Computer Science Illuminated
SECOND EDITION
Nell Dale • John Lewis

Chapter 10,11

Operating Systems

File Systems

Chapter Goals

- Describe the two main responsibilities of an operating system
- Define memory and process management
  - Timesharing creates the virtual machine illusion
  - Explain the relationship between logical and physical addresses
  - Compare memory management techniques
    - Distinguish between fixed and dynamic partitions
    - Define and apply partition selection algorithms
    - Explain how demand paging creates the virtual memory illusion

10-2

Chapter Goals

  - Explain the stages and transitions of the process life cycle
  - Explain the processing of various CPU scheduling algorithms
- Describe the purpose of files, file systems, and directories
  - Distinguish between text and binary files
  - Identify various file types by their extensions
  - Define the basic operations on a file
  - Create absolute and relative paths for a directory tree

10-3

Software Categories

- **Application software** Software written to address specific needs—to solve problems in the real world

  Word processing programs, games, inventory control systems, automobile diagnostic programs, and missile guidance programs are all application software
- **System software** Software that manages a computer system at a fundamental level

  It provides the tools and an environment in which application software can be created and run

10-4

Operating System

- An **operating system**
  - manages computer resources, such as cpu, memory and input/output devices
  - provides an interface through which a human can interact with the computer
  - allows an application program to interact with these other system resources

10-5

Operating System

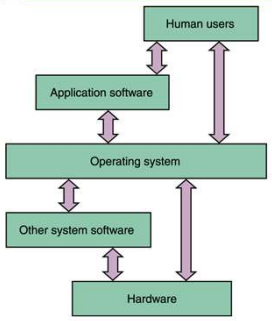


Figure 10.1
An operating system interacts with many aspects of a computer system.

10-6

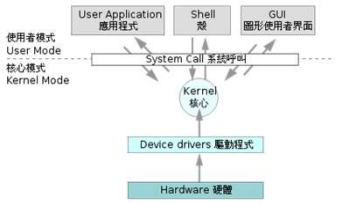
Operating System

- The various roles of an operating system generally revolve around the idea of "sharing nicely"
- An operating system manages resources, and these resources are often shared in one way or another among programs that want to use them

10-7

操作系统的历史

操作系统（英语：Operating System，简称OS）是管理计算机硬件与软件资源的程序，同时也是计算机系统的核心与基石。操作系统身负诸如管理与配置内存、决定系统资源供需的优先次序、控制输入与输出设备、操作网络与管理文件系统等基本事务。

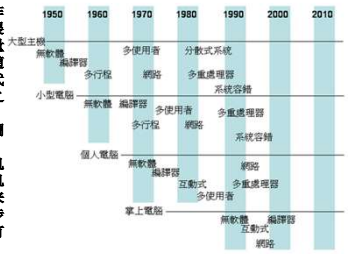


详细见维基百科

10-8

操作系统的历史

綜观计算机之历史，操作系统与计算机硬件的发展息息相关。从最早的批量模式开始，分时机制也随之出现，在多处理器时代来临时，操作系统也随之添加多处理器协调功能，甚至是分散式系统的协调功能。
另一方面，在个人计算机上，操作系统因袭大型机的成长之路，在硬件越来越复杂、强大时，也逐步实践以往只有大型机才有的功能。



总而言之，操作系统的历史就是一部解决计算机系统需求与问题的历史。

10-9

1980年代前

- 第一部计算机并没有操作系统。
  - 早期计算机的创建方式（如同建造机械算盘）与性能不足以运行如此程序。
- 1947年发明了晶体管，以及莫里斯·威尔克斯发明的微程序方法
- 1964年，IBM System/360,都共用代号为OS/360的操作系统。
  - 硬盘驱动器的面世
  - 分时概念的创建
- 1969年，AT&T贝尔实验室的丹尼斯·里奇与肯·汤普逊所创建的Unix系统
  - 1971年在PDP-11上运行

IBM System/360，大型主机的经典之作

10-10

1980年代（个人计算机OS）

- 第一代微型计算机，没有装设操作系统的需求或能力
  - 基本输入输出系统（BIOS）
  - BASIC程序
- 1980年微软公司，比尔·盖茨推出MS-DOS
  - 软盘操作系统（Disk Operating System，DOS）
  - MS-DOS的成功使得微软成为地球上最赚钱的公司之一
- 80年代操作系统异数是Mac OS
  - 此时一位施乐·伯拉图实验室的员工Dominik Hagen访问了苹果，展示了施乐开发的图型用户界面
  - 苹果公司开启了图形化OS进程

10-11

90年代（多媒体界面OS）

- 1997年发布新操作系统——Mac OS X的测试版
  - Steve Jobs风光再现
- 微软发布Windows系列产品
  - 1993年7月27日推出Windows 3.1
  - 1995年8月15日推出Windows 95
  - 2000年所推出的Windows 2000
  - Windows XP在2001年10月25日发布
  - 2004年占领90%PC及PC服务器市场
- 1991 年, 芬兰学生林纳斯·托瓦兹根据类 Unix 系统，发布了 Linux 操作系统内核

现代OS发展趋势

- 手机OS
  - IOS（Apple）
  - Android（Google）
- 互联网及云服务OS
  - Chrome OS（Google）
- 都采用Linux内核
  - 微软何去何从？
  - WIN 8 的挑战？

历史证明，得OS者得天下！

10-13

Resource Management

- **Multiprogramming** The technique of keeping multiple programs in main memory at the same time that compete for access to the CPU so that they can execute
- **Memory management** The process of keeping track of what programs are in memory and where in memory they reside

10-14

Resource Management

- **Process** A program in execution
- The operating system performs **process management** to carefully track the progress of a process and all of its intermediate states
- **CPU scheduling** determines which process in memory is executed by the CPU at any given point

10-15

Batch Processing

- A typical computer in the 1960s and '70s was a large machine
- Its processing was managed by a human *operator*
- The operator would organize various jobs from multiple users into *batches*

10-16

Batch Processing

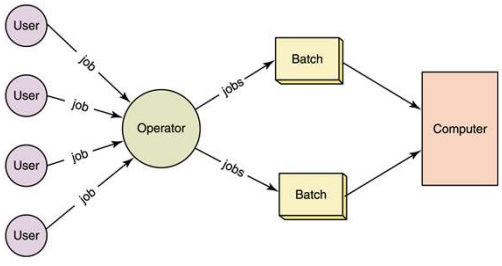


Figure 10.2 In early systems, human operators would organize jobs into batches

10-17

Timesharing

- **Timesharing system** A system that allows multiple users to interact with a computer at the same time
- **Multiprogramming** A technique that allows multiple processes to be active at once, allowing programmers to interact with the computer system directly, while still sharing its resources
- In a timesharing system, each user has his or her own **virtual machine**, in which all system resources are (in effect) available for use

10-18

Memory Management

- Operating systems must employ techniques to
  - Track where and how a program resides in memory
  - Convert **logical addresses** into actual **addresses**
- **Logical address** (sometimes called a virtual or relative address) A value that specifies a generic location, relative to the program but not to the reality of main memory
- **Physical address** An actual address in the main memory device

10-19

Memory Management

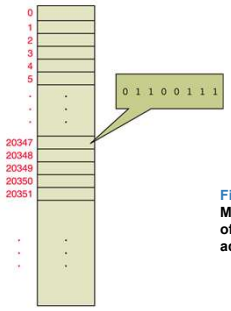


Figure 10.3
Memory is a continuous set of bits referenced by specific addresses

10-20

Single Contiguous Memory Management

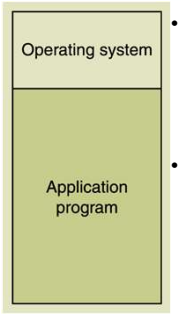


Figure 10.4
Main memory divided into two sections

- There are only two programs in memory
  - The operating system
  - The application program
- This approach is called **single contiguous memory management**

10-21

Single Contiguous Memory Management

- A logical address is simply an integer value relative to the starting point of the program
- To produce a physical address, we add a logical address to the starting address of the program in physical main memory

10-22

Single Contiguous Memory Management

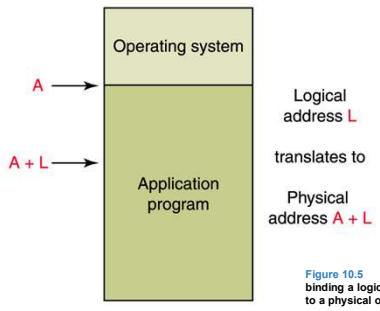


Figure 10.5
binding a logical address to a physical one

10-23

Partition Memory Management

- **Fixed partitions** Main memory is divided into a particular number of partitions
- **Dymanic partitions** Partitions are created to fit the needs of the programs

10-24

Partition Memory Management

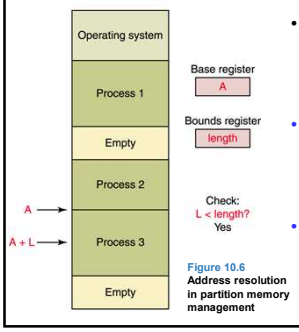


Figure 10.6
Address resolution in partition memory management

- At any point in time memory is divided into a set of partitions, some empty and some allocated to programs
- **Base register** A register that holds the beginning address of the current partition
- **Bounds register** A register that holds the length of the current partition

Partition Selection Algorithms

Which partition should we allocate to a new program?

- **First fit** Allocate program to the first partition big enough to hold it
- **Best fit** Allocated program to the smallest partition big enough to hold it
- **Worst fit** Allocate program to the largest partition big enough to hold it

Paged Memory Management

- **Paged memory technique** A memory management technique in which processes are divided into fixed-size **pages** and stored in memory **frames** when loaded into memory
  - **Frame** A fixed-size portion of *main memory* that holds a process page
  - **Page** A fixed-size portion of a *process* that is stored into a memory frame
  - **Page-map table** (PMT) A table used by the operating system to keep track of page/frame relationships

Paged Memory Management

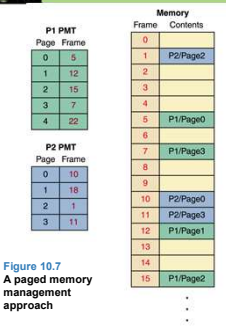


Figure 10.7
A paged memory management approach

- To produce a physical address, you first look up the page in the PMT to find the frame number in which it is stored
- Then multiply the frame number by the frame size and add the offset to get the physical address

Paged Memory Management

- **Demand paging** An important extension of paged memory management
  - Not all parts of a program actually have to be in memory at the same time
  - In demand paging, the pages are brought into memory on demand
- Page swap The act of bringing in a page from secondary memory, which often causes another page to be written back to secondary memory

Paged Memory Management

- The demand paging approach gives rise to the idea of **virtual memory**, the illusion that there are no restrictions on the size of a program
- Too much page swapping, however, is called **thrashing** and can seriously degrade system performance.

Process Management

- The Process States

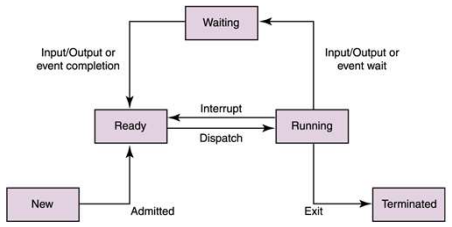


Figure 10.8 The process life cycle

10-31

The Process Control Block

- The operating system must manage a large amount of data for each active process
- Usually that data is stored in a data structure called a process control block **(PCB)**
- Each state is represented by a list of PCBs, one for each process in that state

10-32

The Process Control Block

- Keep in mind that there is only one CPU and therefore only one set of CPU registers
  - These registers contain the values for the currently executing process
- Each time a process is moved to the running state:
  - Register values for the currently running process are stored into its PCB
  - Register values of the new running state are loaded into the CPU
  - This exchange of information is called a **context switch**

10-33

CPU Scheduling

- **CPU Scheduling** The act of determining which process in the *ready* state should be moved to the *running* state
  - Many processes may be in the ready state
  - Only one process can be in the running state, making progress at any one time
- *Which one gets to move from ready to running?*

10-34

CPU Scheduling

- **Nonpreemptive scheduling(非抢占式调度)** The currently executing process gives up the CPU voluntarily
- **Preemptive scheduling(抢占式调度)** The operating system decides to favor another process, preempting the currently executing process
- **Turnaround time(时间片)** The amount of time between when a process arrives in the ready state the first time and when it exits the running state for the last time

10-35

CPU Scheduling Algorithms

First-Come, First-Served(排队服务策略)
- Processes are moved to the CPU in the order in which they arrive in the running state

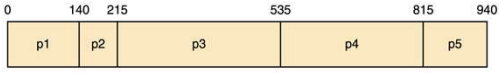
Shortest Job Next(最短时间优先)
- Process with shortest estimated running time in the ready state is moved into the running state first

Round Robin(分时循环)
- Each process runs for a specified time slice and moves from the running state to the ready state to await its next turn if not finished

10-36

First-Come, First-Served

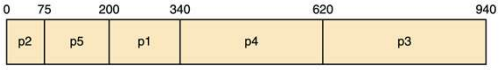
Process	Service time
p1	140
p2	75
p3	320
p4	280
p5	125



10-37

Shortest Job Next

- Looks at all processes in the ready state and dispatches the one with the smallest service time



10-38

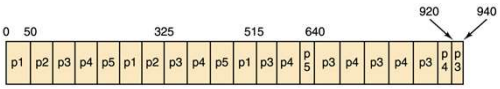
Round Robin

- Distributes the processing time equitably among all ready processes
- The algorithm establishes a particular **time slice** (or time quantum), which is the amount of time each process receives before being preempted and returned to the ready state to allow another process its turn

10-39

Round Robin

- Suppose the time slice was 50



10-40

CPU Scheduling Algorithms

Are they preemptive or non-preemptive? Explain

- *First-Come, First-Served?*
- *Shortest Job Next?*
- *Round Robin?*

10-41

File Systems

- **File** A named collection of related data
- **File system** The logical view that an operating system provides so that users can manage information as a collection of files
- **Directory** A named group of files

11-42

Text and Binary Files

- **Text file** A file in which the bytes of data are organized as characters from the ASCII or Unicode character sets
- **Binary file** A file that contains data in a specific format, requiring interpretation

Text and Binary Files

- The terms ***text file*** and ***binary file*** are somewhat misleading
- They seem to imply that the information in a text file is not stored as binary data
- Ultimately, all information on a computer is stored as binary digits
- These terms refer to how those bits are formatted: as chunks of 8 or 16 bits, interpreted as characters, or in some other special format

File Types

- Most files, whether they are in text or binary format, contain a specific type of information
  - For example, a file may contain a Java program, a JPEG image, or an MP3 audio clip
- The kind of information contained in a document is called the **file type**
  - Most operating systems recognize a list of specific file types

File Types

Extensions	File type
txt	text data file
mp3, au, wav	audio file
gif, tiff, jpg	image file
doc, wp3	word processing document
java, c, cpp	program source files

Figure 11.1 **Some common file types and their extensions**

- File names are often separated, usually by a period, into two parts
  - Main name
  - File extension
- The **file extension** indicates the type of the file

File Operations

- Create a file
- Delete a file
- Open a file
- Close a file
- Read data from a file
- Write data to a file
- Reposition the current file pointer in a file
- Append data to the end of a file
- Truncate a file (delete its contents)
- Rename a file
- Copy a file

File Access

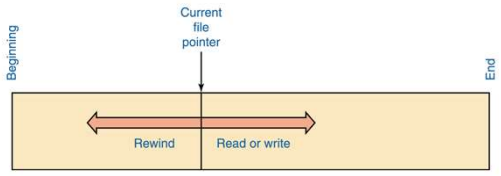


Figure 11.2 **Sequential file access**

File Access

- **Sequential access** Information in the file is processed in order, and read and write operations move the current file pointer as far as needed to read or write the data

  The most common file access technique, and the simplest to implement

11-49

File Access

- **Direct access** Files are conceptually divided into numbered logical records and each logical record can be accessed directly by number

11-50

File Access

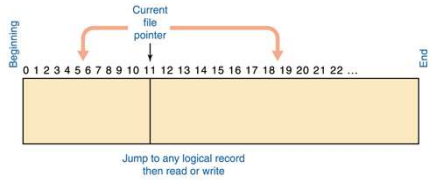


Figure 11.3 **Direct file access**

11-51

File Protection

- In multiuser systems, file protection is of primary importance
- We don't want one user to be able to access another user's files unless the access is specifically allowed
- A file protection mechanism determines who can use a file and for what general purpose

11-52

File Protection

- A file's protection settings in the Unix operating system is divided into three categories
  - Owner
  - Group
  - World

	Read	Write/Delete	Execute
Owner	Yes	Yes	No
Group	Yes	No	No
World	No	No	No

11-53

Directory Trees

- A directory of files can be contained within another directory

  The directory containing another is usually called the *parent directory*, and the one inside is called a *subdirectory*
- **Directory tree** A logical view of a file system; a structure showing the nested directory organization of a file system
- **Root directory** The directory at the highest level

11-54

Directory Trees

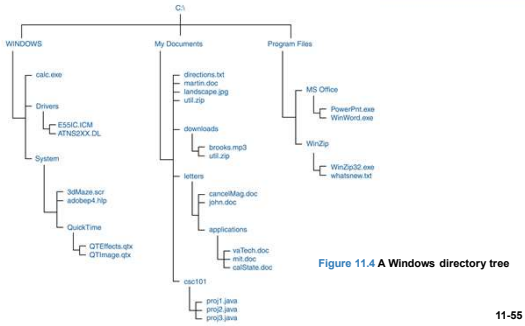
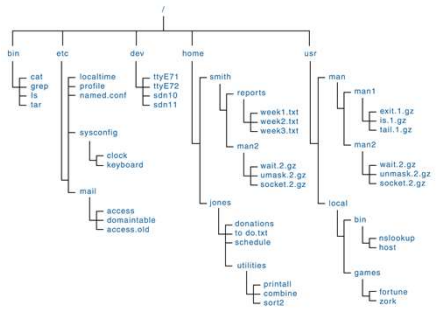


Figure 11.4 A Windows directory tree

Directory Trees

- At any point in time, you can be thought of as working in a particular location (that is, a particular subdirectory)
- **Working directory** The subdirectory in which you are working

Figure 11.5 A Unix Directory Tree



Path Names

- **Path** A text designation of the location of a file or subdirectory in a file system, consisting of the series of directories through which you must go to find the file
- **Absolute path** A path that begins at the root and specifies each step down the tree until it reaches the desired file or directory
- **Relative path** A path name that begins at the current working directory

Path Names

- Examples of absolute path
  - C:\Program Files\MS Office\WinWord.exe
  - C:\My Documents\letters\applications\vaTech.doc
  - C:\Windows\System\QuickTime
- Suppose the current working directory is
  - C:\My Documents\letters
- Then the following relative path names could be used
  - cancelMag.doc
  - applications\calState.doc

课程要点

1、The role of Operation System
2、Key Concept: Multiprogramming, Process, Timesharing
3、Resource management: Memory management , Process management , CPU scheduling
4、Memory management：Logical Address, Physical Address, Fixed partitions, Dynamic partitions, (First, Best, Worst) fit
5、Process management： process states