

CS 162 Over Enrolled (400) for 245 Li Ka Shing (~300 seats)

- CoE Dean Sastry asked us to fulfill the department's responsibilities:
 - To enforce fire code, up to refusing to continue with lecture should that become necessary. Everyone must have a seat in the room
 - To webcast the lecture so in the unexpected event that the room is overcrowded and students are turned away, they will still have access to all course content. <https://calcentral.berkeley.edu/academics/semester/spring-2018>
 - To communicate very prominently in any printed and online materials that if the lecture hall reaches capacity at any time, students will be turned away and directed to the webcast.
 - The link for this will be provided to ensure that students don't miss part of the class searching for the webcast version.
 - To ensure that special provision is afforded to accommodate the needs of any disabled students for whom watching the lecture online would adversely affect their learning outcomes compared to attending the lecture in person.
- Course recordings will be posted on CalCentral within 12 hours
 - <https://calcentral.berkeley.edu/academics/semester/spring-2018>
 - Select COMPSCI 162

1/17/18

Joseph and Ragan-Kelley CS162 © UCB Spring 2018

Lec 1.1

Who Am I?

- Anthony D. Joseph
 - 465 Soda Hall (RISE Lab)
 - Web: <http://www.cs.berkeley.edu/~adj/>
 - Office hours: TBD
- Teach BerkeleyX MOOCs on the side
 - Big Data and Apache Spark ('15/'16 >240k students with >11% finishing)
- Research areas:
 - Cancer Genomics/Precision Medicine (ADAM/Apache Spark), Secure Machine Learning (SecML), DETER security testbed
 - Previous: Cloud computing (Apache Mesos), Peer-to-Peer networking (Tapestry), Mobile computing, Wireless/Cellular networking
- Campus Cyber-Risk Responsible Executive, co-chair dept IT cmte

1/17/18

Joseph and Ragan-Kelley CS162 © UCB Spring 2018

Lec 1.3

CS162 Operating Systems and Systems Programming Lecture 1

What is an Operating System?

January 17th, 2018

Profs. Anthony D. Joseph and Jonathan Ragan-Kelley

<http://cs162.eecs.Berkeley.edu>

Who am I?

- Jonathan Ragan-Kelley
 - 525 Soda Hall (Visual Computing Lab)
 - Web: <http://www.cs.berkeley.edu/~jrk/>
 - Office hours: TBD
- Research areas:
 - Applying systems, compilers, programming languages, and architecture to graphics, vision, machine learning, and computational imaging
 - Making it easy to write faster programs, build more efficient systems

e.g., <http://halide-lang.org>



- Previously: Stanford (boo), MIT, Lucasfilm, Google, ExpanDrive

1/17/18

Joseph and Ragan-Kelley CS162 © UCB Spring 2018

Lec 1.4

“The Magnificent Seven” - CS162 TAs



Eric Hou



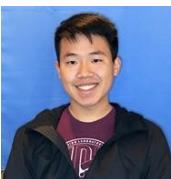
Nitin Manivasagan



Kalyanaraman Shankari



William Sheu



Jack Yang



Patrick Yang



Justin Yum (Head TA)

1/17/18

Joseph and Ragan-Kelley CS162 © UCB Spring 2018

Lec 1.5

1/17/18

Joseph and Ragan-Kelley CS162 © UCB Spring 2018

Lec 1.6

Goals for Today

- What is an Operating System?
 - And – what is it not?
- What makes Operating Systems so exciting?
- Oh, and “How does this class operate?”

Interactive is important!

Ask Questions!

Slides courtesy of David Culler, Anthony D. Joseph, John Kubiatowicz, AJ Shankar, George Necula, Alex Aiken, Eric Brewer, Ras Bodik, Ion Stoica, Doug Tygar, and David Wagner.

1/17/18

Joseph and Ragan-Kelley CS162 © UCB Spring 2018

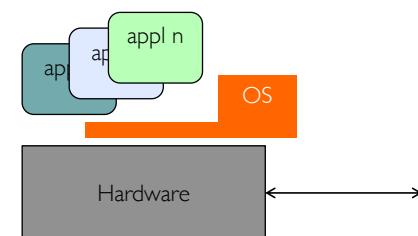
Lec 1.7

This and Next Week

- Sections on Fridays – attend any section you want
 - We'll assign permanent sections after forming project groups
 - This week will help us determine the section balance
- *This is an Early Drop Deadline course (Friday January 26th)*
 - If you are not serious about taking, please drop early
 - Dept will continue to admit students as other students drop
- On the waitlist ???
 - *We are at capacity, but waitlist will be processed through Mon 1/29*

What is an operating system?

- Special layer of software that provides application software access to hardware resources
 - Convenient abstraction of complex hardware devices
 - Protected access to shared resources
 - Security and authentication
 - Communication amongst logical entities



1/17/18

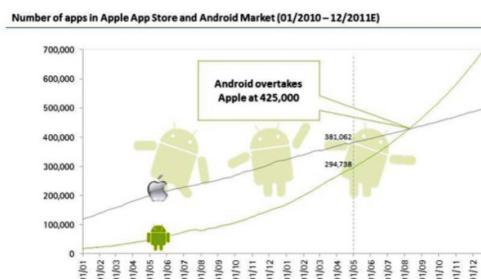
Joseph and Ragan-Kelley CS162 © UCB Spring 2018

Lec 1.8

What Does an OS do?

- Provide abstractions to apps

- File systems
- Processes, threads
- VM, containers
- Naming system
- ...



- Manage resources:

- Memory, CPU, storage, ...

- Achieves the above by implementing specific algos and techniques:

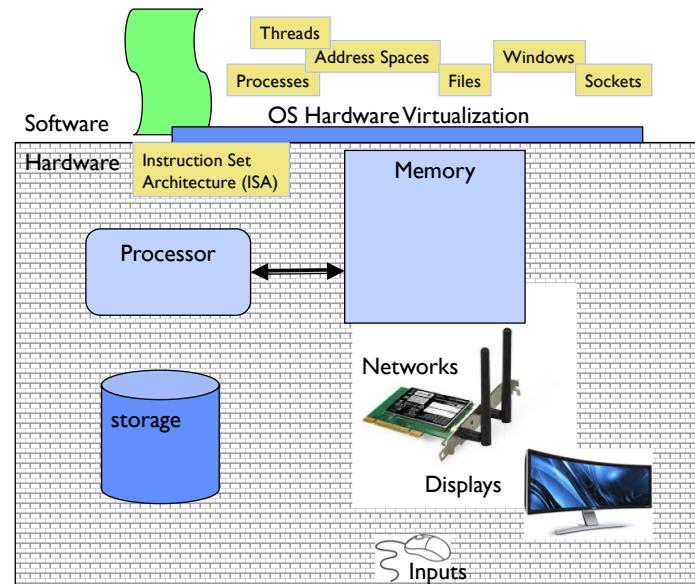
- Scheduling
- Concurrency
- Transactions
- Security
-

1/17/18

Joseph and Ragan-Kelley CS162 © UCB Spring 2018

Lec 1.9

OS Basics: "Virtual Machine" Boundary

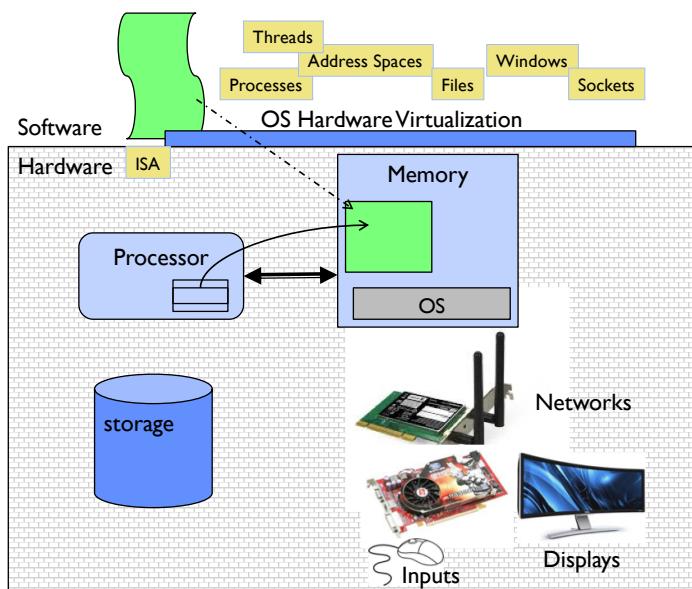


1/17/18

Joseph and Ragan-Kelley CS162 © UCB Spring 2018

Lec 1.10

OS Basics: Program => Process

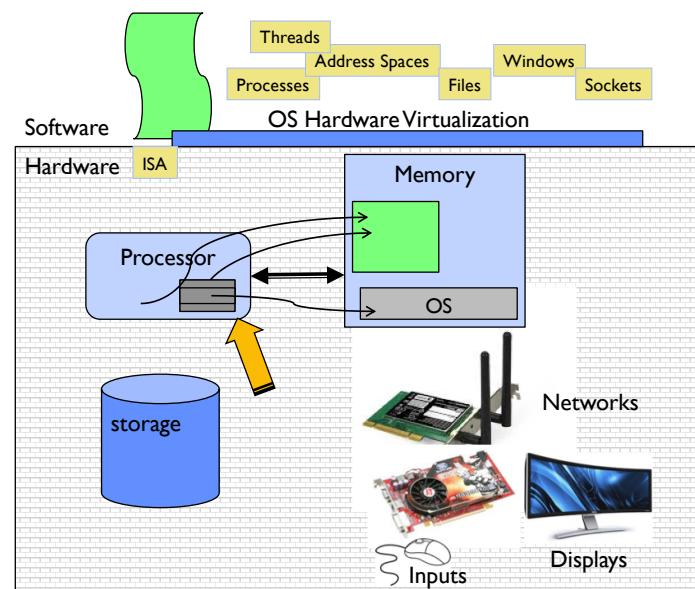


1/17/18

Joseph and Ragan-Kelley CS162 © UCB Spring 2018

Lec 1.11

OS Basics: Context Switch

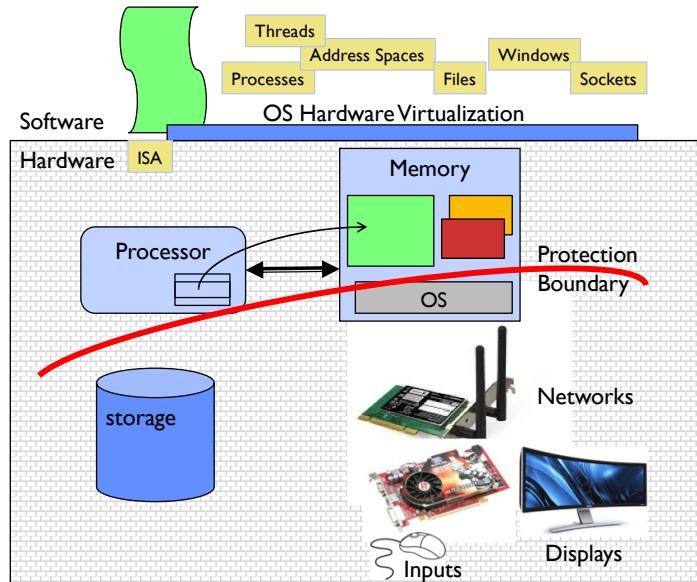


1/17/18

Joseph and Ragan-Kelley CS162 © UCB Spring 2018

Lec 1.12

OS Basics: Scheduling, Protection

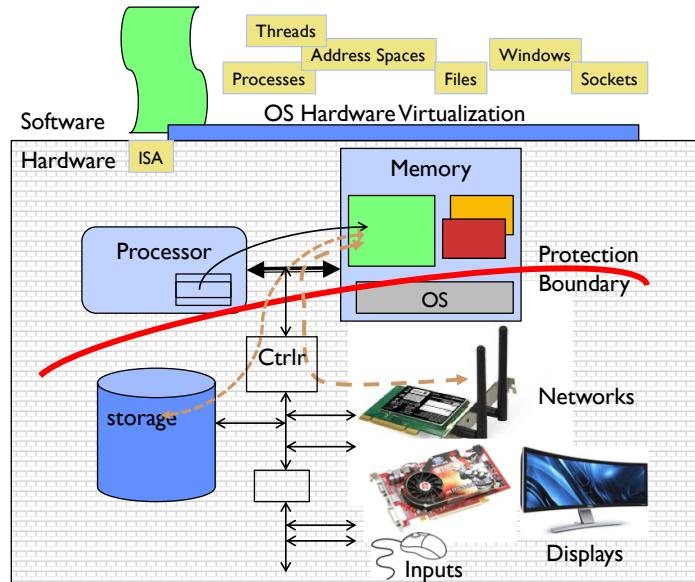


I/17/18

Joseph and Ragan-Kelley CS162 © UCB Spring 2018

Lec 1.13

OS Basics: I/O

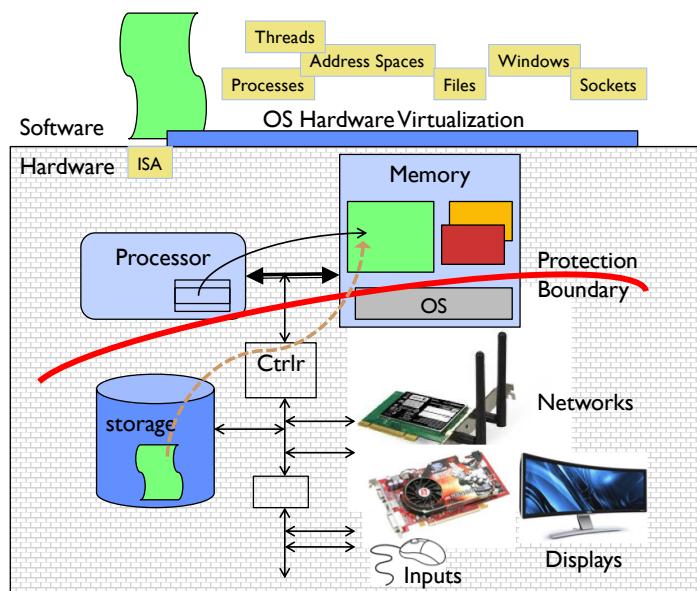


I/17/18

Joseph and Ragan-Kelley CS162 © UCB Spring 2018

Lec 1.14

OS Basics: Loading



I/17/18

Joseph and Ragan-Kelley CS162 © UCB Spring 2018

Lec 1.15

What makes Operating Systems Exciting and Challenging?



SPECTRE

```
if (x < array1_size)
    y = array2[array1[x] * 256];
```



MELTDOWN

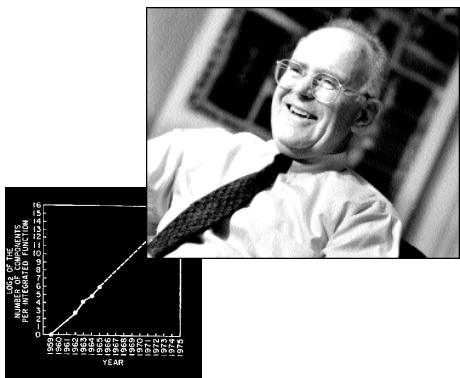
```
meltdown:
    mov al, byte [rcx]
    shl rax, 0xc
    jz meltdown
    mov rbx, qword [rbx + rax]
```

I/17/18

Joseph and Ragan-Kelley CS162 © UCB Spring 2018

Lec 1.16

Technology Trends: Moore's Law



Gordon Moore (co-founder of Intel) predicted in 1965 that the transistor density of semiconductor chips would double roughly every 18 months

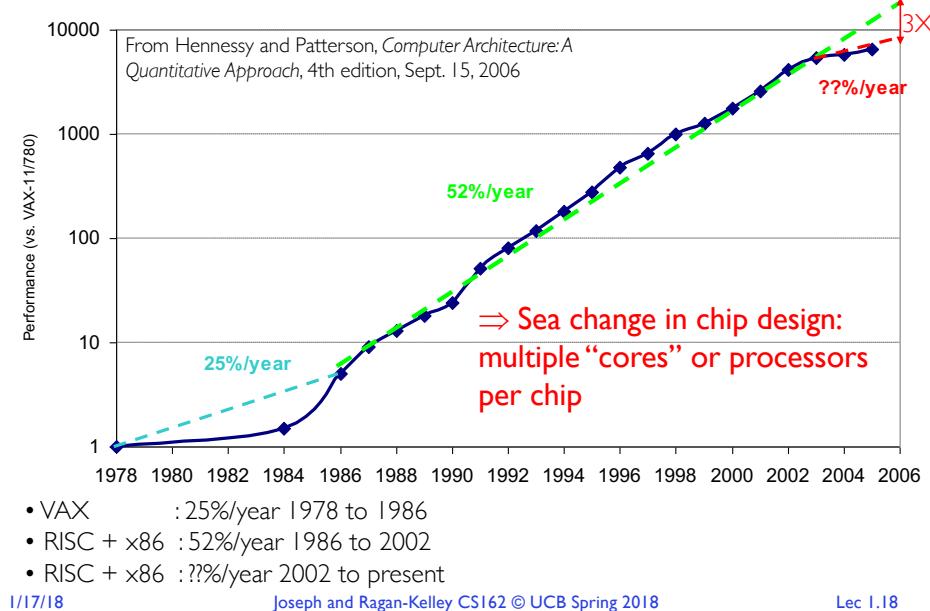
I/17/18

Joseph and Ragan-Kelley CS162 © UCB Spring 2018

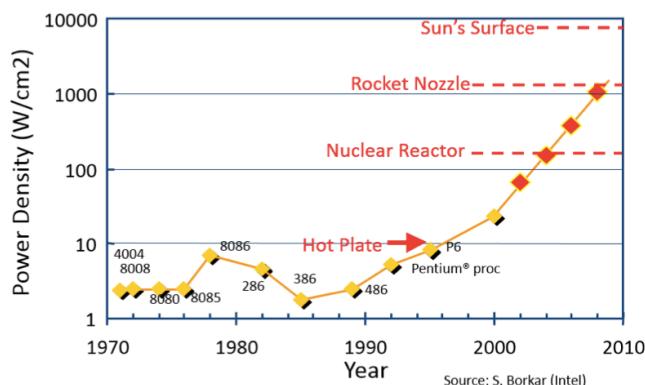
Lec 1.17

Microprocessors have become smaller, denser, and more powerful

New Challenge: Slowdown in Joy's law of Performance



Another Challenge: Power Density



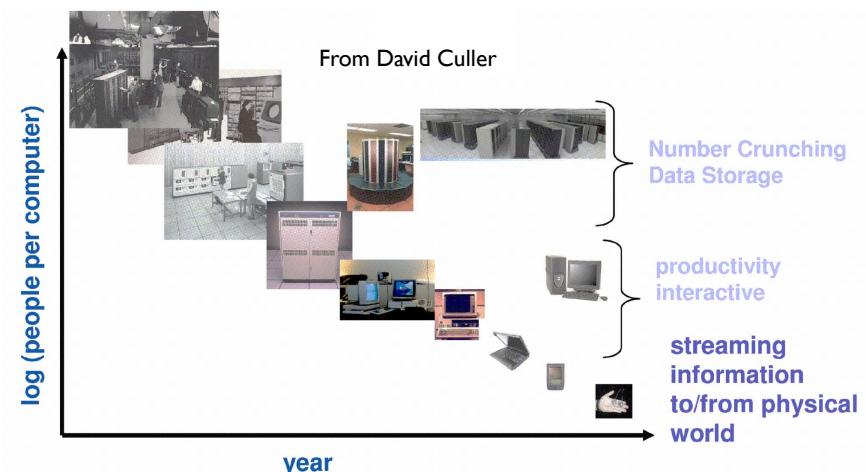
- Moore's law extrapolation
 - Potential power density reaching amazing levels!
- Flip side: battery life very important
 - Moore's law can yield more functionality at equivalent (or less) total energy consumption

I/17/18

Joseph and Ragan-Kelley CS162 © UCB Spring 2018

Lec 1.19

People-to-Computer Ratio Over Time



- Today: multiple CPUs/person!
 - Approaching 100s?

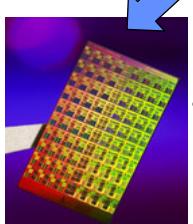
I/17/18

Joseph and Ragan-Kelley CS162 © UCB Spring 2018

Lec 1.20

ManyCore Chips: The future is here

- Intel 80-core multicore chip (Feb 2007)



- 80 simple cores
- Two FP-engines / core
- Mesh-like network
- 100 million transistors
- 65nm feature size

- Intel Single-Chip Cloud Computer (August 2010)

- 24 "tiles" with two cores/tile
- 24-router mesh network
- 4 DDR3 memory controllers
- Hardware support for message-passing

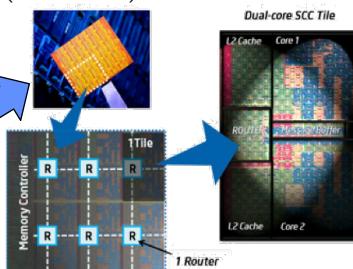
- Amazon X1 instances

- 128 virtual cores, 2 TB RAM

- How to program these?

- Use 2 CPUs for video/audio
- Use 1 for word processor, 1 for browser
- 76 for virus checking???

- Parallelism must be exploited at all levels



I/17/18

Joseph and Ragan-Kelley CS162 © UCB Spring 2018

Lec 1.21

The End of Moore's Law...



- Moore's Law has (officially) ended -- Feb 2016
 - No longer getting 2 x transistors/chip every 18 months...
 - or even every 24 months
- May have only 2-3 smallest geometry fabrication plants left:
 - Intel and Samsung and/or TSMC
- Vendors moving to 3D stacked chips
 - More layers in old geometries

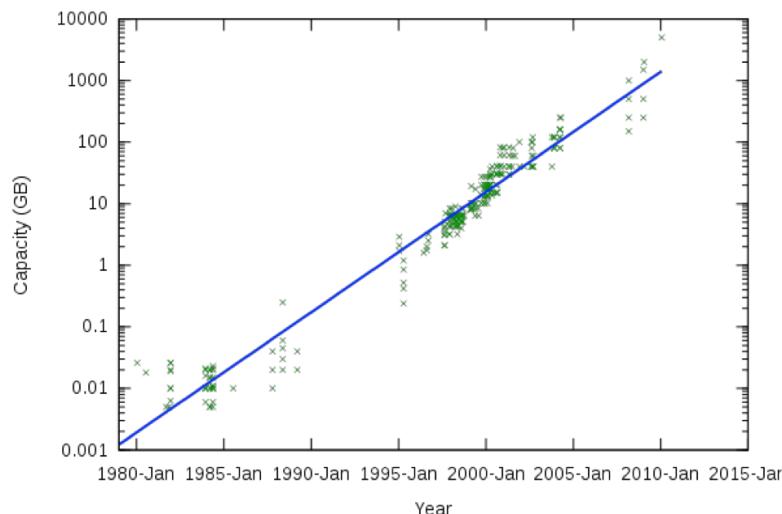
<http://files.shareholder.com/downloads/INTC/867590276x0xS50863-16-105/50863/filing.pdf>

I/17/18

Joseph and Ragan-Kelley CS162 © UCB Spring 2018

Lec 1.22

Storage Capacity



- Retail hard disk capacity in GB

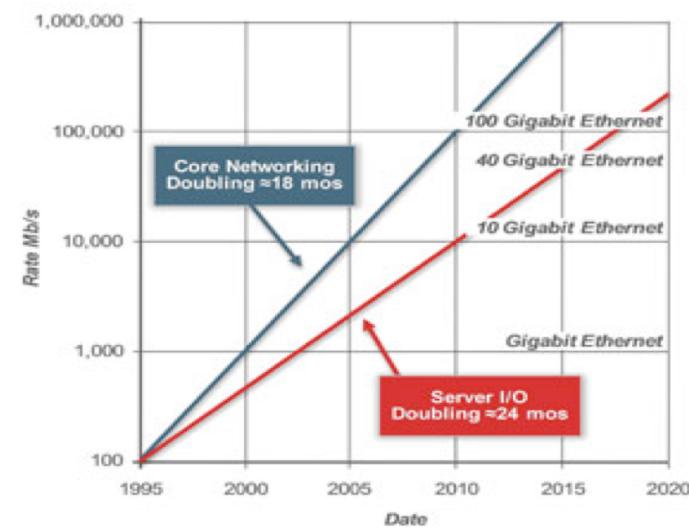
(source: <http://www.digitaltonto.com/2011/our-emergent-digital-future/>)

I/17/18

Joseph and Ragan-Kelley CS162 © UCB Spring 2018

Lec 1.23

Network Capacity



(source: <http://www.ospmag.com/issue/article/Time-Is-Not-Always-On-Our-Side>)

I/17/18

Joseph and Ragan-Kelley CS162 © UCB Spring 2018

Lec 1.24

Challenge: Complexity

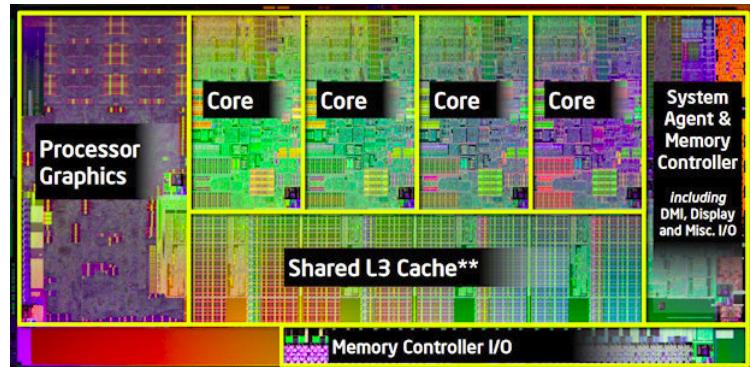
- Applications consisting of...
 - ... a variety of software modules that ...
 - ... run on a variety of devices (machines) that
 - » ... implement different hardware architectures
 - » ... run competing applications
 - » ... fail in unexpected ways
 - » ... can be under a variety of attacks
- Not feasible to test software for all possible environments and combinations of components and devices
 - The question is not whether there are bugs but how serious are the bugs!

I/17/18

Joseph and Ragan-Kelley CS162 © UCB Spring 2018

Lec 1.25

A Modern Processor: Intel Sandy Bridge



- Package: LGA 1155
 - 1155 pins
 - 95W design envelope
- Cache:
 - L1: 32K Inst, 32K Data (3 clock access)
 - L2: 256K (8 clock access)
 - Shared L3: 3MB – 20MB (not out yet)
- Transistor count:
 - 504 Million (2 cores, 3MB L3)
 - 2.27 Billion (8 cores, 20MB L3)
- Note that ring bus is on high metal layers
 - above the Shared L3 Cache

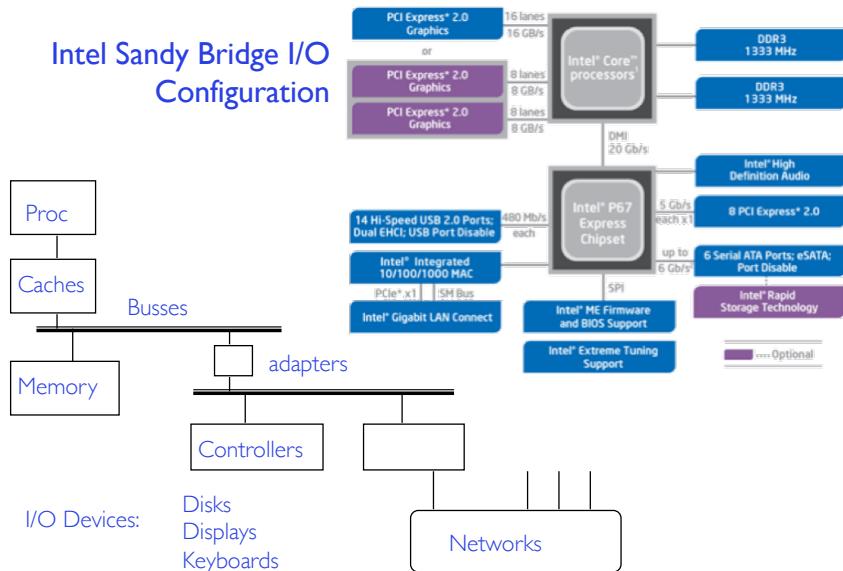
I/17/18

Joseph and Ragan-Kelley CS162 © UCB Spring 2018

Lec 1.26

HW Functionality comes with great complexity!

Intel Sandy Bridge I/O Configuration



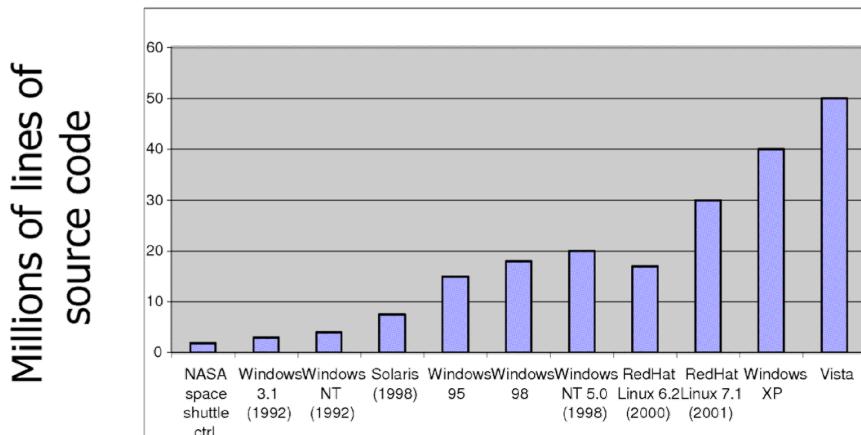
I/17/18

Joseph and Ragan-Kelley CS162 © UCB Spring 2018

Lec 1.27

5 min break

Increasing Software Complexity



From MIT's 6.033 course

1/17/18

Joseph and Ragan-Kelley CS162 © UCB Spring 2018

Lec 1.29

How do we tame complexity?

- Every piece of computer hardware different
 - Different CPU
 - » Pentium, PowerPC, ColdFire, ARM, MIPS
 - Different amounts of memory, disk, ...
 - Different types of devices
 - » Mice, Keyboards, Sensors, Cameras, Fingerprint readers
 - Different networking environment
 - » Cable, DSL, Wireless, Firewalls, ...
- Questions:
 - Does the programmer need to write a single program that performs many independent activities?
 - Does every program have to be altered for every piece of hardware?
 - Does a faulty program crash everything?
 - Does every program have access to all hardware?

1/17/18

Joseph and Ragan-Kelley CS162 © UCB Spring 2018

Lec 1.31

OS Tool: Virtual Machine Abstraction

Application

Virtual Machine Interface

Operating System

Physical Machine Interface

Hardware

- Software Engineering Problem:
 - Turn hardware/software quirks ⇒ what programmers want/need
 - Optimize for convenience, utilization, security, reliability, etc...
- For any OS area (e.g. file systems, virtual memory, networking, scheduling):
 - What's the hardware interface? (physical reality)
 - What's the application interface? (nicer abstraction)

1/17/18

Joseph and Ragan-Kelley CS162 © UCB Spring 2018

Lec 1.32

Virtual Machines

- Software emulation of an abstract machine
 - Give programs illusion they own the machine
 - Make it look like hardware has features you want
- Two types of "Virtual Machine"s
 - Process VM: supports the execution of a single program; this functionality typically provided by OS
 - System VM: supports the execution of an entire OS and its applications (e.g., VMWare Fusion, Virtual box, Parallels Desktop, Xen)



1/17/18

Joseph and Ragan-Kelley CS162 © UCB Spring 2018

Lec 1.33

Process VMs

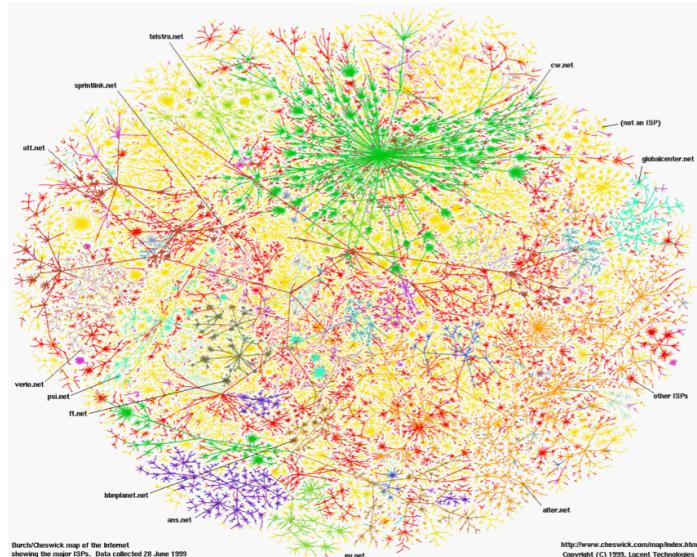
- Programming simplicity
 - Each process thinks it has all memory/CPU time
 - Each process thinks it owns all devices
 - Different devices appear to have same high level interface
 - Device interfaces more powerful than raw hardware
 - » Bitmapped display \Rightarrow windowing system
 - » Ethernet card \Rightarrow reliable, ordered, networking (TCP/IP)
- Fault Isolation
 - Processes unable to directly impact other processes
 - Bugs cannot crash whole machine
- Protection and Portability
 - Java interface safe and stable across many platforms

1/17/18

Joseph and Ragan-Kelley CS162 © UCB Spring 2018

Lec 1.34

Greatest Artifact of Human Civilization...



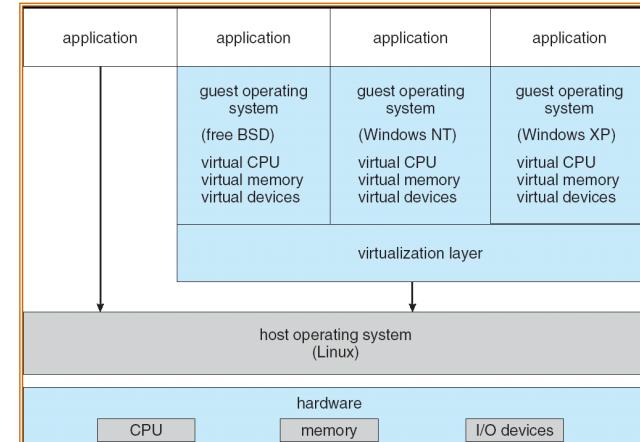
1/17/18

Joseph and Ragan-Kelley CS162 © UCB Spring 2018

Lec 1.36

System Virtual Machines: Layers of OSs

- Useful for OS development
 - When OS crashes, restricted to one VM
 - Can aid testing programs on other OSs

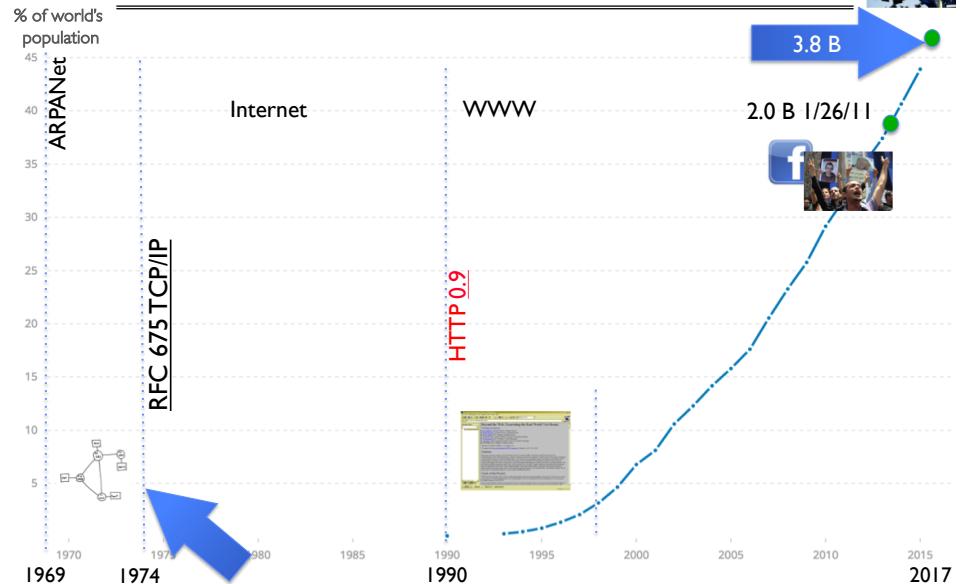


1/17/18

Joseph and Ragan-Kelley CS162 © UCB Spring 2018

Lec 1.35

Internet Scale: Over 3.8 Billion Users!



1/17/18

Joseph and Ragan-Kelley CS162 © UCB Spring 2018

Lec 1.37

Internet Scale: Over 3.8 Billion Users!

WORLD INTERNET USAGE AND POPULATION STATISTICS JUNE 30, 2017 - Update						
World Regions	Population (2017 Est.)	Population % of World	Internet Users 30 June 2017	Penetration Rate (% Pop.)	Growth 2000-2017	Internet Users %
Africa	1,246,504,865	16.6 %	388,104,452	31.1 %	8,497.0%	10.1 %
Asia	4,148,177,672	55.2 %	1,909,408,707	46.0 %	1,570.5%	49.8 %
Europe	822,710,362	10.9 %	650,558,113	79.1 %	519.0%	17.0 %
Latin America / Caribbean	647,604,645	8.6 %	392,215,155	60.6 %	2,070.7%	10.2 %
Middle East	250,327,574	3.3 %	146,972,123	58.7 %	4,374.3%	3.8 %
North America	363,224,006	4.8 %	320,059,368	88.1 %	196.1%	8.3 %
Oceania / Australia	40,479,846	0.5 %	28,180,356	69.6 %	269.8%	0.7 %
WORLD TOTAL	7,519,028,970	100.0 %	3,835,498,274	51.0 %	962.5%	100.0 %

NOTES: (1) Internet Usage and World Population Statistics updated as of June 30, 2017. (2) CLICK on each world region name for detailed regional usage information. (3) Demographic (Population) numbers are based on data from the [United Nations Population Division](#). (4) Internet usage information comes from data published by [Nielsen Online](#), by ITU, the [International Telecommunications Union](#), by [GfK](#), by local ICT Regulators and other reliable sources. (5) For definitions, navigation help and disclaimers, please refer to the [Website Surfing Guide](#). (6) Information from this site may be cited, giving the due credit and placing a link back to [www.internetworldstats.com](#). Copyright © 2017, Miniwatts Marketing Group. All rights reserved worldwide.

(source: <http://www.internetworldstats.com/stats.htm>)

1/17/18

Joseph and Ragan-Kelley CS162 © UCB Spring 2018

Lec 1.38

Not Only PCs connected to the Internet

- Smartphone shipments exceed PC shipments!

373.1 million in Q3 2017

- 2011 shipments:

– 487M smartphones

– 414M PC clients

➢ 210M notebooks

➢ 112M desktops

➢ 63M tablets

– 25M smart TVs

70.6 million in Q4 2017

52.9 million in Q4 2017

222 million in 2016

- 4 billion phones in the world → smartphone over next decade

1/17/18

Joseph and Ragan-Kelley CS162 © UCB Spring 2018

Lec 1.39

Societal Scale Information Systems (Or the "Internet of Things")

- The world is a large distributed system
 - Microprocessors in everything
 - Vast infrastructure behind them

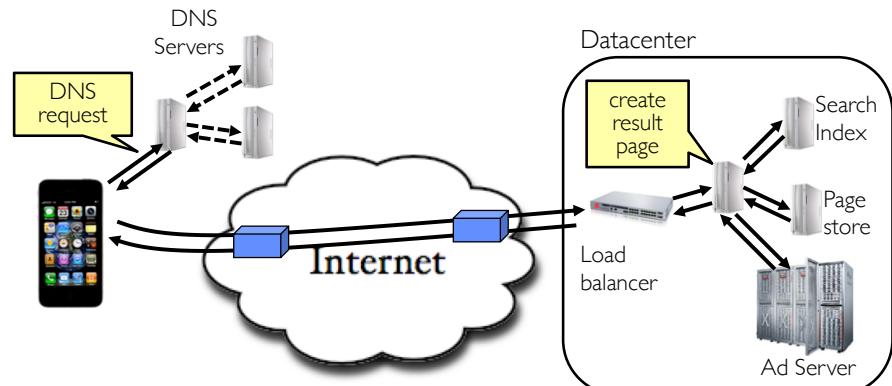


1/17/18

Joseph and Ragan-Kelley CS162 © UCB Spring 2018

Lec 1.40

Example: What's in a Search Query?



- Complex interaction of multiple components in multiple administrative domains
 - Systems, services, protocols, ...

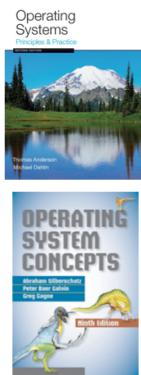
1/17/18

Joseph and Ragan-Kelley CS162 © UCB Spring 2018

Lec 1.41

Infrastructure, Textbook & Readings

- Infrastructure
 - Website: <http://cs162.eecs.berkeley.edu>
 - Piazza: <https://piazza.com/berkeley/spring2018/cs162>
 - Webcast: Cal Central - <https://calcentral.berkeley.edu/academics/teaching-semester/spring-2018/class/compsci-162>
- Textbook: Operating Systems: Principles and Practice (2nd Edition) Anderson and Dahlin
- Recommend: Operating Systems Concepts, 9th Edition Silberschatz, Galvin, Gagne
 - Copies in Bechtel
- Online supplements
 - See course website
 - Includes Appendices, sample problems, etc.
 - Networking, Databases, Software Eng, Security
 - Some Research Papers!



Lec 1.42

1/17/18

Joseph and Ragan-Kelley CS162 © UCB Spring 2018

Syllabus

- OS Concepts: How to Navigate as a Systems Programmer!
 - Process, I/O, Networks and Virtual Machines
- Concurrency
 - Threads, scheduling, locks, deadlock, scalability, fairness
- Address Space
 - Virtual memory, address translation, protection, sharing
- File Systems
 - I/O devices, file objects, storage, naming, caching, performance, paging, transactions, databases
- Distributed Systems
 - Protocols, N-Tiers, RPC, NFS, DHTs, Consistency, Scalability, multicast
- Reliability & Security
 - Fault tolerance, protection, security
- Cloud Infrastructure

1/17/18

Joseph and Ragan-Kelley CS162 © UCB Spring 2018

Lec 1.43

Learning by Doing

- Individual Homeworks: Learn Systems Programming
 - 0. Tools, Autograding, recall C, executable
 - 1. Simple Shell
 - 2. Web server
 - 3. Memory allocation
- Three Group Projects (Pintos in C)
 - 1. Threads & Scheduling
 - 2. User-programs
 - 3. File Systems

Lec 1.44

1/17/18

Joseph and Ragan-Kelley CS162 © UCB Spring 2018

Group Project Simulates Industrial Environment

- Project teams have 4 members (try really hard to find 4 members – 3 members requires serious justification)
 - Must work in groups in “the real world”
 - Same section much preferred
- Communicate with colleagues (team members)
 - Communication problems are natural
 - What have you done?
 - What answers you need from others?
 - You must document your work!!!
- Communicate with supervisor (TAs)
 - What is the team’s plan?
 - What is each member’s responsibility?
 - Short progress reports are required
 - Design Documents: High-level description for a manager!

1/17/18

Joseph and Ragan-Kelley CS162 © UCB Spring 2018

Lec 1.45

Getting started

- Start homework 0, this Friday, 1/19
 - Github account
 - Registration survey
 - Vagrant virtualbox – VM environment for the course
 - » Consistent, managed environment on your machine
 - Get familiar with all the cs162 tools
 - Submit to autograder via git
- Start forming a project group

1/17/18

Joseph and Ragan-Kelley CS162 © UCB Spring 2018

Lec 1.46

Personal Integrity

- UCB Academic Honor Code: "As a member of the UC Berkeley community, I act with honesty, integrity, and respect for others."

<http://asuc.org/honorcode/resources/HC%20Guide%20for%20Syllabi.pdf>

1/17/18

Joseph and Ragan-Kelley CS162 © UCB Spring 2018

Lec 1.48

Grading

- Point distribution subject to change
- 42% three midterms (14% each) – ALL DATES AND TIMES ARE TENTATIVE
 - Wednesday, 2/28, 6:30-8p, TUESDAY, 3/20, 6:30-8p (no class), Monday, 4/23, 6:30-8p (no class)
- 35% projects
- 15% homework
- 8% participation
- No final exam
- Projects
 - Initial design document, Design review, Code, Final design
 - Submission via git push triggers autograder

1/17/18

Joseph and Ragan-Kelley CS162 © UCB Spring 2018

Lec 1.47

CS 162 Collaboration Policy

Explaining a concept to someone in another group

Discussing algorithms/testing strategies with other groups



Helping debug someone else's code (in another group)

Searching online for generic algorithms (e.g., hash table)



Sharing code or test cases with another group

Copying OR reading another group's code or test cases

Copying OR reading online code or test cases from prior years

We compare all project submissions against prior year submissions and online solutions and will take actions (described on the course overview page) against offenders

1/17/18

Joseph and Ragan-Kelley CS162 © UCB Spring 2018

Lec 1.49

Lecture Goal

Interactive!!!

1/17/18

Joseph and Ragan-Kelley CS162 © UCB Spring 2018

Lec 1.50

What is an Operating System?



- Referee
 - Manage sharing of resources, Protection, Isolation
 - » Resource allocation, isolation, communication

- Illusionist
 - Provide clean, easy to use abstractions of physical resources
 - » Infinite memory, dedicated machine
 - » Higher level objects: files, users, messages
 - » Masking limitations, virtualization



Glue



- Common services
 - » Storage, Window system, Networking
 - » Sharing, Authorization
 - » Look and feel

1/17/18

Joseph and Ragan-Kelley CS162 © UCB Spring 2018

Lec 1.51

What is an Operating System,... Really?

- Most Likely:
 - Memory Management
 - I/O Management
 - CPU Scheduling
 - Communications? (Does Email belong in OS?)
 - Multitasking/multiprogramming?
- What about?
 - File System?
 - Multimedia Support?
 - User Interface?
 - Internet Browser? ☺
- Is this only interesting to Academics??

1/17/18

Joseph and Ragan-Kelley CS162 © UCB Spring 2018

Lec 1.52

Operating System Definition (Cont.)

- No universally accepted definition
- “Everything a vendor ships when you order an operating system” is good approximation
 - But varies wildly
- “The one program running at all times on the computer” is the **kernel**
 - Everything else is either a system program (ships with the operating system) or an application program

1/17/18

Joseph and Ragan-Kelley CS162 © UCB Spring 2018

Lec 1.53

"In conclusion..."

- Operating systems provide a virtual machine abstraction to handle diverse hardware
 - Operating systems simplify application development by providing standard services
- Operating systems coordinate resources and protect users from each other
 - Operating systems can provide an array of fault containment, fault tolerance, and fault recovery
- CS162 combines things from many other areas of computer science:
 - Languages, data structures, hardware, and algorithms