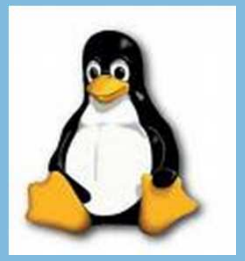
IRA A. FULTON SCHOOLS OF engineering

school of computing, informatics, & decision systems engineering

Introduction to Engineering Using Robotics Experiments

Lecture L10 Operating Systems

Dr. Yinong Chen



Functions of Operating Systems

Operating System is the interface between human and computer

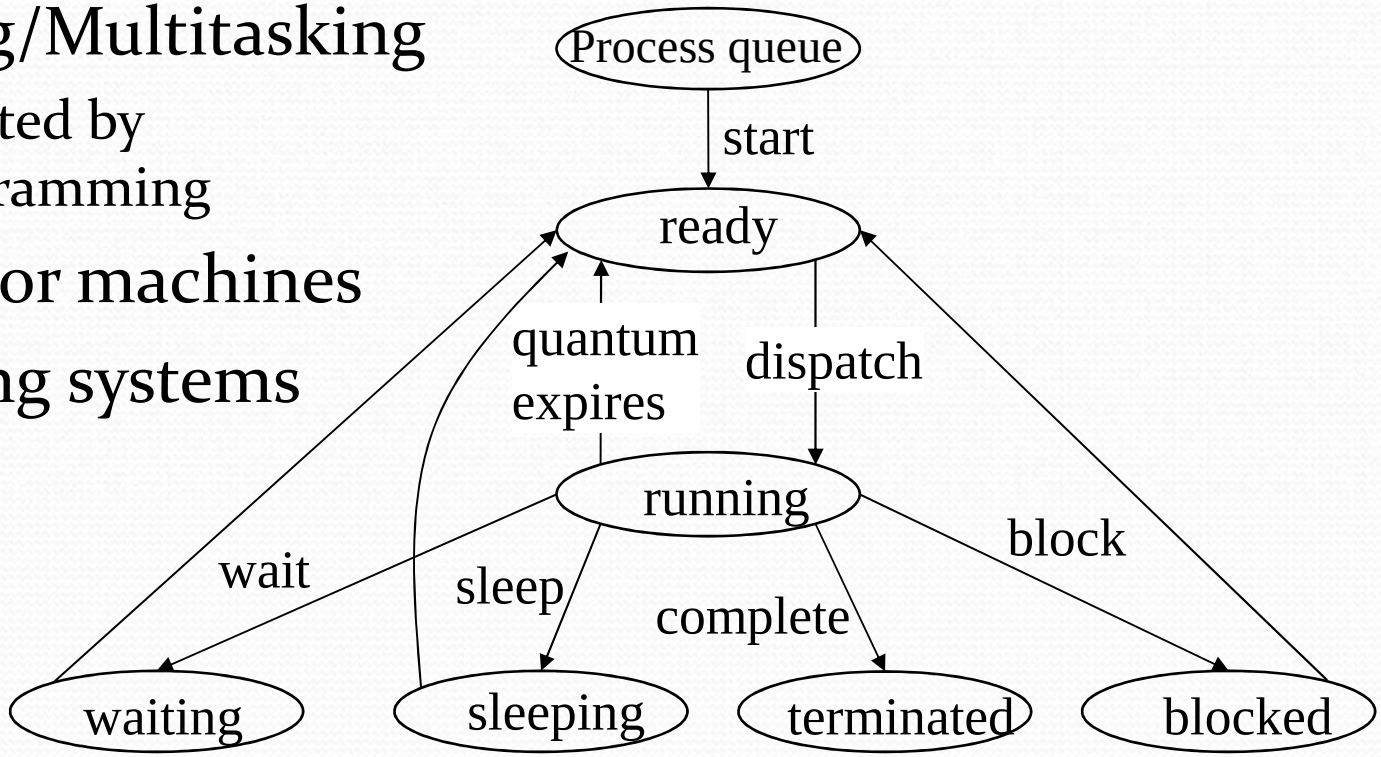
- Oversee operations of a computer
- Store and retrieve files
- Schedule programs for execution
- Coordinate the execution of programs

What is the interface between hardware and software?

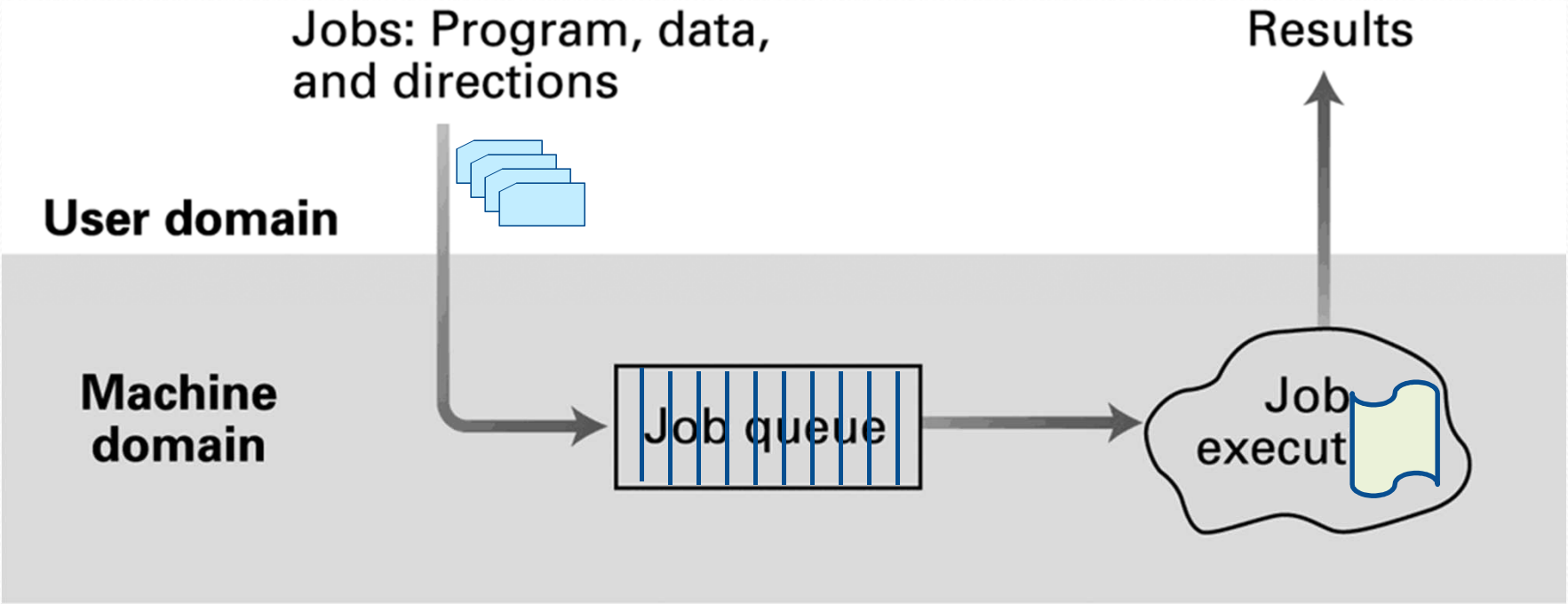
An OS course will discuss OS design and implementation in detail

Evolution of Shared Computing

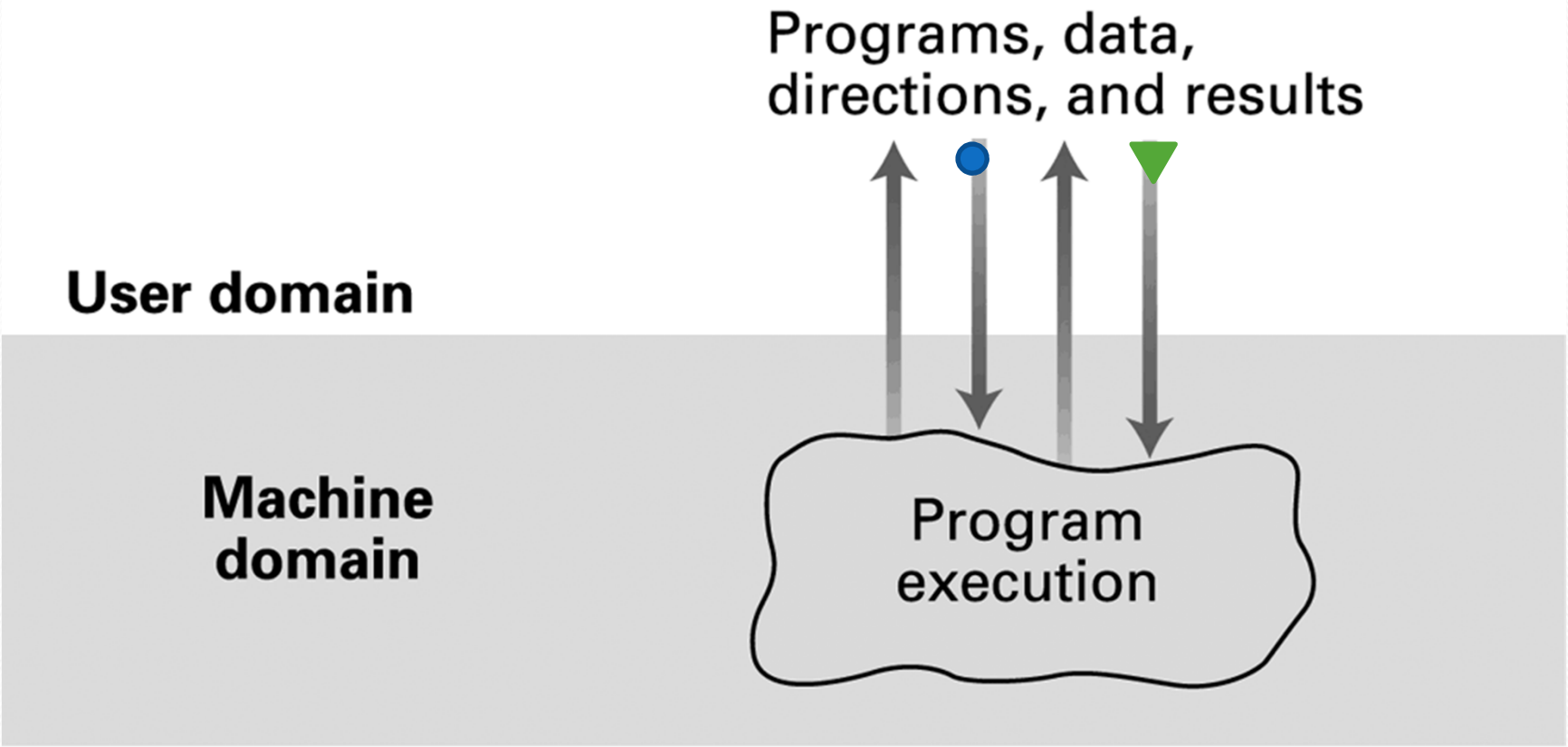
- Batch processing
- Interactive processing
  - Requires real-time processing
- Time-sharing/Multitasking
  - Implemented by multiprogramming
- Multiprocessor machines
- Web operating systems



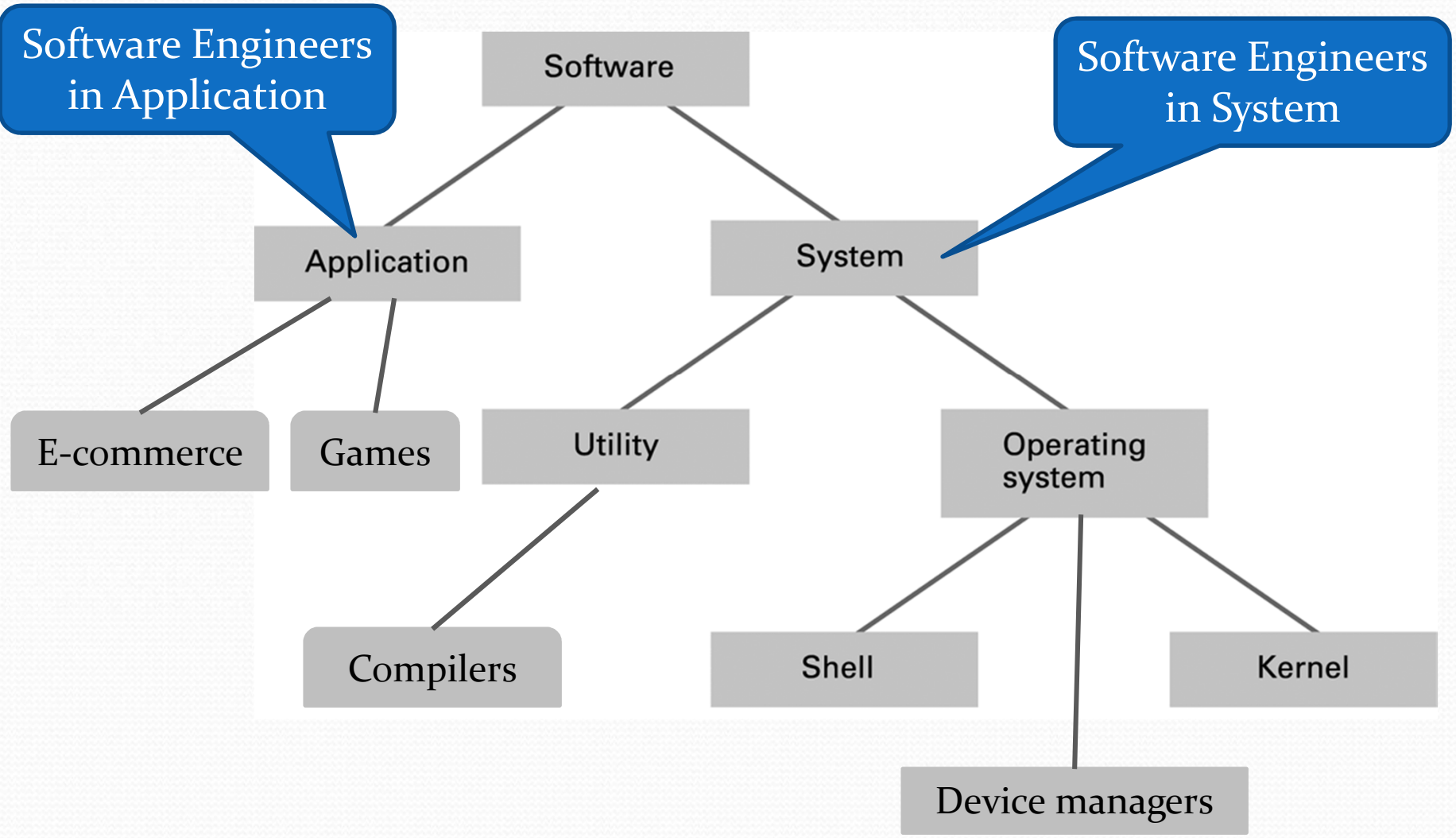
Batch processing



Interactive processing



Software Classification



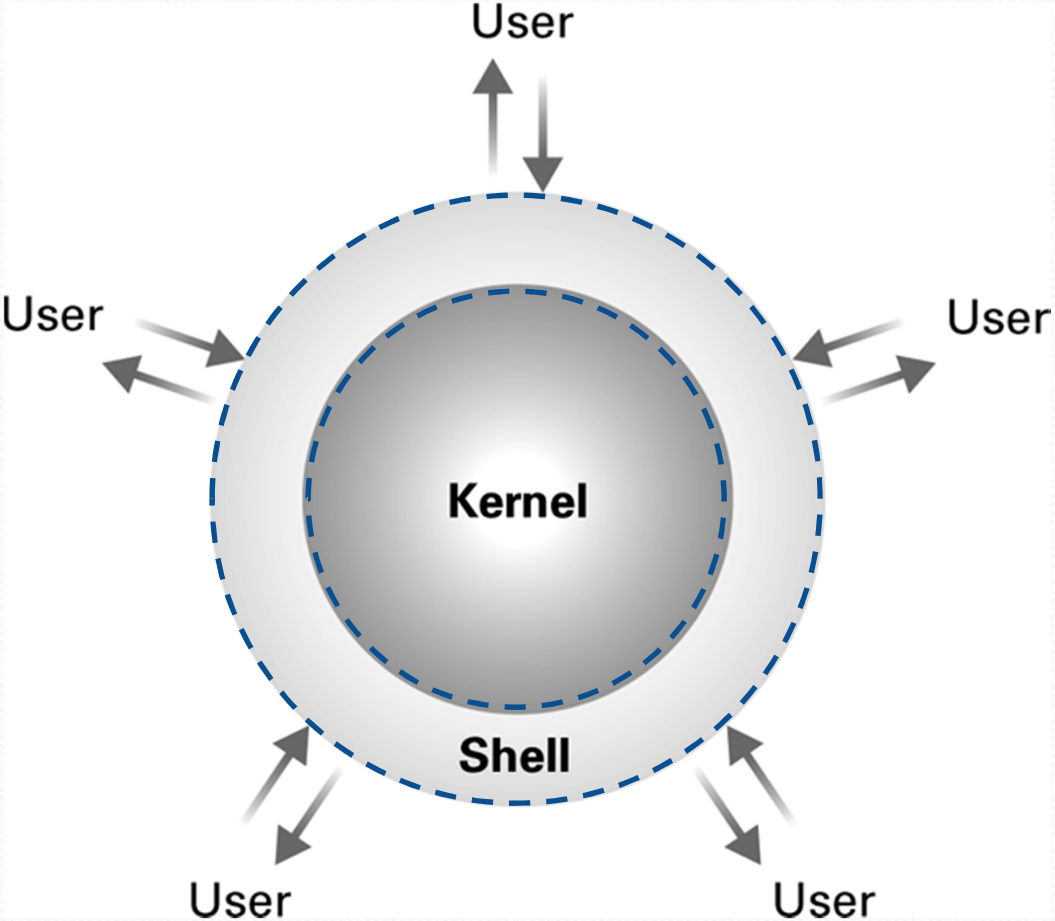
Operating System Components

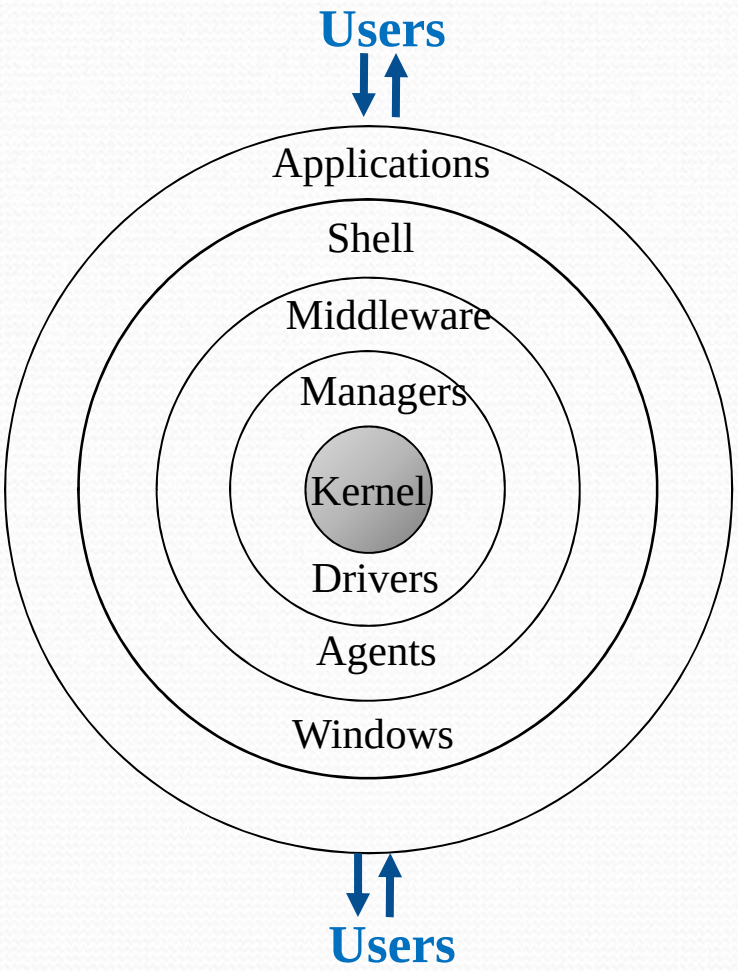
- **Shell:** Communicates with users
  - Text based
  - Graphical user interface (GUI)
- **Kernel:** Performs basic required functions
  - File manager
  - Memory manager
  - Scheduler and dispatcher
- **Device managers**
  - Drivers that can be installed and uninstalled by users

Not replaceable

Replaceable

The shell as an interface between users and the operating system





File Manager

- **Directory** (or **Folder**): A user-created bundle of files and other directories (subdirectories)
- **Directory Path:** A sequence of directories within directories
- Example: **DOS** (Disk Operating System) is basically a file manager, as the shell and other kernel mangers are very simple. DOS runs one program at a time, and thus memory manager is almost nothing:

640K ought to be enough for anybody.	***Bill Gates, Microsoft***

Memory Manager

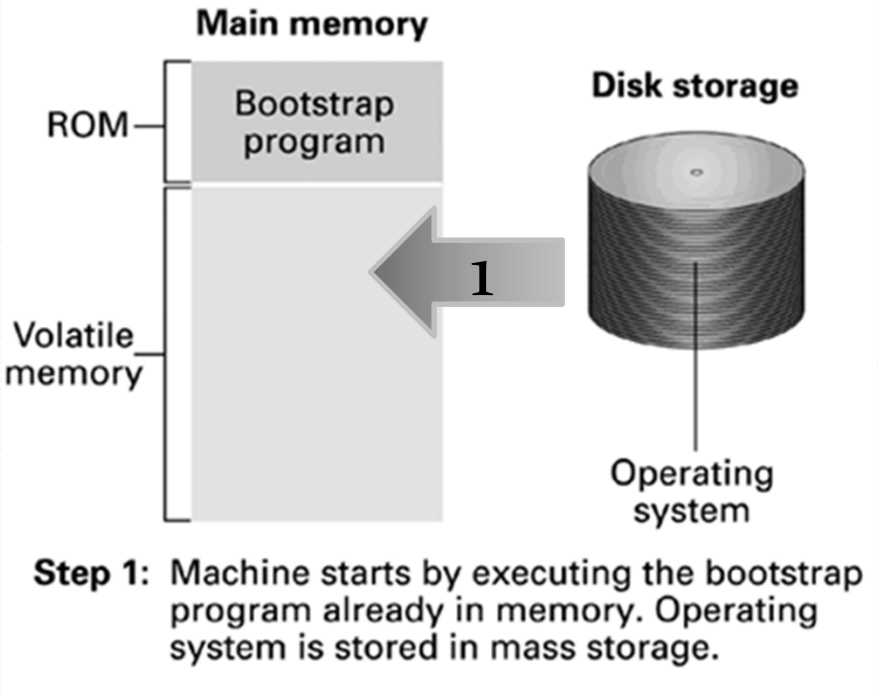
- Allocates space in main memory
- May create the illusion that the machine has more memory than it actually does (**virtual memory**) by playing a "shell game", in which blocks of data (**pages**) are shifted back and forth between main memory and mass storage (disk)
- Memory manger is complex in multitasking and multi-processor system
  - Memory sharing
  - Faster memory → Cache → Level 1 and Level 2

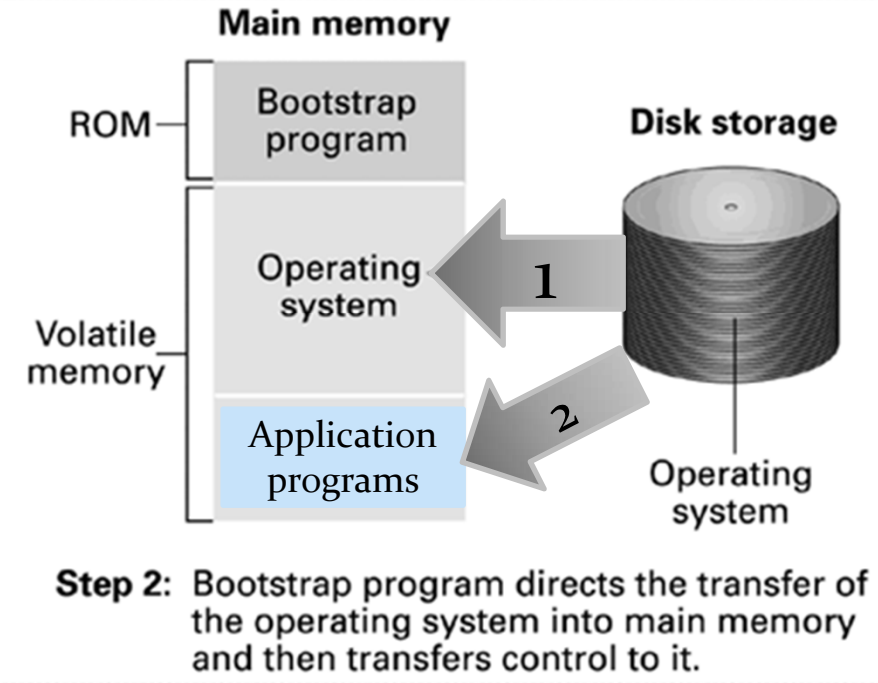
Computer Organization and Architecture courses will discuss the topics in detail

Getting it Started (Bootstrapping)

- **Bootstrap:** Program in ROM (example of firmware)
  - Run by the CPU when power is turned on
  - Transfers operating system from mass storage (disk) to main memory (RAM)
  - Executes jump to operating system

The booting process





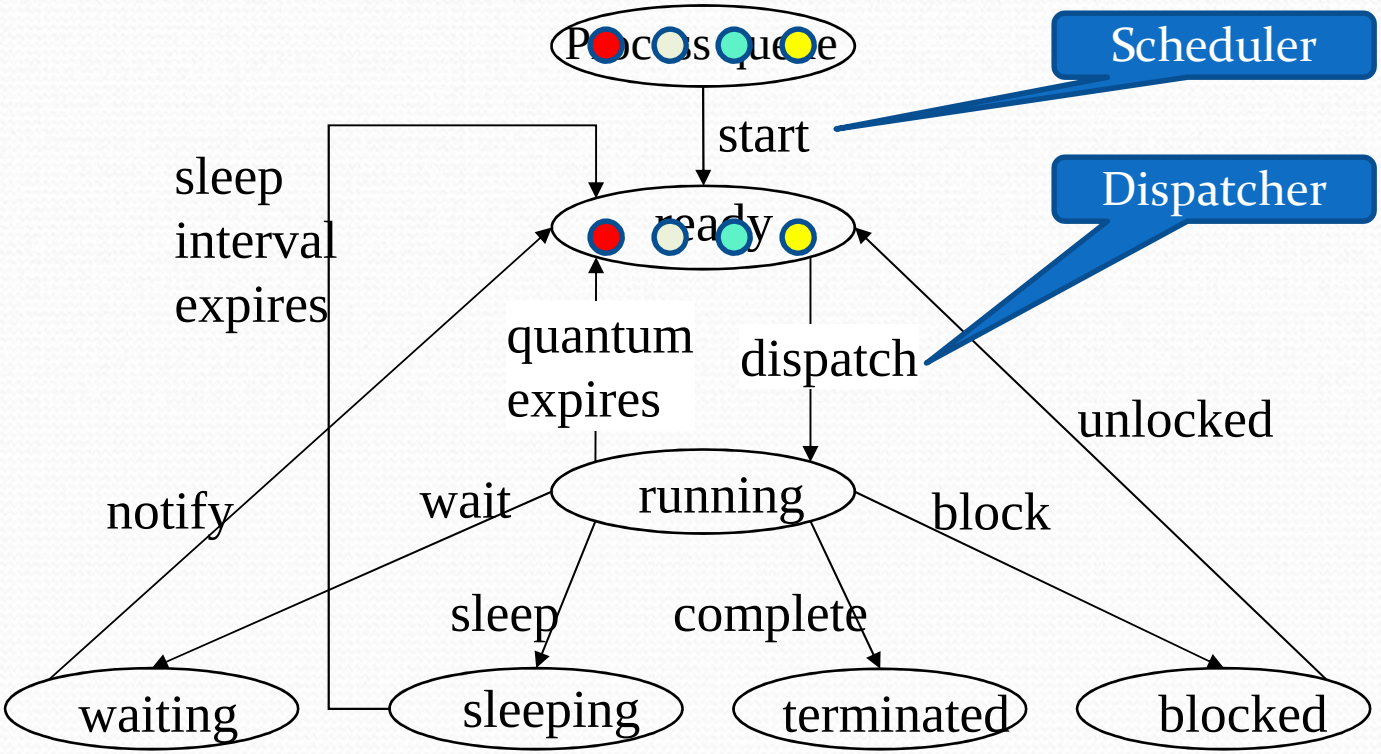
Step 3: Load application programs

Programs vs. Processes

- **A program** is a sequence of instructions
- **Process:** a program in execution, with intermediate results (process state)
- **Process State:** Current status of the process
  - Program counter (what instruction is to be executed next?)
  - Register values (temporary space for values being processed)
  - Related portion of main memory contents

Process Administration in Multitasking OS

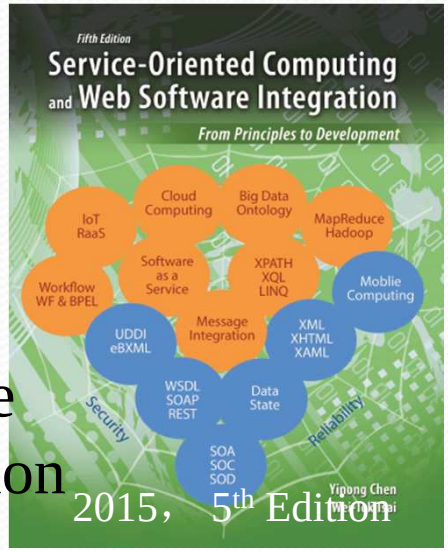
- **Scheduler:** Adds new processes to the process table and removes completed processes from the process table
- **Dispatcher:** Controls the allocation of time slices to the processes in the process table (ready state)
  - The end of a time slice is signaled by an interrupt.



Web Operating System

- Started around 1999 at UC Berkeley
- Started to challenge desktop OS, when Web-based computing started to compete with desktop-based computing in the recent years
- Web-based computing concepts;
  - Service-oriented computing, e-commerce applications
  - Web 2.0: Web as computing platform
  - Web 3.0: Semantic Web
  - Cloud computing
    - Software as a Service (SaaS)
    - Infrastructure as a Service (IaaS)
    - Platform as a Service (PaaS)
  - Big Data

SOC & Software Integration

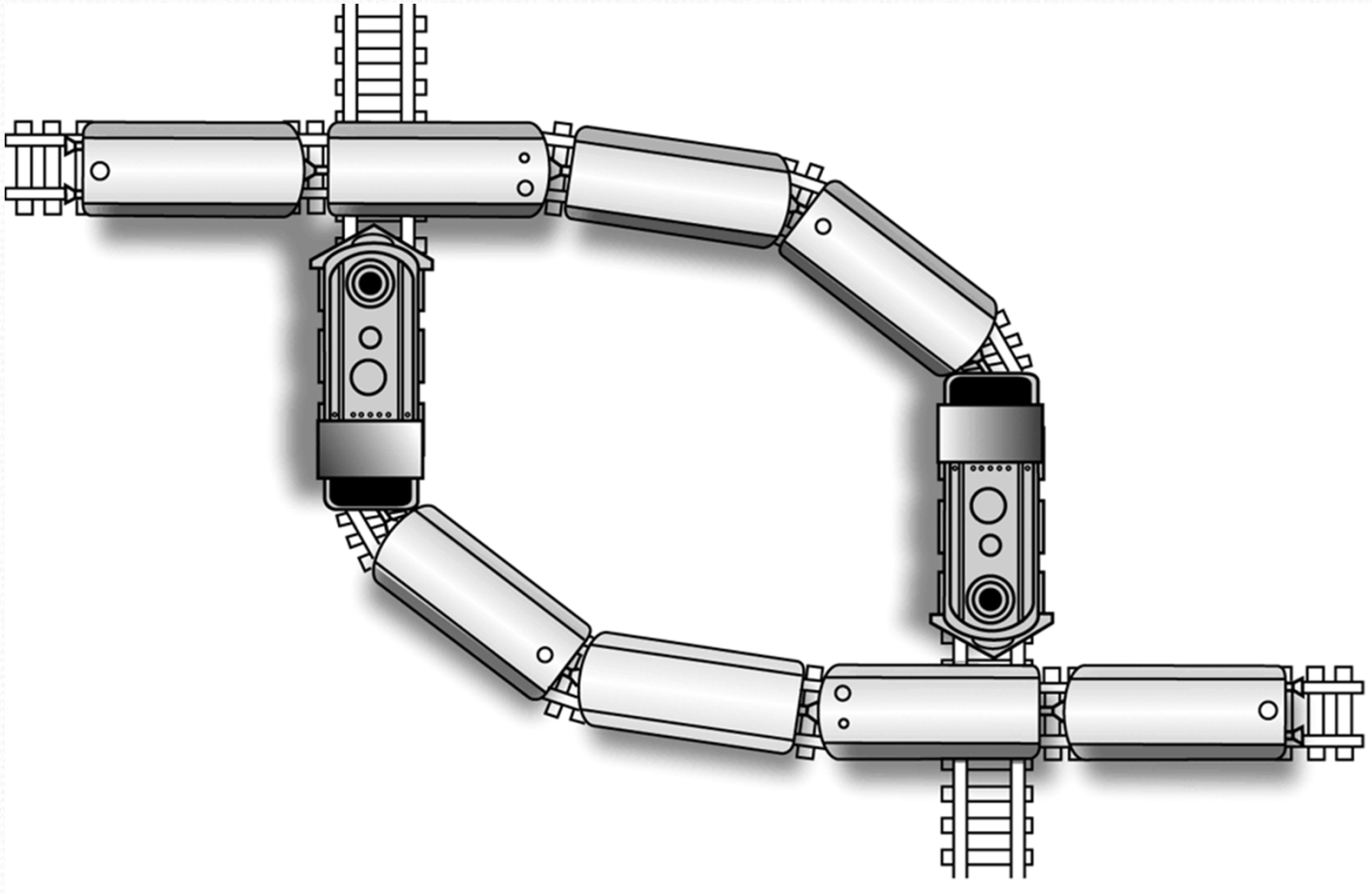


2015， 5th Edition

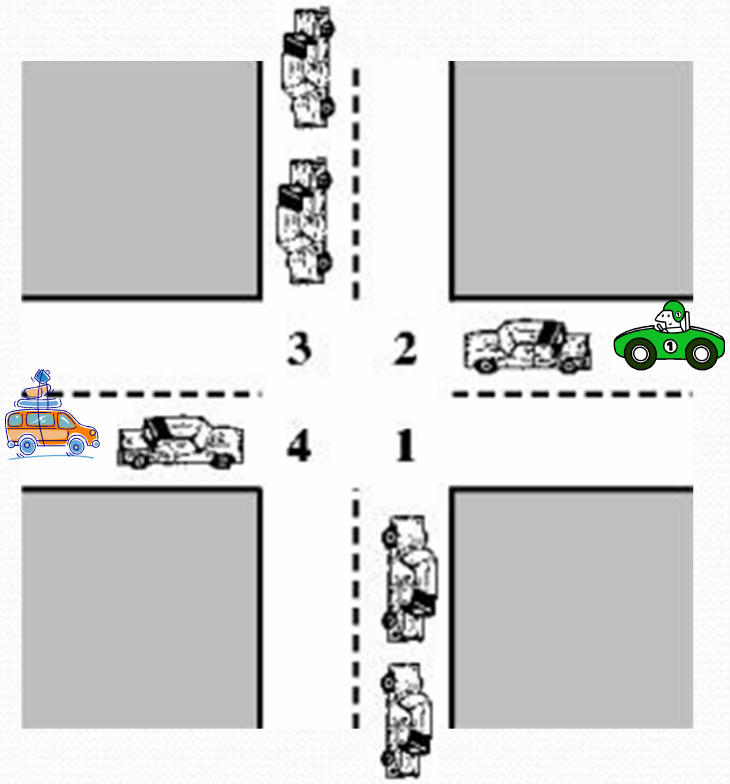
A Main Issue in OS Design: Deadlock

- A **deadlock** is a situation wherein two or more competing actions are waiting for the other to finish, and thus neither ever does.
- A typical situation is, two or more actions need more than one resource to proceed, and each holds one resource while waiting for others to release the resources.

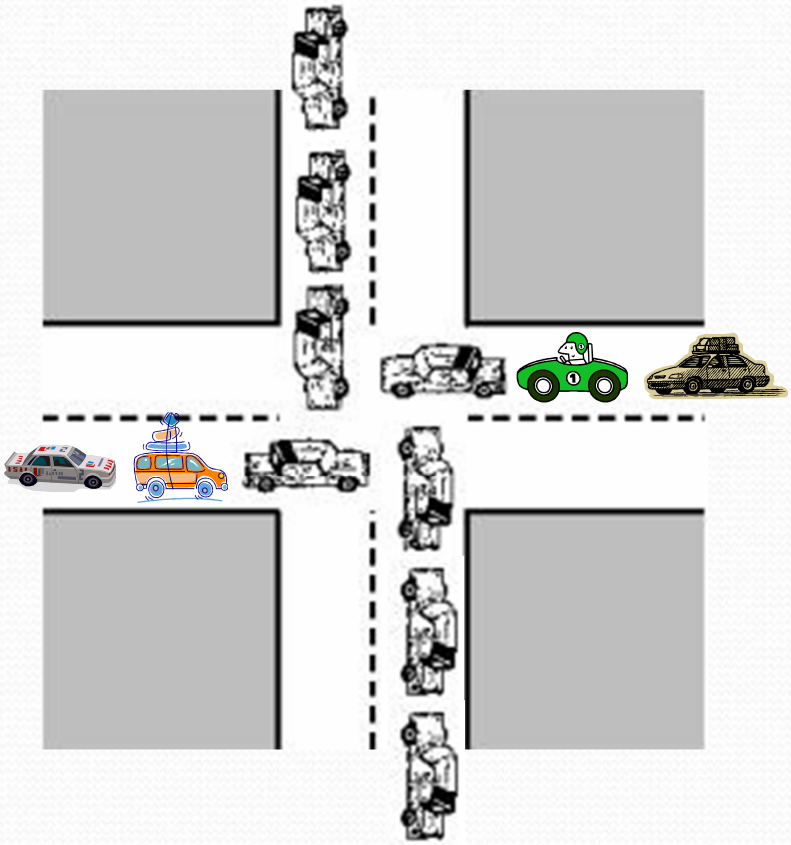
A deadlock resulting from competition for nonshareable railroad intersections



Traffic Deadlock



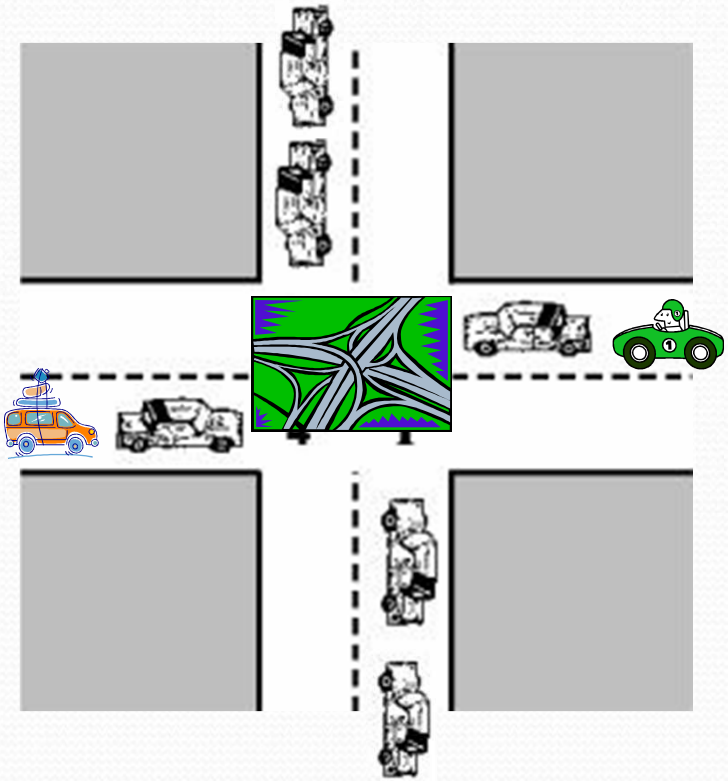
Deadlock possible:
Each vehicle needs two sections of the road to proceed.



Deadlock occurs:
Each vehicle hold one section of the road, waiting for the second section to clear.

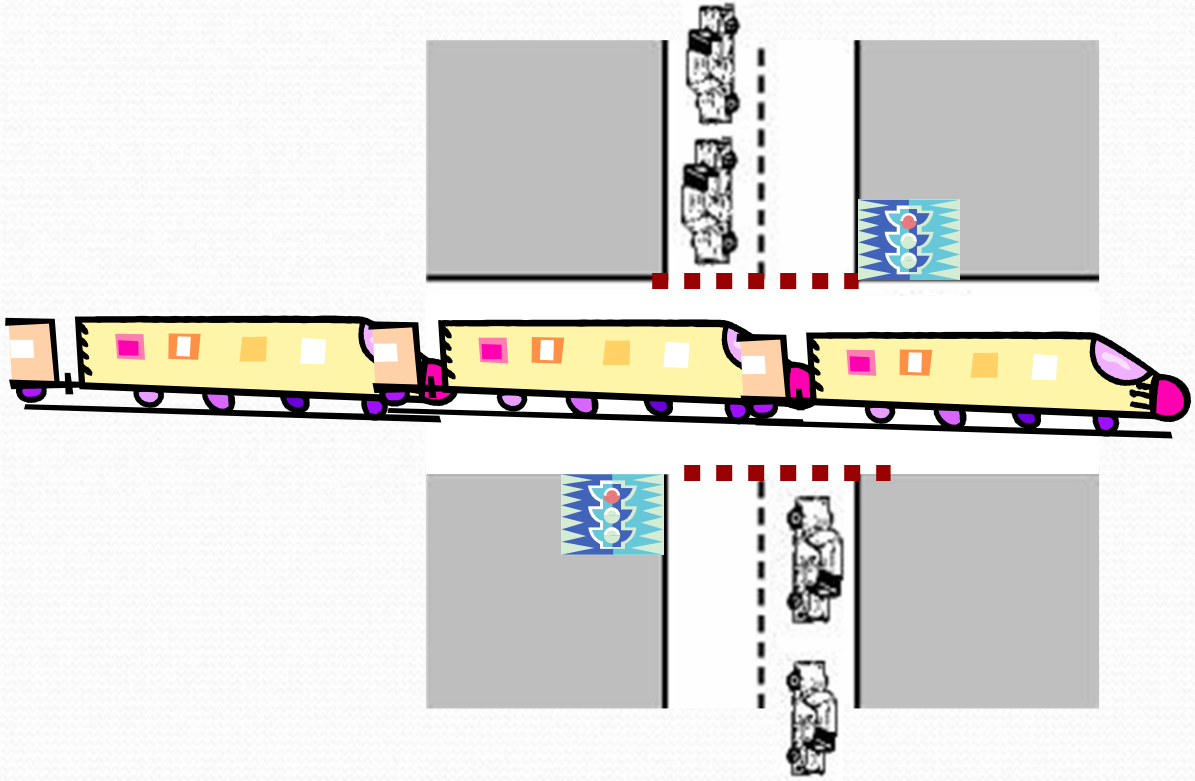
Resolving Deadlock (1)

- **Deadlock prevention**: use an algorithm which can guarantee that no deadlock can occur.



Resolving Deadlock (2)

- **Deadlock avoidance**: use an algorithm which will anticipate that a deadlock is likely to occur and therefore refuse a resource request.



Resolving Deadlock (3)

- **Deadlock detection and recovery**: use an algorithm to detect the occurrence of a deadlock and force the actions to release the resources that are hold while waiting.