1

CSMC 412

Operating Systems
Prof. Ashok K Agrawala

Set 1

Spring 2001

2

Today

- Review Syllabus
  - read the warning about the size of the project
  - make sure you get the 5th edition of the book
- Program #0 Handout
  - its due in just over one week
  - purpose is to get familiar with the compiler/debugger
- Discussion Sections
  - will focus on the project and meet only once a week (W)
- Reading
  - Chapter 1
  - Chapter 2

Spring 2001

3

Catalog Description

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

Spring 2001

4

Objective

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

Spring 2001

5

Prerequisites

- CMSC 311
- CMSC 330

Spring 2001

6

Topics Covered

- Introduction to Operating Systems (1 week)
- 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 (1 week)

Spring 2001

Text

- Required
  - *Applied Operating System Concepts* 1st Edition, Siberschatz, Galvin and Gagne, John Wiley 2000.

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.

Grading

- Final Exam 30%
- Midterms (2 each worth 15%) 30%
- Programming Assignments 40%
-
- *Exams:*
- Midterm #1 - March 9(tentative) in class
- Midterm #2 - April 20(tentative) in class
- Final - ???

What is an Operating System?

- Resource Manager
  - Resources include: CPU, memory, disk, network
  - OS allocates and de-allocates these resources
- Virtual Machine
  - provides an abstraction of a larger (or just different machine)
  - Examples:
    - » Virtual memory - looks like more memory
    - » Java - pseudo machine that looks like a stack machine
    - » IBM VM - a complete virtual machine (can boot multiple copies of an OS on it)
- Multiplexor
  - allows sharing of resources and protection
  - motivation is cost: consider a $40M supercomputer

What is an OS (cont)?

- Provider of Services
  - includes most of the things in the above definition
  - provide “common” subroutines for the programmer
    - » windowing systems
    - » memory management
- The software that is always loaded/running
  - generally refers to the Os kernel.
    - » small protected piece of software
- All of these definitions are correct
  - but not all operating have all of these features

Closely Related to an Operating System

- Hardware
  - OS is managing hardware resources so needs to know about the ugly details of the hardware
    - » interrupt vectors
    - » page tables
    - » I/O registers
  - some features can be implemented either in hardware or the OS
    - » Example: page tables on MIPS
- Languages
  - can you write an OS in any language?
    - » No: need to be able to explicitly layout data structures to match hardware

13

OS Related Topics (cont)

- Language Runtime systems
  - memory management requirements
    - » explicit heap management
    - » garbage collection
    - » stack layout
  - concurrency and synchronization
  - calling convention (how are parameters passed)
- Data Structure and Algorithms
  - efficient access to information in an OS
    - » for most things need linear time and space
    - » for many things want log or constant time

14

Usability Goals

- Robustness
  - accept all valid input
  - detect and gracefully handle all invalid input
  - should not be possible to crash the OS
- Consistency
  - same operation should mean the same thing
    - » read from a file or a network should look the same
    - » a "-" flag should be the same in different commands
  - conventions
    - » define the convention
    - » follow the convention when adding new items

15

Usability Goals (cont)

- Proportionality
  - simple, common cases are easy and fast
    - » good default values
  - complex, rare cases are possible but more complex and slower
    - » "rm *" should give a warning
    - » formatting the disk should not be on the desktop next to the trash can

16

Cost Goals

- Good Algorithms
  - time/space tradeoff are important
  - use special hardware where needed
    - » smart disk controllers, memory protection
- Low maintenance cost
  - should not require constant attention
- Maintainability
  - most of cost in OS is in maintenance so make it easy to maintain the software base

17

Adaptability Goals

- Tailored to the environment
  - server vs. workstation
  - multi-media vs. data entry
- Changes over time
  - added memory
  - new devices
- Extensible
  - third parties can add new features
    - » database vendors often need custom features
  - end customers can extend the system
    - » new devices
    - » new policies