ECE397A – Operating Systems

- Welcome!
- Instructor: Professor Csaba Andras Moritz
- Class webpage: http://www.ecs.umass.edu/ece/andras/courses/ECE397
- Instructors webpage: http://www.ecs.umass.edu/ece/andras
- 3 homeworks (10%), 4 projects/labs in Java/C (30%)
- 2 exams and pop quizzes (60%)
- All information will be put on the web, check the class webpage regularly.
- Acknowledgements: *material leveraged from Silberschatz, Galvin, and Gagne.*

Chapter 1: Introduction

- What is an Operating System?
- Mainframe Systems (OS390, z/OS, Linux for S/390, available on G5, G6 and Multiprise 3000 procs.)
- Desktop Systems (WinNT,Me,2000,XP, Linux, Solaris, BeOS, OS2-Warp, MAC OS X …)
- Multiprocessor Systems (FreeBSD SMP, Raw OS,..)
- Distributed Systems – Middleware (CORBA, .NET, J2EE)
- Clustered System (SGI CXFS 2.0, TeraGrid project)
- Embedded Systems
- Real -Time Systems (RTOS, lots of specialized OS)
- Handheld Systems (Pocket PC2002, PalmOS)

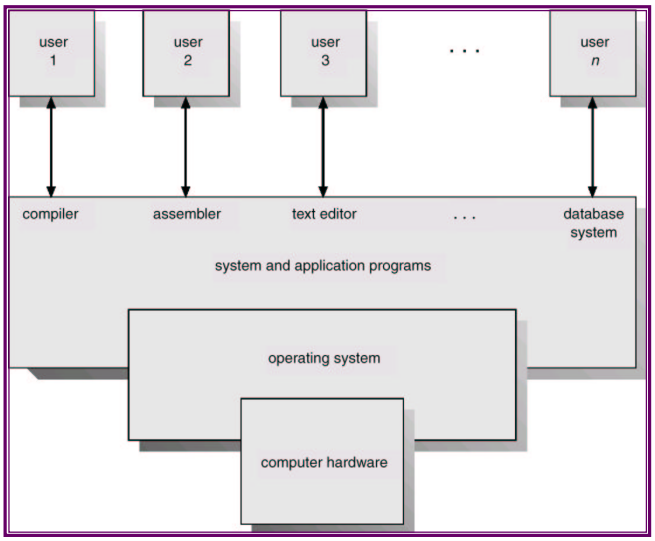
What is an Operating System?

- A software that acts as an intermediary between a user of a computer and the computer hardware.
- Operating system goals:
  - Execute user programs and make solving user problems easier.
  - Make the computer system convenient to use. (G)UI interface.
  - Manage hardware resources
  - Use the computer hardware in an efficient manner.

Computer System Components

1. Hardware – provides basic computing resources (CPU, cache, memory, disk, I/O devices).
2. Operating system – controls and coordinates the use of the hardware among the various application programs for the various users.
3. Applications programs – define the ways in which the system resources are used to solve the computing problems of the users (compilers, database systems, video games, business programs).
4. Users (people, other computers).

Abstract View of System Components



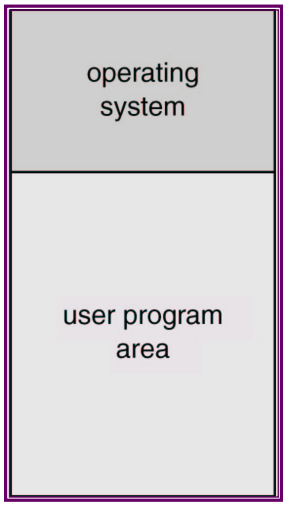
Operating System Definitions

- Resource allocator – manages and allocates resources.
  - Why do we need to manage resources?
- Control program – controls the execution of user programs and operations of I/O devices .
  - Why can't we have all IO controls implemented in the applications directly?
- Kernel – the one program running at all times (all else being application programs).
  - Why do we need a kernel?

Mainframe Systems

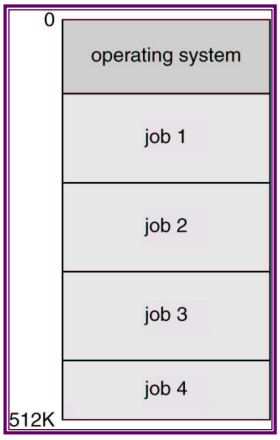
- Reduce setup time by batching similar jobs
- Automatic job sequencing – automatically transfers control from one job to another. First rudimentary operating system.
- Resident monitor
  - initial control in monitor
  - control transfers to job
  - when job completes control transfers pack to monitor

Memory Layout for a Simple Batch System



Multiprogrammed Batch Systems

Several jobs are kept in main memory at the same time, and the CPU is multiplexed among them.



OS Features Needed for Multiprogramming

- I/O routine supplied by the system.
- Memory management – the system must allocate the memory to several jobs.
- CPU scheduling – the system must choose among several jobs ready to run.
- Allocation of devices.

Time-Sharing Systems–Interactive Computing

- 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)

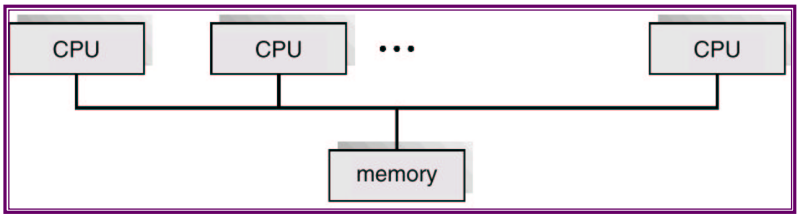
Parallel Systems

- Multiprocessor systems with more than on 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 (Cont.)

- *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 allocated work to slave processors.
  - More common in extremely large systems

Symmetric Multiprocessing Architecture



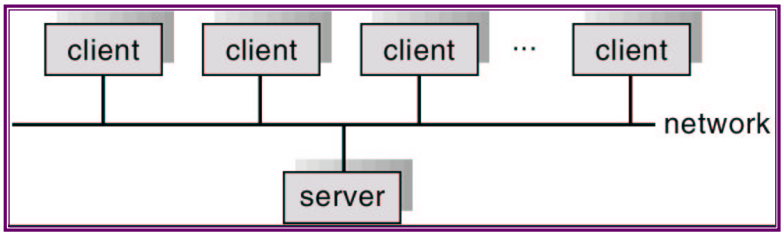
Distributed Systems

- 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 (cont)

- 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

- 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.

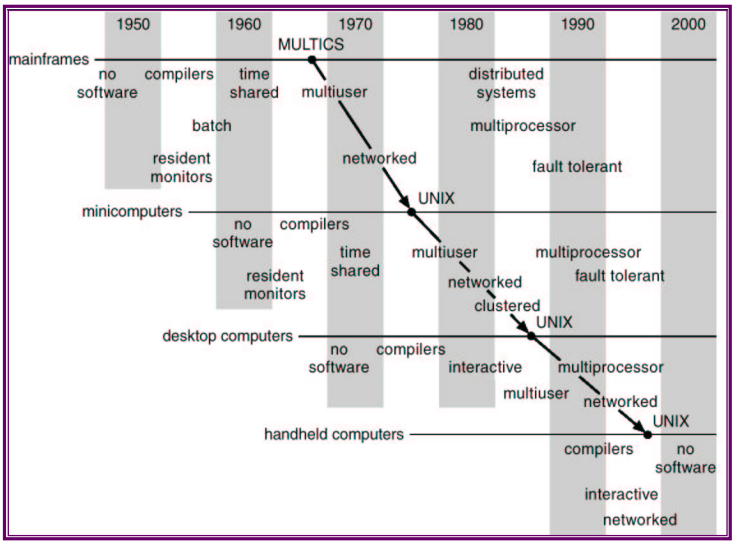
Real-Time Systems (Cont.)

- 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.

Migration of Operating-System Concepts and Features



Computing Environments

- Traditional computing
- Web-Based Computing
- Embedded Computing