CS-3013 — Operating Systems

Professor Hugh C. Lauer

CS-3013 — Operating Systems

Slides include copyright materials *Modern Operating Systems*, 3rd ed., by Andrew Tanenbaum and from *Operating System Concepts*, 7th and 8th ed., by Silbershatz, Galvin, & Gagne

Outline for Today

- **Details and logistics of this course**
- **Discussion**
  - What is an Operating System?
  - What every student should know about them
- **Project Assignment**
    - Virtual Machines
- **Introduction to Concurrency**

This Course

- **Two 2-hour classes per week**
  - 8:00 – 10:00 AM, Tuesdays and Fridays
  - January 17 – March 4, 2014
- **Very similar to first half of CS-502**
  - First graduate course in Operating Systems
- **Concentrated reading and project work**

Concentrated reading and project work

- **A number of students report that this is their hardest CS course at WPI (so far).**
  - The programming is demanding (even though not many lines of code)
  - *C* language is unforgiving (costing you many hours of frustration)
  - If you wait till a day or two before an assignment is due, you have very little chance to complete it

This Course

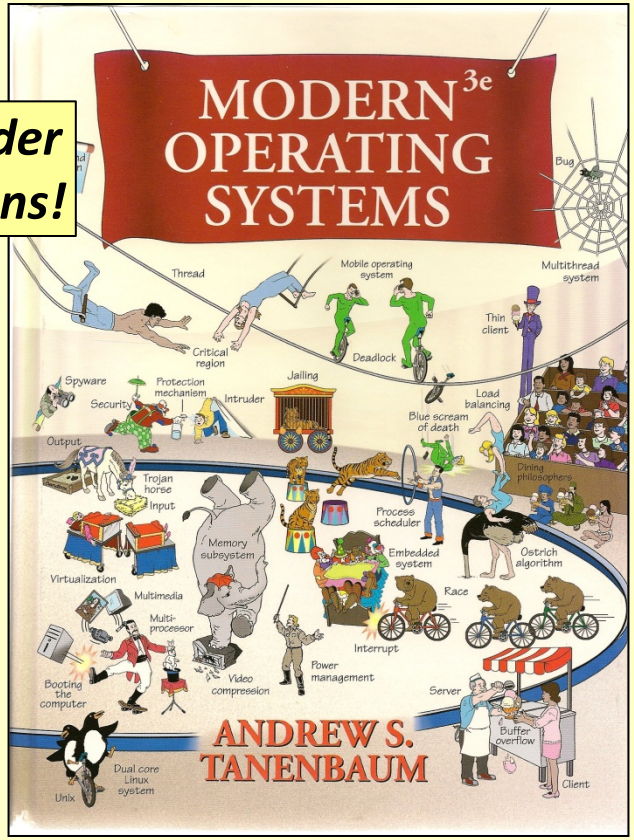
- **Two 2-hour classes per week**
  - 8:00 – 10:00 AM, Tuesdays and Fridays
  - January 17 – March 4, 2014
- **Very similar to first half of CS-502**
  - First graduate course in Operating Systems
- **Concentrated reading and project work**
- **Course web site:–**
  - *http://web.cs.wpi.edu/~cs3013/c14/*

Parts of web site are protected in order to comply with copyright regulations

Textbooks

Modern Operating Systems, 3rd edition, by Andrew S. Tanenbaum, Pearson Prentice-Hall, 2008

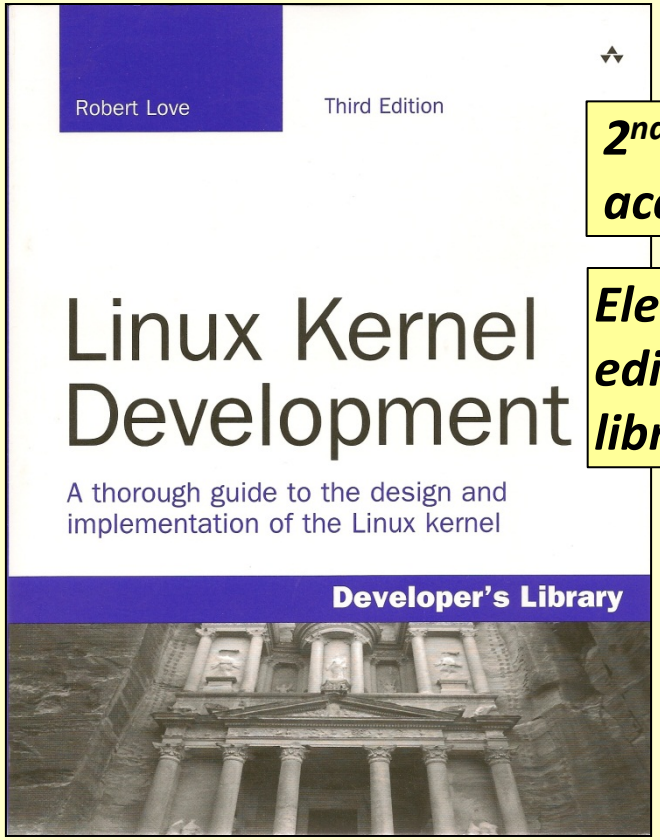
No older editions!



Linux Kernel Development, 3rd edition, by Robert Love, Novell Press, 2010

2nd edition acceptable

Electronic edition in library



Supplemental Text

- **Daniel Bovet & Marco Cesari, *Understanding the Linux Kernel*, 3rd edition, O'Reilly Media, 2006**
  - Denser and more encyclopedic
  - Aimed at professional Linux kernel developers
  - Two copies on reserve in Gordon Library

Recommended Background

- **Computer Programming:–**
  - *C*/*C*++ programming
    - Especially a low-level language such as *C*
    - *CS-2301* or *CS-2303*
  - Data structures
    - Linked lists
  - Computer Organization and Assembly Language
    - CS-2011
  - Unix/Linux user experience

C Programming Language

- **Almost *all* difficulties that students have in this course go back to the *C* language**
  - Insufficient preparation and understanding
  - Especially modules and include files
  - **`printf()`** formats of data types
  - **`static`** versus global variables

  - Lack of experience with debugger

Recommended Background

- **Computer Programming:–**
  - *C*/*C*++ programming
    - Especially a low-level language such as *C*
  - Data structures
  - Computer Organization — CS-2011
  - Unix/Linux user experience

- ***Reading assignment***
  - Tanenbaum Chapter 1
  - *Linux Kernel Development*, Chapter 1 & pp. 337–348
  - Quiz on Friday, January 24!

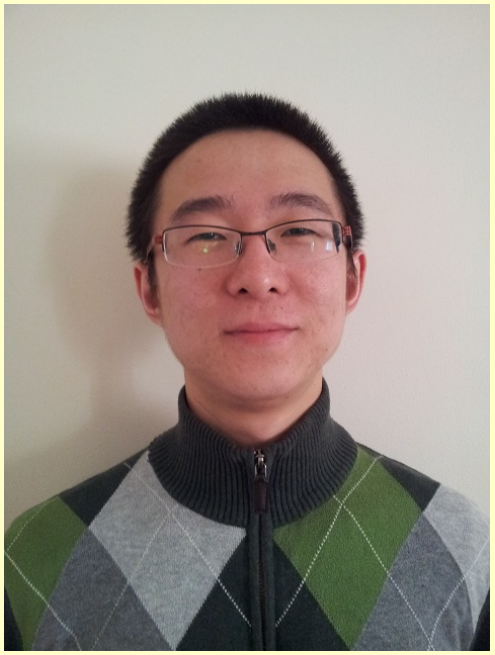
Schedule & Logistics

- **Schedule**
  - Fuller Labs 320
  - 8:00 – 10:00 AM, Tuesdays and Fridays
  - One 5 minute break
  - 14 classes
- **~ 4 Programming Projects**
  - 1-3 weeks each
- **Weekly quizzes**
- **Mobile Phones, pagers and other similar devices SILENT during class**
- **Prof's Office Hours**
  - By appointment, *or*
  - TBD
  - Office:– Fuller 144
- **Contact**
  - **lauer in the domain cs.wpi.edu**
- **Course e-mail list**
  - **cs3013-all (in the same domain)**
  - **CS-3013-staff (Prof & TAs)**

Teaching Assistants

- **Jia "Joe" Wang**
  - Office hours: Mon 1:00 – 3:00 PM
    Wed 3:00 PM – 5:00 PM
- **Angel Trifonov**
  - Office hours:
    Mon: 3:00 – 5:00
    Thur: 1:00 – 3:00

Office hours in Fuller A22 (or Zoo Lab)



Jia Wang

Angel Trifonov

Lecture Capture

- Lectures (voice and slides) will be captured automatically
- Can be viewed at

  TBD
- Link is also on course web-site

Please remind me *every class* to turn on microphone!

Weekly Quizzes

Testing portion of grade based on best four of first five quizzes plus last quiz

- **One quiz each week!**

Friday, Jan 24	Friday, Feb 14
Friday, Jan 31	Friday, Feb 21
Friday, Feb 7	Tuesday, Mar 4

- **20-25 minutes each**
  - Except 35-45 minutes on March 4
- **No final exam!**
  - *Must take March 4 quiz to pass course*

Open book, open notes

May use electronic books

Projects

- Install Virtual Machine, build Linux kernel, add Linux Kernel Module

1. *Fork* — learn how to create and manage processes
2. Adding functionality to Linux Kernel
3. Serious multithreaded application

Option 1

4. *Kernel messaging system*

Option 2

4. *Thread-safe* malloc (tentative)

Additional project points may be available from time to time

Grading

- **Grading**
  - Quizzes ~ 40%
  - Projects ~ 40%
  - Class participation ~ 20%

- **Good-faith attempt & submission of all projects *required* to pass this course!**

WPI Academic Honesty Policy

http://www.wpi.edu/Pubs/Policies/Honesty/policy.html

More on Prerequisites

- **C programming is essential**
  - Java-only programmers will find it very challenging
- **Time required**
  - 17+ hours per week, 7 weeks total
- **Computing resources required**
  - Modern PC or Mac with > 20 gigabytes of free disk space
  - Preferably dual- or quad-core
  - Ability to install VMware Workstation, VMware Player software, or (for Mac) VMware Fusion

OR

  - Zoo Lab (with do-it-yourself storage)

Ground Rule

- There are no “stupid” questions.
- It is a waste of your time and the class’s time to proceed when you don’t understand the basic terms.
- If you don’t understand it, someone else probably doesn’t, either.

Ground Rule #2

- **Help each other!**
- **Even if a project or assignment is specified as *individual*, ask your colleagues about stuff you don't understand.**
- **It is a waste of your time try to figure out some obscure detail on your own when there are lots of resources around.**
- **When you have the answer, *write it in your own words* (or own coding style)**

Questions?

Teaching staff

Hugh C. Lauer
Adjunct Professor

Office hours:– Tue 12:00 – 2:00 PM
Fri 10:00 – 11:00 AM
Fri 3:00 – 4:00 PM

- **Ph. D. Carnegie-Mellon, 1972-73**
  - Dissertation "Correctness in Operating Systems"
- **Faculty at University of Newcastle upon Tyne, UK**
- **Approximately 30 years in industry in USA**
- **WPI since 2006**
- **21 US patents issued**
- **2 seminal contributions to Computer Science**

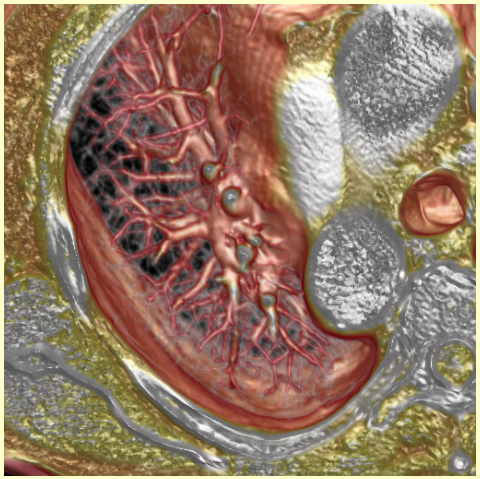
Systems Experience

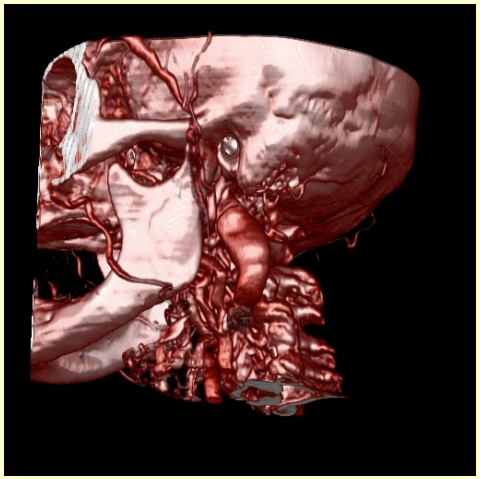
- **IBM Corporation**
- **University of Newcastle**
- **Systems Development Corporation**
- **Xerox Corporation (Palo Alto)**
- **Software Arts, Inc.**
- **Apollo Computer**
- **Eastman Kodak Company**
- **Mitsubishi Electric Research Labs (MERL)**
- **Real-Time Visualization**
  - Founded and spun out from MERL
  - Acquired by TeraRecon, Inc.
- **SensAble Technologies, Inc.**
- **Dimensions Imaging, Inc. (recent start-up, now defunct)**

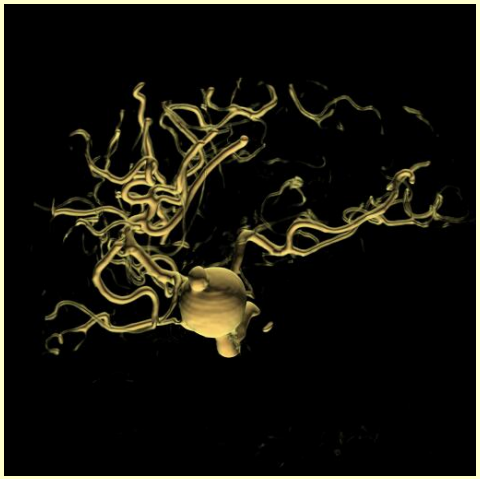
VolumePro™

- **Interactive volume rendering of 3D data such as**
  - MRI scans
  - CT scans
  - Seismic scans
- **Two generations of ASICs, boards, software**
  - VolumePro 500 – 1999
  - VolumePro 1000 – 2001
- **CTO, Chief Architect of VolumePro 1000**
  - 7.5-million gate, high-performance ASIC
  - 10^9 Phong-illuminated samples per second

Sample images from VolumePro™







Operating Systems I Have Known

- IBSYS (IBM 7090)
- OS/360 (IBM 360)
- TSS/360 (360 mod 67)
- Michigan Terminal System (MTS)
- CP/CMS & VM 370
- MULTICS (GE 645)
- Alto (Xerox PARC)
- *Pilot (Xerox STAR)*
- CP/M
- MACH
- *Apollo DOMAIN*
- Unix (System V & BSD)
- Apple Mac (v.1 – v.9)
- MS-DOS
- Windows NT, 2000, XP, Vista, 7, etc.
- various embedded systems
- Linux
- …

Outline for Today

- Details and logistics of this course
- **Discussion**
  - What is an Operating System?
  - What every student should know about them
- Project Assignment
  - Virtual Machines
- Introduction to Concurrency

Class Discussion

What is an Operating System?

(Laptops closed, please!)

What is an Operating System?

- **Characteristics**
  - Large, complex set of programs
  - Long-lived, evolutionary
  - Worked on by many people over many years
- **Functions**
  - Creates *abstractions*
  - Multiplexes concurrent activities
  - Manages resources
  - Mediates access to hardware devices
  - Provides a variety of services to users and applications
  - …

Definition – *Abstraction*

- The distillation of a complex mechanism into a simple, conceptual model
- *User* of abstraction does not need to worry about details
- *Implementer* of abstraction does not need to worry about how user will use it (within limits)

Abstraction

The most important word in this course!

What is an operating system? (continued)

- ***Abstractions:–***
  - *Processes, threads*, and concurrent computation
  - *Virtual memory.* For managing memory
  - *Files*. Persistent storage of information
  - *Sockets* & *connections* for network communication
- **Controls I/O & peripherals**
- **Implements security and accessibility**

- **See §1.1 of Tanenbaum**
- **Definition — Same as judicial definition of pornography**
- **“I cannot define it, but I sure can recognize one when I see it!”**

OS and Hardware

- **OS mediates programs' access to hardware**
  - Computation – CPU
  - Storage – volatile (memory) and persistent (disk)
  - Networks – NIC, protocols
  - I/O devices – sound cards, keyboards, displays
- **OS creates uniform abstractions**
  - Processes
  - Files
  - Sockets
  - Streams

Operating Systems – a Study of Evolution

- Simple managing of time of expensive computers
- Managing concurrency between I/O and computation

  … and users

  … and applications
- Managing memory
- Managing files, communication, GUIs
- Creating abstractions for all of the above
- … and more!

What should every student of the Computational Sciences know about Operating Systems?

- Processes, threads, concurrent computation, & how to use them
- Memory Management, fragmentation, allocation, and virtual memory
- Files, persistent storage, and what they can do for you
- Protection, authentication, and what are those silly little keys they ask us about
- Different kinds of operating systems and what they are good for

All of these are embodied in the Course Outcomes

http://web.cs.wpi.edu/~cs3013/c14/Resources/Outcomes.htm

Questions or Comments?

Outline for Today

- Details and logistics of this course
- Discussion
  - What is an Operating System?
  - What every student should know about them
- **Project Assignment**
  - Virtual Machines
- Introduction to Concurrency

More about Programming Projects

- **Project work based on *Ubuntu 12.04.3***
- **Each student will use a "virtual machine"**
  - What is a virtual machine? (§1.7.5 & 8.3 in Tanenbaum)
  - See *Setting up your Virtual Machine* (docx, pdf)

- **Build, modify, install Linux kernel on your virtual machine**
  - Debug, analyze, crash
  - Restore, try again

Using a Virtual Machine

- **Use *VMware Player* on your own PC**
- ***VMware Fusion* on your Macintosh**
- **Any other virtualization platform**
  - You are on your own for support!
- **Zoo Lab**
  - See Professor if you need it!
- **(Replaces Fossil Lab)**
  - *Free Open Source Software Lab*
  - No longer a “laboratory”

What is a *Virtual Machine?*

- **An application that simulates a computer system with enough performance and fidelity to mimic actual hardware**
- **Concept originated in 1960s, and has been used occasionally in large systems**
- **Established in mainstream of enterprise systems by *VMware* in early 2000s**
  - By 2012, a number of high quality virtualization systems are available

Virtual Machine Definitions

- ***Host* system:– The hardware and operating system that supports the virtualization application**
  - E.g., your own or company PC or Mac
  - E.g., a departmental server
- ***Guest* system:– The virtual hardware and the operating system that is being simulated**
  - E.g., *Ubuntu 12.04.3* for this course

Questions?

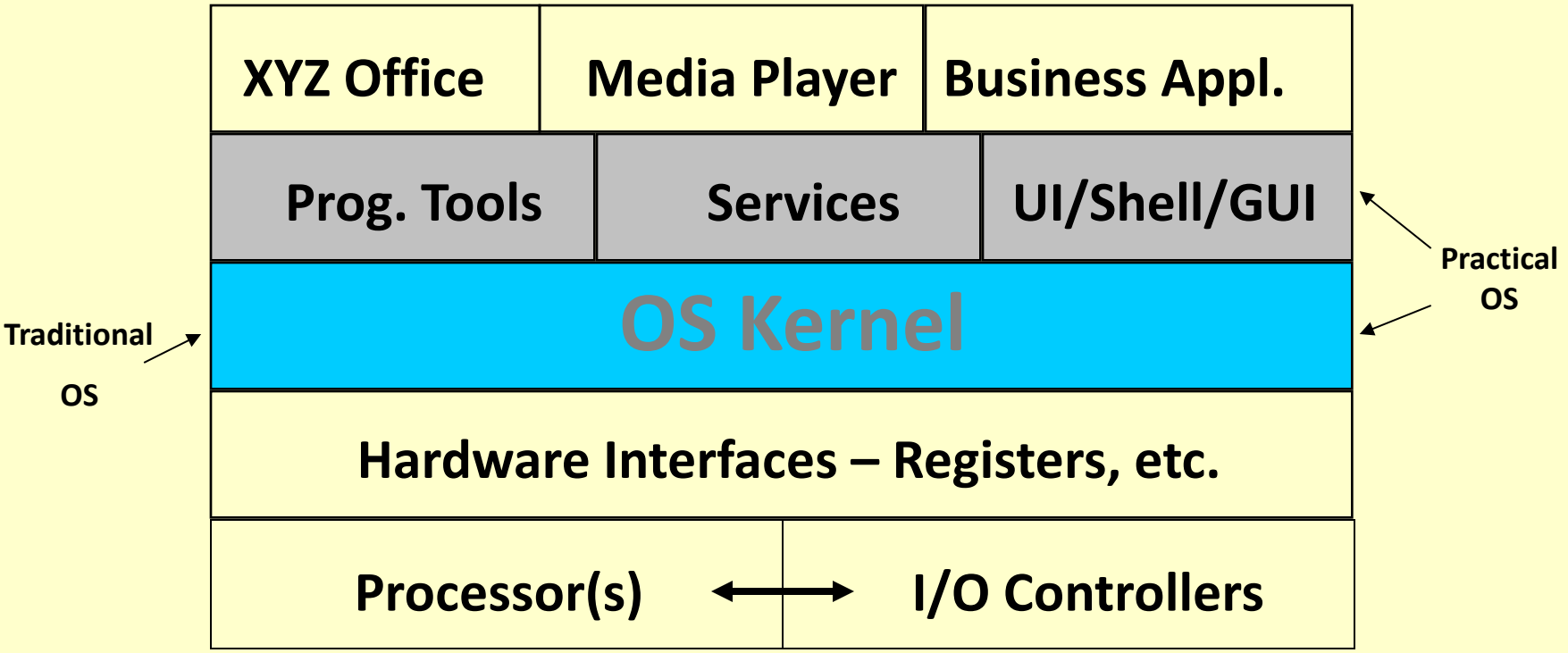
Before the Break

- **Photos**
  - To help me learn your names!
- **Survey**
  - To help me understand your background

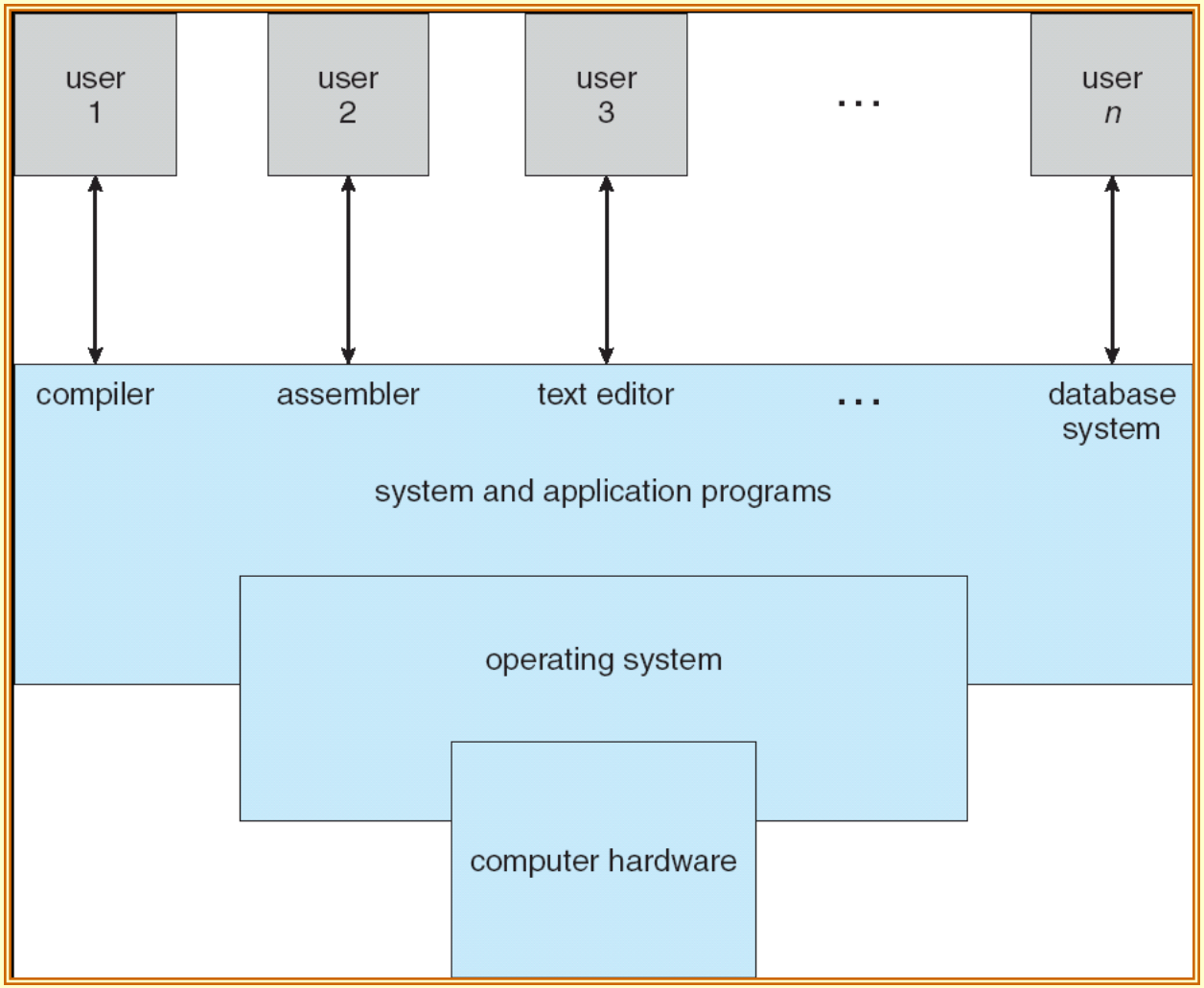
Short Break

(Not enough time to go to Campus Center for coffee!)

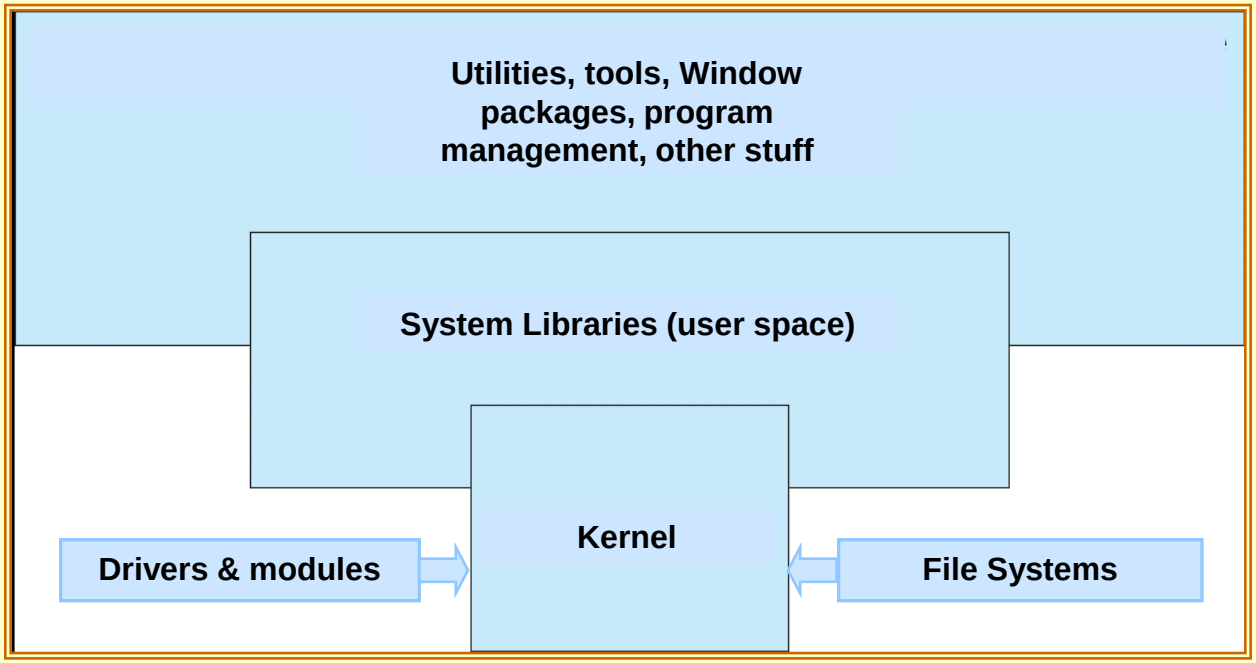
What is an Operating System?



Computer System Organization



Operating System Organization



Major Topics in Traditional OS Courses

- structure: how is the OS organized?
- sharing: how are resources shared across users?
- naming: how are resources named (by users or programs)?
- security: how is the integrity of the OS and its resources ensured?
- protection: how is one user/program protected from another?
- performance: how do we make it all go fast?
- reliability: what happens if something goes wrong – hardware or software
- extensibility: can we add new features?
- communication: how do programs exchange information
- concurrency: how are parallel activities created and controlled?
- scale: what happens as demands or resources increase?
- persistence: how do you make data last longer than program executions?
- distribution: how do multiple computers interact with each other?
- accounting: how do we keep track of resource usage, and charge for it?

- *Is user interface package part of operating system?*

Kinds of operating systems

- **See §1.4 of Tanenbaum – *Operating System Zoo***
  - Mainframe Operating Systems
  - Server Operating Systems
  - Multiprocessor Operating Systems
  - Personal Computer Operating Systems
  - Handheld Computer Operating Systems
  - Embedded Operating Systems
  - Sensor Node Operating Systems
  - Real-time Operating Systems
  - Smart-card Operating Systems
  - **…**

Two Important Operating Systems

- Linux — Chapter 10
- Windows — Chapter 11

- Spans PCs, servers, multiprocessors, etc.

OS History – Unix & Linux

- **Unix**
  - Descendant of Multics
  - First “C” version in 1973 (DEC PDP-11)
    - Timesharing for < 10 users on 32K Memory
    - Many Unix versions at Bell Labs – different goals
    - Source code made available to Universities – BSD
  - Posix (start 1981) defines standard Unix system calls
  - AT&T licensing!

OS History - Linux

- **Open Source – Linux.org**
- **First Version 1991, Linus Torvalds, 80386 processor**
  - v.01, limited devices, no networking,
  - with proper Unix process support!
- **1994, v1.0**
  - networking (Internet)
  - enhanced file system
  - many devices, dynamic kernel modules

OS History — Linux

- **1996, v2.0**
  - multiple architectures, multiple processors
  - threads, memory management ….
- **Gnome UI – introduced in 1999**
- **Recent**
  - V2.4 - 3 million lines of code
  - 7-10 million users
  - Growth by 25%/year through 2003
  - Growing use in business server market
- **Note: development convention**
  - Odd numbered minor versions “development”
  - Even numbered minor versions “stable”

Linux Versions

- **Linux 2.6.xx.yy has been the "stable" version for many years!**
- **Many revisions in *xx* and *yy!***
  - Including some rather major changes!
- **No magic in rollover from 2.x.y to 3.x.y**
  - Simply celebrating 20th anniversary of Linux development
- **Typical "social dynamic" of numbering systems!**

- **Version for this course:– 3.8.0-35**

OS History – Windows NT/2000/XP

- **Key designer – David Cutler also designed VAX/VMS**
- **1988, v1 - Win32 API – "microkernel"**
- **1990, v3.1- Server and Workstation versions**
- **1996, v4**
  - Win95 interface
  - Graphics moved into kernel
  - More NT licenses sold than all Unix combined
  - Microkernel de-emphasized

OS History – Windows NT/2000/XP

- **Windows 2000 – NT5.0**
  - Multi-user (with terminal services)
  - Professional - desktop
  - Server and Advanced Server - Client-server application servers
  - Datacenter Server - Up to 32 processors, 64 GB RAM
- **Windows XP**
  - Windows 2000 code base
  - Revised UI
  - EOL for DOS/Windows line

OS History – Windows NT/2000/XP/etc.

- **Microsoft has 80% to 90% of OS market for …**
  - Desktops, laptops, servers, data centers, etc.
- ***Wintel* – Windows + X86**
- **WinNT 4.x is 12 million lines of code**
- **Win2000 is 18 million lines of code**
- **Windows XP – approaching 10^8 lines of code**
- **Windows Vista – early 2006**
- **Windows 7 – 2010**
- **Windows 8 – 2013**

Questions?