

CS 4410 / CS4411 OPERATING SYSTEMS

Prof. Surer & Prof. Bracy
Fall 2017
Cornell University

WHAT'S THIS COURSE ABOUT?

- Ostensibly, it's about operating systems
 - architecting complex software
 - identifying needs and priorities
 - separating concerns
 - implementing artifacts with desired properties
- It's really about software design principles
 - It just so happens that OSes illustrate organizational principles and design patterns

WHAT KIND OF A COURSE IS THIS?

- Constructive, top-down
 - We start from first principles and re-derive the design of every component of a complex system

- Deconstructive, bottom-up
 - We dissect existing systems, learn what tradeoffs they make, what patterns they use

COURSE OBJECTIVE

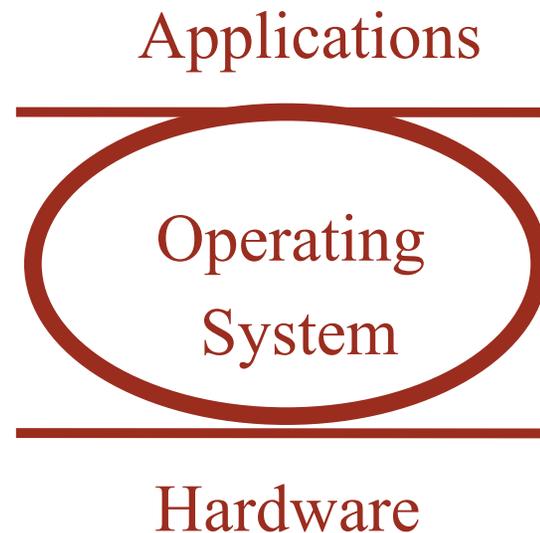
Establish a foundation for building complex programs

- Architect systems
- Identify desirable properties
- Build large systems
- Debug complex systems

Understand all layers of the software stack between hardware and applications

WHAT IS AN OPERATING SYSTEM?

- An operating system (OS) provides a virtual execution environment on top of hardware that is more convenient than the raw hardware interface
 - “All of the code you did not write”
 - Simpler
 - More reliable
 - More secure
 - More portable
 - More efficient



WHAT DO OSES DO?

- Manage physical resources
- Provide virtual resources
- Implement mechanisms and enforce policies to arbitrate access to resources
- Mediate the interaction of mutually distrusting applications
- Provide an extensible, general-purpose platform for a variety of applications

WHAT RESOURCES DO OSES CONTROL?

- Physical Resources
 - CPU, memory, disks, networks, I/O devices, ...
- Virtual Resources
 - Files, directories, sockets, names, ...

ISSUES IN OS DESIGN

- **Structure:** how is an OS organized?
- **Concurrency:** how are parallel activities created and controlled?
- **Sharing:** how are resources shared?
- **Naming:** how are resources named by users?
- **Protection:** how are distrusting parties protected from each other?
- **Security:** how to authenticate, authorize and ensure privacy?
- **Performance:** why is it so slow?

MORE ISSUES

- **Reliability:** how do we deal with failures?
- **Extensibility:** how do we add new features?
- **Communication:** how do we exchange information?
- **Scale:** what happens as demands increase?
- **Persistence:** how do we make information outlast the processes that created it?
- **Accounting:** who pays the bills and how do we control resource usage?

WHY LEARN OPERATING SYSTEMS?

- At most 1% of software developers will work on OS code
- The material taught in this course is critical not just for building Oses, but for building correct, high-performance applications (clients of Oses), as well as building reusable platforms for others
- Course will go far beyond OS design, covering all aspects of computer organization: including concurrency, synchronization, input/output, filesystems, networking, routing, distributed systems and so forth
- Engineering pride alone requires full understanding

FACT

- There has never been as exciting a time to work on systems hardware and software as now!!!
- The world is increasingly dependent on computer systems
 - Connected, networked, interlinked
- People just do not know how to build **robust systems**

BSOD

```
*** STOP: 0x0000001E (0x80000003,0x80106fc0,0x8025ea21,0xfd6829e8)
Unhandled Kernel exception c0000047 from fa8418b4 (8025ea21,fd6829e8)

Dll Base Date Stamp - Name                               Dll Base Date Stamp - Name
80100000 2be154c9 - ntoskrnl.exe                                           80400000 2bc153b8 - hal.dll
80250000 2bd49628 - nscrc710.sys                                           8025c000 2bd49668 - SCSIPOST.SYS
80267000 2bd49683 - scsidisk.sys                                           802a6000 2bd496b9 - Fastfat.sys
fa800000 2bd49666 - Floppy.SYS                                             fa810000 2bd496db - Hpfs_Rec.SYS
fa820000 2bd49676 - Sull.SYS                                               fa830000 2bd4965a - Beep.SYS
fa840000 2bd4aabb0 - i8042prt.SYS                                          fa850000 2bd5a020 - SERMCGSE.SYS
fa860000 2bd4966f - kbdcclass.SYS                                         fa870000 2bd49671 - MOOCLASS.SYS
fa880000 2bd905be - Videoprt.SYS                                          fa890000 2bd49628 - HCCI701E.SYS
fa8a0000 2bd4a4ce - Vga.SYS                                                fa8b0000 2bd496d0 - Msfs.SYS
fa8c0000 2bd496c3 - Spfs.SYS                                               fa8e0000 2bd496c9 - Htfs.SYS
fa940000 2bd496df - SDIS.SYS                                               fa930000 2bd49707 - wdlan.sys
fa970000 2bd49712 - TDI.SYS                                                fa950000 2bd5a7fb - nbf.sys
fa980000 2bd72406 - streams.sys                                           fa9b0000 2bd4975f - ubnh.sys
fa9c0000 2bd5bffd7 - usbser.sys                                             fa9d0000 2bd4971d - nethoc.sys
fa9e0000 2bd49678 - Parallel.sys                                          fa9f0000 2bd4969f - serial.SYS
faa00000 2bd49739 - mup.sys                                                faa40000 2bd4971f - SMTSTAT.SYS
faa10000 2bd6f2a2 - srv.sys                                               faa50000 2bd4971a - afd.sys
faa60000 2bd6fd80 - rdr.sys                                           faaa0000 2bd49735 - bowser.sys

Address dword dump Dll Base                               - Name
801afc20 80106fc0 80106fc0 00000000 00000000 80149905 : fa840000 - i8042prt.SYS
801afc24 80149905 80149905 ff8e6b8c 80129c2c ff8e6b94 : 8025c000 - SCSIPOST.SYS
801afc2c 80129c2c 80129c2c ff8e6b94 00000000 ff8e6b94 : 80100000 - ntoskrnl.exe
801afc34 801240f2 80124f02 ff8e6df4 ff8e6f60 ff8e6c58 : 80100000 - ntoskrnl.exe
801afc54 80124f16 80124f16 ff8e6f60 ff8e6c3c 8015ac7e : 80100000 - ntoskrnl.exe
801afc64 8015ac7e 8015ac7e ff8e6df4 ff8e6f60 ff8e6c58 : 80100000 - ntoskrnl.exe
801afc70 80129bda 80129bda 00000000 80088000 80106fc0 : 80100000 - ntoskrnl.exe

Kernel Debugger Using: COM2 (Port 0x2f8, Baud Rate 19200)
Restart and set the recovery options in the system control panel
or the /CRASHDEBUG system start option. If this message reappears,
contact your system administrator or technical support group.
```

BSOD



quakecon 2002 - mesquite, texas, USA - august 2002 - photo: yossarian holmberg (yossman@yossman.net)

BSOD

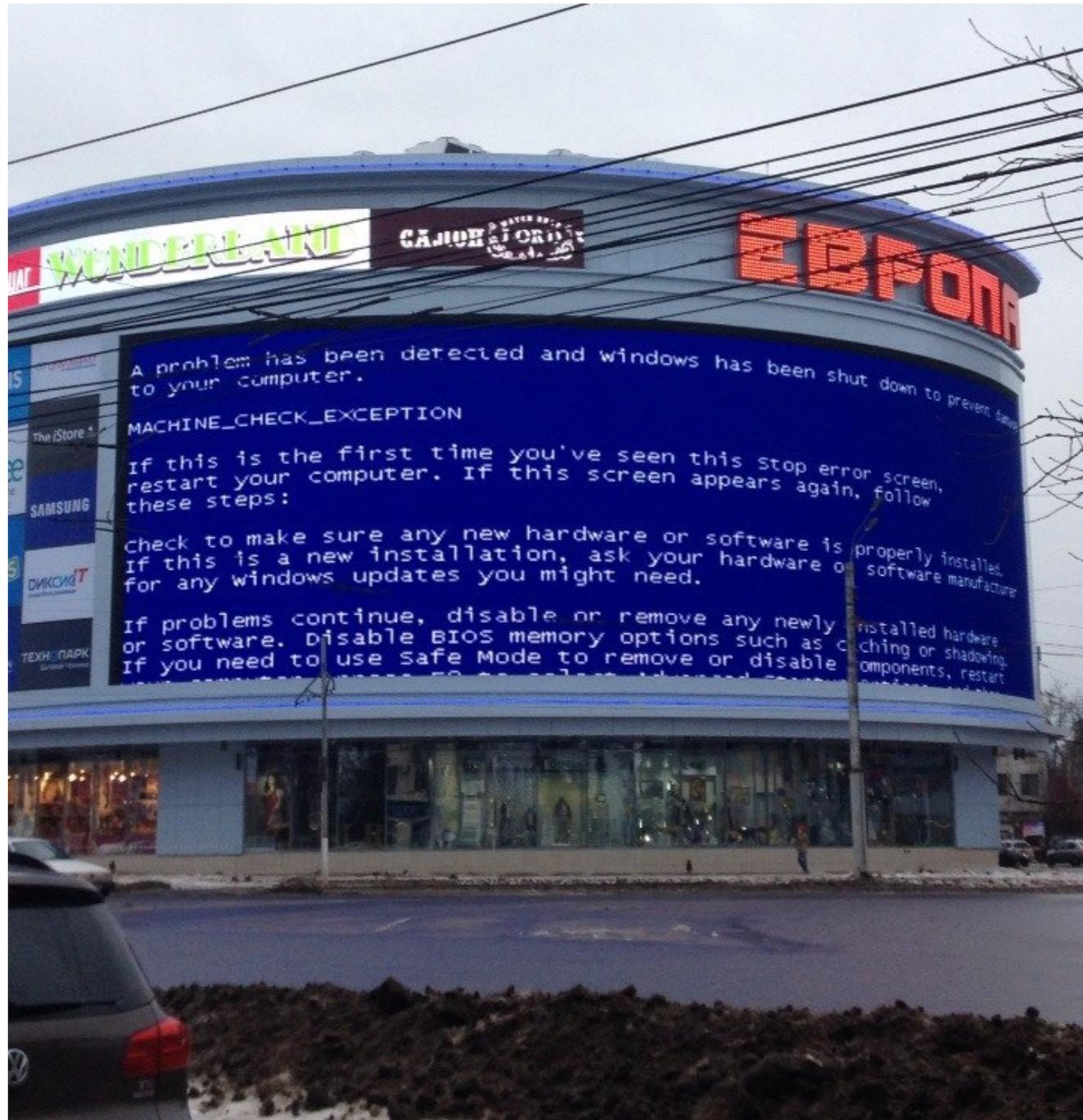


Photo courtesy of zem.

BSOD



BSOD

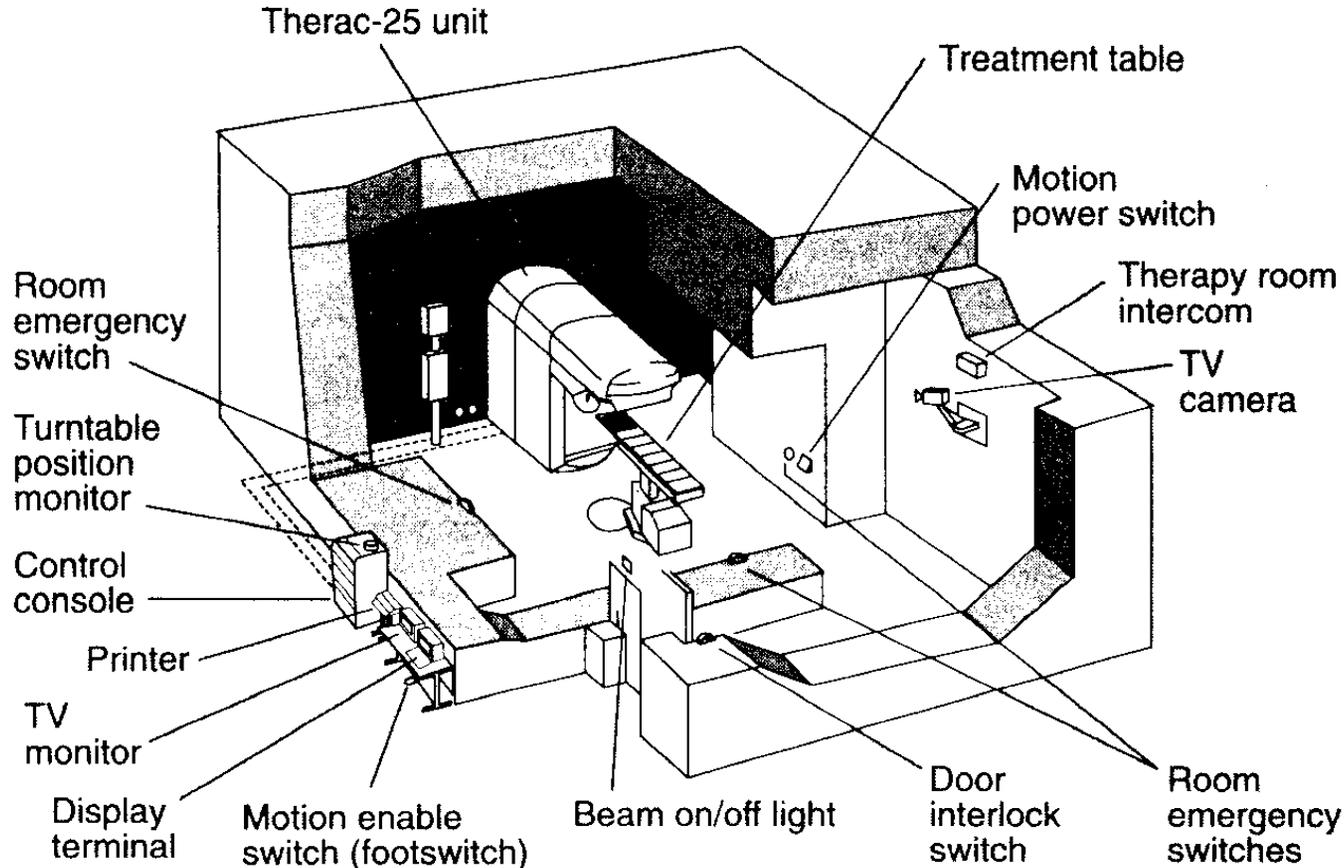


BSOD



THERAC-25

- A safety-critical system with software interlocks
- Beam controlled entirely through a custom OS



THERAC-25

- Old system used a hardware interlock
 - A lever that could either be in the “zap” or “x-ray” position
- New system was computer controlled
- A synchronization failure was triggered when competent nurses used the back arrow to change the data on the screen “too quickly”

THERAC-25 OUTCOME

- Beam killed one person directly, burned others, and may have given inadequate treatment to cancer patients
- Problem was very difficult to diagnose; initial fix involved removal of the back arrow key from the keyboard
- People died because a programmer could not write correct code for a concurrent system

DENVER AIRPORT

- The most modern, most expensive airport in recent history
 - Cost overrun in excess of \$2B
- Highly automated luggage handling system was supposed to deliver your luggage to you at arrival
- Lack of persistence caused luggage carts to “forget” their contents, sprinkling the luggage on the runway

BITCOIN BANKS

- There is a new crop of systems for holding data known as NoSQL databases
- We cautioned about the errors people were committing with NoSQL systems
- The simplest attack against banks, of simultaneous withdrawals, actually works! Attackers stole millions of dollars from Bitcoin banks
 - One bank lost a million and folded
 - Another lost \$500K, took losses

OTHER SYSTEMS

- FAA air traffic control system
- IRS data management system
- IBM “Microkernel”
- Pentagon data security
- Many others, too numerous to list

SUMMARY

- We do not have the necessary technologies and know-how to build robust computer systems
- There is huge demand for people who deeply understand and can build robust systems

PHILOSOPHY

- Not your grandparents' OS course
- We believe that the following topics are critical for every software engineer
 - Building concurrent systems
 - Architecting networked components
 - Building transactional systems
- Not widely shared, the course is unique

LOGISTICS

- Lectures
 - Tuesday, Thursday 2:55-4:10pm, Uris G01
- Instructors:
 - Prof. Sierer & Prof. Bracy
 - cs4410-prof@cornell.edu: *goes to both of us*
 - Office Hours
 - Prof. Sierer: Tuesday 4:10-5:10pm
 - Prof. Bracy: Mondays 10:30-11:30am, Tuesdays 1:15-2:15 pm
- TAs

COMMUNICATION

- Course Web Page
 - Office hours, assignments, lectures, and other supplemental materials will be on the web site
- Piazza:
 - For 99% of the communication
 - Private posts should be visible to *all* course staff
 - Please do not contact course staff by other means (facebook, texts, email, etc.)

ADMINISTRATIVE

- Course has three components
 - Lectures and Readings
 - Exams
 - Assignments
- Textbook
 - Anderson and Dahlin
- You are expected to keep up with all three

GRADING

- CS4410 Breakdown
 - ~45% Assignments
 - ~50% Exams (best 2 of 3)
 - ~5% Flexgrade (participation, attitude, effort)
- CS4411 Breakdown
 - ~90% Projects
 - ~10% Flexgrade
- Grading will not be done on a curve
 - It is our goal to be able to give everyone an A+
 - Help us achieve this

ASSIGNMENTS IN 4410

- 4 assignments
 - *Note:* if you are taking CS 4410 and 4411 this semester, you are exempt from the 4th 4410 assignment
- To be done individually
- Start early, time management is key

PROJECTS IN CS4411

- Projects will be done in two-person teams
 - You may indicate a desired partner
 - If they also indicate you, we will pair you up
 - If you don't have a preferred partner, we'll pair you up with someone suitable
- Working in groups
 - Start early, time management is key
 - Manage the team effort
 - Part of what you are supposed to learn is how to manage to get work done in a small team

ACADEMIC INTEGRITY AND HONOR CODE

- All submitted work must be your own
 - All homeworks must be your own independent work
 - OK to study together
 - Cannot share solutions, ever
- Project groups submit joint work
 - All group assignments must represent solely the work of the two people in that group
 - Cannot be in possession of someone else's solution
- Violations will be prosecuted to the fullest extent
- Closed-book exams, no calculators

OUR EXPECTATIONS

- Code of Silence
 - Absolute quiet during lectures
 - If you have a question, please speak up
 - Chances are 100% that someone else has the same question
- No electronics, Luddite zone
 - Scientific studies show that such classrooms are far more effective

PREREQUISITE

- CS3410 or equivalent required. Or permission of instructor.
- Required means required.
- If you did not take CS3410 or equivalent, you must contact course staff, explain your situation and request permission.

DRAFT SYLLABUS

- Introduction
- Architectural Support for Operating Systems
- Processes and Threads
- Scheduling
- Synchronization, Mutual Exclusion, Spin Locks, Semaphores, Condition Variables
- Deadlocks, Detection and Avoidance
- Memory Management
- Networking, LANs, WANs, Ethernet, ARP, IP, UDP, TCP
- Disks and RAID
- Filesystems, UFS, LFS
- Security

PROJECT PLAN FOR CS4411

- Threads and Concurrency
- Scheduling
- Basic Datagram Networking
- Reliable Streaming Protocols
- Filesystems

ABOUT PROF. SIRER

- **Interests:** OSes, distributed systems, self-organizing and peer-to-peer systems
- Bitcoin, flaws within, fixes to
- Sailing, wind, sea, woodworking (aka body modification)



ABOUT PROF. BRACY

- **Interests:** computer architecture, microarchitecture, processor performance
- Swimming, hiking, wrangling small children



QUESTIONS?

- And demographics...

The slides for this semester are the product of many rounds of teaching CS 4410 by Professors Agarwal, Alvisi, Anderson, Bershada, Bracy, George, Levy, Sirer, and Van Renesse.