Operating System

1 Overview

"What is Operating System ?"

- An operating system acts an intermediary(trung gian) between the user of a computer and the computer hardware.
- Operating system provide an environment in which a user can execute programs in a convenient and efficient manner.
- An Operating System is software that manages the computer hardware, the hardware must provide appropriate mechanisms to ensure the correct operation and prevent user from interfering with the proper operation of the system.

1.1 What Operating System can do ?

1. A computer system can be divided into 4 components: the hardware, the operating system, the application programs, and the users.
    - The Hardware includes the central unit processing (CPU), the memory, and the input-output (IO) devices
    - Hardware provides the basic computing resources for the system.
    - The Application programs define the ways in which these resources are used to solve user's computing problem.
    - The operating system controls the hardware and coordinates it use among the various application programs for various users.
    - In system's view, the operating system is the program most intimately involved with the hardware. an operating system can be viewed as resource allocator.
    - An operating system is a control program that manages the execution of user programs to prevent errors and improper use of the computer.
    - An operating system is a program running at all times on the computer - usually called the kernel
2. Computer-System Operation
    - For a computer to start running, it needs to have an initial program to run or bootstrap program.
    - Bootstrap program is stored in ROM (read-only memory) or EEPROM (electrically erasable programmable read-only memory) or firmware.
    - Bootstrap program initializes all aspects of the system, it knows how to load the operating system and start executing that system.
    - The operating system then starts executing the first process and waits for the occurrence events.
    - The occurrence of an event is usually signaled by an interrupt from either software or hardware.
        - Hardware may trigger an interrupt at any time by sending a signal to the CPU.
        - Software may trigger a special operation called a system call or monitor call.

3. Interrupt:

   - When the CPU is interrupted, it stops what it is doing and immediately transfers execution to a fixed location.
   - The fixed location usually contains the starting address where service routine for the interrupt is located.
   - The interrupt must transfer control to the appropriate interrupt service routine.
   - The interrupt vector provide the address of the interrupt service routine for the interrupting device.
   - The interrupt architecture must also save the address of the interrupted instruction.
   - Incoming interrupts are disabled while another interrupt is being processed to prevent a lost interrupt
   - A trap is a software-generated interrupt caused either by an error or a user request.
   - An operating system is interrupt driven.

4. Storage Structure

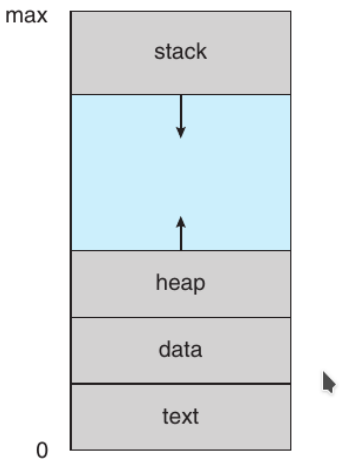
   - General-purpose computers run most of their program from rewritable memory, called main memory (or random-access memory (RAM)).
   - Main memory is commonly implemented in a semiconductor technology called dynamic random-access memory (DRAM).
   - Main memory is usually too small to store all needed programs and data permanently.
   - Main memory is a volatile storage device that loses its contents when power is turned off or otherwise lost. → secondary storage for larges quantities of data permanently.
   - magnetic disk: provide storage for both program and data.
   - Most programs (system and application) are storage on a disk until they are loaded into memory.
   - Differences between among the various storage systems lie on speed, cost, size and volatility.

   - In the absence of power and general backup systems, some data must be written to nonvolatile storage.
   - Some nonvolatile disks are electronic disk, NVRAM (DRAM with battery backup power).

Chapter 3: Process

Process Concept

1. The Process
    - The process:
        - A process is a program in execution.
        - The status of the current activity of a process is represented by the value of the program counter and the contents of the processor's register.
        - The memory layout of a process is typically divided into multiple sections
            * Text section: - the executable code
            * Data section: - global variables
            * Heap section: - memory that is dynamically allocated duirng program run time.
            * Stack section: - temporary data storage when invoking functions (parameters, return val,



... etc).

        - The size of text and data section is fixed, while the stack and heap section can shrink and grow dynamically during program execution.
        - the stack and heap sections grow toward one another, the operating system must ensure they do not overlap one another.
        - A program itself is not a process, which is called passive entity while process is called active entity with a program counter specifying the next instruction to execute and a set of associated resources.
        - A program become a process when when an executable file is loaded into a memory.
        - Two processes may be associated with the same program, they are nevertheless considered two separate execution sequences.
        - A process can itself be an execution environment for another code. For instance, the JVM
2. Process State
    - The state of the process is defined in part of the current activity of that process.
        - New: The process is being created.

- Running: Instructions are being executed.
- Waiting: The process is waiting for some events to occur (such as I/O completion or reception of a signal).
- Ready: The process is waiting to be assigned to a processor.
- Terminated: The process has finished execution.

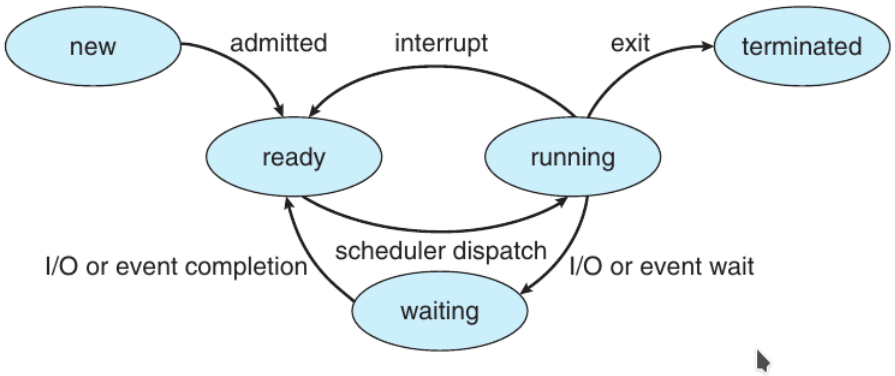


Figure 3.2 Diagram of process state.

- Only one process can be running on any processor core at any instant. Many process may be ready and waiting.

3. Process Control Block
   - Each process is represented in the operating system by a process control block (PCB) or task control block.
     - Process state
     - Program counter
     - CPU register
     - CPU-scheduling information
     - Memory-management information
     - Accounting information
     - I/O status information
4. Threads
   - A process is a program that performs a single thread of execution.

Process Scheduling

- The objective of multiprogramming is to have some process running at all the times so as to maximize CPU utilization.
- The objective for time sharing is to switch a CPU core among processes so frequently that users can interact with each program while it is running.
- To meet those objective, process scheduler is used to select an available process for program execution on a core.

- The number of processes currently in memory is known as the degree of multiprogramming.
- Most processes can be described as either I/O bound or CPU bound.
  - An I/O bound process is one that spends more of its time doing I/O than it spends doing computations.
  - A CPU-bound process generates I/O requests infrequently, using more of its time doing computations.

1. Scheduling Queues
   - As a processes enter the system, they are put into the ready queue, where they are ready and waiting to execute on a CPU's core. This queue is a linked-list.
   - When a process is allocated a CPU core, it executes for a while and eventually terminates, is interrupted, or waits for the occurrence of a particulat event - I/O request. Therefore, the process has to wait for the completion of I/O - which is placed in a wait queue.

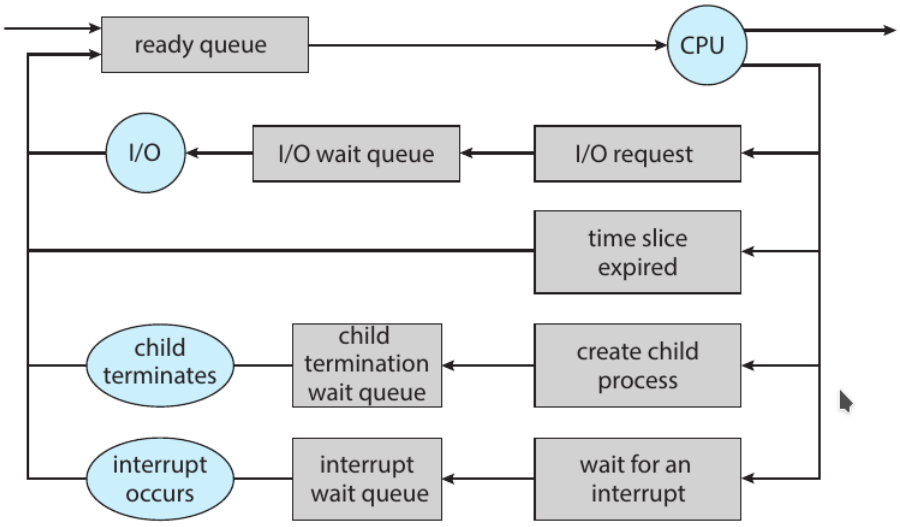


Figure 3.5 Queueing-diagram representation of process scheduling.

2. CPU Scheduling
   - The role of CPU scheduler is to select from among the processes that are in the ready queue and allocate a CPU core to one of them.
   - The scheduler must select a new process for the CPU frequently.