Unit 1 – Part 1 of 3

Core Operating Concepts

Based on Chapter 1 of
Modern Operating Systems (2nd. Edition), Andrew S. Tanenbaum)

- What is an operating system?
- A Taxonomy of Operating Systems
- Operating system concepts

Introduction

- A modern computer system consists of:
  - One, or more processors
  - Memory
  - Disks (storage devices)
  - Input devices (keyboard, mouse, etc.)
  - Network Interfaces
  - Other I/O Devices (display monitor)
- Keeping track of all the parts of a computer system is a complex task

Introduction

- The layered approach helps to control this complexity
  - Hardware is built up in layers
    - Digital Logic layer made up of gates
    - The microarchitecture (microprogram) layer is made up circuits that are built from gates
    - The machine layer is built on top of the microarchitecture level and implements the Instruction Set Architecture (ISA) of the computer system.

The Machine Level

- Each layer of this “tower of abstraction” is a virtual machine that:
  - Is implemented on top of the layer(s) below and utilizes the services provided by the layer below
  - Provides services to the layer above
- At the machine level, a programmer may write programs in machine language to instruct the hardware

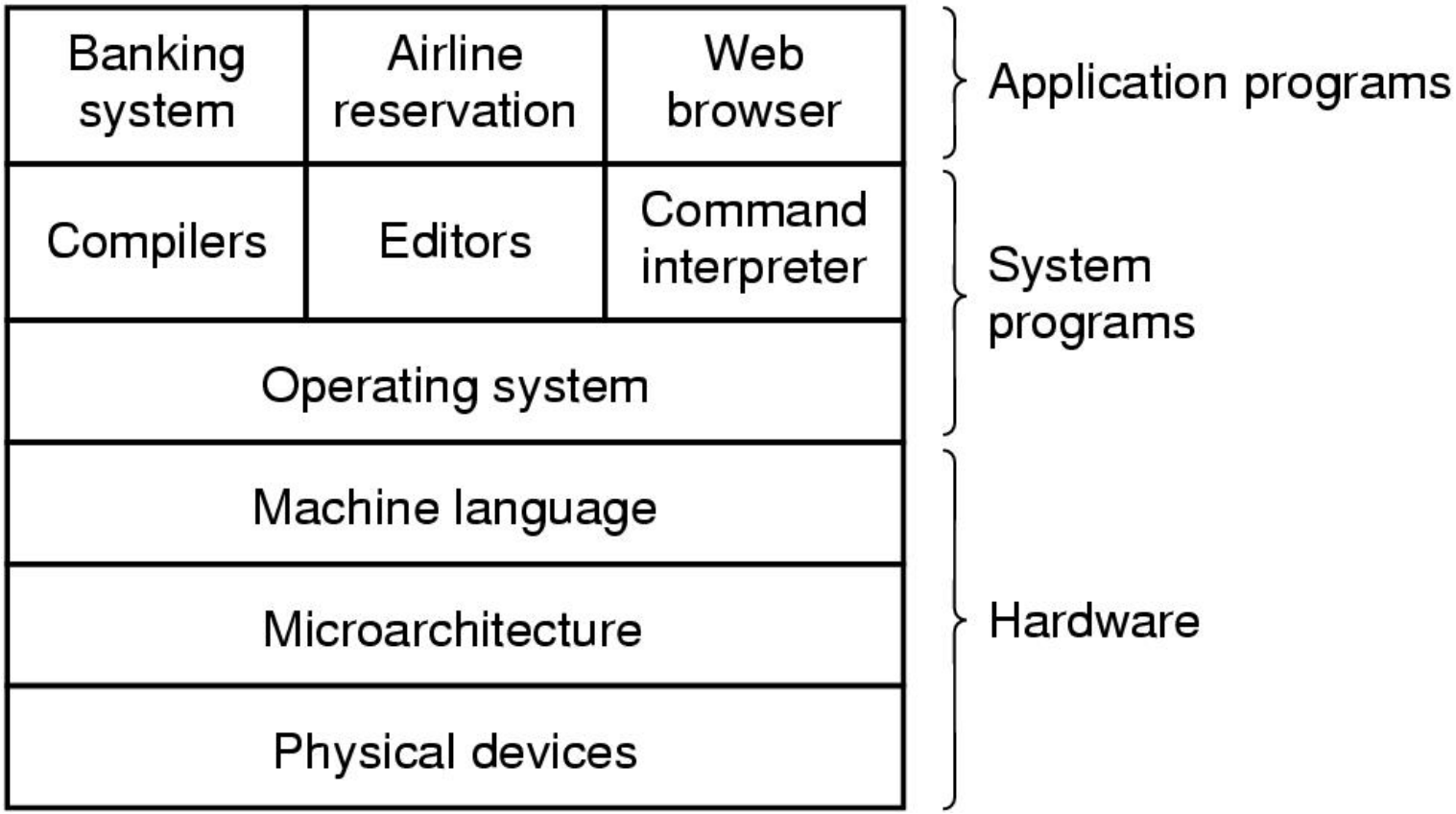
The Machine Level

- Machine instructions are primitive and awkward to use
- For example, to copy bytes from one device (e.g. a CD-ROM to another (e.g. a fixed disk) a programmer must issue several instructions that:
    1. Check to see if the device is available
    2. Copy a chunk of bits from the CD-ROM drive into memory
    3. Set a status bit to 1 to indicate that there is data in memory that must now be transferred to the disk
    4. Generate a signal to the processor to indicate that bytes can now be transferred from memory to the disk
- Every program that needs to carry out this operation would have to include code to do so
- Also, program might have to be modified when devices are changed (e.g. a faster CD-ROM is installed in the computer).

The Operating System

- A layer of software is built on top of the hardware to manage the hardware and to (partially) hide the complexities of the hardware from programmers.
- This layer is referred to as the *Operating System*
- Programs that are run on a computer system do so under the control of the operating system.
- The operating system may be seen in that case as the interface between the programs that are run on a computer system and the hardware that makes up the computer system.
- **Note:** The operating system is itself a program (usually very large and messy) that is run on the hardware

Introduction



What is an Operating System

- *It is an extended machine*
  - The O/S Hides the messy details which must be performed
    - For example the O/S presents a disk as a collection of files that may be read from, or written to, the disk,
    - This “pretty is a far cry from tracks, sectors, and bytes
  - The O/S implements a virtual machine on top of the machine that is easier to use (i.e. write programs for)
  - This is the “top down” (or external view) of an operating system.

What is an Operating System

- *It is a resource manager*
  - Each program gets time with the resource (e.g. processor)
  - Each program gets space on the resource (e.g. memory)
  - The operating system co-ordinates the use of the computer system in order to ensure that system resources are used in an optimal manner
  - In cases where computers do several things at the same (this is the norm today) the O/S "directs" these activities to prevent one activity from interfering with the other.
- *This is the "bottom-up" (or internal) view of an O/S*

A Taxonomy of Operating Systems

- Mainframe operating systems
  - Very powerful systems used on large computers (e.g. in in nsurance companies, banks)
  - Support batch processing, timesharing)
- Server operating systems
  - Large PC's that provide services (usually over a network)
  - Typically UNIX or Windows 2000
- Multiprocessor operating systems
  - Special versions of server O/S for multiprocessor hardware

A Taxonomy of Operating Systems

- Personal computer operating systems
  - Windows, Linux, MacOS
- Real-time operating systems
  - Special O/S's that operate in real time
  - Response of hardware must meet strict time requirements
  - VxWorks and QNX are two "well known" O/S in this group
- Embedded operating systems
  - O/S for small devices (PDA's cell phones)
  - PALMOS, Windows CE
- Smart card operating systems
  - O/S which run on credit card size systems
  - Mostly JVM based.

A Taxonomy of Operating Systems

- Two dominant O/S families today
  - UNIX
    - Source code was available
    - Many different versions developed (many incompatible)
    - Two dominant versions
      - AT&T Unix System V
      - University of California Berkley Software Distribution BSD
  - IEEE developed POSIX standard defining minimal set of interface functions that a UNIX system should implement.
  - Linux is free UNIX

A Taxonomy of Operating Systems

- A Bit About Linux
  - MINIX O/S developed for teaching by Tenenbaum
  - Linus Torvald developed a production version called Linux
  - Both Linux and MINIX are free
  - GNU license allows free distribution of O/S Source
    - Source must be distributed with binary
    - Red Hat, Caldera, Mandrake
    - Knoppix is Linux run from a single CD
    - Ubuntu is a version of Linux that was intended to be “more friendly”

A Taxonomy of Operating Systems

- Windows is the other Dominant Family of O/S’
  - CP/M was first PC Operating System
  - DOS (Disk Operating System) later to be renamed MS-DOS dominated the PC Market.
  - Windows provided GUI for DOS (up to Windows 3.x)
  - Windows 95 was first free standing version of Windows
  - Followed by Windows 98
  - Windows NT was first 32-bit O/S
  - Windows 2000 was introduced as Windows NT version 4
  - Windows VISTA

Operating System Concepts

Every modern operating systems provides support for:

- Creating and Managing Processes
- Inter-Process Communication
- Preventing/Resolving Conflicts (Deadlocks)
- Memory Management
- Communicating with peripherals (Input/Output)
- Management of Persistent Storage (Files)
- Security
- User Interface (Shell)

Operating System Concepts

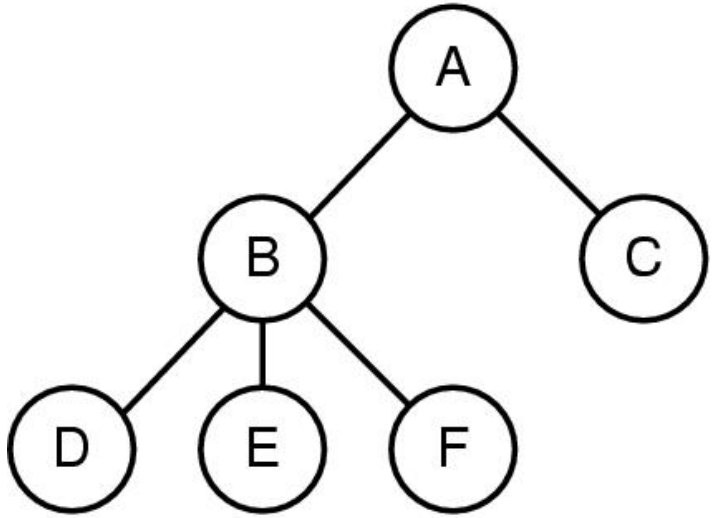
- Processes
  - A program is run on a computer system as a process (a task)
    - A process is a running instance of a program
  - Associated with each process are:
    - The program's address space (instructions, data, stack)
    - The values stored in a set of registers including Program Counter, stack pointer
    - Other information needed to run the program
  - The O/S uses this data to manage the process
    - E.g when process is temporarily suspended data is saved so that when the process is restarted it continues where it left off.

Operating System Concepts

- Processes
  - Each process is uniquely identified with a process id
  - The O/S will in most cases store information in a process table
  - The process id is used to locate this information.
  - A process can cause other processes to be created
  - Unix for example implements processes as a hierarchy
    - Process 0 is the O/S memory manager (swapper)
    - Process 1 is the process that creates all others
    - All processes are descendants of process 1

Operating System Concepts

- Processes
- A process tree
  - A created two child processes, B and C
  - B created three child processes, D, E, and F



Operating System Concepts

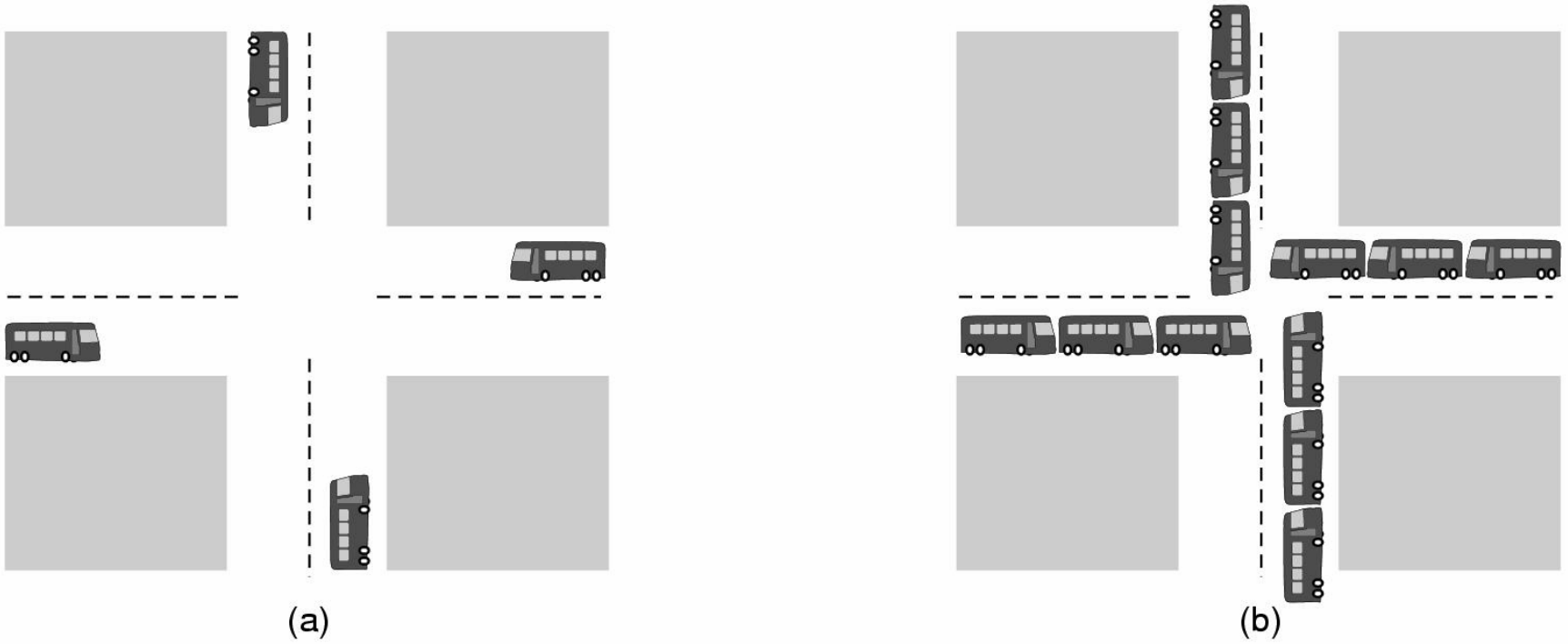
- Inter-Processes Communication
- Most modern operating systems are multiprocessing systems
  - Several processes can be run at the “same time”
- Processes can perform tasks in cooperation with each other
- For example one process can be writing data to a file that is being read by another process
- These processes must communicate with other in other to synchronize their activities
- The O/S facilities this by providing inter-process communication (IPC) facility
  - E.g. on UNIX one process can terminate another process (given the requisite authority)

Operating System Concepts

- Deadlock
  - As processes run they require the use of various resources
  - Consider the following:
    - Process A is using a resource that process B needs
    - Process B requires the use of the resource that process A is using
  - This occurrence is referred to as deadlock
  - Prevention of deadlocks is the duty of the O/S
  - If deadlocks occur the O/S must mediate in order to resolve the situation.

Operating System Concepts

- A real-life depiction of deadlock.



(a) A potential deadlock. (b) an actual deadlock.

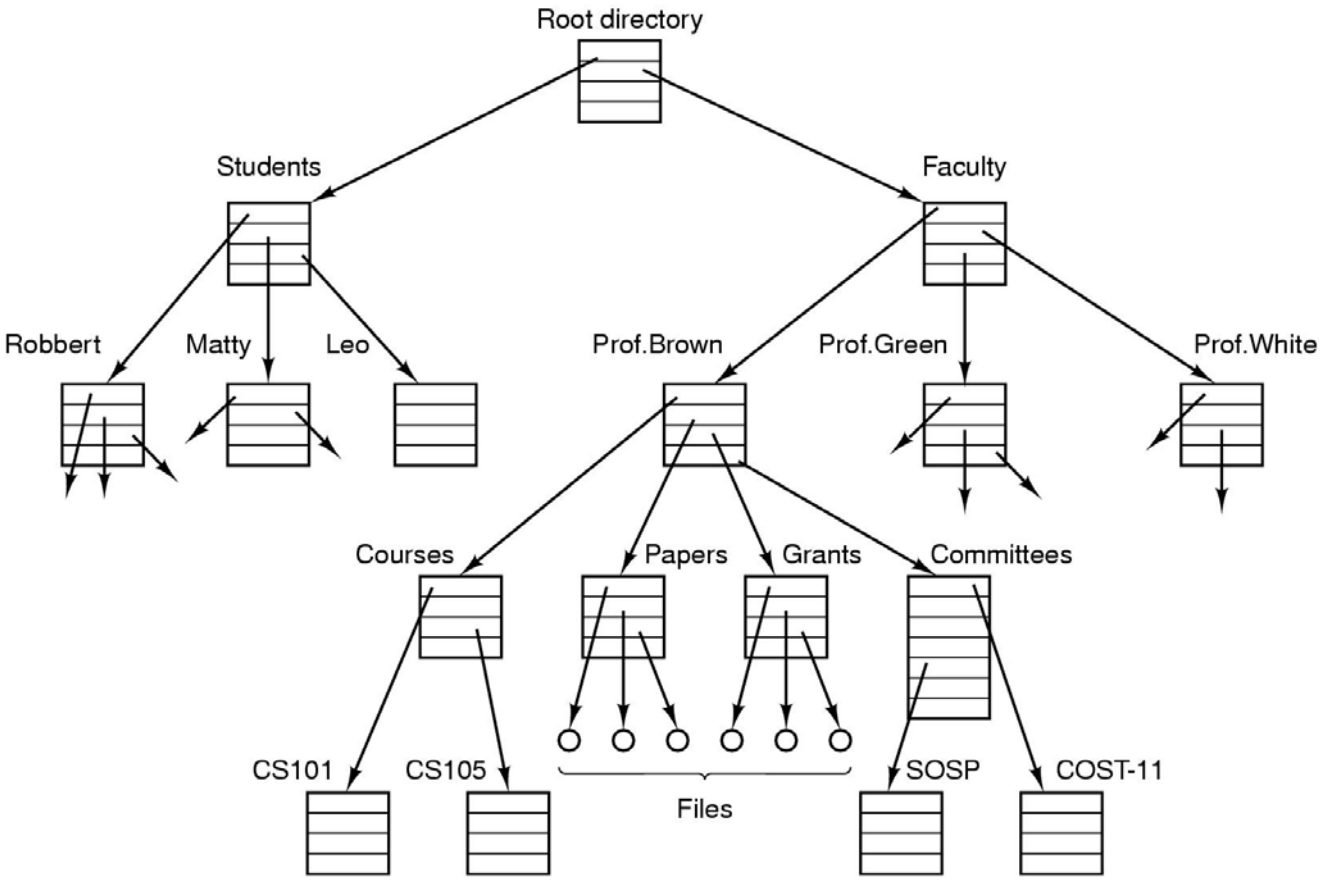
Operating System Concepts

- Memory Management
  - Modern O/S allow more than one program to run at a time
  - Memory is partitioned into units that hold data and instructions for each running process
  - Memory management is a function of the O/S
  - In older systems simply swapped programs in and out of memory as needed
    - Programs had to be no larger than physical memory
  - Modern computer systems use Virtual Memory
    - Portions of program are loaded into memory as needed
      - Paging
      - Segmentation

Operating System Concepts

- Files, Directories and File Systems
  - Secondary storage provides persistent storage in a computer system
  - O/S implements an abstraction that is called a file to represent bytes stored on secondary storage
  - Files are organized into directories
  - Directories make up file systems
- The O/S provides services for managing files, directories, and file systems on secondary storage,

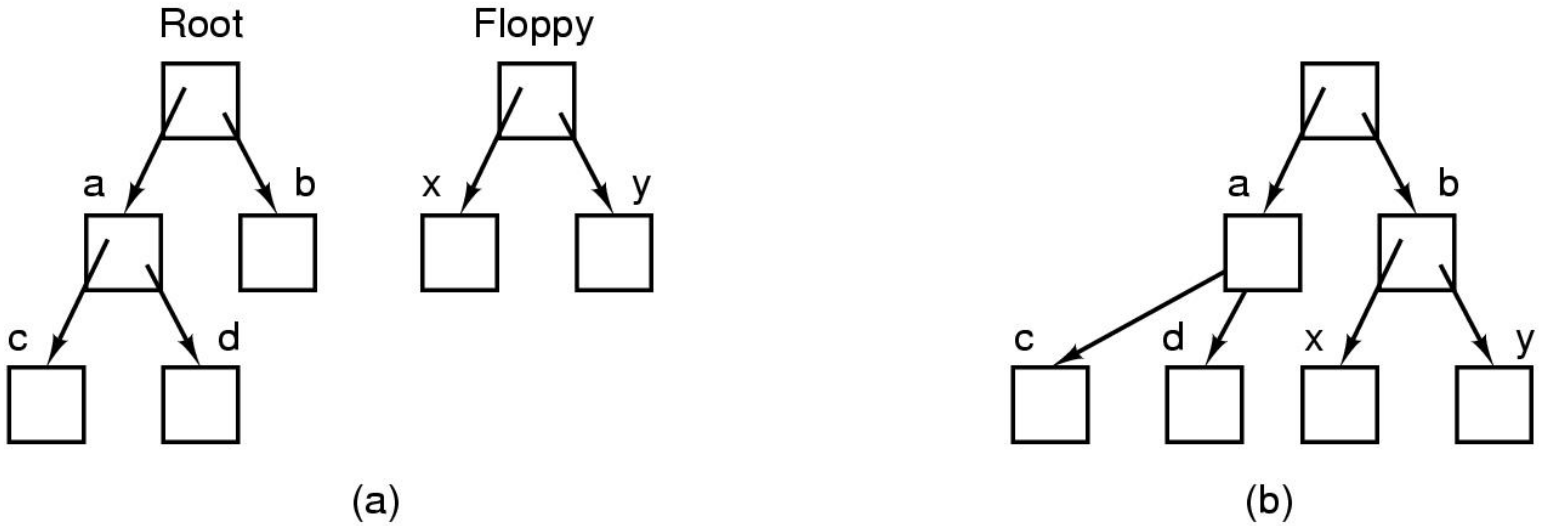
Operating System Concepts



File system for a university department

Operating System Concepts

Unix file systems can be extended to include files from additional secondary storage devices.



- Before mounting,
  - files on floppy are inaccessible
- After mounting floppy on b,
  - files on floppy are part of file hierarchy

Operating System Concepts

- Input/Output
  - Every O/S has an I/O sub-system th
  - O/S must understand how to communicate with many types of peripherals (keyboard, mouse, monitor, printer)
  - The O/S must provide an interface to I/O devices that enables programmers to perform I/O without being concerned with H/W details
  - Modern operating systems interface with the hardware through device drivers
  - Device drivers provide structures that represent devices and operation that can be performed on these structures.

Operating System Concepts

- Security
  - O/S must prevent unauthorized access too resources
  - Must also prevent one user from interfering with another user's resources (e.g. files)
  - O/S provides mechanisms for
    - Determining who can gain access to the computer system
    - What resources a user can access
    - What action(s) the user can carry out on resource

Operating System Concepts

- Shell
  - Interpreter for commands issued by the user
  - Allows user to run programs
  - For example:
    - the user issues a command to display a list of files
    - The operating system responds by displaying the files
    - The shell than waits for another command.
  - An O/S shell also allows a user to start a process that runs without interaction with the user (in the background)
    - ps &
    - start *command* (Windows/DOS command)
- Programmers can write programs using shell commands to perform batch tasks.