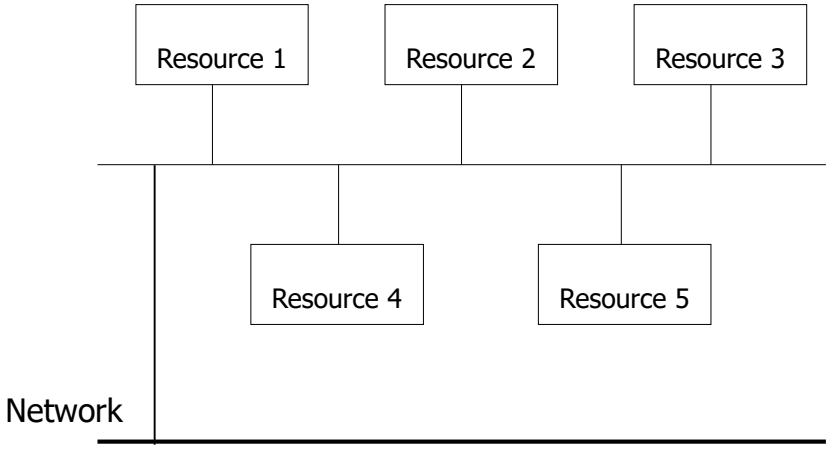
OS Types (History)

- Networking of Distributed Computing Resources
  - Tightly Coupled
    - Multi/Parallel Processing
  - Loosely Coupled
    - Distributed Systems
      - Network OS
      - Distributed OS

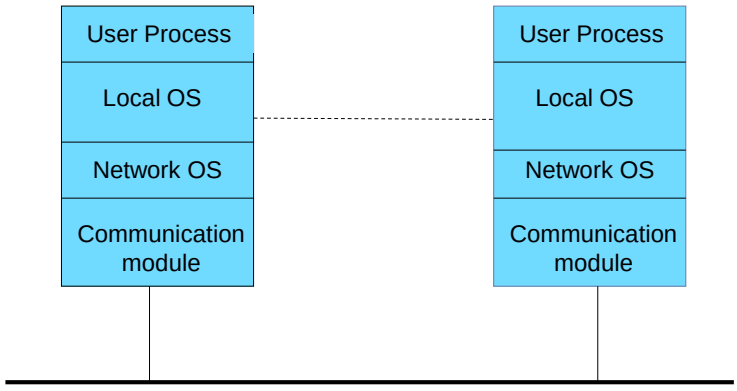


Distributed System





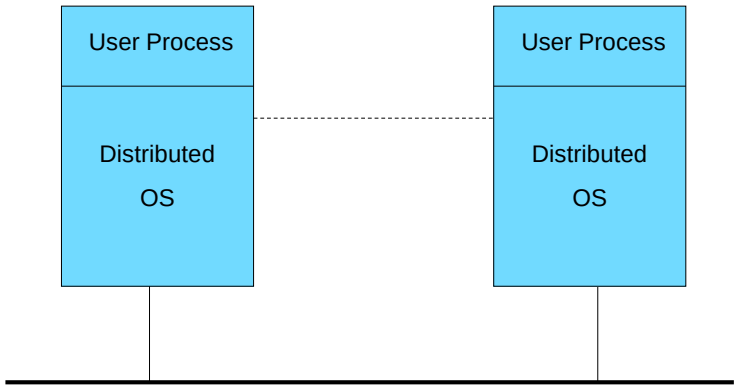
Network OS



Explicit File Transfer, Visible, Explicit Resource Usage

45

Distributed OS



Virtual computer, hides resources and location
Like centralized OS

46

OS Types (History)

- Embedded OS
  - Specific purpose operating system used in the computer system's embedded hardware
  - Designed to work on dedicated devices
- Real-Time OS
- Multiprocessor OS
- PC OS
- Server OS
- Mobile-Device OS
- …

Process Model

- Program(Static)
  - A set of instructions
- Process (Dynamic)
  - A program in action

- A process includes:
  - Program Counter (PC)
  - Stack
  - Data Section

Process Model

Process States

- As a process executes, it changes state

New
: Process is being created

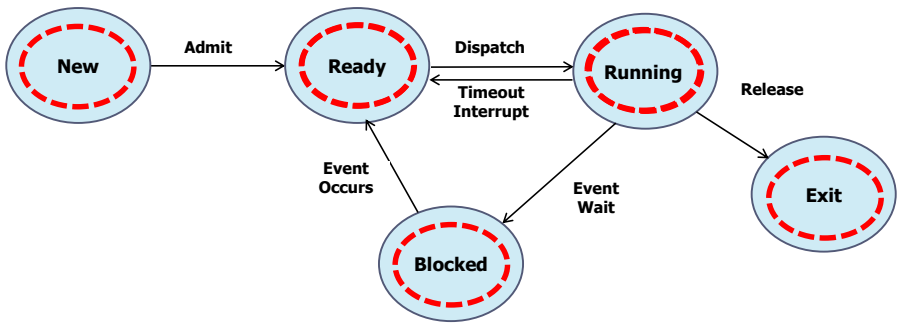
Ready
: Process is waiting to run

Running
: Instructions are being executed

Waiting
: Process waiting for some event to occur

Terminated
: Process has finished execution

Process Model



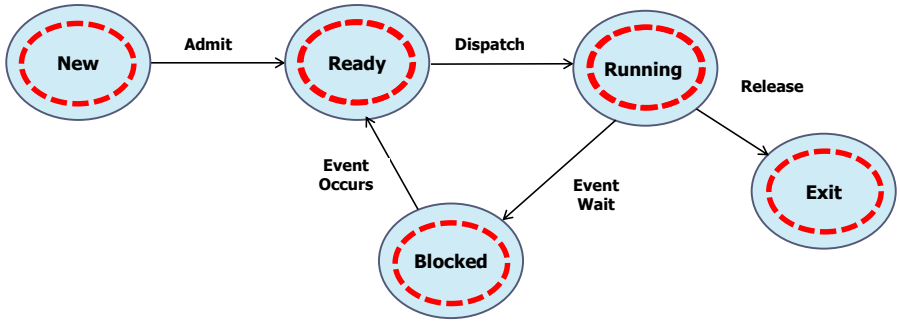
Process Model

- Process: a program in action
- Process execution consists of a cycle of CPU execution (CPU burst) and I/O (burst) wait

- Compute Bound

- IO Bound

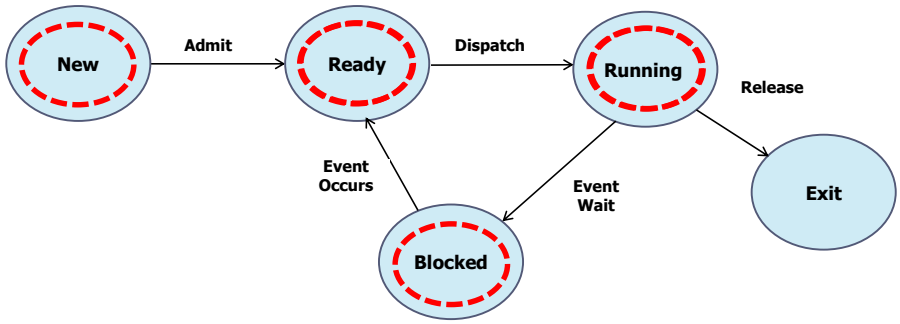
Batch OS



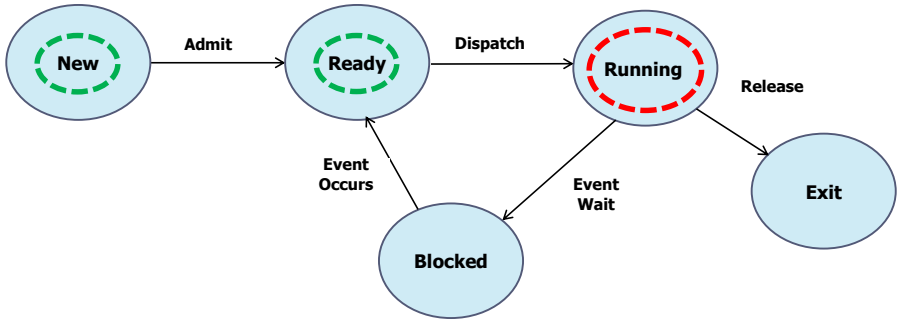
Pure Multiprogramming OS

- Non-Preemptive CPU Scheduling
  - Process in Running stare voluntarily leave
    - Running to Exit state
    - Running to Blocked state
  - Context Switch
    - Running to Exit/Blocked
    - Ready to Running

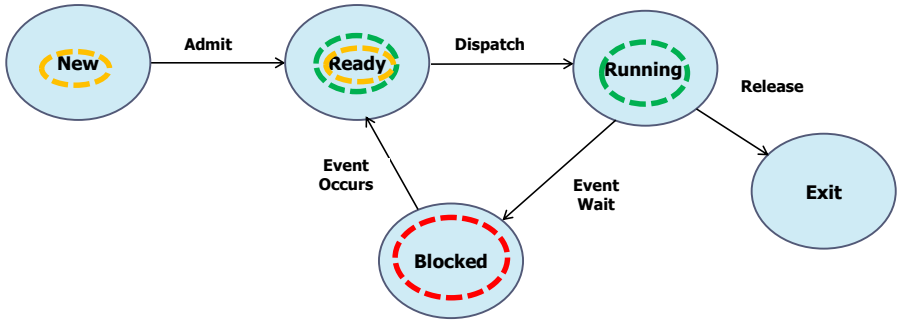
Pure Multiprogramming OS



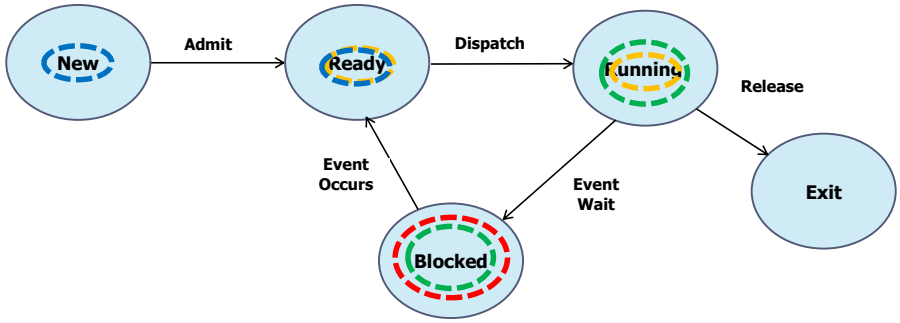
Pure Multiprogramming OS



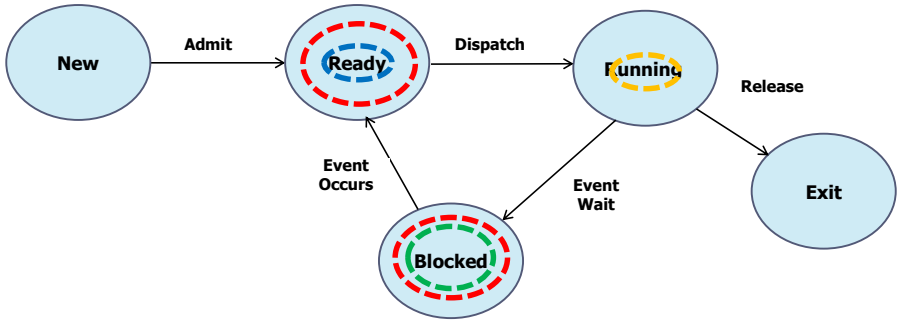
Pure Multiprogramming OS



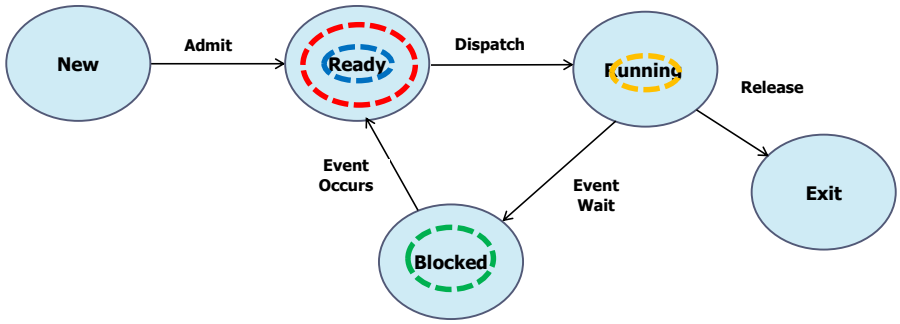
Pure Multiprogramming OS



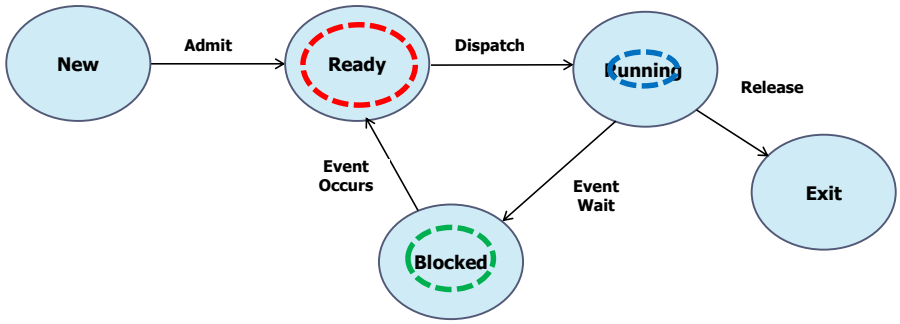
Pure Multiprogramming OS



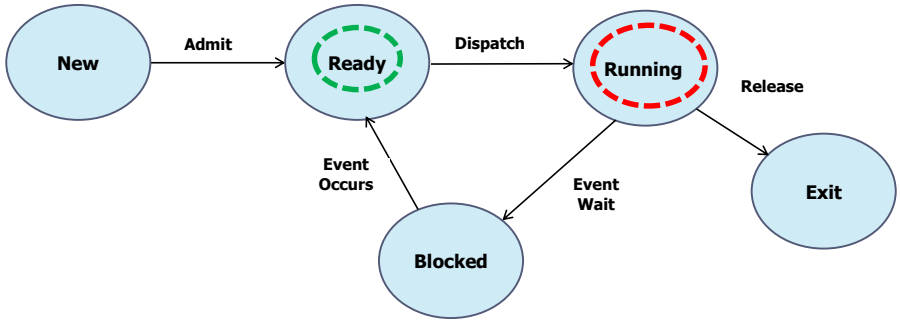
Pure Multiprogramming OS



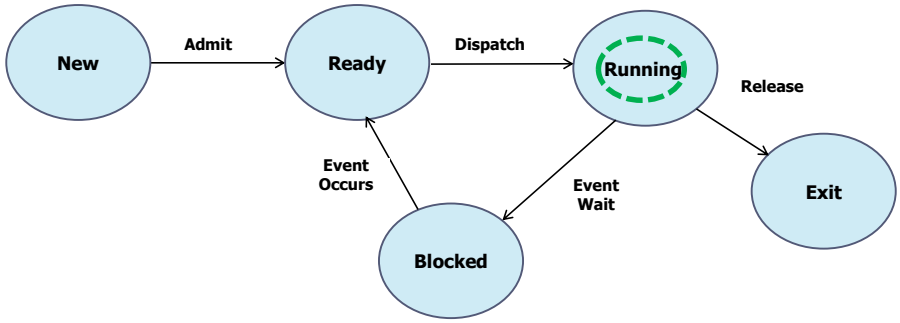
Pure Multiprogramming OS



Pure Multiprogramming OS



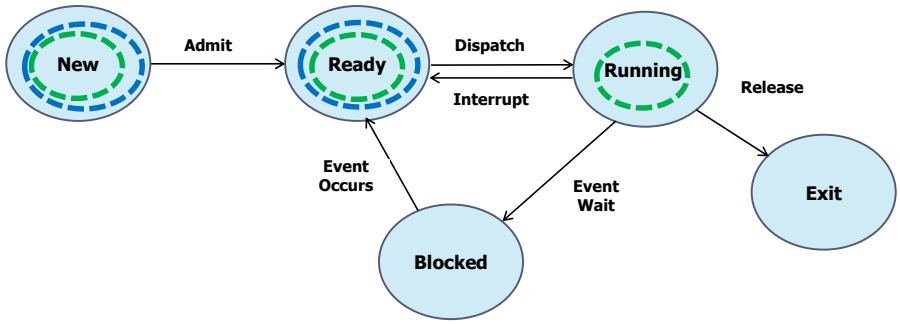
Pure Multiprogramming OS



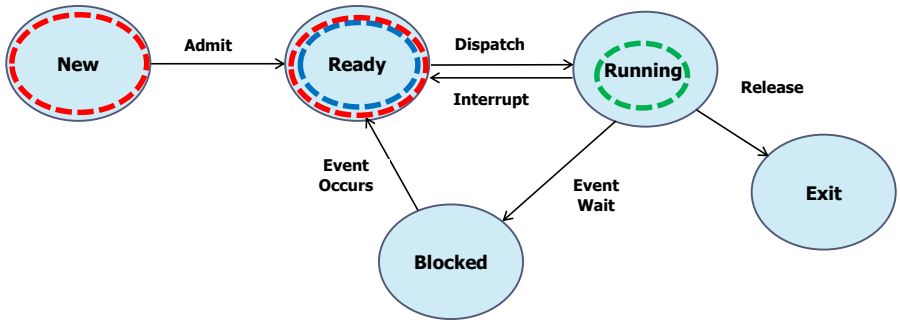
Multiprogramming with Priorities

- Preemptive CPU Scheduling
  - Low priority process in Running state
  - High priority process enters Ready state
  - Context Switch
    - Running (LP) to Ready
    - Ready (HP) to Running

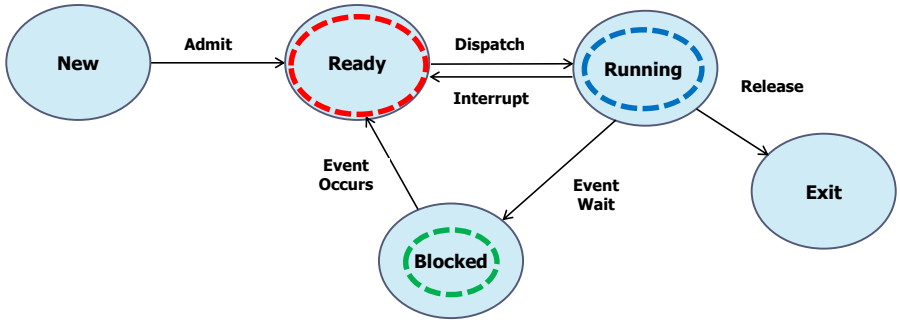
Multiprogramming with Priorities



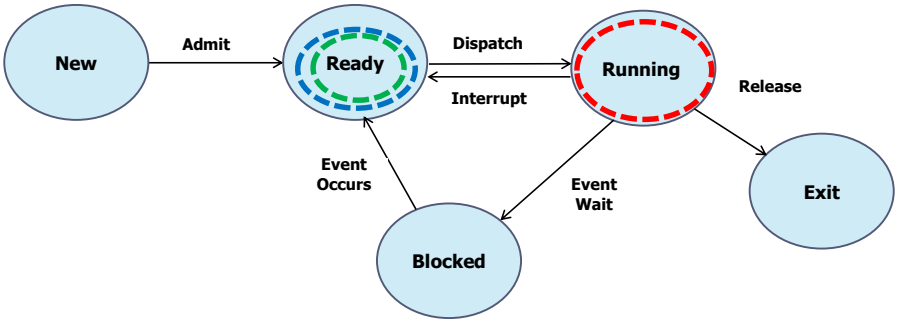
Multiprogramming with Priorities



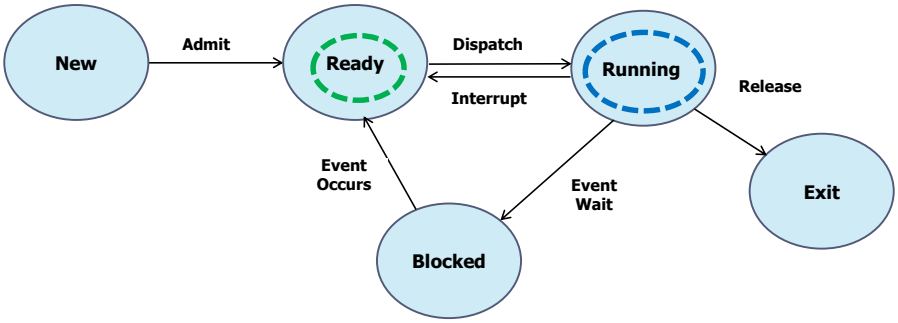
Multiprogramming with Priorities



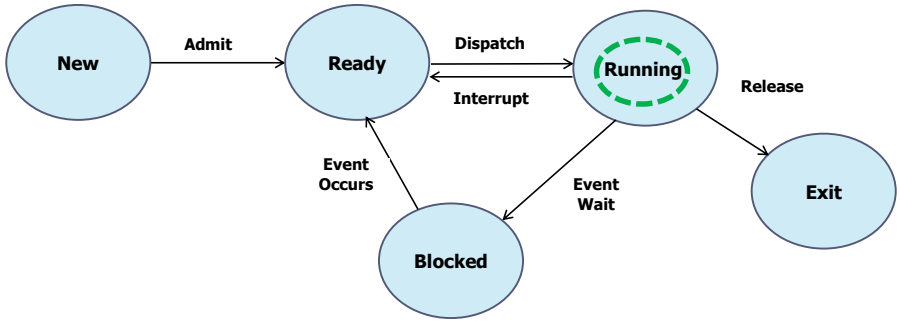
Multiprogramming with Priorities



Multiprogramming with Priorities



Multiprogramming with Priorities



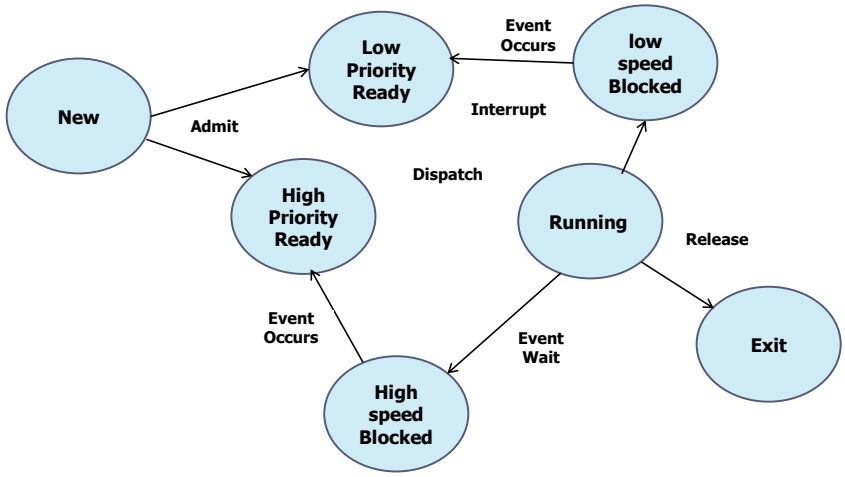
Multiprogramming with Priorities

- In a two types of priorities (Low and High) OS, there is mixed types (IO bound and Compute bound) processes
- To which type of processes, you will assign high priority ?

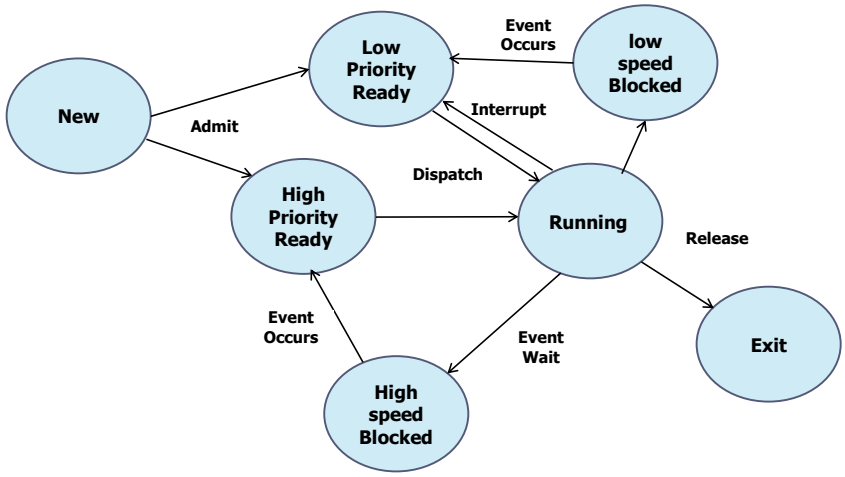
Multiprogramming with Priorities

- Multiprogramming with priorities OS, priorities are assigned statically (at process creation time)
- Two types of priorities (Low and High)
- IO devices with Two speeds (Low and High)
- Low/High priority processes use Low/High speed IO devices respectively
- Different lists for Ready and Blocked states of different priorities and different speed IO devices
- Process Model

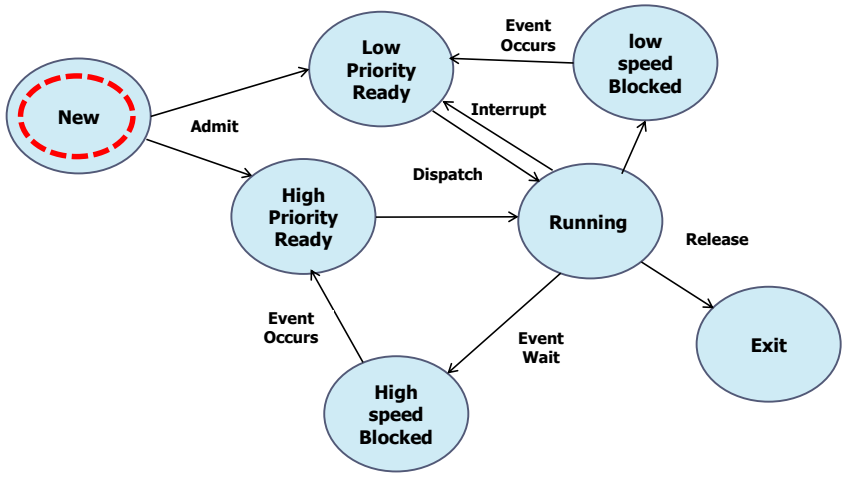
Multiprogramming with Priorities



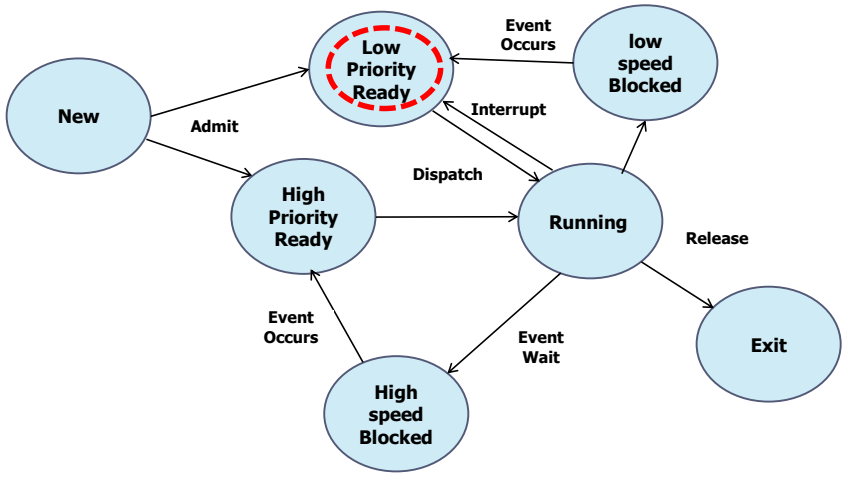
Multiprogramming with Priorities



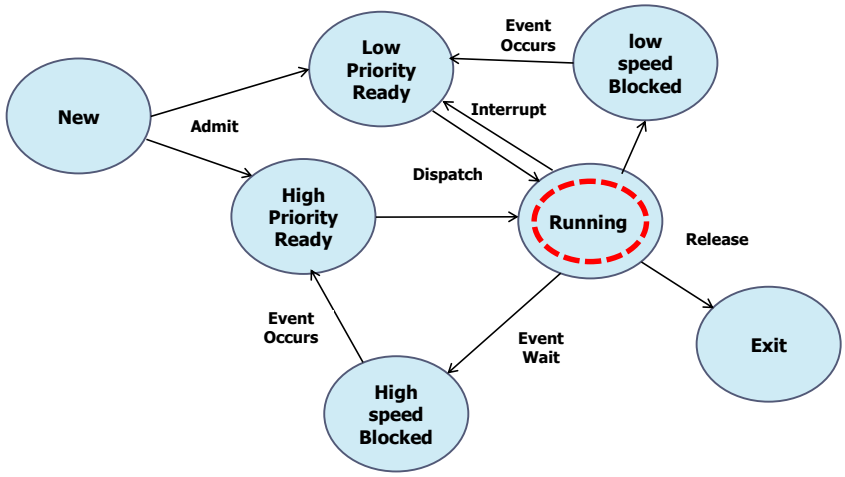
Multiprogramming with Priorities



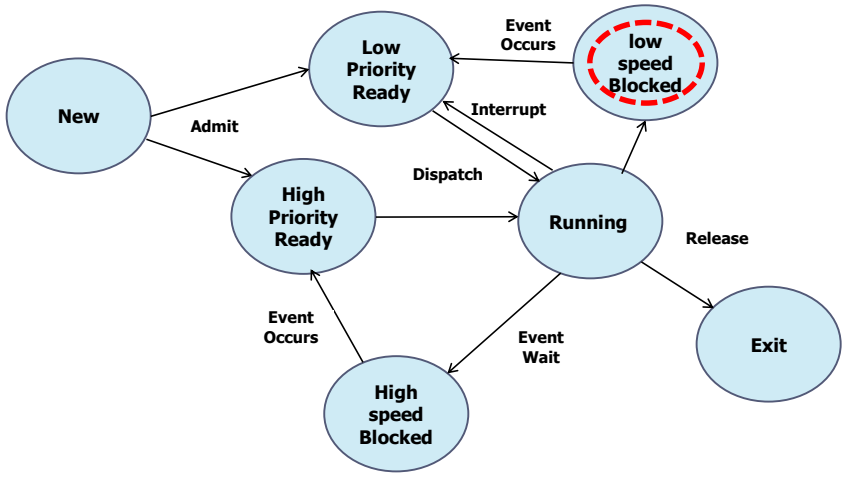
Multiprogramming with Priorities



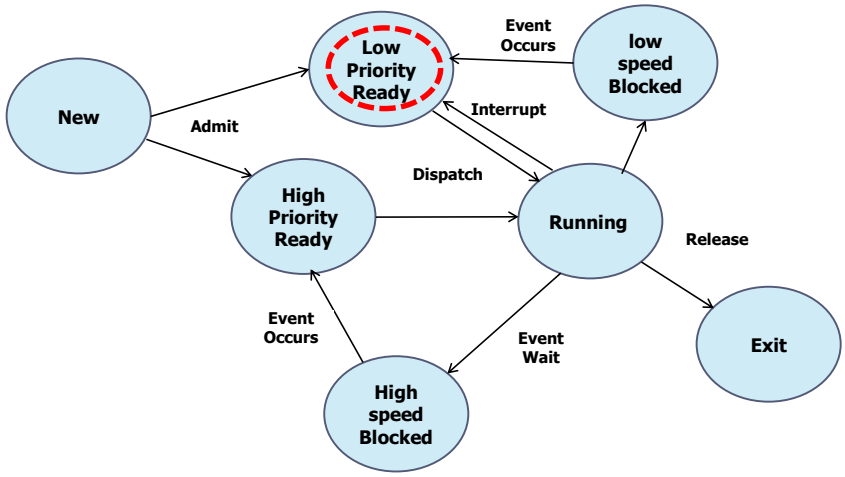
Multiprogramming with Priorities



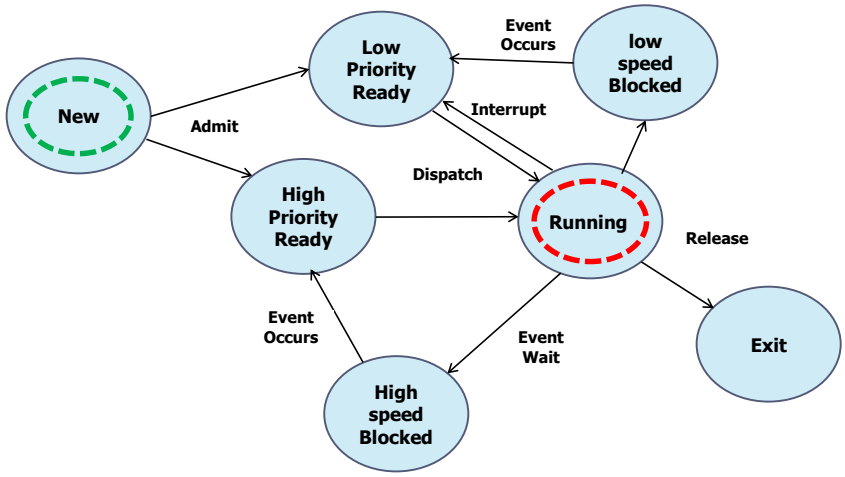
Multiprogramming with Priorities



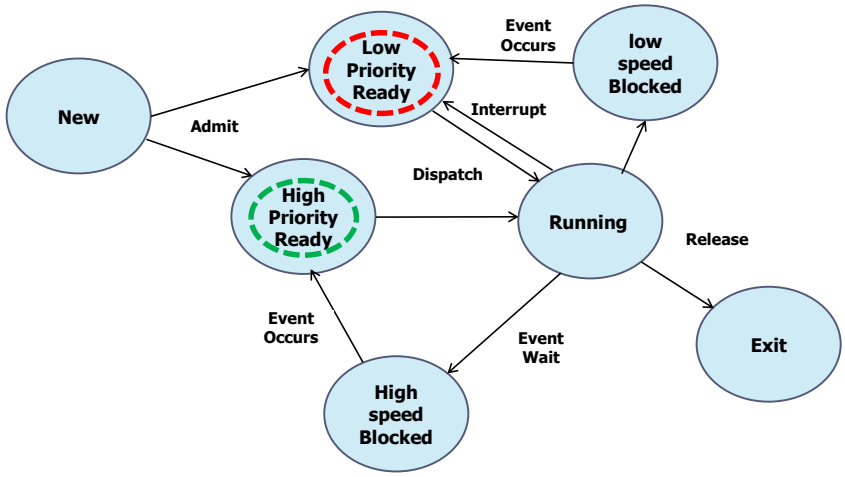
Multiprogramming with Priorities



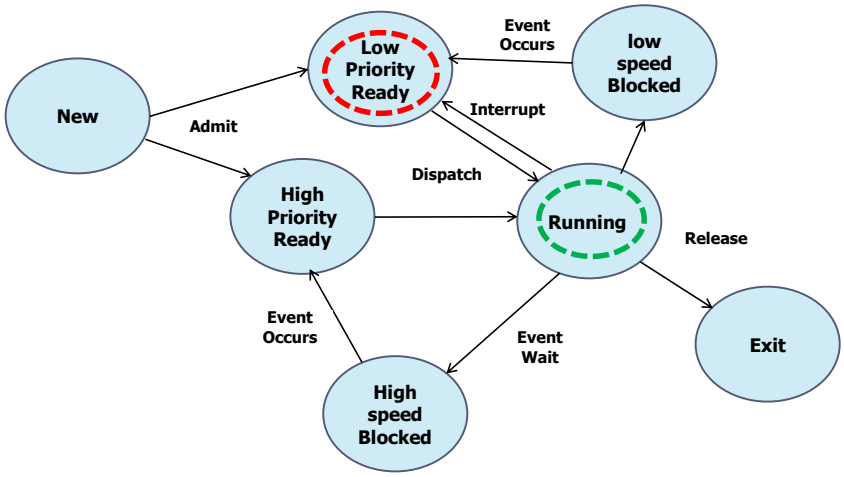
Multiprogramming with Priorities



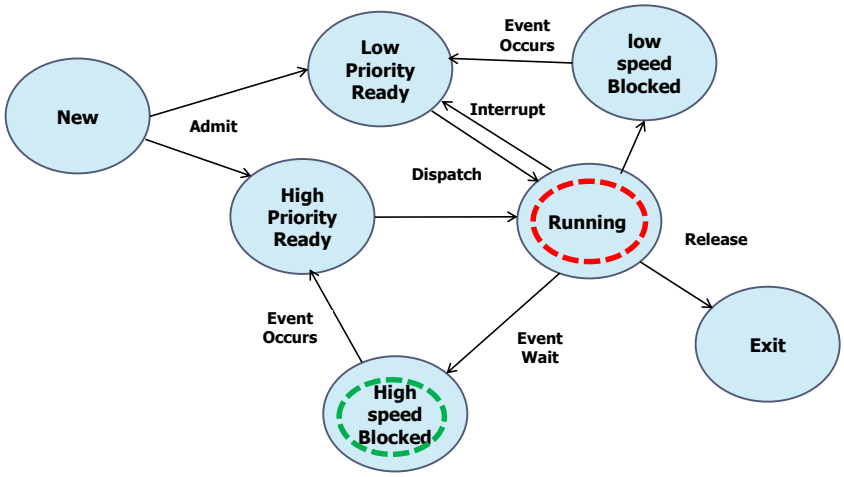
Multiprogramming with Priorities



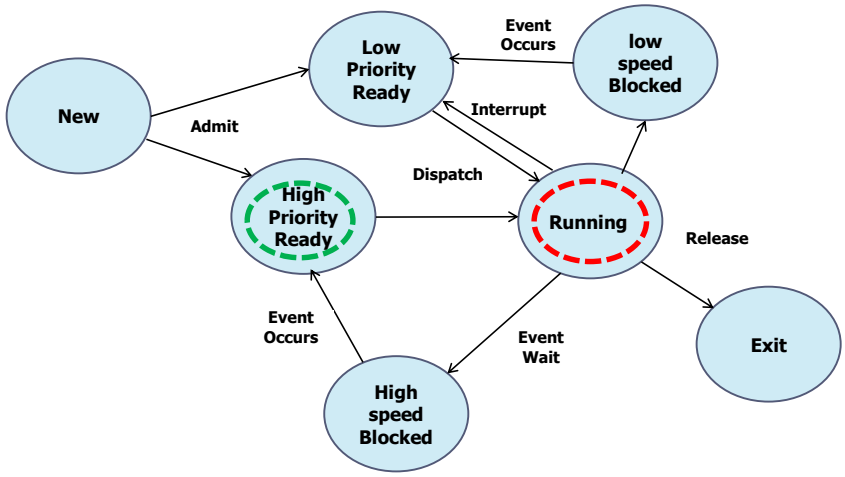
Multiprogramming with Priorities



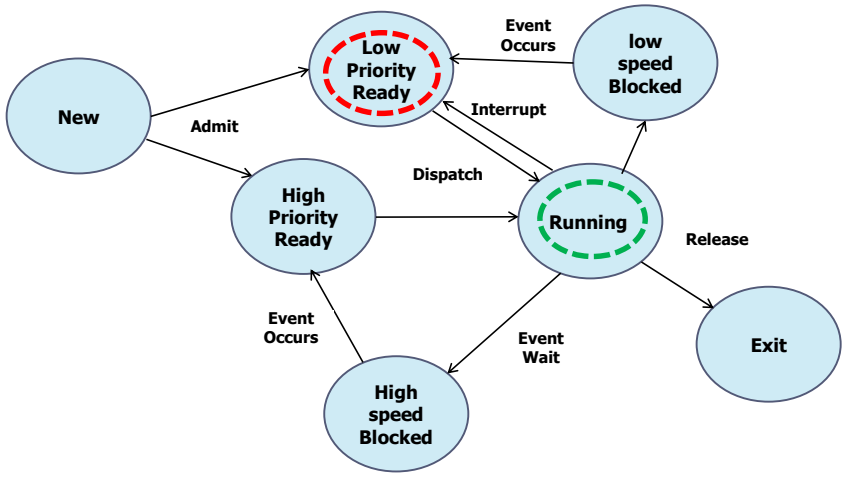
Multiprogramming with Priorities



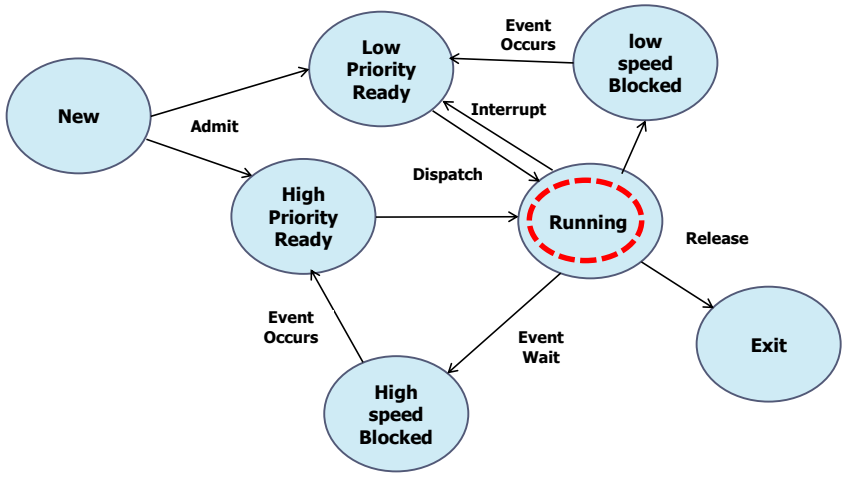
Multiprogramming with Priorities



Multiprogramming with Priorities



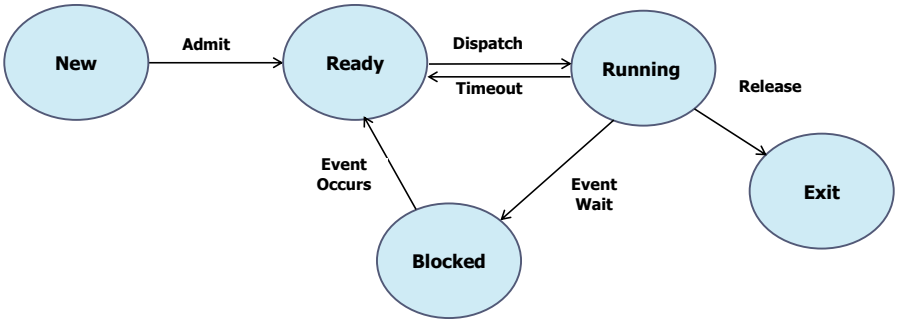
Multiprogramming with Priorities



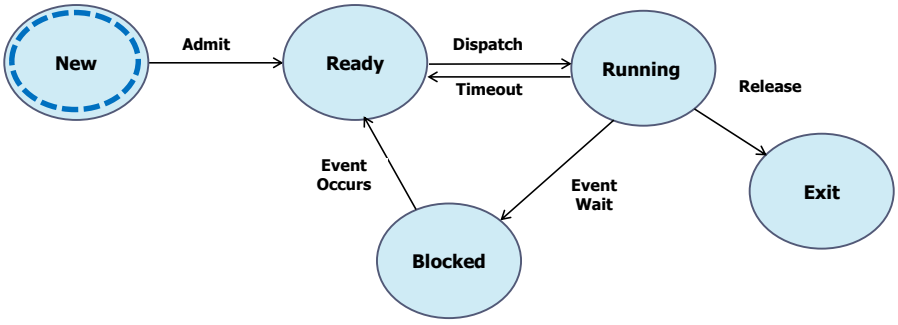
Time-sharing OS

- Preemptive CPU Scheduling
- FIFO Ready List
- Context Switch at time-out (timeslot completion)
  - Running to Ready (at rear pointer of list)
  - Ready (from front pointer) to Running

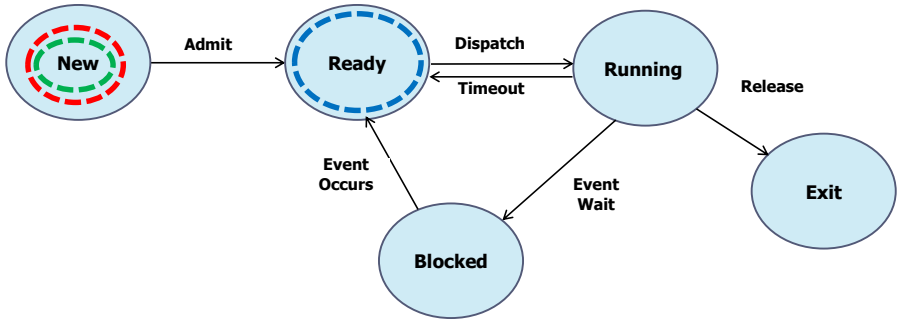
Time-sharing OS



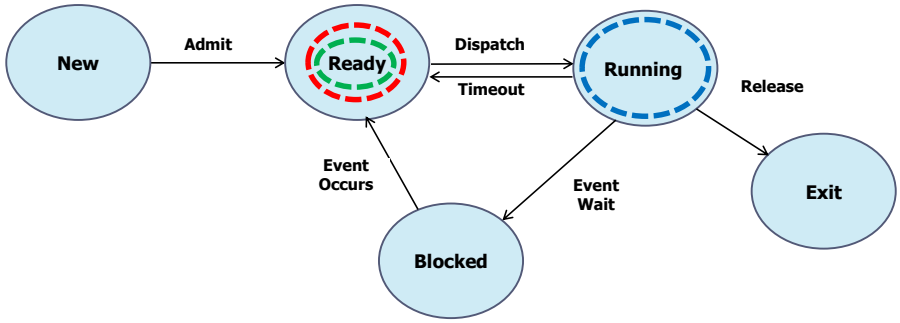
Time-sharing OS



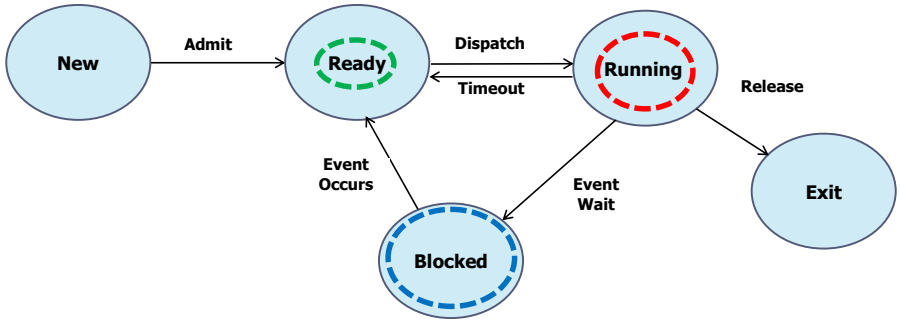
Time-sharing OS



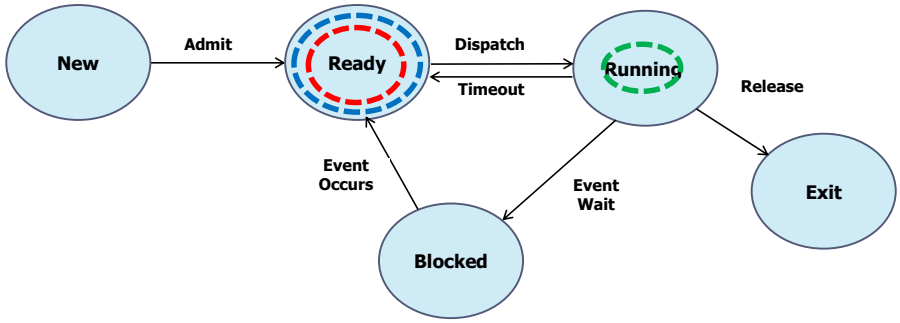
Time-sharing OS



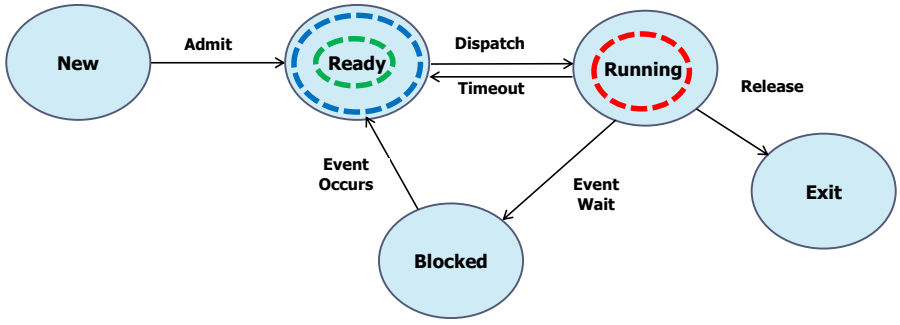
Time-sharing OS



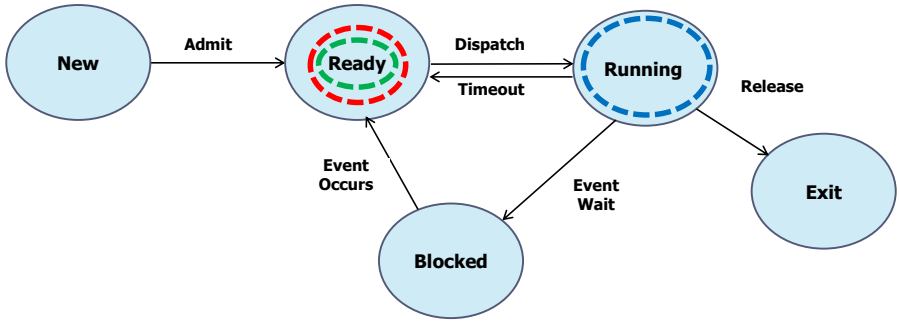
Time-sharing OS



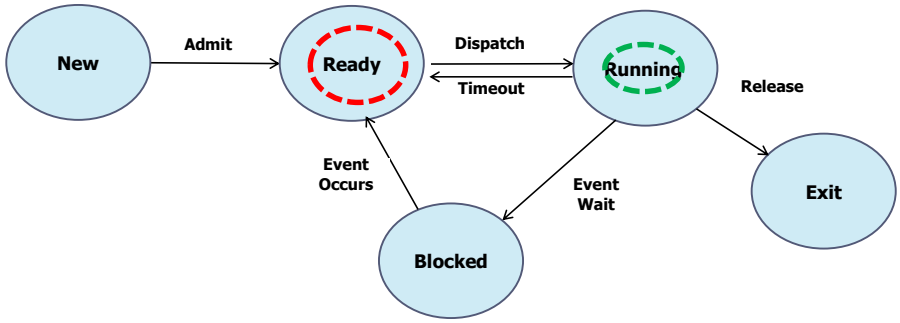
Time-sharing OS



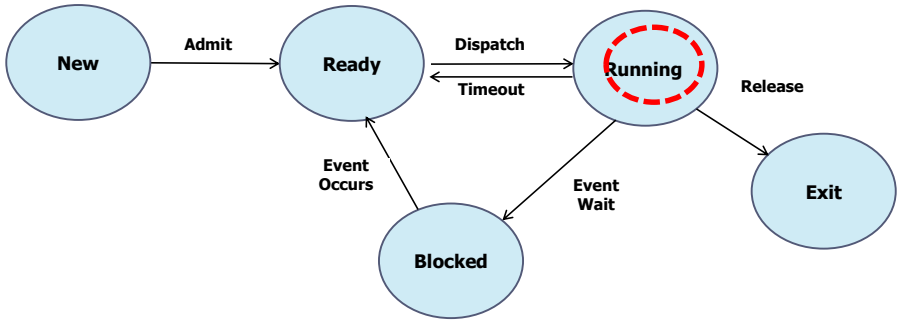
Time-sharing OS



Time-sharing OS



Time-sharing OS



General Purpose OS