UMassAmherst

# Operating Systems
# CMPSCI 377
# Introduction

**Prashant Shenoy**

**University of Massachusetts Amherst**



## Today's Class

- Organizational meeting
  - Course organization & outline
  - Policies
  - Prerequisites & course sign-up
- Intro to (operating) systems

# Organizational Information

- Course web page
  - Visit www.cs.umass.edu/~shenoy/courses/377
- Contact info
  - shenoy@cs.umass.edu
- TA:
  - Shashi Singh (shashi@cs.umass.edu)
- Discussion section



# Prerequisites and Syllabus

- CMPSCI 187 (Data structures) and CMPSCI 201 (Architecture)
- Textbook: Operating System Concepts 1ed (Silberschatz,Galvin,Gagne) 7$^{th}$ or 8$^{th}$ ed
- Course requirements
  - 4-5 homeworks plus few in-class assignments (20%)
  - 4 programming assignments (40%)
  - 2 exams (40%)
- Strict late policies and policies on cheating

# Course Organization: Misc

- Accounts in the Ed-lab: 30+ Linux-based PCs
- Discussion section to help you with Lab assignments
- Office hours:
  - Instructor: TuThu: 2:30-3:30, CS 336 or by appt
  - TA: Shashi Singh
  - Off hrs and location: Mon, Fri 1 -2, LGRT 220



# Course Requirements

- Note: Percentages are subject to revision.
- Programming projects: 40%
  - Strict late policy!
  - Autograder
- In-class exams: 40%

# Projects and Autograder

- 4 Projects
- Projects are autograded
  - Provides fast feedback
  - 3 Bonus submissions per-project
  - (3 late days across all projects)
- Computer Lab Accounts



# Plagiarism

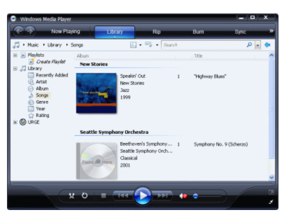
- Cheaters will be found & executed
  - We use sophisticated detection software
- Sign form this class
- Cheating includes:
  - "Borrowing" code from someone
    - This includes reading previous solutions
  - Giving code to someone (even next year)
  - Copying code from anyone (including the net)
  - Hiring someone to write your code
  - Submitting someone else's code as your own
  - Looking at anyone else's code

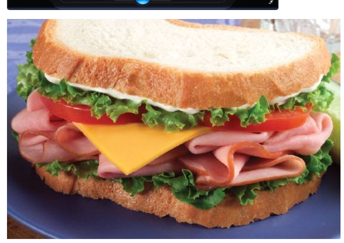
# What's An Operating System?

- Definition has changed over years
  - Originally, very bare bones
  - Now, includes more and more



# What's an OS? Bill Gates says…

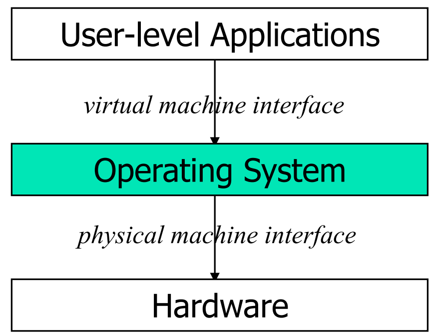





"even
a ham sandwich"
(Steve B.)

# OS: Traditional View

- Interface between user and architecture
  – Hides architectural details
- Implements virtual machine:
  – Easier to program than raw hardware
- Illusionist
  – Bigger, faster, reliable
- Government
  – Divides resources
  – "Taxes" = overhead






# New Developments in OS

- Operating systems: active field of research
  – Demands on OS's growing
  – New application spaces (Web, Grid)
  – Rapidly evolving hardware
- Advent of open-source operating systems
  – Linux etc.
  – You can contribute to and develop OS's!
  – Excellent research platform

# Build Large Computer Systems

- Goals: Fast, reliable, large scale
- To build these systems, you need to know
  - Each computer:
    - Architectural details that matter
    - C and C++ (nitty gritty & more)
    - Memory management & locality
    - Concurrency & scheduling
    - Disks, network, file systems
  - Across cluster:
    - Server architectures
    - Distributed computing, file systems



# History of Operating Systems

- And now, for some historical context
  - From mainframes to web-based systems in nine slides

# 1. Single-User Computers

- Hardware: expensive; humans: cheap
- One user at a time on console
  - Interacting with as program runs
- Computer executes one function at a time
  - No overlap: computation & I/O
- User must be at console to debug

- Multiple users = inefficient use of machine



# 2. Batch Processing

- Execute multiple "jobs" in batch:
  - Load program
  - Run
  - Print results, dump machine state
  - Repeat
- Users submit jobs (on cards or tape)
- Human schedules jobs
- Operating system loads & runs jobs

- More efficient use of machine

# 3. Overlap I/O and Computation

- Before: machine waits for I/O to complete
- New approach:
  - Allow CPU to execute while waiting
  - Add buffering
    - Data fills "buffer" and then output
  - and interrupt handling
    - I/O events trigger a signal ("interrupt")
- More efficient use of machine
  - still one job at a time



# 4. Multiprogramming

- Several programs to run simultaneously
  - Run one job until I/O
  - Run another job, etc.
- OS manages interactions
  - Which jobs to run (schedule)
  - Protects program's memory from others
  - Decides which to resume when CPU available

# OS Complexity

- Increased functionality & complexity
- First OS failures
  - Multics (GE & MIT):
    announced 1963, released 1969
  - OS/360 released with 1000 known bugs
- Need to treat OS design scientifically
- Managing complexity becomes key to…



# The Renaissance (1970's)

- Hardware: cheap; humans: expensive
- Users share system via terminals
- The UNIX era
  - Multics:
    - army of programmers, six years
  - UNIX:
    - three guys, two years
    - "Shell": composable commands
    - No distinction between programs & data
- But: response time & thrashing

# Industrial Revolution (1980's)

- Hardware very cheap;
- humans expensive
- Widespread use of PCs
  - IBM PC: 1981, Macintosh: 1984
- Simple OS (DOS, MacOS)
  - No multiprogramming, concurrency, memory protection, virtual memory, …
  - Later: networking, file-sharing, remote printing…
  - GUI added to OS ("WIMP")



# The Modern Era (1990's-now)

- Hardware cheap; processing demands increasing
- "Real" operating systems on PC's
  - NT (1991); Mac OS X; Linux
- Different modalities:
  - Real-time: Strict or loose deadlines
  - Sensor/Embedded: Many small computers
  - Parallel: Multiple processors, one machine
  - Distributed: Multiple networked processors
    - Think P2P, the Web, Google

# Architectural Trends

- Big Changes
  - In 50 years, almost every computer component now
    9 orders of magnitude faster, larger, cheaper

| *examples* | **1983** | **1999** |
|---|---|---|
| MIPS | 0.5 | 500 |
| cost/MIP | $100,000 | $500 |
| memory | 1 MB | 1 GB |
| network | 10 Mbit/s | 1 Gb/s |
| disk | 1 GB | 1 Tbyte |



# History Lesson

This degree of change has no counterpart in any other area of business.

**Examples:**

- Transportation -- over the last 200 years, we have gone from horseback (10 miles/hour) to the Concorde (1000 miles/hour) - 2 orders of magnitude.
- Communication -- at the invention of the telephone (voice), TV (video) and fax (text \& pictures), communication went from the speed of transportation to nearly the speed of light - 7 orders of magnitude.