CS162
Operating Systems and Systems Programming
Lecture 1

What is an Operating System?

August 22rd , 2018

Prof. David Culler

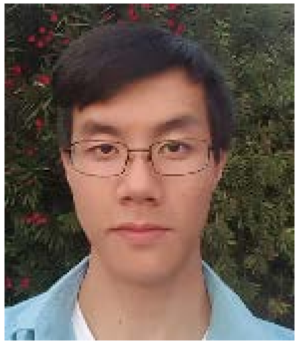
http://cs162.eecs.Berkeley.edu

Instructor: Ion Stoica

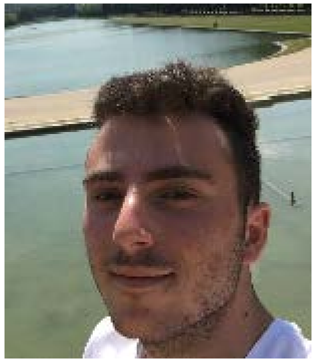
- 465 Soda Hall (RISE Lab)
- Web: http://www.cs.berkeley.edu/~istoica/
- Office hours (tentative):
  - Mondays 1-2pm, and Thursdays 11-12pm in 465F Soda
- Research areas:
  - ML systems (Ray, Pywren, Clipper, …)
  - Big Data systems (Apache Spark, Succinct, …)
  - Previous: Cloud computing (Apache Mesos, Alluxio), Peer-to-Peer networking (Chord), Networking QoS

CS162 TAs

Alon Amid

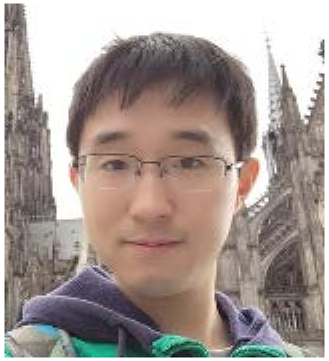

Jason Chin

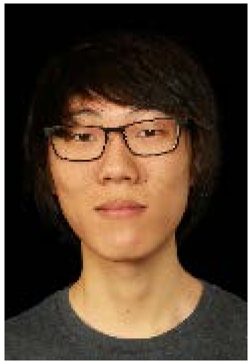
Eric Hou


Alexander Kozarian

Devin Petersohn

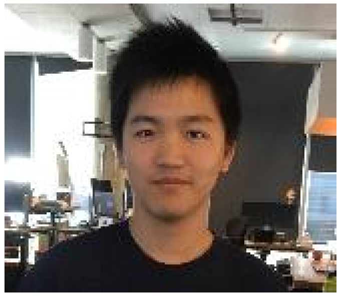
William Sheu


Yang You


Eric Zhou

CS162 Readers

Natalia Layson


Michael Luo

Denny Liang

Alexander Wu

This and Next Week

- Sections Tuesdays/Wednesdays – 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 (September 1)
  - If you are not serious about taking, please drop early
  - Dept will continue to admit students as other students drop
- On the waitlist ???
  - Unfortunately, we maxed out sections and this room capacity

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?"

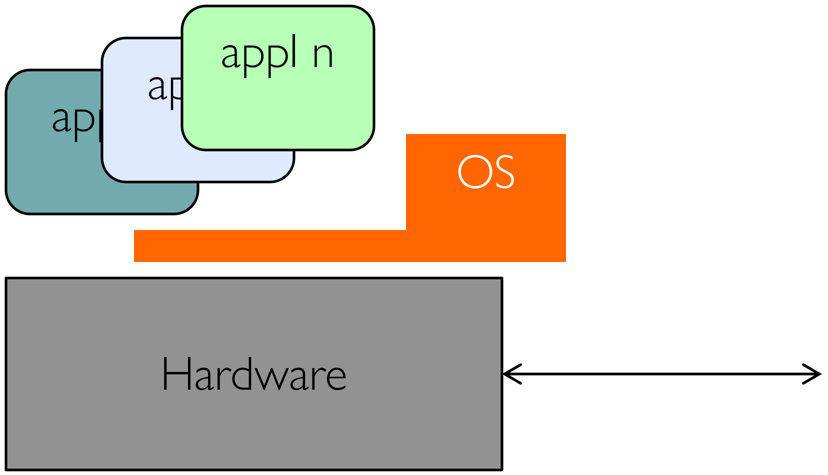
Interactive is important!

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.

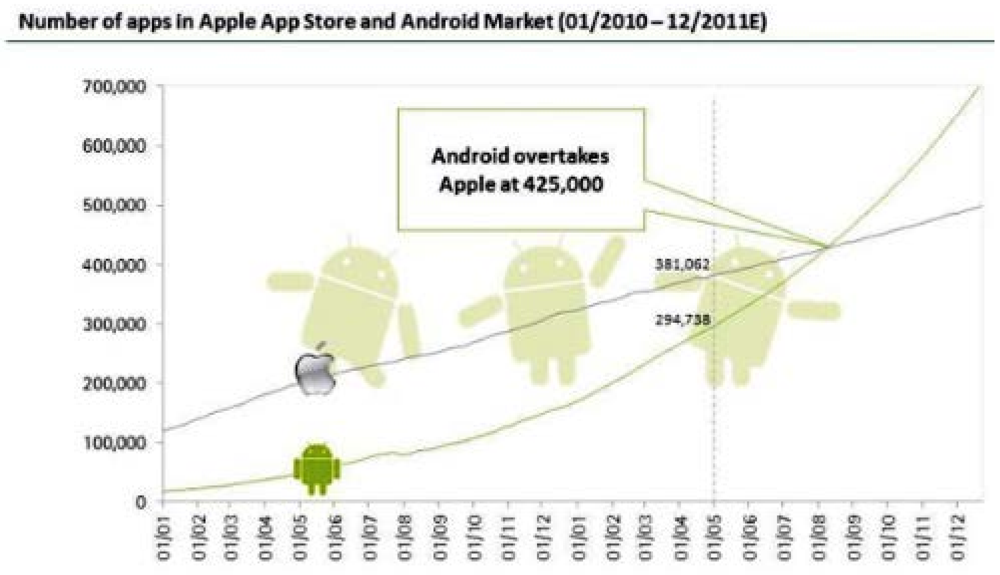
What is an operating system?

- 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

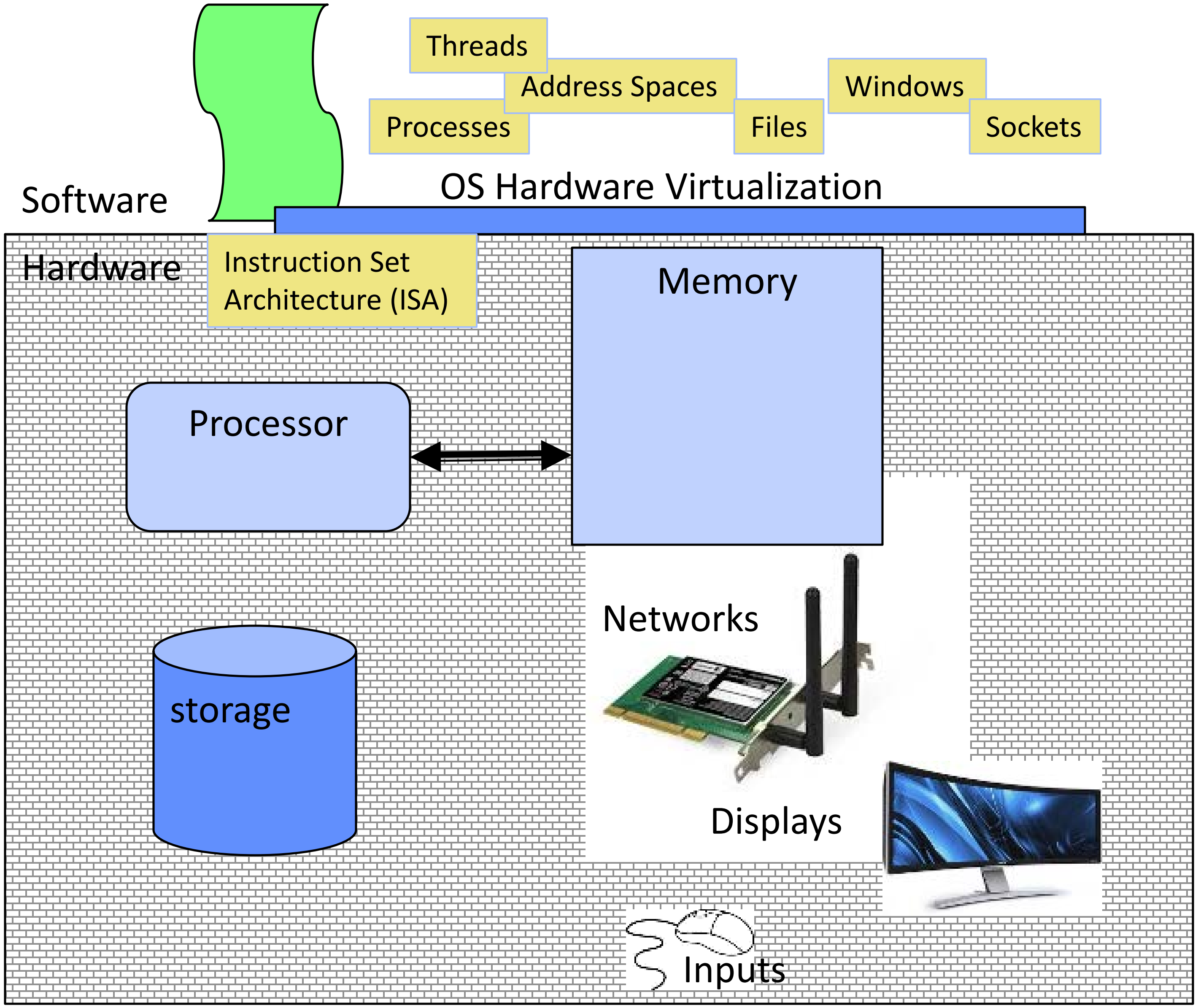


What Does an OS do?

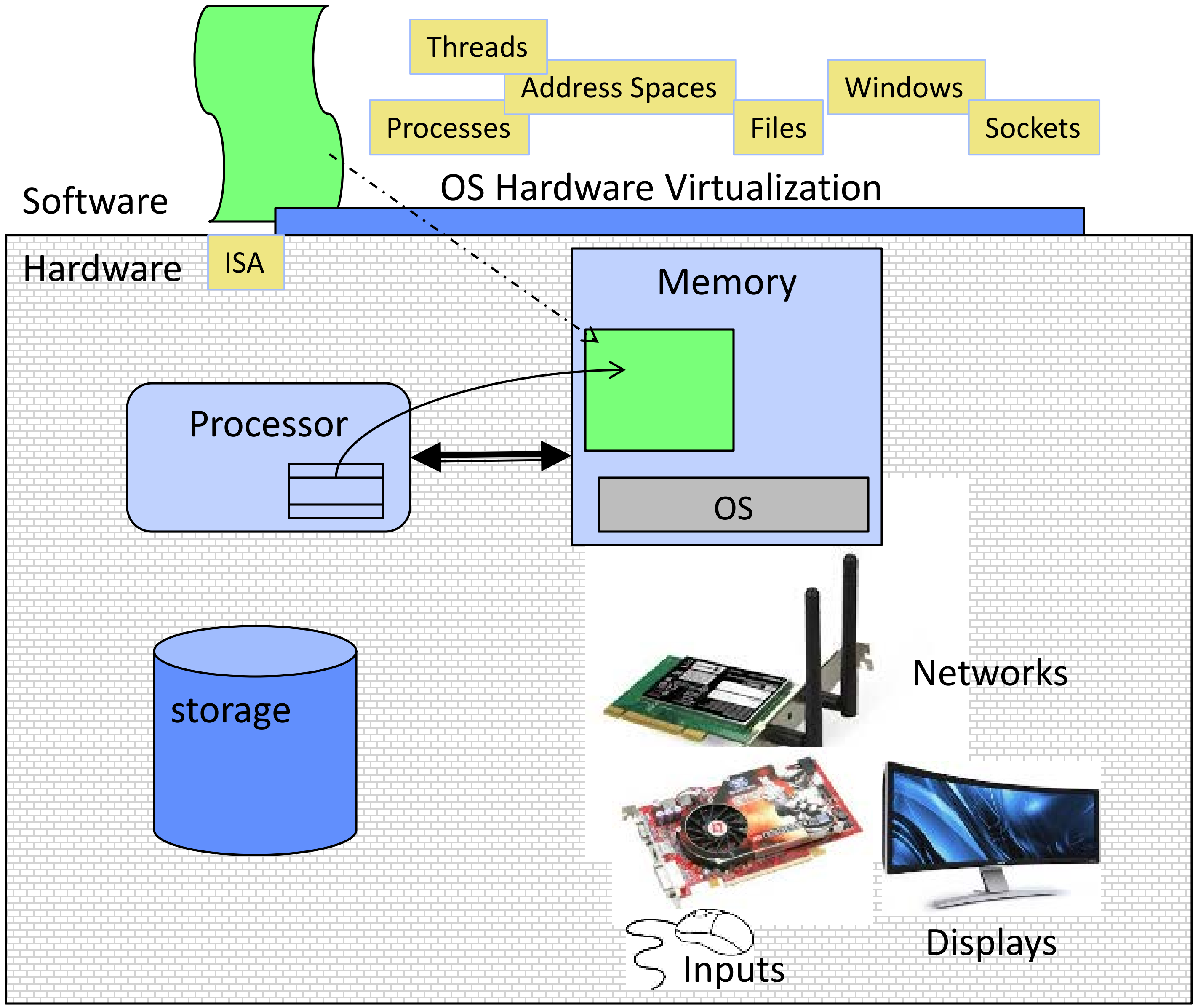
- Provide abstractions to apps
  - File systems
  - Processes, threads
  - VM, containers
  - Naming system
  - …
- Manage resources:
  - Memory, CPU, storage, …
- Achieves the above by implementing specific algos and techniques:
  - Scheduling
  - Concurrency
  - Transactions
  - Security
  - …..



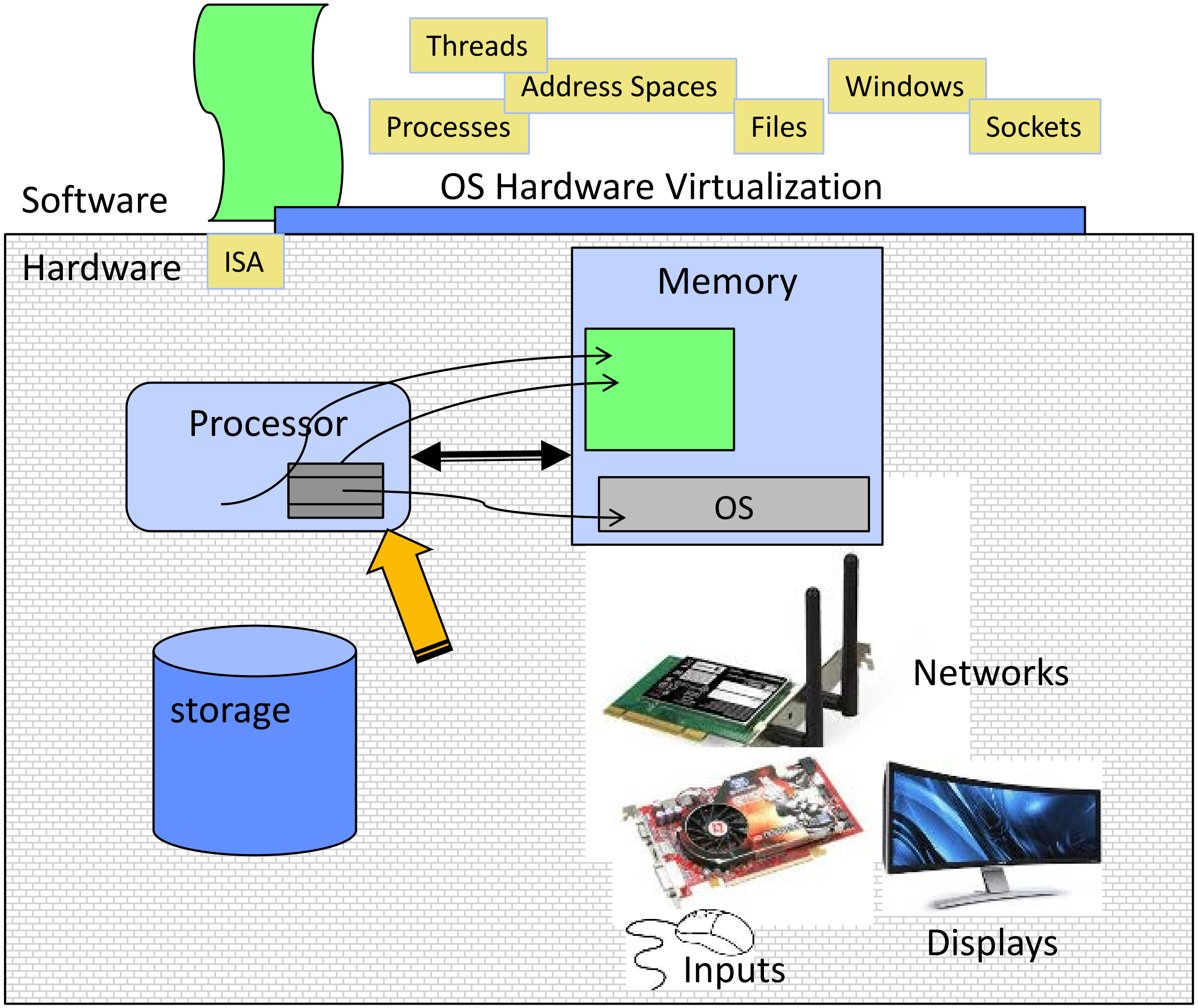
OS Basics: "Virtual Machine" Boundary



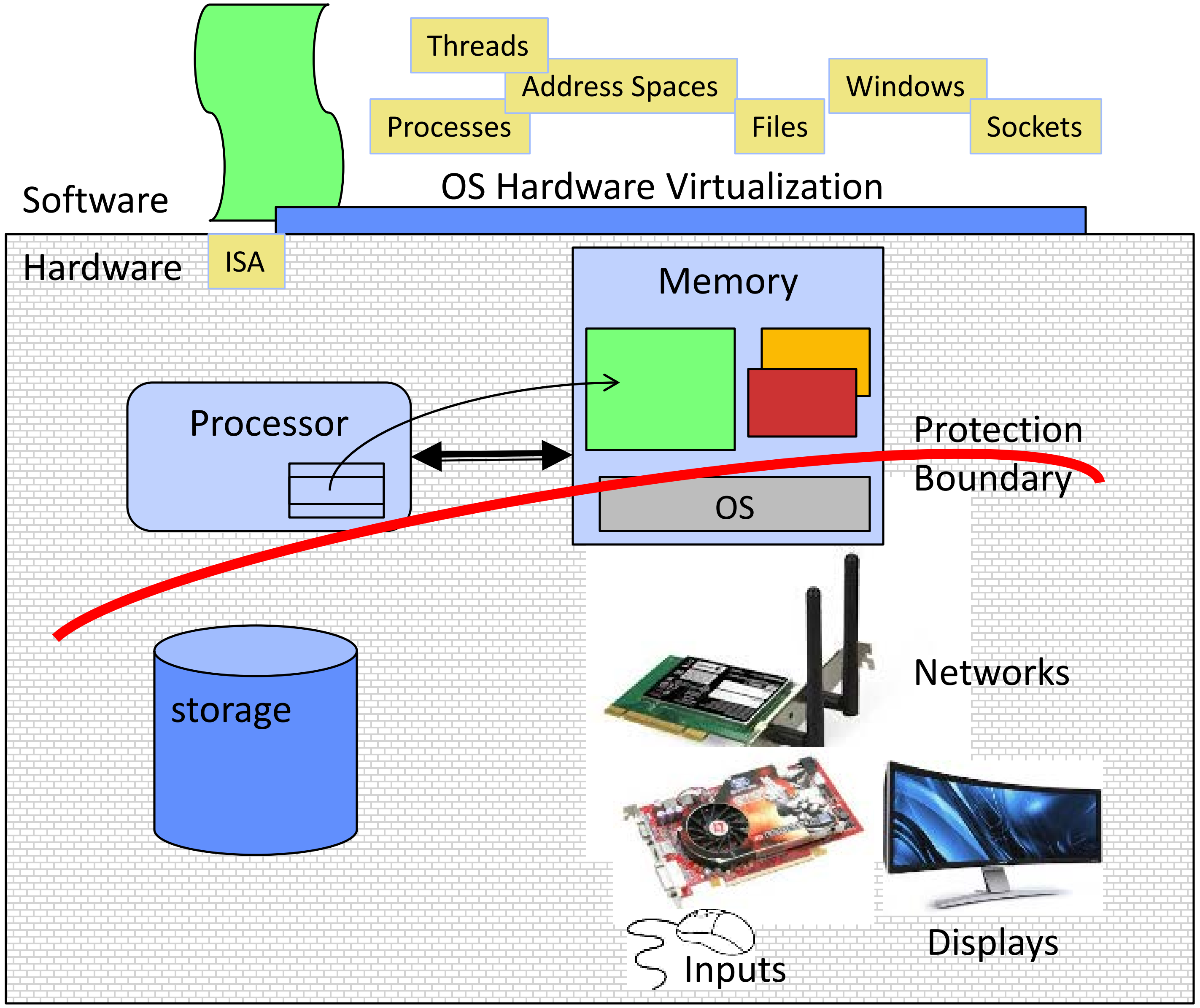
OS Basics: Program => Process



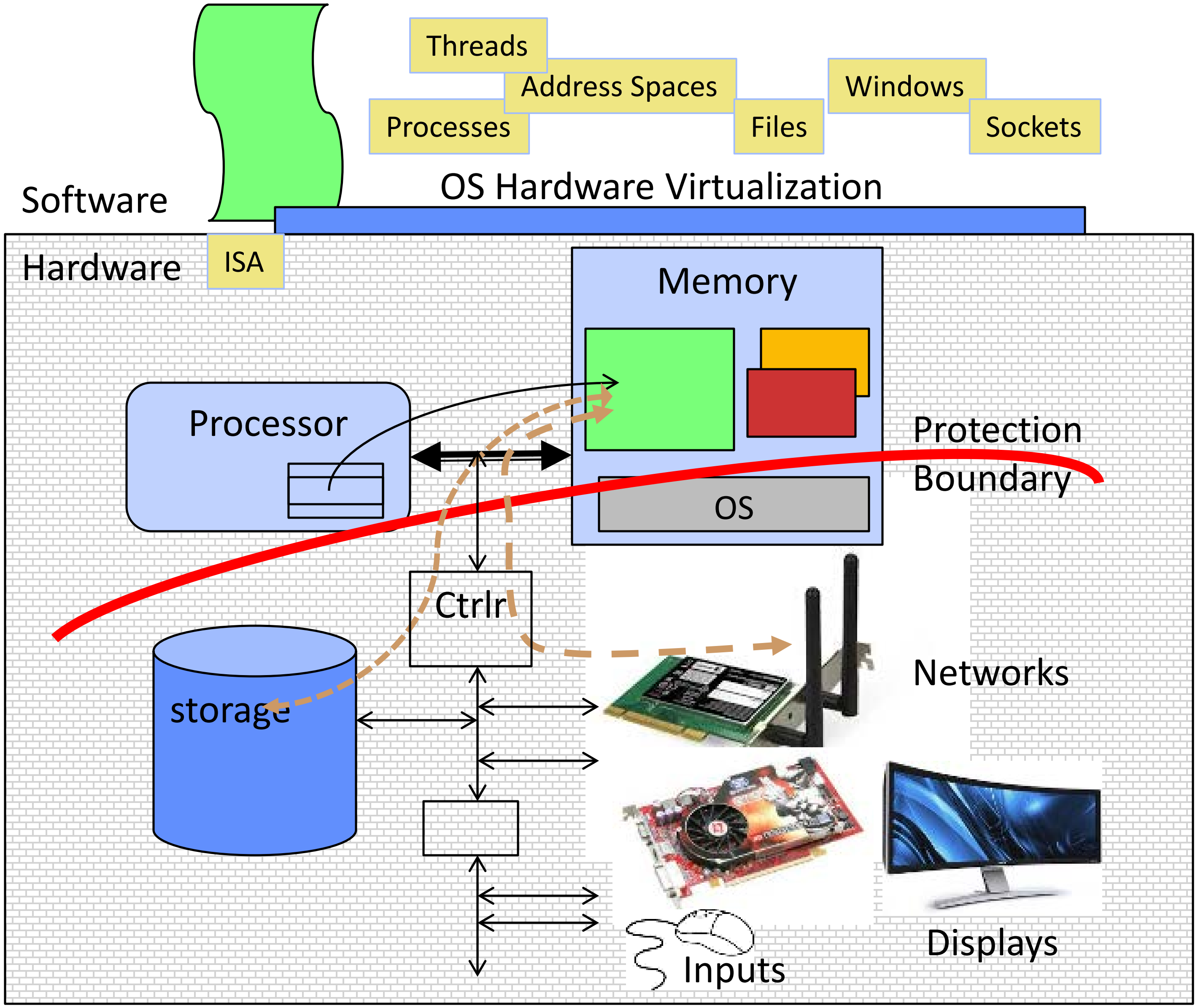
OS Basics: Context Switch



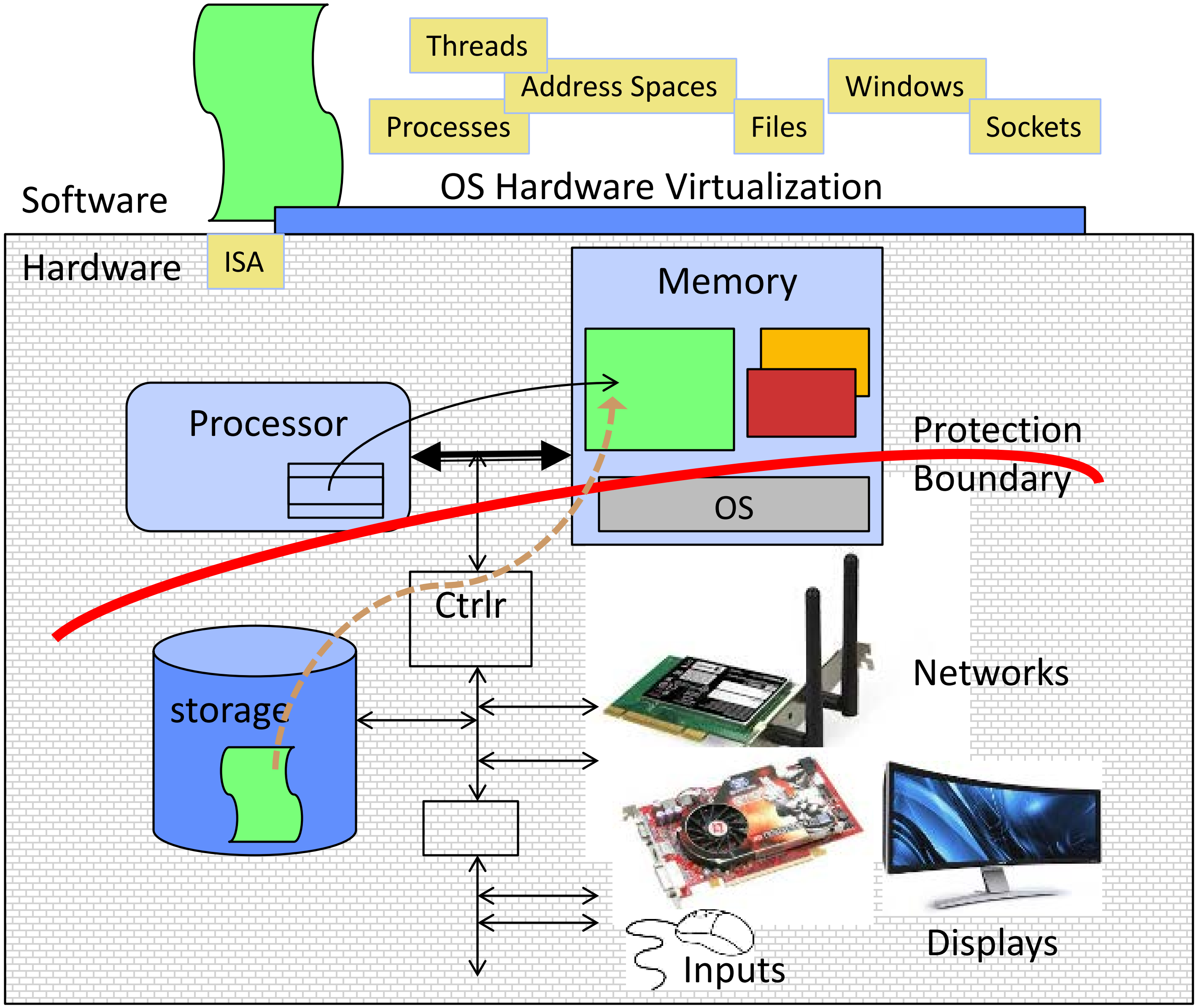
OS Basics: Scheduling, Protection



OS Basics: I/O

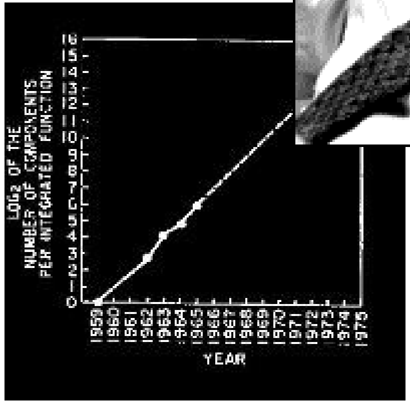

Threads
Address Spaces
Windows
Processes
Files
Sockets
Software
OS Hardware Virtualization
Hardware
ISA
Memory
Processor
Protection
Boundary
OS
Ctrlr
storage
Networks
Inputs
Displays

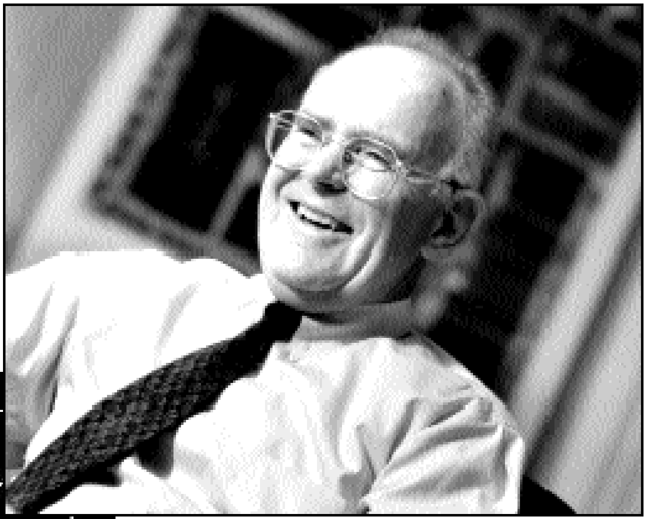
OS Basics: Loading

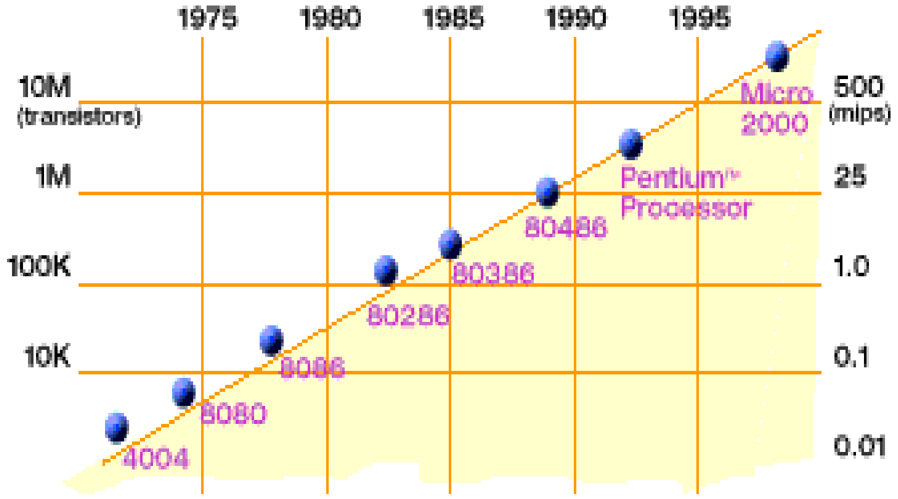


What makes Operating Systems Exciting and Challenging?

Technology Trends: Moore's Law







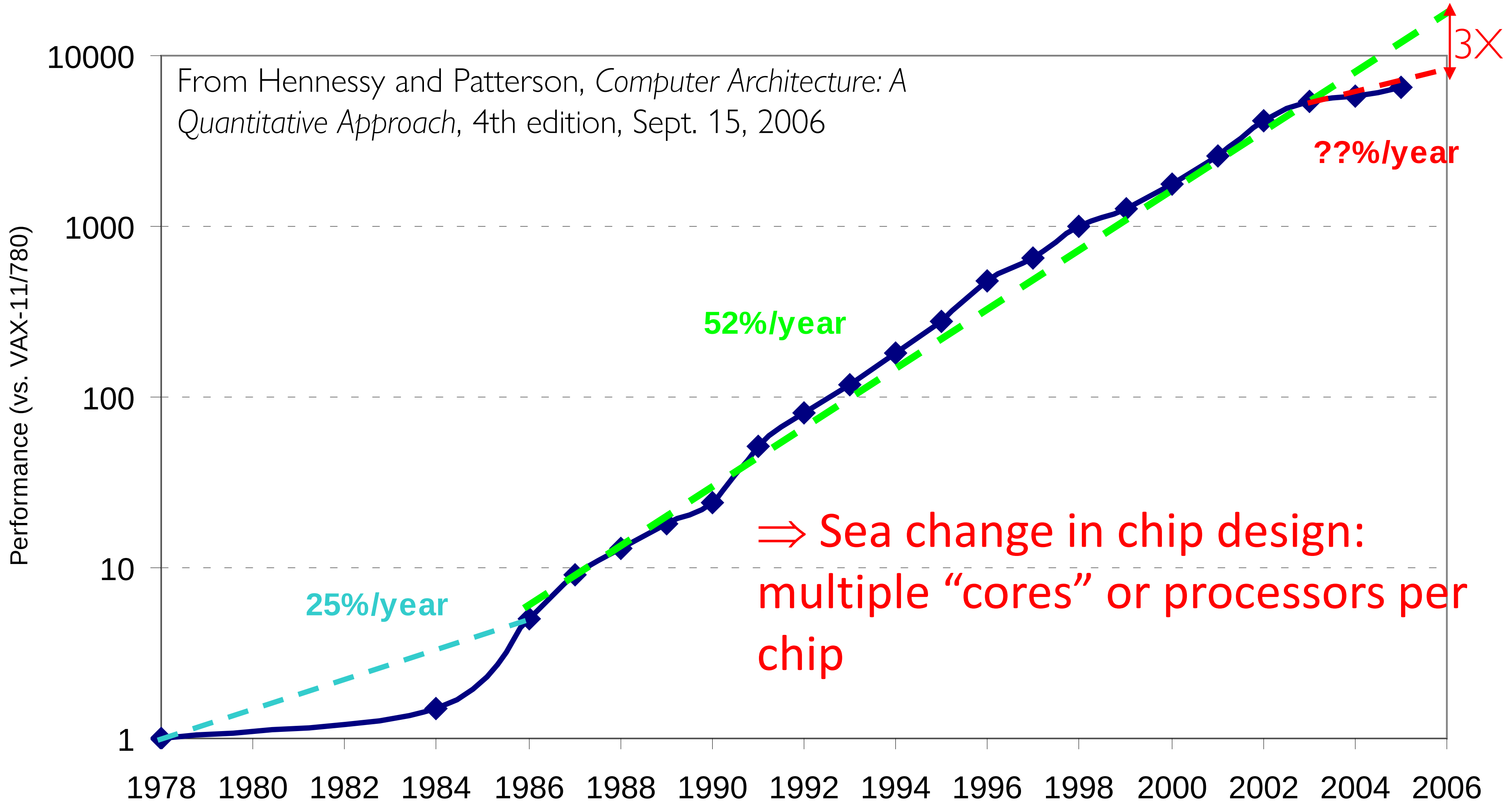
2X transistors/Chip Every 1.5 years

Called "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

Microprocessors have become smaller, denser, and more powerful

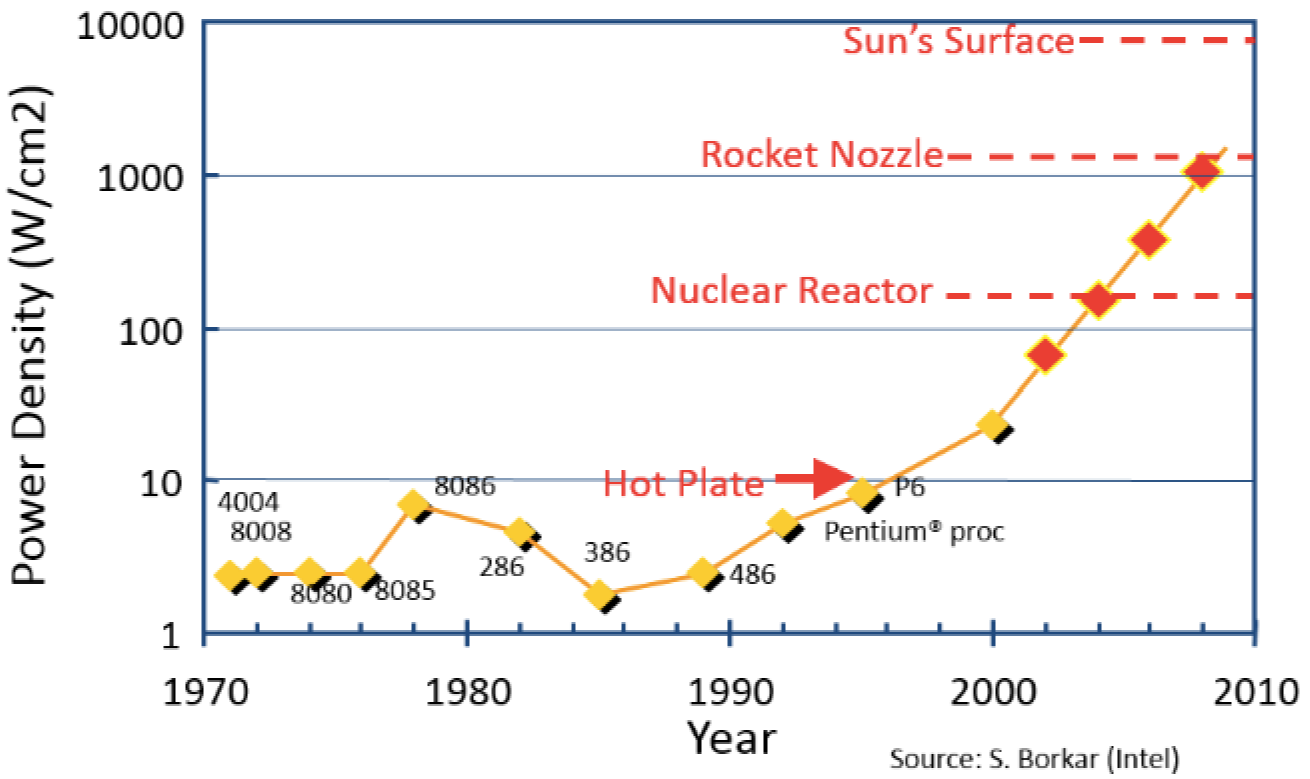
New Challenge: Slowdown in Joy's law of Performance



⇒ Sea change in chip design: multiple "cores" or processors per chip

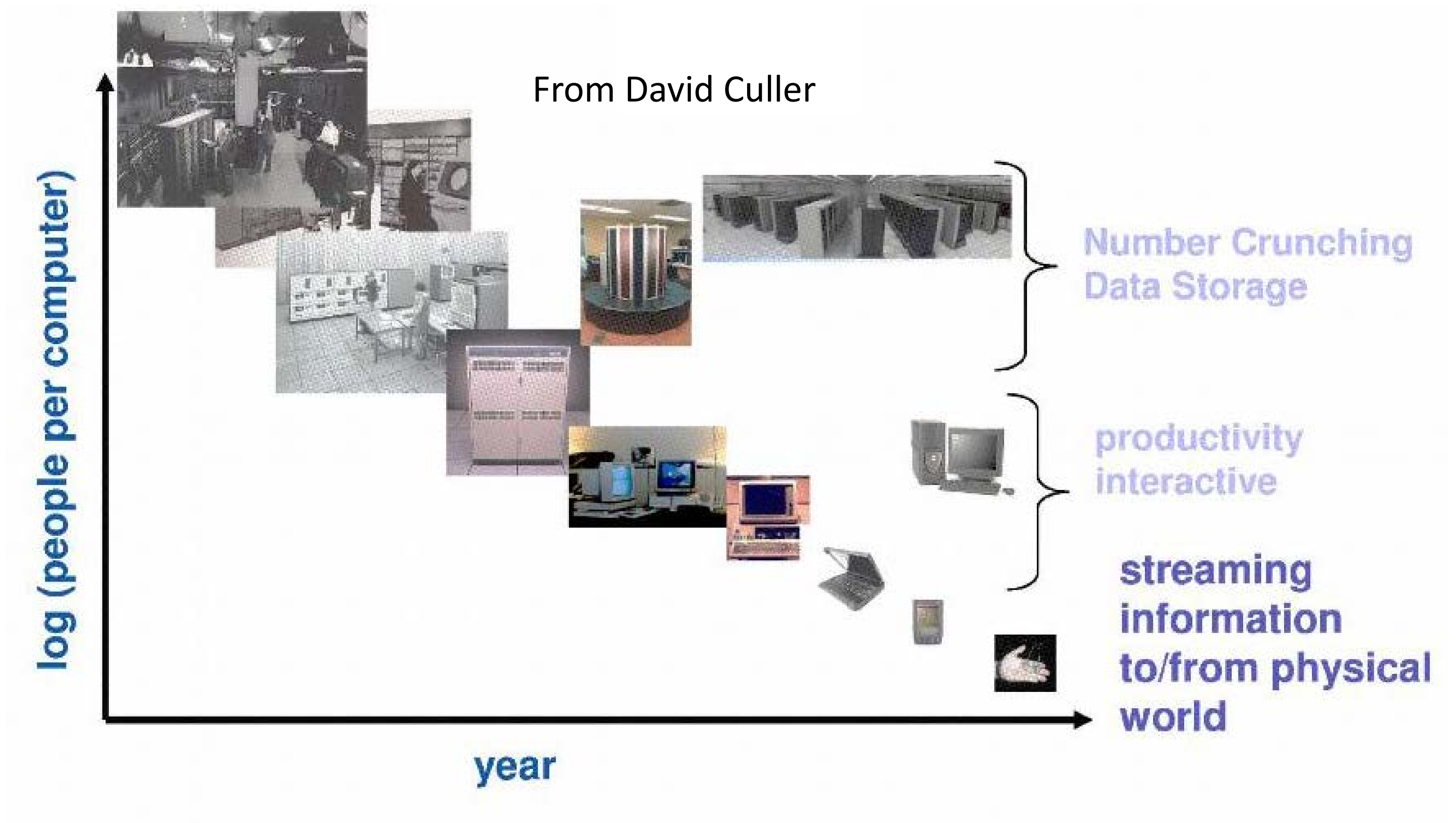
- VAX : 25%/year 1978 to 1986
- RISC + x86 : 52%/year 1986 to 2002
- RISC + x86 : ??%/year 2002 to present

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

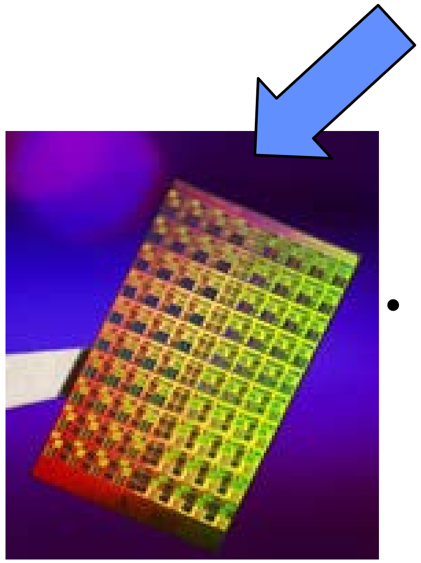
People-to-Computer Ratio Over Time

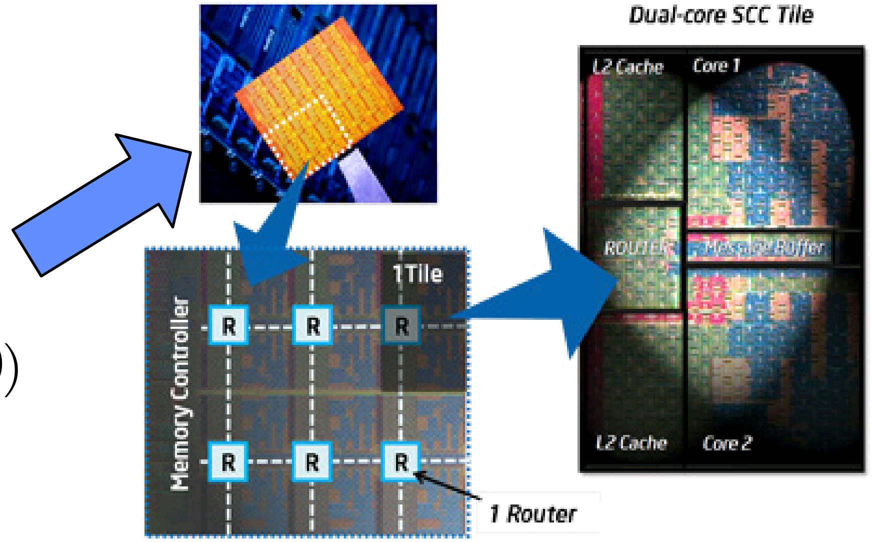


- Today: multiple CPUs/person!
  - Approaching 100s?

ManyCore Chips: The future is here

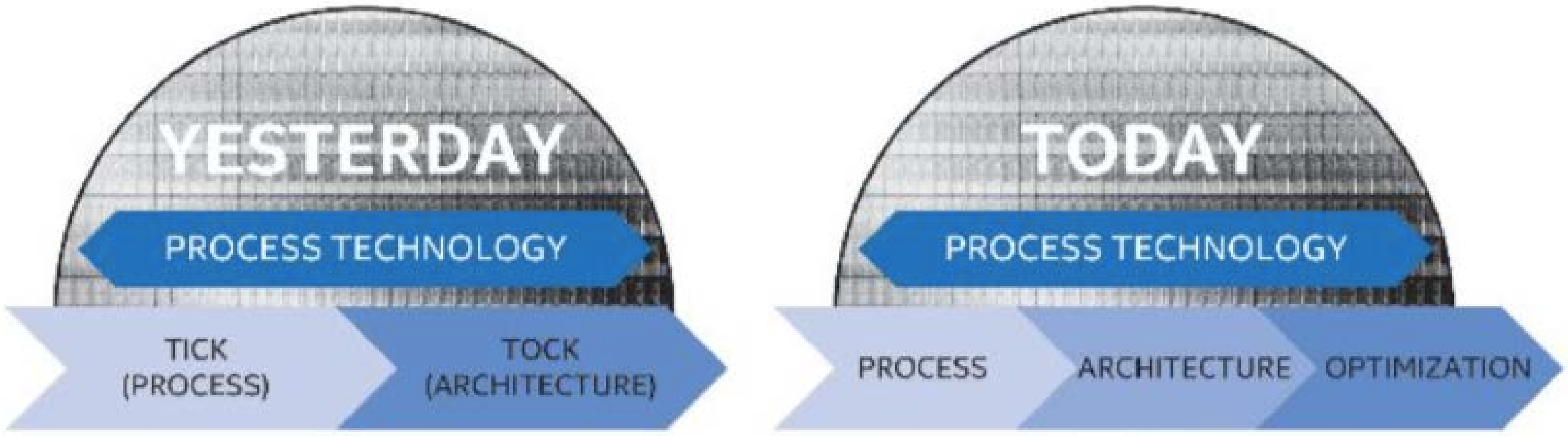
- Intel 80-core multicore chip (Feb 2007)
  - 80 simple cores
  - Two FP-engines / core
  - Mesh-like network
  - 100 million transistors
  - 65nm feature size
- Intel Single-Chip Cloud Computer (August 2010)
  - 24 "tiles" with two cores/tile
  - 24-router mesh network
  - 4 DDR3 memory controllers
  - Hardware support for message-passing





- Amazon X1 instances (2016)
  - 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

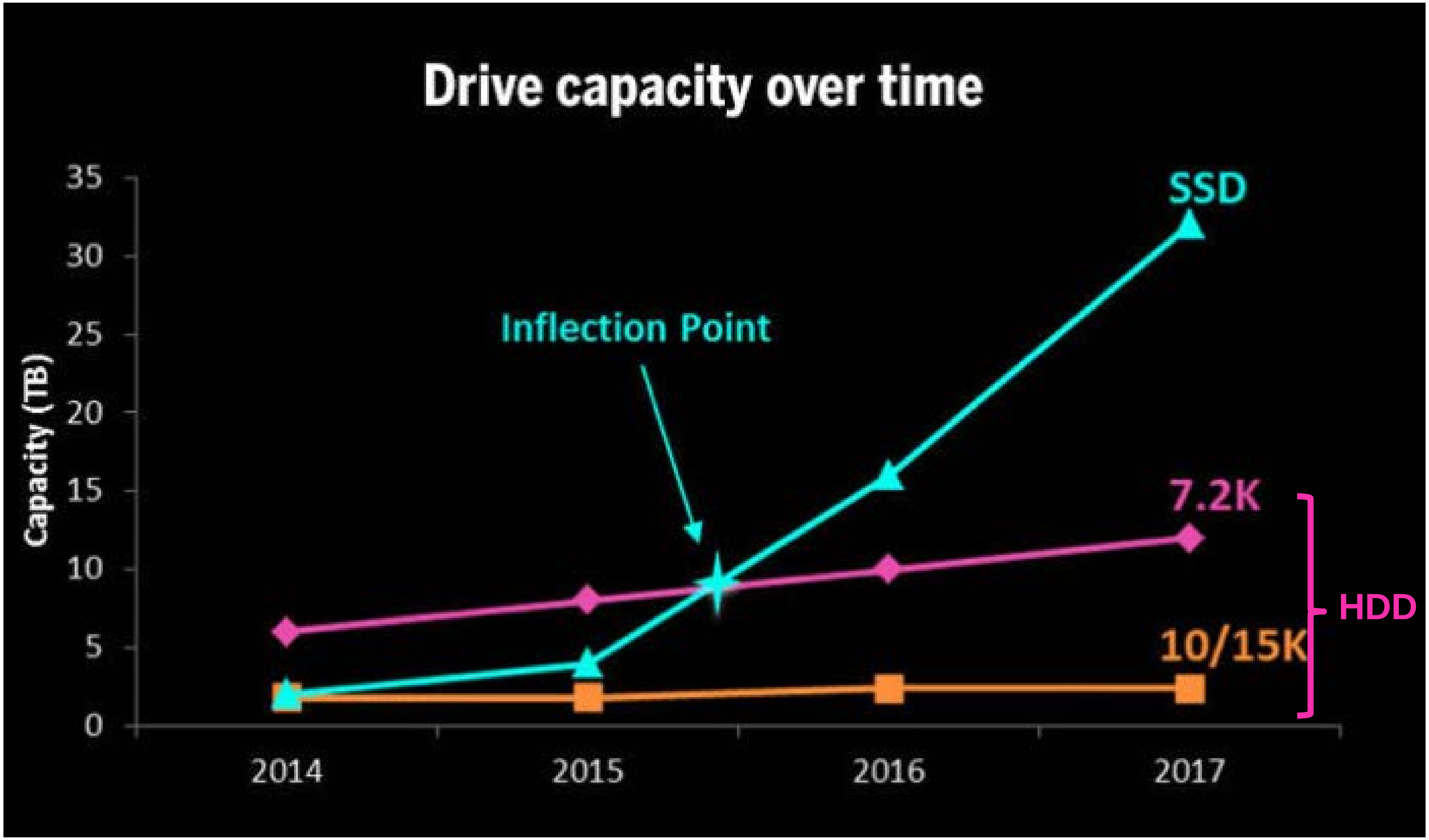
The End of Moore's Law…



- 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

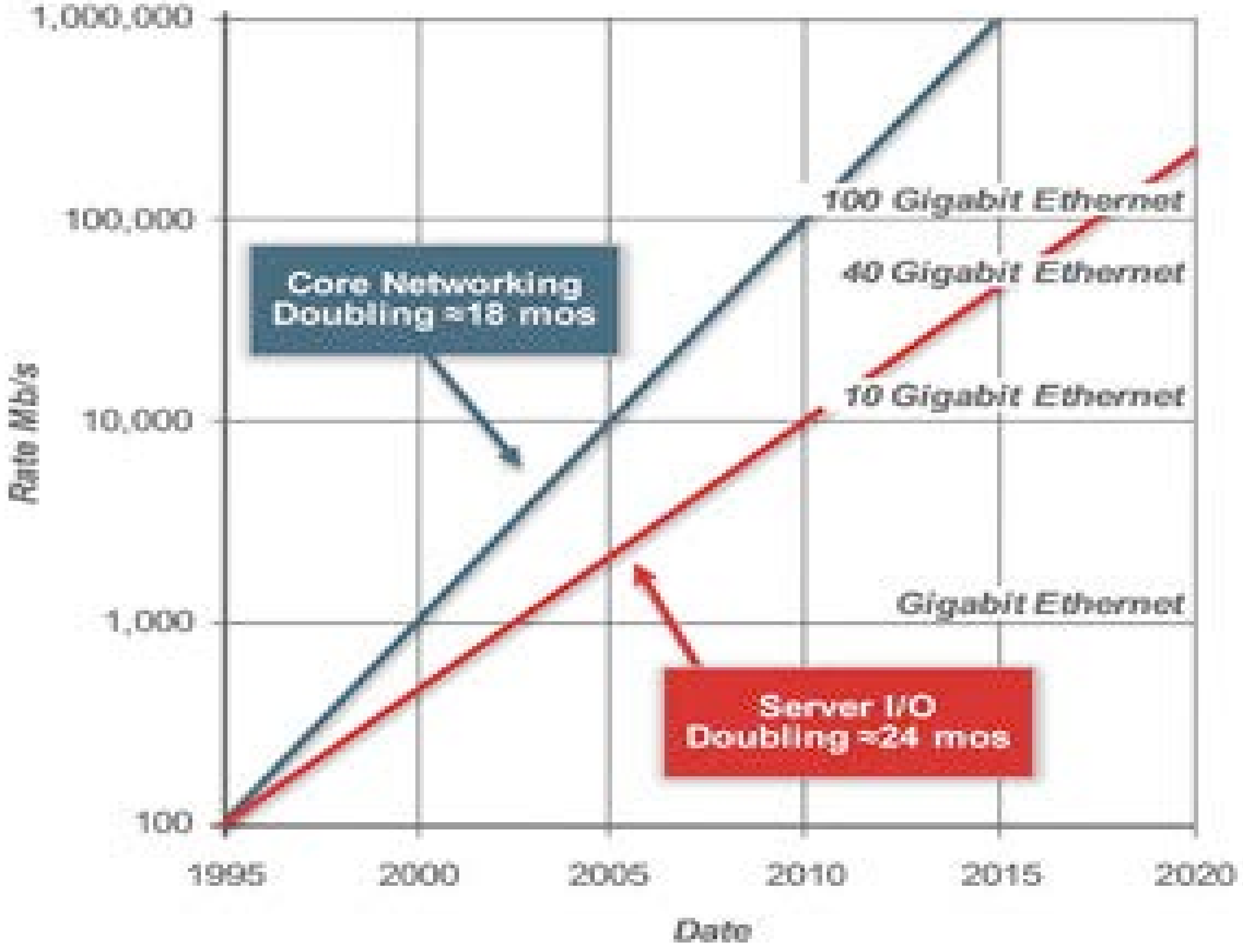
http://files.shareholder.com/downloads/INTC/867590276x0xS50863-16-105/50863/filing.pdf

Storage Capacity



(source: https://www.networkworld.com/article/3153244/data-center/solid-state-drives-are-now-larger-than-hard-disk-drives-the-impact-for-your-data-center.html)

Network Capacity



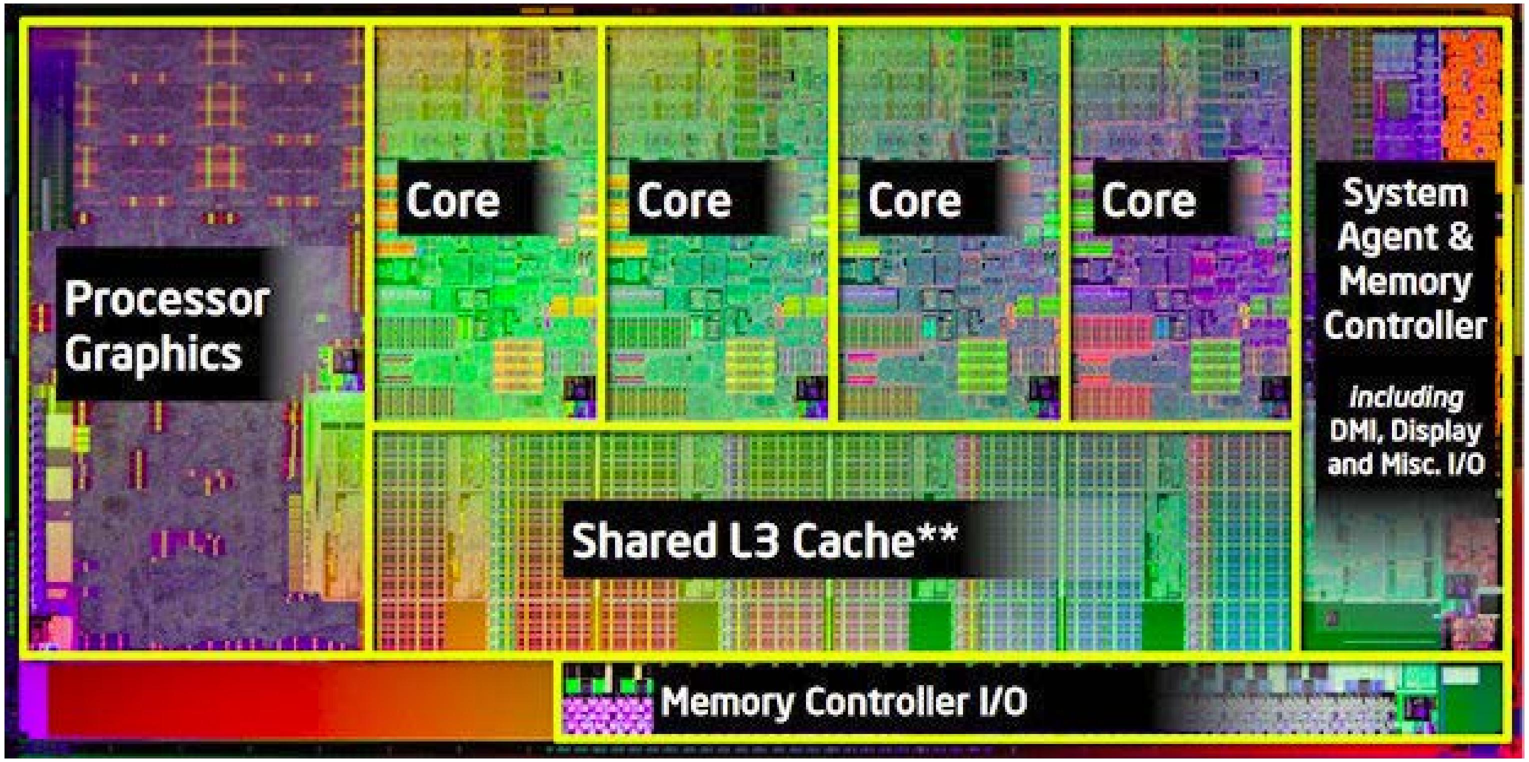
(source: http://www.ospmag.com/issue/article/Time-Is-Not-Always-On-Our-Side)

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!

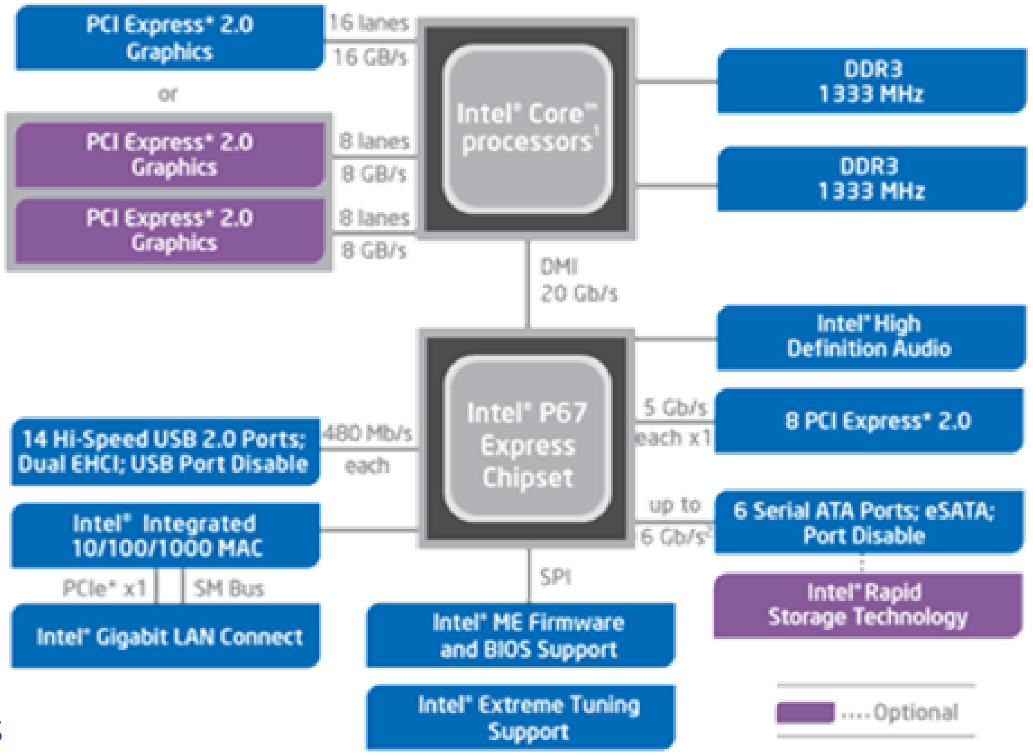
A Modern Processor: Intel Sandy Bridge

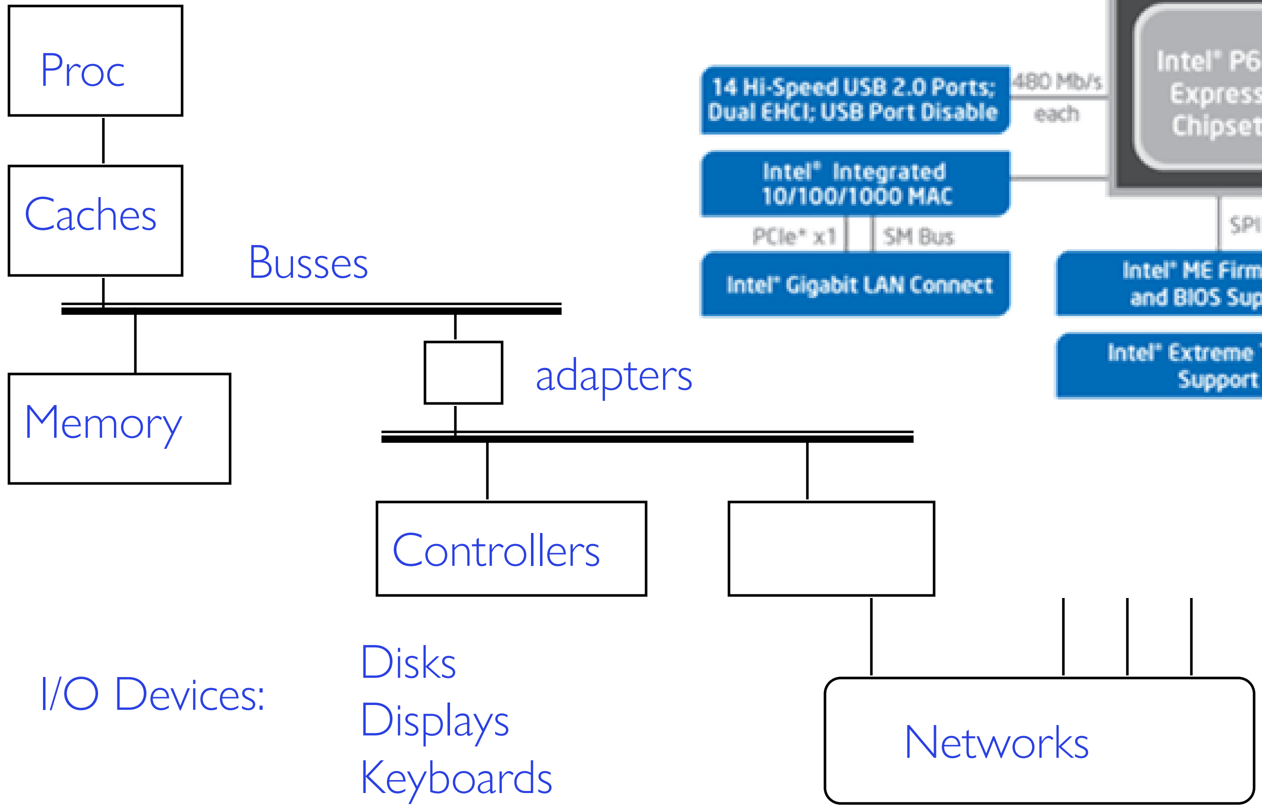


- Package: LGA 1155
  - 1155 pins
  - 95W design envelope
- Cache:
  - L1: 32K Inst, 32K Data (3 clock access)
  - L2: 256K (8 clock access)
  - Shared L3: 3MB – 20MB
- Transistor count:
  - 504 Million (2 cores, 3MB L3)
  - 2.27 Billion (8 cores, 20MB L3)
- Note that ring bus is on high metal layers – above the Shared L3 Cache

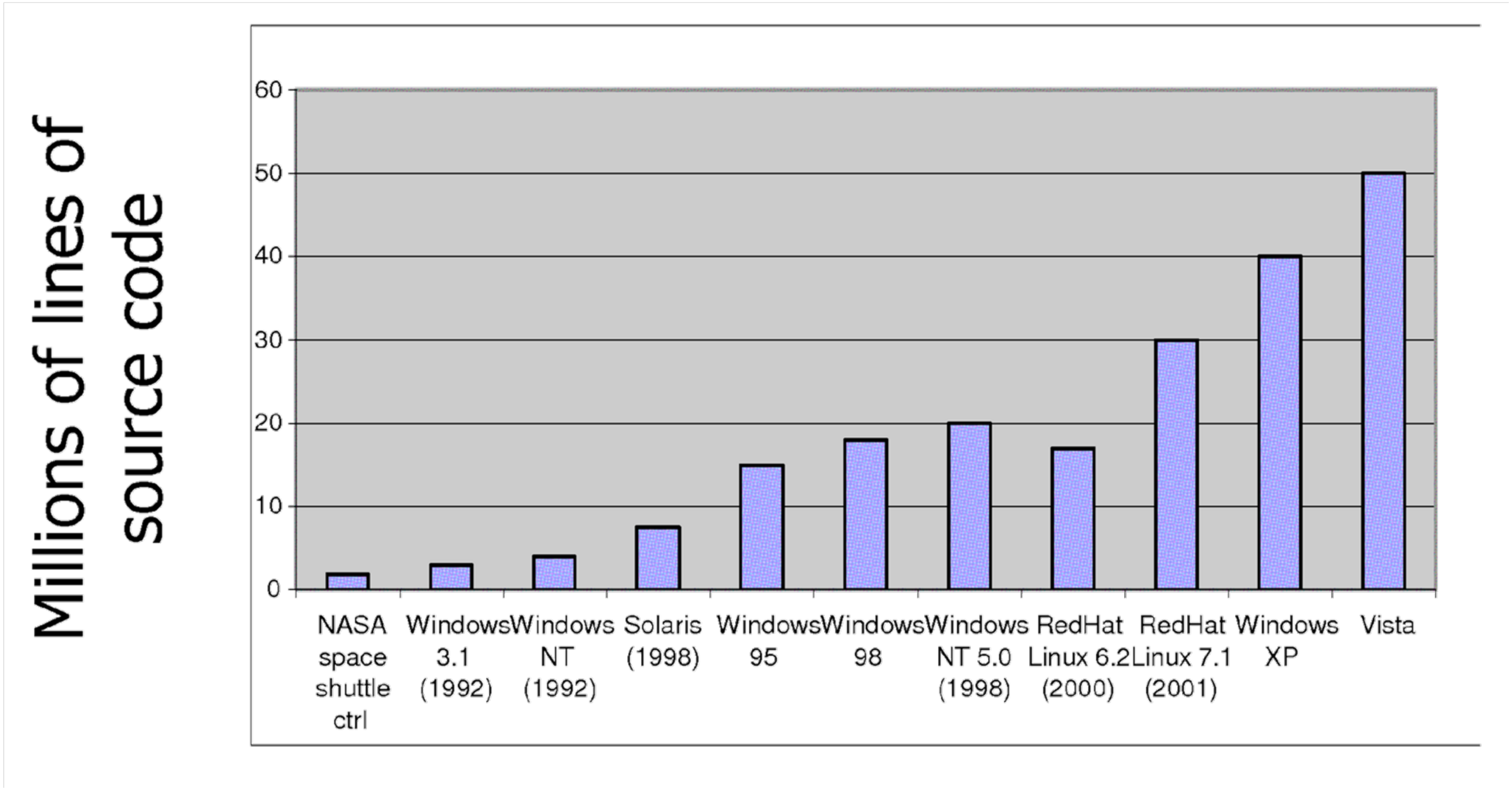
HW Functionality comes with great complexity!

Intel Sandy Bridge I/O Configuration


PCI Express* 2.0 Graphics
16 lanes
16 GB/s
or
PCI Express* 2.0 Graphics
8 lanes
8 GB/s
PCI Express* 2.0 Graphics
8 lanes
8 GB/s
Intel® Core™ processors
DDR3 1333 MHz
DDR3 1333 MHz
DMI 20 Gb/s
Intel® High Definition Audio
Intel® P67 Express Chipset
5 Gb/s each x1
8 PCI Express* 2.0
14 Hi-Speed USB 2.0 Ports; Dual EHCI; USB Port Disable
480 Mb/s each
up to 6 Gb/s
6 Serial ATA Ports; eSATA; Port Disable
Intel® Integrated 10/100/1000 MAC
PCIe* x1
SM Bus
Intel® Gigabit LAN Connect
SPI
Intel® ME Firmware and BIOS Support
Intel® Rapid Storage Technology
Intel® Extreme Tuning Support
....Optional



Increasing Software Complexity

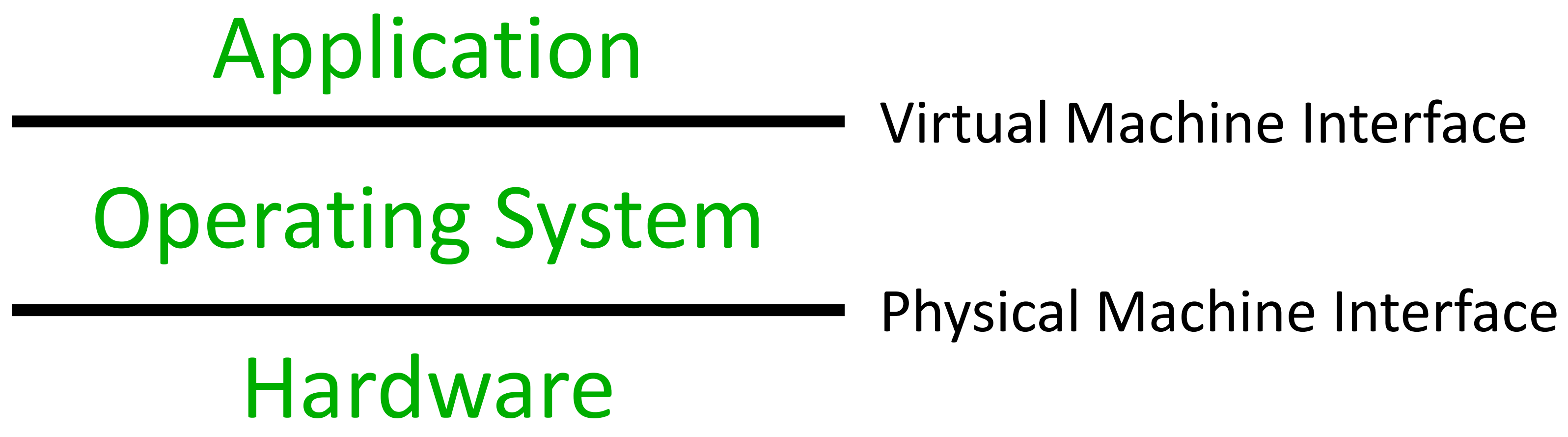


From MIT's 6.033 course

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?

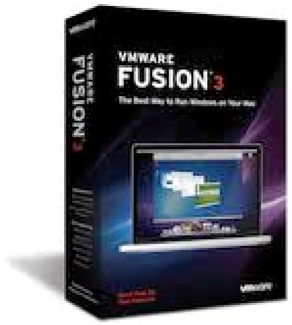
OS Tool: Virtual Machine Abstraction



- 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)

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)

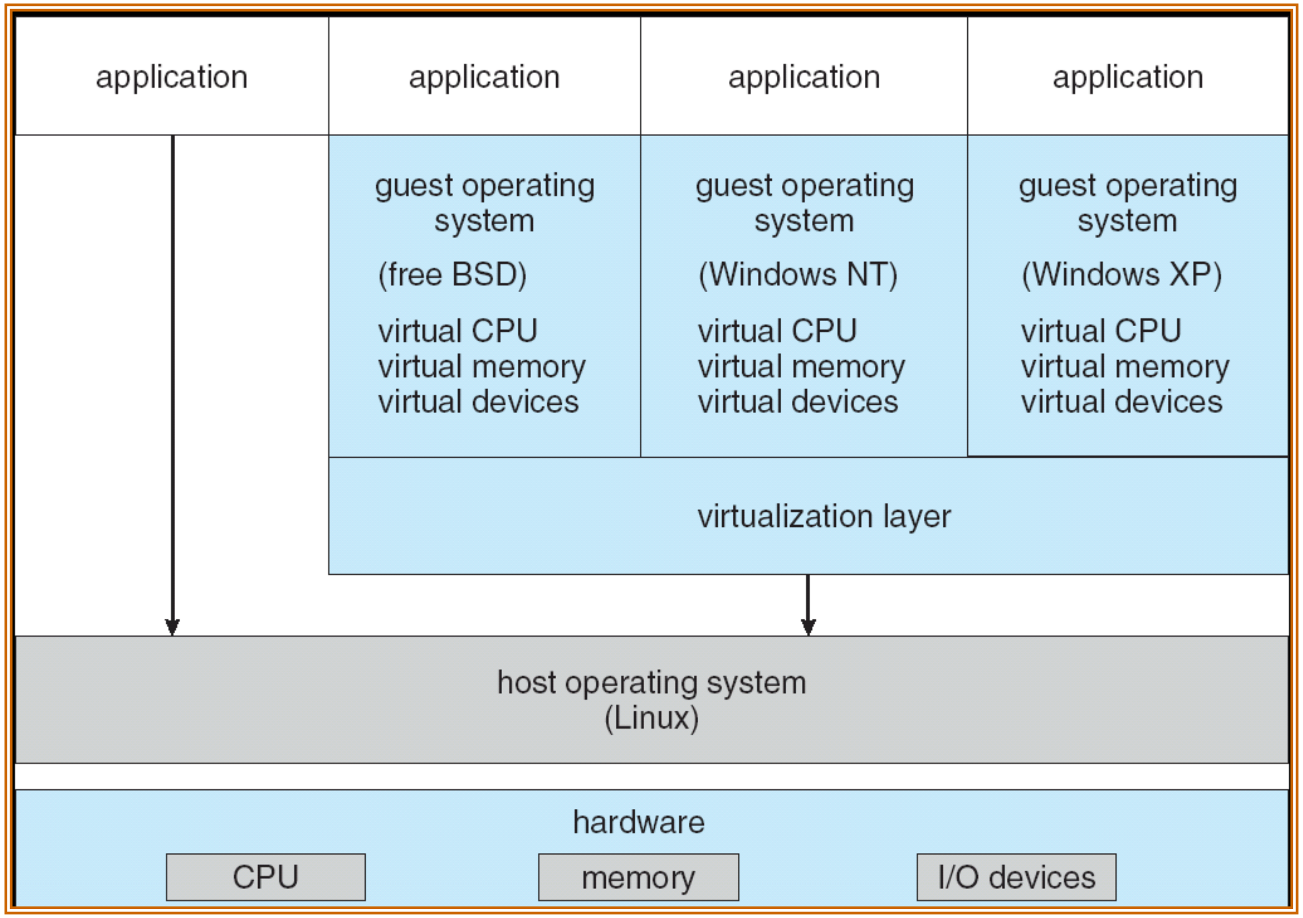


Process VMs

- 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 ⇒ windowing system
    - » Ethernet card ⇒ 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

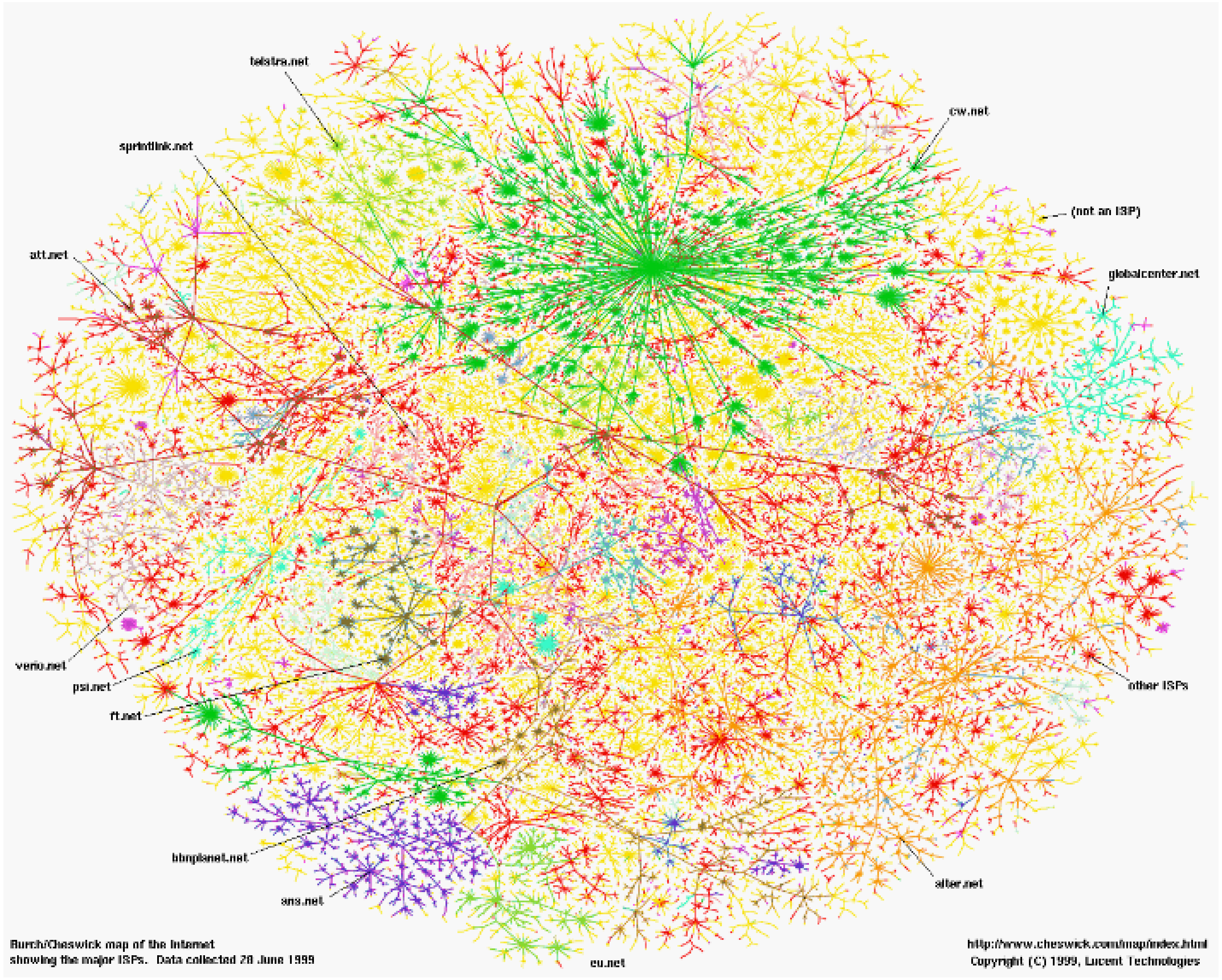
System Virtual Machines: Layers of OSs

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

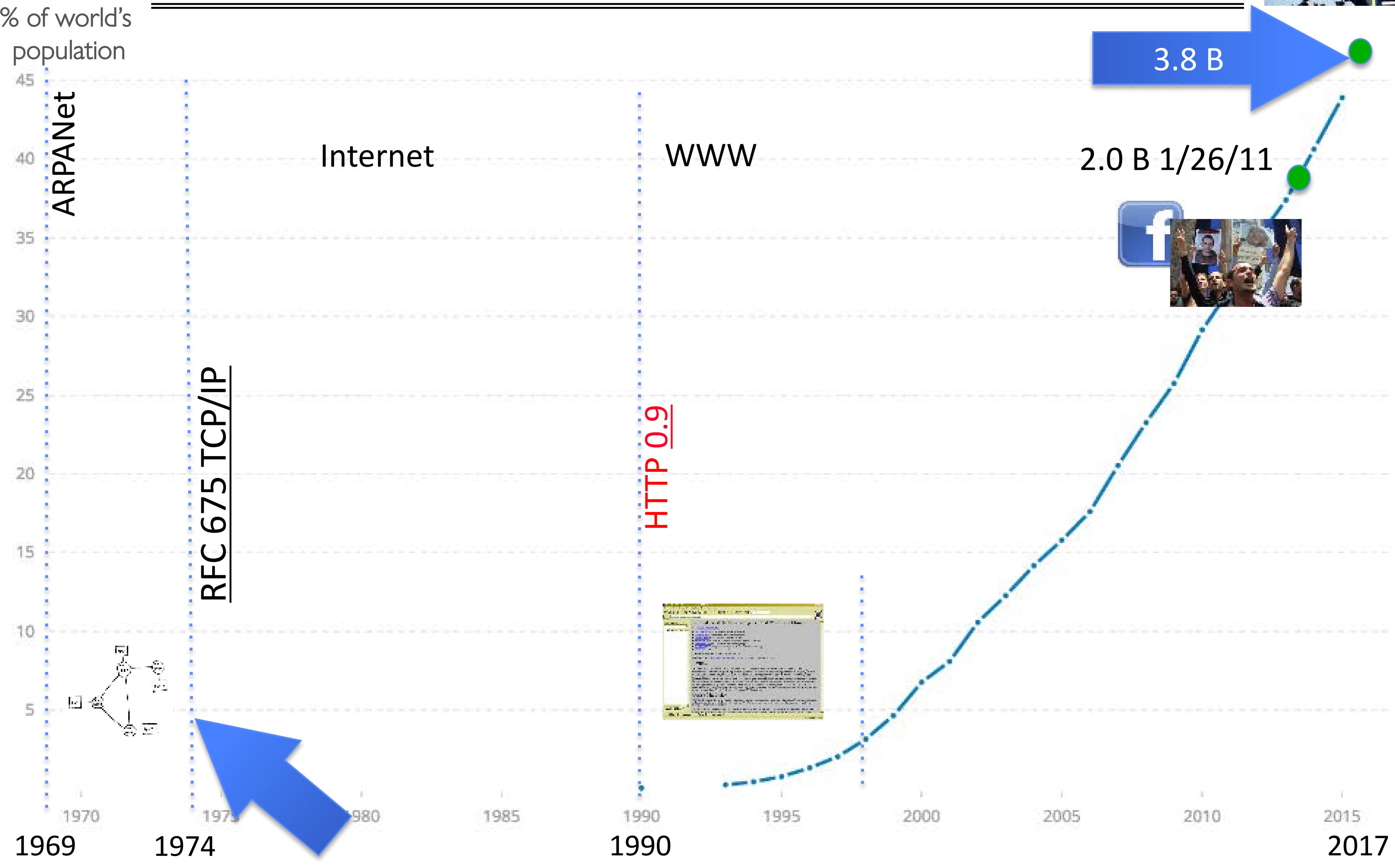


5 min break

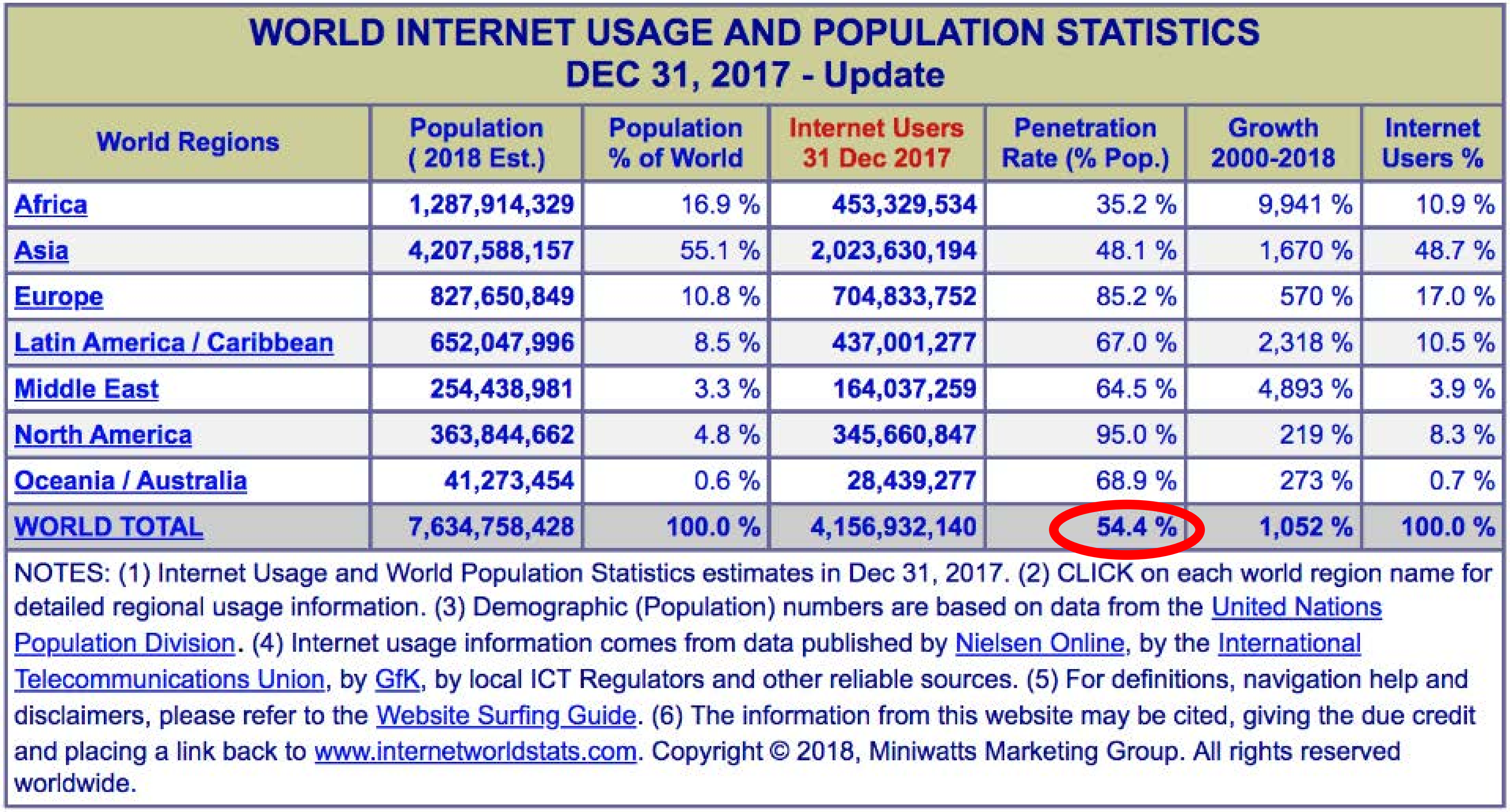
Greatest Artifact of Human Civilization…


telstra.net
cw.net
sprintlink.net
(not an ISP)
att.net
globalcenter.net
verio.net
psi.net
ft.net
other ISPs
bbnplanet.net
ans.net
alter.net
eu.net
Burch/Cheswick map of the Internet
showing the major ISPs. Data collected 28 June 1999
http://www.cheswick.com/map/index.html
Copyright (C) 1999, Lucent Technologies

Internet Scale: Over 3.8 Billion Users!



Internet Scale: Over 3.8 Billion Users!



WORLD INTERNET USAGE AND POPULATION STATISTICS
DEC 31, 2017 - Update

World Regions	Population (2018 Est.)	Population % of World	Internet Users 31 Dec 2017	Penetration Rate (% Pop.)	Growth 2000-2018	Internet Users %
Africa	1,287,914,329	16.9 %	453,329,534	35.2 %	9,941 %	10.9 %
Asia	4,207,588,157	55.1 %	2,023,630,194	48.1 %	1,670 %	48.7 %
Europe	827,650,849	10.8 %	704,833,752	85.2 %	570 %	17.0 %
Latin America / Caribbean	652,047,996	8.5 %	437,001,277	67.0 %	2,318 %	10.5 %
Middle East	254,438,981	3.3 %	164,037,259	64.5 %	4,893 %	3.9 %
North America	363,844,662	4.8 %	345,660,847	95.0 %	219 %	8.3 %
Oceania / Australia	41,273,454	0.6 %	28,439,277	68.9 %	273 %	0.7 %
WORLD TOTAL	7,634,758,428	100.0 %	4,156,932,140	54.4 %	1,052 %	100.0 %

NOTES: (1) Internet Usage and World Population Statistics estimates in Dec 31, 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 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) The information from this website may be cited, giving the due credit and placing a link back to www.internetworldstats.com. Copyright © 2018, Miniwatts Marketing Group. All rights reserved worldwide.

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

Not Only PCs connected to the Internet

- Smartphone shipments exceed PC shipments!
- 2011 shipments:
  - 487M smartphones
  - 414M PC clients
    - » 210M notebooks
    - » 112M desktops
    - » 63M tablets
  - 25M smart TVs

1.53B in 2017

262.5M in 2017

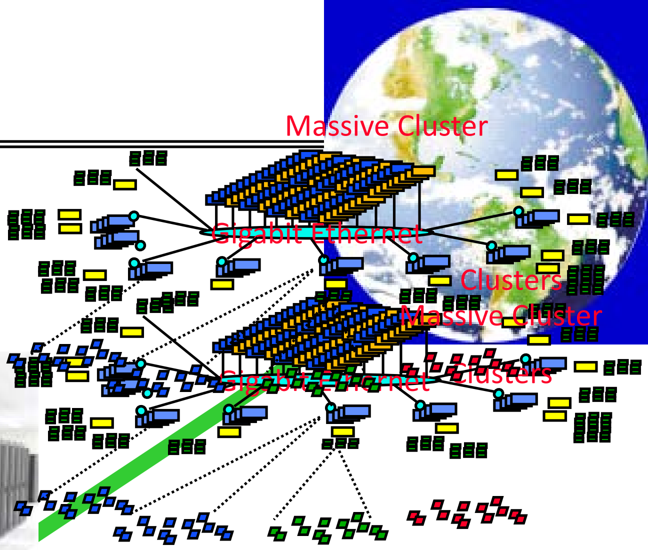
164M in 2017

39.5M in 2017

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

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

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



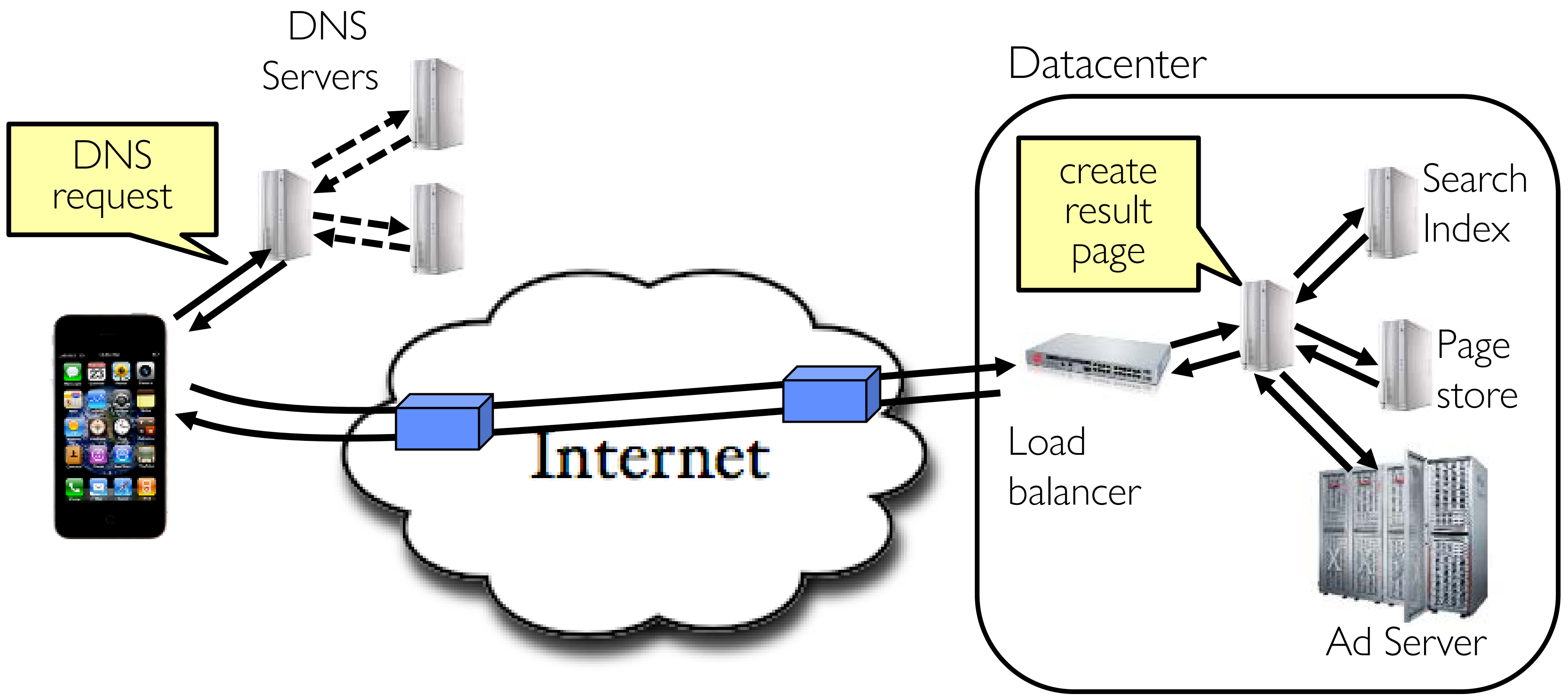
Internet Connectivity

MEMS for Sensor Nets

Scalable, Reliable, Secure Services

Databases
Information Collection
Remote Storage
Online Games
Commerce
…

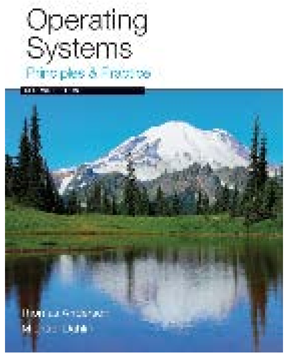
Example: What's in a Search Query?

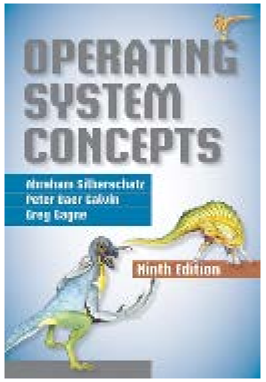


- Complex interaction of multiple components in multiple administrative domains
  - Systems, services, protocols, …

Infrastructure, Textbook & Readings

- Infrastructure
  - Website: http://cs162.eecs.berkeley.edu
  - Piazza: https://piazza.com/berkeley/fall2018/cs162
  - Webcast: Cal Central - https://calcentral.berkeley.edu/academics/teaching-semester/fall-2018/class/compsci-162
- Textbook: Operating Systems: Principles and Practice (2nd Edition) Anderson and Dahlin
- Recommend: Operating Systems Concepts, 9th Edition Silbershatz, Galvin, Gagne
  - Copies in Bechtel
- Online supplements
  - See course website
  - Includes Appendices, sample problems, etc.
  - Networking, Databases, Software Eng, Security
  - Some Research Papers!





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

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

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!

Getting started

- Start homework 0, this Monday, 9/3
  - 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

Grading

- 42% three midterms (14% each)
  - Thursday, 9/26, 5-7pm (no class)
  - Monday, 10/29, 5-7pm (no class)
  - Wednesday, 11/28, 5-7pm (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

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

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

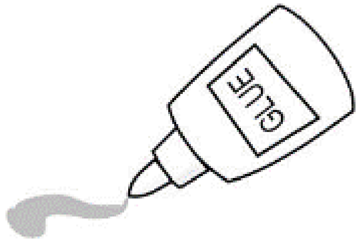
Lecture Goal

Interactive!!!

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

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??

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

"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