Introduction to Operating Systems (Topic 1-1)

홍 성 수

서울대학교 공과대학 전기 공학부
Real-Time Operating Systems Laboratory

Introduction (1)

- Why are operation systems interesting?
  - Bring together many areas for Computer Science: Programming languages, data structures, computer architecture/ hardware,algorithms.
  - Magician: Generator of illusions.
  - Large, complex systems.

Introduction (2)

- Overall course goal:
  - Learn general techniques and be able to apply them elsewhere.

    Question: What are the chances you will be called on to design a new operating system?
- Defining the term “Operating System” is difficult.
  - Discipline arose historically from a set of problems.
- OS history divides into three phases.

Introduction (3)

- Phase 1: Hardware expensive, humans cheap.
  - Goal: Make efficient use of the hardware.

    (1) Single user working at console.
    - Debugging done interactively.
    - OS was a shared subroutine library.

    (2) Simple batch monitor.
    - Debugging done off-line.
    - OS was to load and run user jobs, take dumps.

Phase 1 Computers and I/O

Introduction (4)

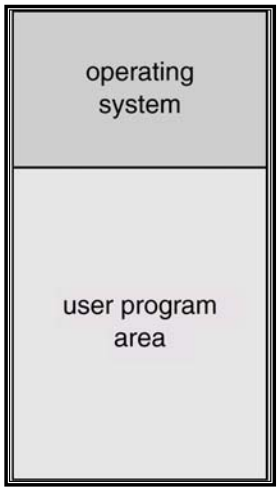
(3) Monitor allowed overlap of I/O and computation.

- Buffering and interrupt handling added to OS.
- Jobs spooled on disk or drum.
- Still single job, so utilization often bad.

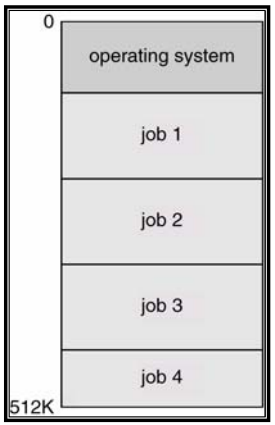
(4) Multiprogramming: several users share the system.

- Memory protection and relocation added to OS.
- Higher utilization because of multiple jobs.
- Concurrent programming becomes necessary.
- OS becomes an focus of study.

Batch System



Multiprogrammed Batch Systems



Introduction (5)

- Phase 2: Hardware cheap, humans expensive.
  - Goal: Make efficient use of people's time.

    (1) Terminals are cheap: Interactive time-sharing.
    - Lets users interact with the system again.
    - Fancy filing systems added to OS.
    - Response time, protection become important.

Introduction (6)

(2) Computers are cheap: Personal computers.
- Computer in every terminal.
- OS becomes a subroutine library again.

(3) Networking.
- Allow different machines to share resources easily.

Introduction (7)

- Phase 3: Everything gets connected.
  - Goal: Provide multimedia services for end users.

    (1) Network is an essential component:
    - Multitasking becomes important again.
    - Internet protocols added to PC OS.
    - Internet programming is important (Web, CGI, Java, ...).

Introduction (8)

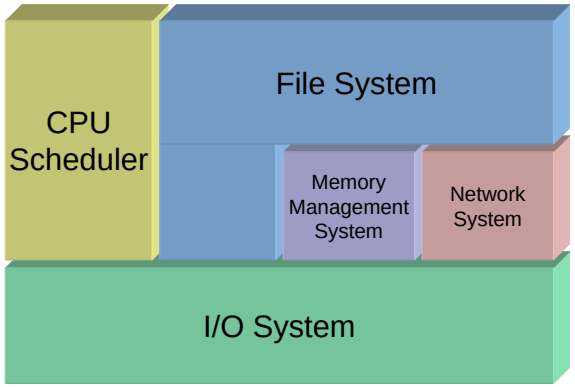
(2) Computers are extremely cheap:
- Even PC has sophisticated architecture.
- OS becomes a complex again.

(3) Multimedia.
- Demands lots of computer and network resources
- Human perception is extremely important.
- QoS (Quality of Service) is a buzz work. (Real-Time OS)
- Home appliances and computers are merged.

Characteristics of OS

- Characteristics of current operating systems:
  - Large: 100k's of lines of code, 100-1000 man-years of work.
  - Complex: Asynchronous, hardware idiosyncrasies,
    conflicting needs of different users, performance.
  - Poorly understood:
    The system outlives any of its builders.
    Too complex to totally debug – often unreliable.
    Behavior is hard to predict – tuning is done by guessing.

OS Components



Functions of OS

- Operating systems have three general functions:

(1) Coordinator:
- Allow things to work together efficiently and fairly.

(2) Illusion generator:
- Exports cleaner, higher – level interface to hardware.

(3) Standard library:
- Provide standard facilities that everyone needs.

Coordination (1)

- Make many things work well together.

(1) Concurrency: The notion of a process.
- Several users working at the same time.
- One user doing many things at the same time.

(2) I/O devices: I/O devices run concurrently with the CPU
- Devices interrupts the CPU when done.
- Interrupt processing complicates the OS.

Coordination (2)

(3) Memory: Each process needs some memory to execute.

- OS must coordinate the memory usage.
- Swap information between memory and disk.

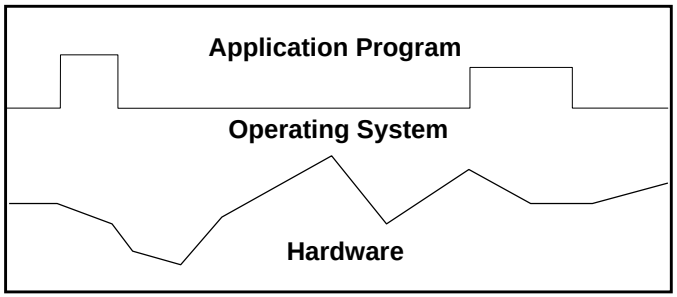
(4) Files: Each user owns a collection of files.

- OS must coordinate how space is allocated.
- Control shared accesses to files.

(5) Network: Allow groups of computers to work together.

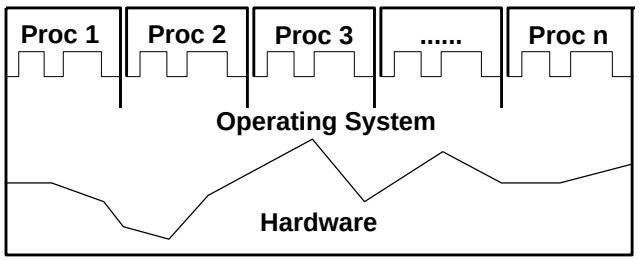
Illusion Generator (1)

OS presents an illusion: “Cleaner abstraction”



Illusion Generator (2)

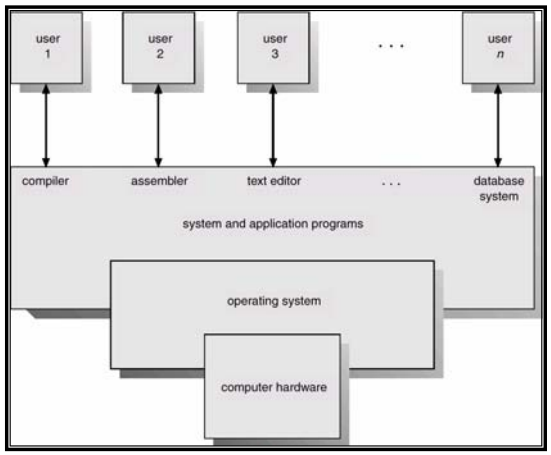
OS presents an illusion: “Multiple processors”



Illusion Generator (3)

- Examples that work:
  - Timesharing, virtual memory.
- Sometimes the illusions fail:
  - You can’t fake what you don’t get.

System Components



Mainframe Systems

- Reduce setup time by batching similar jobs
- Automatic job sequencing – automatically transfers control from one job to another. First rudimentary operating system.
- Resident monitor
  - initial control in monitor
  - control transfers to job
  - when job completes control transfers pack to monitor

Time Sharing Systems

- The CPU is multiplexed among several jobs that are kept in memory and on disk (the CPU is allocated to a job only if the job is in memory).
- A job swapped in and out of memory to the disk.
- On-line communication between the user and the system is provided. When the operating system finishes the execution of one command, it seeks the next “control statement” from the user’s keyboard.
- On-line system must be available for users to access data and code.

Desktop Systems

- *Personal computers* – computer system dedicated to a single user.
- I/O devices – keyboards, mice, display screens, small printers.
- User convenience and responsiveness.
- Can adopt technology developed for larger operating system. Often individuals have sole use of computer and do not need advanced CPU utilization of protection features.
- May run several different types of operating systems (Windows, MacOS, UNIX, Linux)

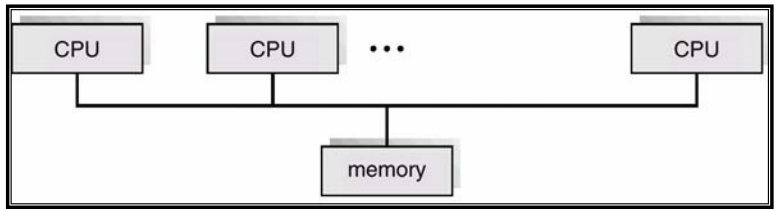
Parallel Systems (1)

- Multiprocessor systems with more than one CPU in close communication.
- *Tightly coupled system* – processors share memory and a clock. Communication usually takes place through the shared memory.
- Advantages of parallel system:
  - Increased *throughput*
  - Economical
  - Increased reliability
    - graceful degradation
    - fail-soft systems

Parallel Systems (2)

- *Symmetric multiprocessing (SMP)*
  - Each processor runs and identical copy of the operating system.
  - Many processes can run at once without performance deterioration.
  - Most modern operating systems support SMP
- *Asymmetric multiprocessing*
  - Each processor is assigned a specific task. Master processor schedules and allocates work to slave processors.
  - More common in extremely large systems

Symmetric Multiprocessor



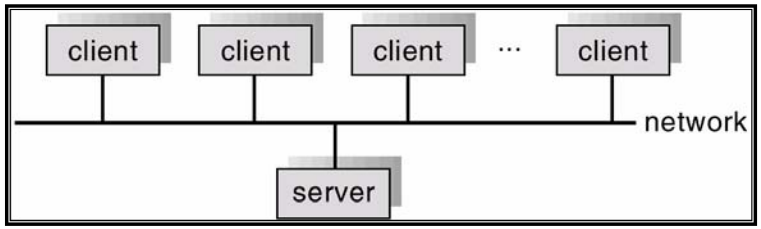
Distributed Systems (1)

- Distribute the computation among several physical processors.
- *Loosely coupled system* – each processor has its own local memory. Processors communicate with one another through various communications lines, such as high-speed buses or telephone lines.
- Advantages of distributed systems.
  - Resources Sharing
  - Computation speed up – load sharing
  - Reliability
  - Communications

Distributed Systems (2)

- Requires networking infrastructure.
- Local area networks (LAN) or Wide area networks (WAN)
- May be either client-server or peer-to-peer systems.

General Structure of Client-Server



Clustered Systems

- Clustering allows two or more systems to share storage.
- Provides high reliability.
- *Asymmetric clustering*: one server runs the application while other servers standby.
- *Symmetric clustering*: all N hosts are running the application.

Real-Time Systems (1)

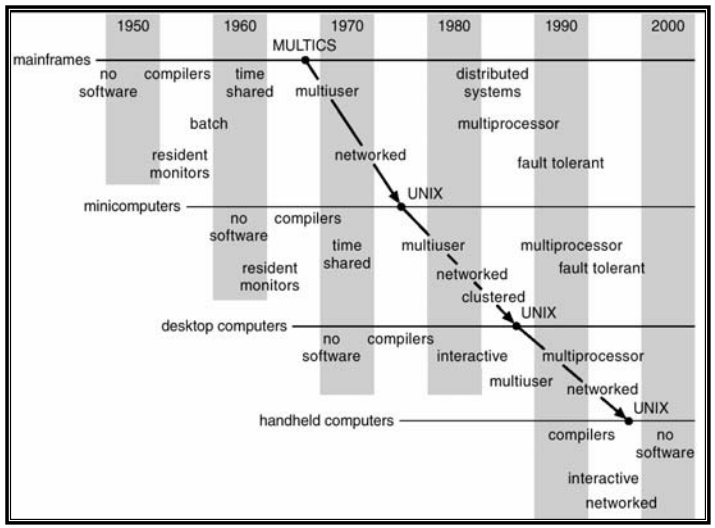
- Often used as a control device in a dedicated application such as controlling scientific experiments, medical imaging systems, industrial control systems, and some display systems.
- Well-defined fixed-time constraints.
- Real-Time systems may be either *hard* or *soft* real-time.

Real-Time Systems (2)

- Hard real-time:
  - Secondary storage limited or absent, data stored in short term memory, or read-only memory (ROM)
  - Conflicts with time-sharing systems, not supported by general-purpose operating systems.
- Soft real-time
  - Limited utility in industrial control of robotics
  - Useful in applications (multimedia, virtual reality) requiring advanced operating-system features.

Handheld Systems

- Personal Digital Assistants (PDAs)
- Cellular telephones
- Issues:
  - Limited memory
  - Slow processors
  - Small display screens.



Computing Environments

- Traditional computing
- Web-Based Computing
- Embedded Computing