

CSMC 412

Operating Systems Prof. Ashok K Agrawala

© 2004 Ashok Agrawala

Today

- Review Syllabus
 - ☞ read the warning about the size of the project
 - ☞ make sure you get the 6th edition of the book
- Program #0 Handout (posted Aug 31)
 - ☞ its due in just over one week
 - ☞ purpose is to get familiar with the simulator/compiler/debugger
- Class Grades Server
 - ☞ Grades.cs.umd.edu
- Discussion Sections
 - ☞ will focus on the project and meet only once a week (W)
 - 📖 Probably will have three times per week; you can attend the section of your choice. See web page for details.
- Reading
 - ☞ Chapter 1
 - ☞ Chapter 2

Cyclone

- Based on C
 - Low-level control over memory management, data representation, and access to the machine
- But type-safe !!
 - Rules out many hard-to-find bugs and security holes
 - ▢ Buffer overflows
 - ▢ Dangling pointers
 - These bugs have killed many projects

Cyclone in this class

- You are required to use it for project 0, to get familiar with it
- From then on it is optional
 - Could very well make your project more reliable, and prevent bugs
 - But, it's a research compiler, and it will have bugs itself, and may have cryptic error messages.
 - ▢ If a major problem in the compiler inhibits a good-faith attempt to complete the project, you will not be penalized.

Cyclone in this class

- Completing at least one project using Cyclone (from project 1 to 6) will net you some extra credit
 - ☞ You must include a write up of your experience, including
 - ▣ what features you used,
 - ▣ what difficulties you ran into and
 - ▣ suggestions for improvements
 - ☞ See <http://www.cs.umd.edu/projects/cyclone/>

Class Grades Server

<http://grades.cs.umd.edu>

- Get your LinuxLab account from here
 - ☞ CS computing cluster. Projects must work and be submitted on these machines
- Complete grade information
- Interface for requesting regrades on exams and projects

Catalog Description

- A hands-on introduction to operating systems, including topics in: multiprogramming, communication and synchronization, memory management, IO subsystems, and resource scheduling policies. The laboratory component consists of constructing a small kernel, including functions for device IO, multi-tasking, and memory management.

Prerequisites

- CMSC 311
- CMSC 330

Topics Covered

- • Introduction to Operating Systems (1 week)
- • The Cyclone Programming Language (0.5 weeks)
- • Concurrent Processes (2 weeks)
- • Kernel implementation techniques (1 week)
- • CPU scheduling (1 week)
- • Memory Management (2 weeks)
- • File and I/O Systems (2 weeks)
- • Security and Protection (1 week)
- • Networking and Distributed Systems (2 weeks)
- • Objects and Naming (1 week)
- • Window and Display Services (0.5 weeks)

Text

- Required
 - *Operating System Concepts* 6th Edition, Siberschatz, Galvin and Gagne, John Wiley 2003.

Programming Projects:

- Understanding operating system concepts is a hands-on activity. This class will include several substantial programming projects that will require students to read and understand provided code, write new modules, and debug the resulting system. The programming assignments will be time consuming and students taking this class should plan their class schedules accordingly.
- The instructor reserves the right to fail, regardless of overall numeric score, students who do not submit a good faith attempt to complete all programming assignments.
- You will have the option to use Cyclone language
 - Next lecture will be an introduction to Cyclone

Grading

- Final Exam 30%
- Midterms (2 each worth 15%) 30%
- Programming Assignments 40%
-
- *Exams:*
- Midterm #1 -
- Midterm #2 -
- Final – Dec 16, 10:30 – 12:30

Introduction

- What is an Operating System?
- Mainframe Systems
- Desktop Systems
- Multiprocessor Systems
- Distributed Systems
- Clustered System
- Real -Time Systems
- Handheld Systems
- Computing Environments

Hardware Support Required

- Many of the functions carried out by the operating system require explicit hardware support
 - ☞ Privileged Instructions
 - ☞ Processor State
 - ☞ System Call
 - ☞ Memory Protection
 - ☞ ...

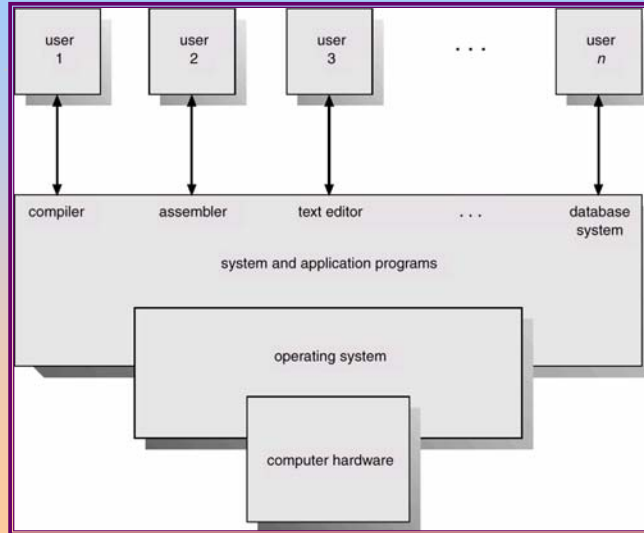
What is an Operating System?

- A program 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.
- Use the computer hardware in an efficient manner.

Computer System Components

1. Hardware – provides basic computing resources (CPU, memory, 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, machines, other computers).

Abstract View of System Components



Operating System Concepts

1.17

Operating System Definitions

- Resource allocator – manages and allocates resources.
- Control program – controls the execution of user programs and operations of I/O devices .
- Kernel – the one program running at all times (all else being application programs).

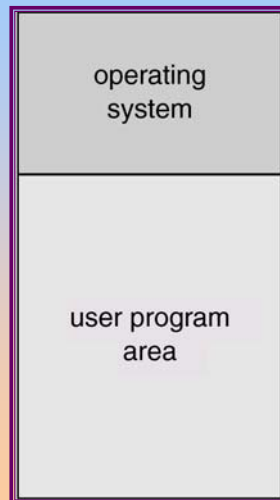
Operating System Concepts

1.18

Mainframe Systems

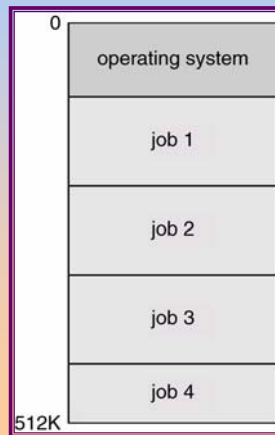
- Reduce setup time by batching similar jobs
- Automatic job sequencing – automatically transfers control from one job to another. First rudimentary operating system.
- Resident monitor
 - initial control in monitor
 - control transfers to job
 - when job completes control transfers back to monitor

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.



Operating System Concepts

1.21

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.

Operating System Concepts

1.22

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)

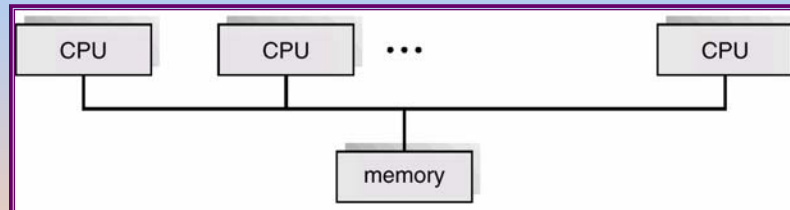
Parallel Systems

- Multiprocessor systems with more than 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 (Cont.)

- *Symmetric multiprocessing (SMP)*
 - ☞ Each processor runs an 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 Multiprocessing Architecture



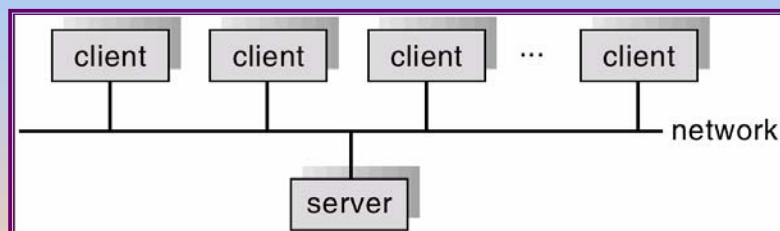
Distributed Systems

- 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.

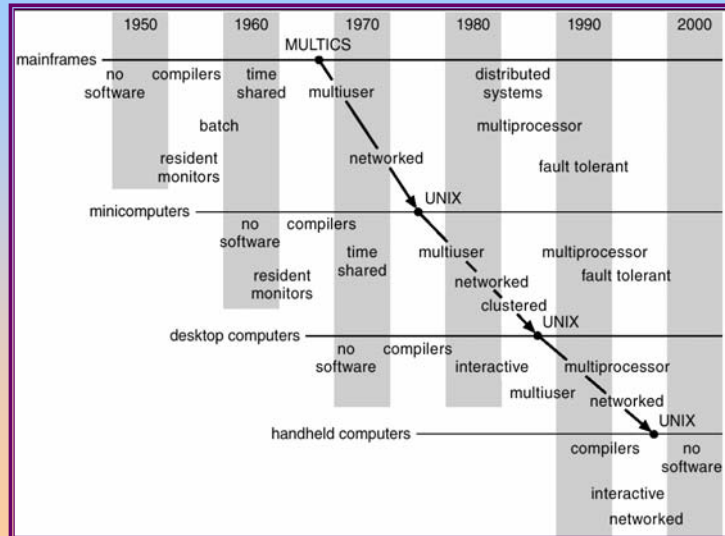
Real-Time Systems (Cont.)

- **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