LECTURE - I

INTRODUCTION

Tevfik Koşar

Louisiana State University
January 16th, 2007

Contact Information

• Instructor: Prof. Tevfik Kosar
  - Office: 292 Coates (also 333 Johnston)
  - Phone: 578-9483
  - Email: kosar@lsu.edu
  - Web: http://www.cct.lsu.edu/~kosar
  - Office hours: Wed & Thu, 1:00pm - 2:00pm
    (Or anytime by appointment)

• Teaching Assistant: Anindya Poddar
  - Email: anindya@csc.lsu.edu
  - Anytime by appointment
Logistics

- Course web page: [http://www.cct.lsu.edu/~kosar/csc4103](http://www.cct.lsu.edu/~kosar/csc4103)
  - All lecture notes will be available online
  - Of course also homework assignments, projects and other important course information

- Course mailing list: [CS4103@cct.lsu.edu](mailto:CS4103@cct.lsu.edu)
  - Important course announcements including projects, homework assignments, and exams will be sent to this mailing list
  - Provide me with your active email address to be added to the class mailing list

Textbooks
Grading

- The end-of-semester grades will be composed of:
  - Pop Quizzes: 10% (3-5)
  - Homework: 15% (5)
  - Projects: 20% (2)
  - Midterm: 25% (1)
  - Final: 30% (1)

You are expected to attend the classes and actively contribute via asking and/or answering questions.

Teaching Philosophy

- Goal:
  - For instructor: teaching the material
  - For student: learning and applying the material in real life
- Grades are of second degree importance
- Do not memorize, understand the material
- You are only responsible from material
  - Covered in the class
  - Part of projects or homework assignments
**Rules**

- Late submission of projects/homeworks will be penalized. (unless otherwise stated!)
- No computers/laptops will be allowed in regular class as well as exam.
- Academic dishonesty will be treated seriously.

**What Expect to Learn?**

- Basic Concepts of Operating Systems
- Operation, Resource Utilization, Management
- Processes, Threads and Concurrency
- CPU and I/O Scheduling
- Memory and Storage Management
- File Systems
- Synchronization and Deadlocks
- Protection and Security
- Distributed OS and Related Issues
- Special Purpose Systems (Real Time & Multimedia)
Introduction

What is an Operating System?

- A type of dinosaur.
- A program that manages the computer hardware.
- An intermediary between the computer user and the computer hardware.
- Manages hardware and software resources of a computer.
Operating System Goals

• From the user perspective
  - Execute user programs and make solving user problems easier
  - Make the computer system convenient to use

• From the System Perspective
  - Manage the resources
  - Use the computer hardware in an efficient manner

Four Components of a Computer System

1. User
2. Compiler
3. System and Application Programs
4. Operating System
5. Computer Hardware
6. Database System
Loading the OS

- **bootstrap program** is loaded at power-up or reboot
  - Typically stored in ROM or EEPROM, generally known as **firmware**
  - Initializes all aspects of system
  - Loads operating system kernel and starts execution

Computer System Organization

- Computer-system operation
  - One or more CPUs, device controllers connect through common bus providing access to shared memory
  - Concurrent execution of CPUs and devices competing for memory cycles
**Computer-System Operation**

- I/O devices and the CPU can execute concurrently.
- Each device controller is in charge of a particular device type.
- Each device controller has a local buffer.
- CPU moves data from/to main memory to/from local buffers
  - If no CPU involved → DMA
- I/O is from the device to local buffer of controller.
- Device controller informs CPU that it has finished its operation by causing an *interrupt*.

**Common Functions of Interrupts**

- Interrupt transfers control to the interrupt service routine generally, through the *interrupt vector*, which contains the addresses of all the service routines (*interrupt handlers*).
- Interrupt architecture must save the address of the interrupted instruction. (also save state of CPU, eg. registers, PC)
- Incoming interrupts are *disabled* while another interrupt is being processed to prevent a *lost interrupt*.
- A *trap* is a software-generated interrupt caused either by an error or a user request.
- An operating system is *interrupt driven*. 
I/O Structure

- After I/O starts, control returns to user program only upon I/O completion ➔ **synchronous**
  - Wait instruction idles the CPU until the next interrupt
  - Wait loop (contention for memory access).
  - At most one I/O request is outstanding at a time, no simultaneous I/O processing.

- After I/O starts, control returns to user program without waiting for I/O completion ➔ **asynchronous**
  - *System call* - request to the operating system to allow user to wait for I/O completion.
  - *Device-status table* contains entry for each I/O device indicating its type, address, and state.
  - Operating system indexes into I/O device table to determine device status and to modify table entry to include interrupt.
Two I/O Methods

Synchronous

- User
  - Requesting process waiting
  - Device driver
  - Interrupt handler
  - Hardware
  - Data transfer

Asynchronous

- User
  - Requesting process
  - Device driver
  - Interrupt handler
  - Hardware
  - Data transfer

Reading Assignment

- Read chapter 1 from Silberschatz.
Acknowledgements