Introduction to CS162

David E. Culler

CS162 – Operating Systems and Systems Programming

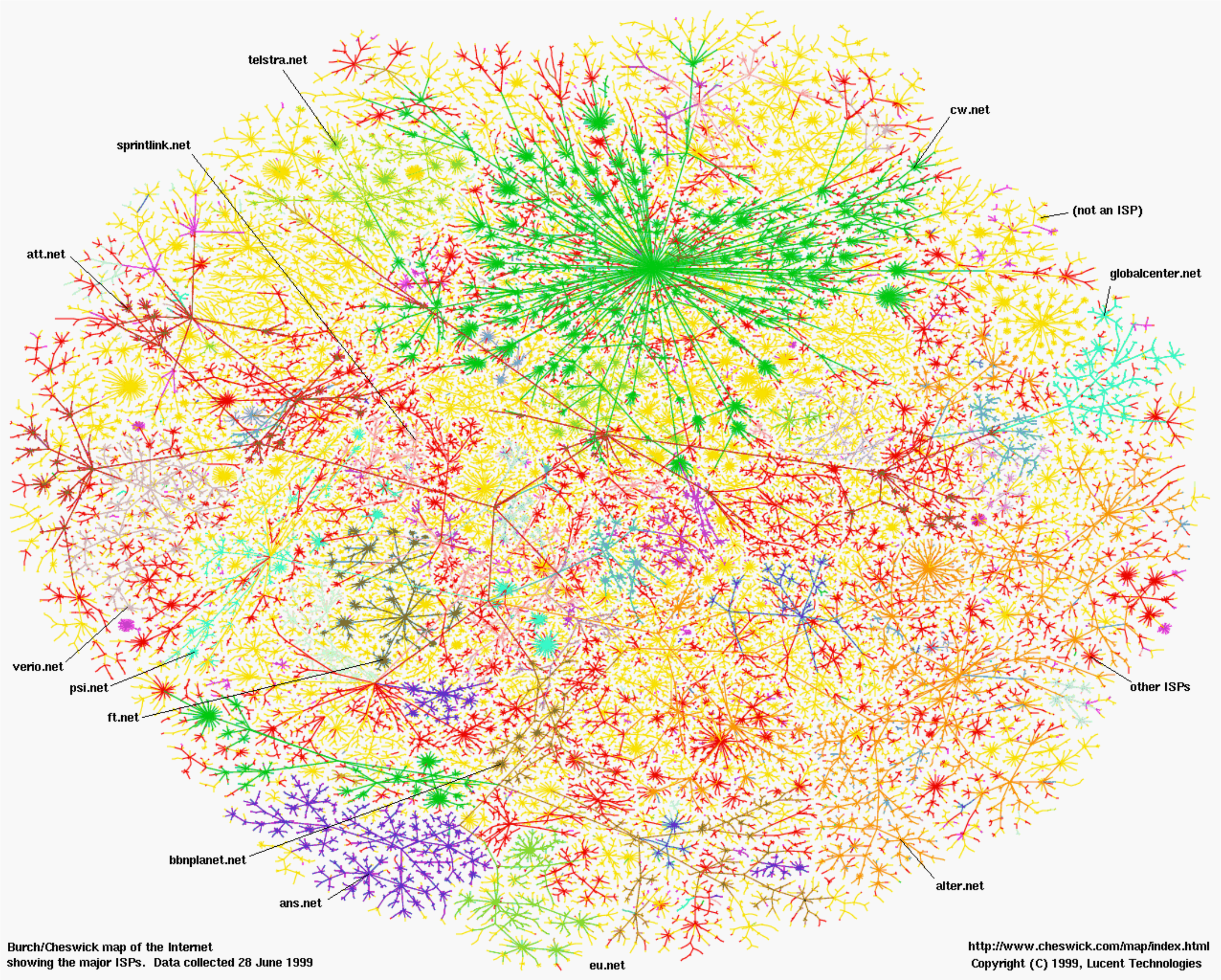
http://cs162.eecs.berkeley.edu/

Lecture 1

August 29, 2014

Read A&D Ch1
HW0 out, due 9/8

Greatest Artifact of Human Civilization …



3 Billion by …

2.8 B

ARPANet

Internet

WWW

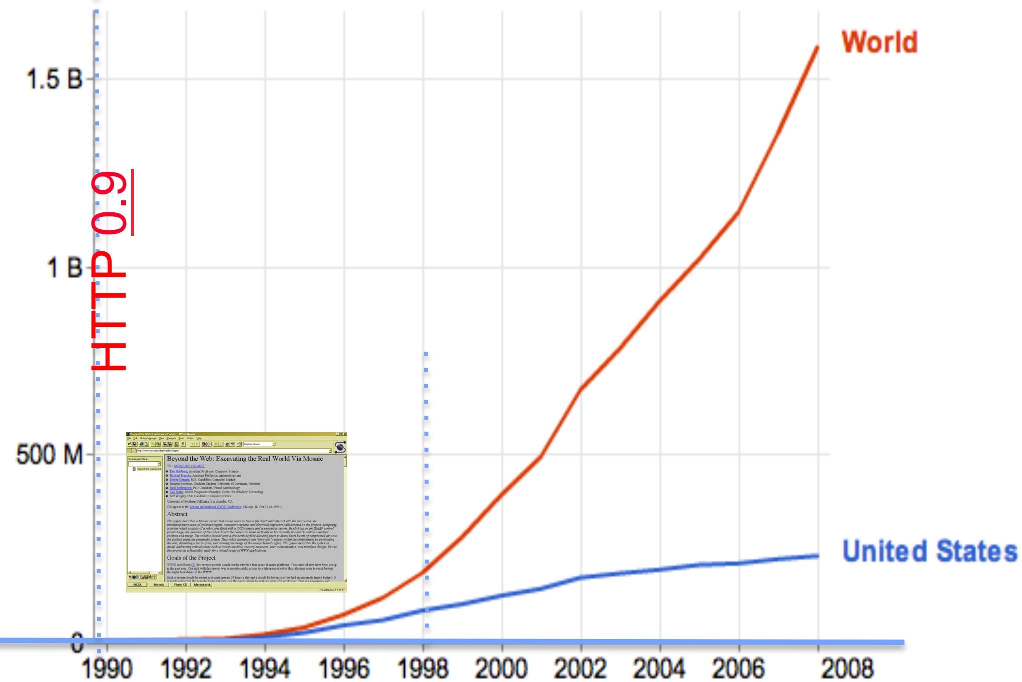
2.0 B 1/26/11

RFC 675 TCP/IP

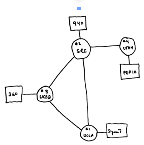
HTTP 0.9

Internet users

People with access to the Internet. More info »



Data source: World Bank, World Development Indicators - Last updated December 21, 2010



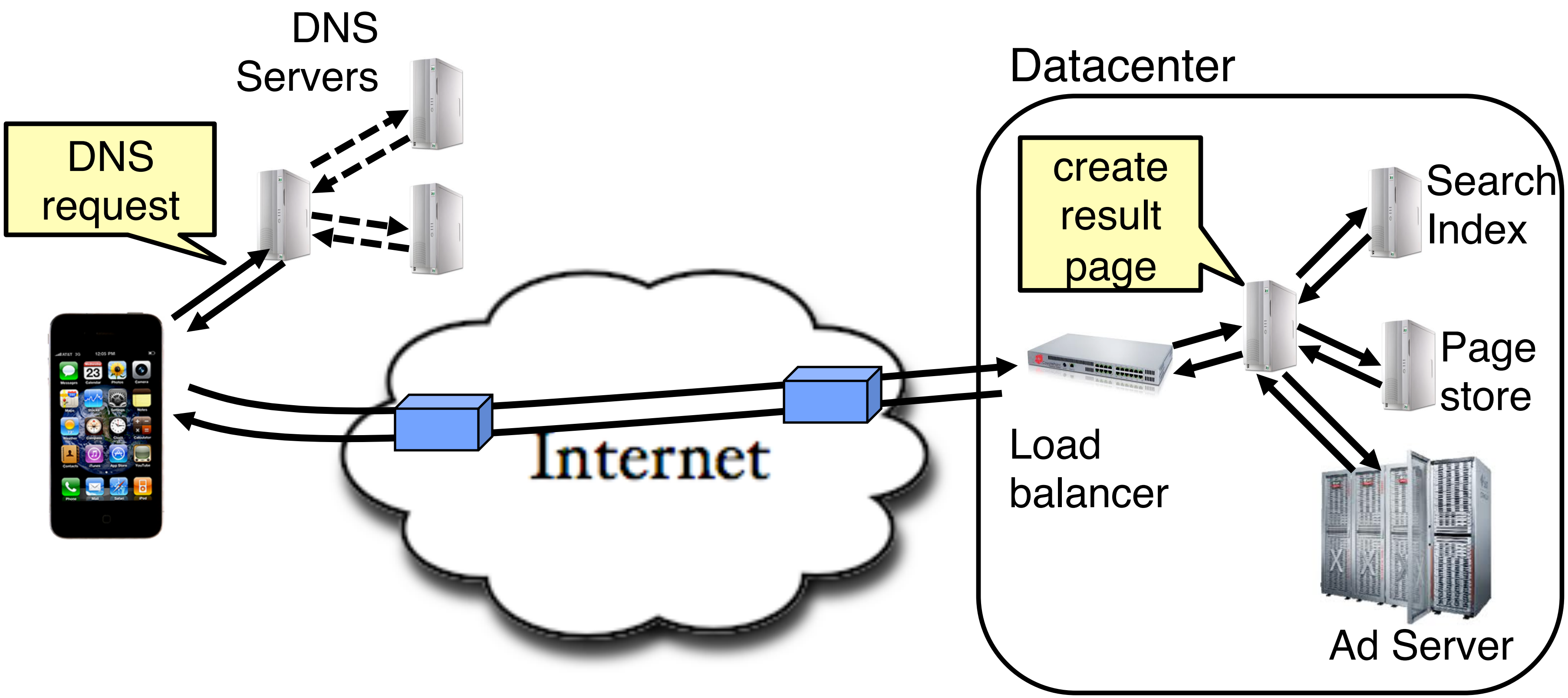
1969 1974 1990 2010

Operating Systems at the heart of it all …

- **Make the incredible advance in the underlying hardware available to a rapid evolving body of applications.**
  - Processing, Communications, Storage, Interaction

- **The key building blocks**
  - Scheduling
  - Concurrency
  - Address spaces
  - Protection, Isolation, Security
  - Networking, distributed systems
  - Persistent storage, transactions, consistency, resilience
  - Interfaces to all devices

Example: What's in a Search Query?



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

Why take CS162?

- Some of you will actually design and build operating systems or components of them.
  - Perhaps more now than ever
- Many of you will create systems that utilize the core concepts in operating systems.
  - Whether you build software or hardware
  - The concepts and design patterns appear at many levels
- All of you will build applications, etc. that utilize operating systems
  - The better you understand their design and implementation, the better use you'll make of them.

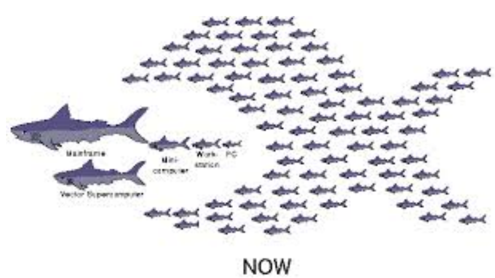
Today's Objectives

- Introduce you to Operating System design
- Introduce the CS162 instructional team
- Establish expectations and logistics
- Maybe get a little excited about how OS is so essential in creating and advancing this "connected world"

CS162 Team - me

- **David Culler (culler@berkeley.edu)**
  - 465 Soda Hall (amplab)
  - http://www.cs.berkeley.edu/~culler
  - Office hours: M 2-3, Tu 9-10, W 2-3 in 449 Soda

- **Bit'o systems experience**
  - Cray Time Sharing System
  - Active Messages
  - Massive High Performance Clusters
  - TinyOS / Berkeley Motes
  - BOSS



CS162 Team - GSIs

- Vaishaal Shankar
  - Head GSI,
  - cs162-ta@inst.eecs.berkele
  - Sec 108: W 4-5
  - OH: Tu/Th 5-6, 751 Soda

- Arka Bhattacharya
  - cs162-tb@inst.eecs.berkeley.
  - Sec: 107 W 3-4, 105 Th 12-1
  - OH: W 5:30-7:30, 411 Soda

- Kaifei Chen
  - cs126-tc@inst.eecs.berkeley.ed
  - Sec: 109 Th 10-11
  - OH: Tu 9-11am, 411 Soda

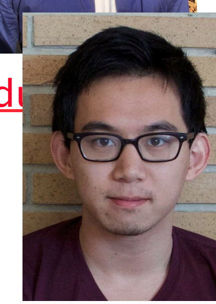
- Stanley Hung
  - cs162-td@inst.eecs.berkeley
  - Sec: 106 Th 1-2
  - OH: Tu/Th 2-3, 611 Soda

- Jason Jia
  - cs162-te@inst.eecs.berkeley.
  - Sec: 101 W 3-4
  - OH: Tu/Th 1-2pm, Soda 651

- Will Liu
  - cs162-tf@inst.eecs.berkeley.ed
  - Sec: 102 W 4-5
  - OH: Tu/Th 3-4 pm, 611 Soda

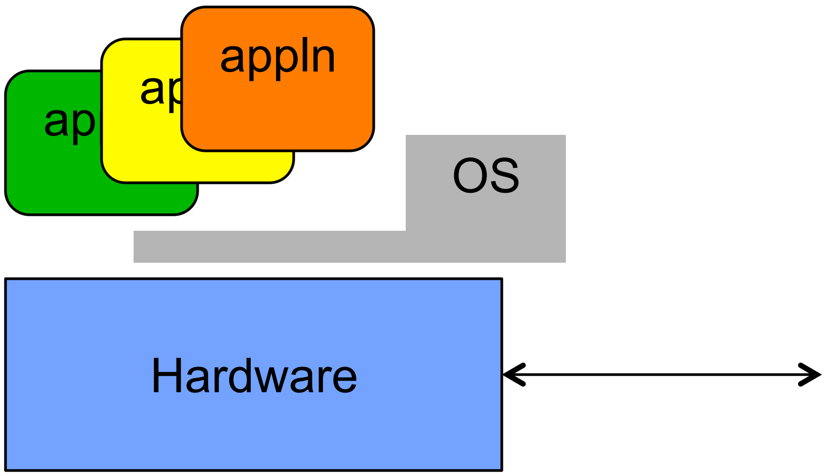
- Issac Tian
  - cs162-tg@inst.eecs.berkeley.
  - Sec: 103 Th 10-11
  - OH: Wed 1:30-2:30, 651 Soda

- Caleb Wang
  - cs162-th@inst.eecs.berkeley.edu
  - Sec: 104 Th 11-12
  - OH: Mon 2-4 pm, 751 Soda

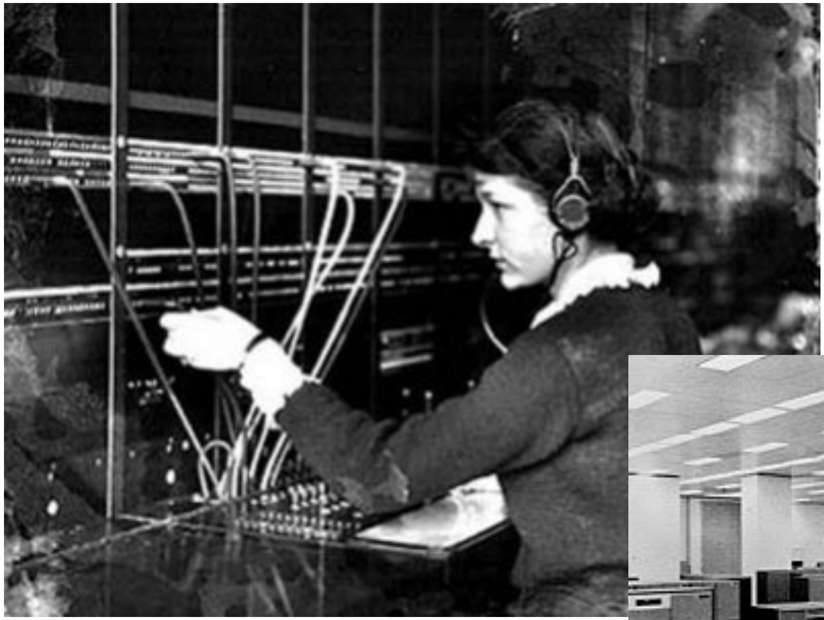


What is an operating system?

- 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

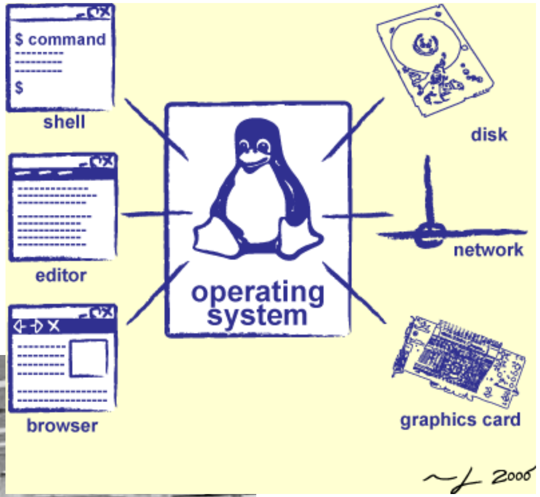


Operator …



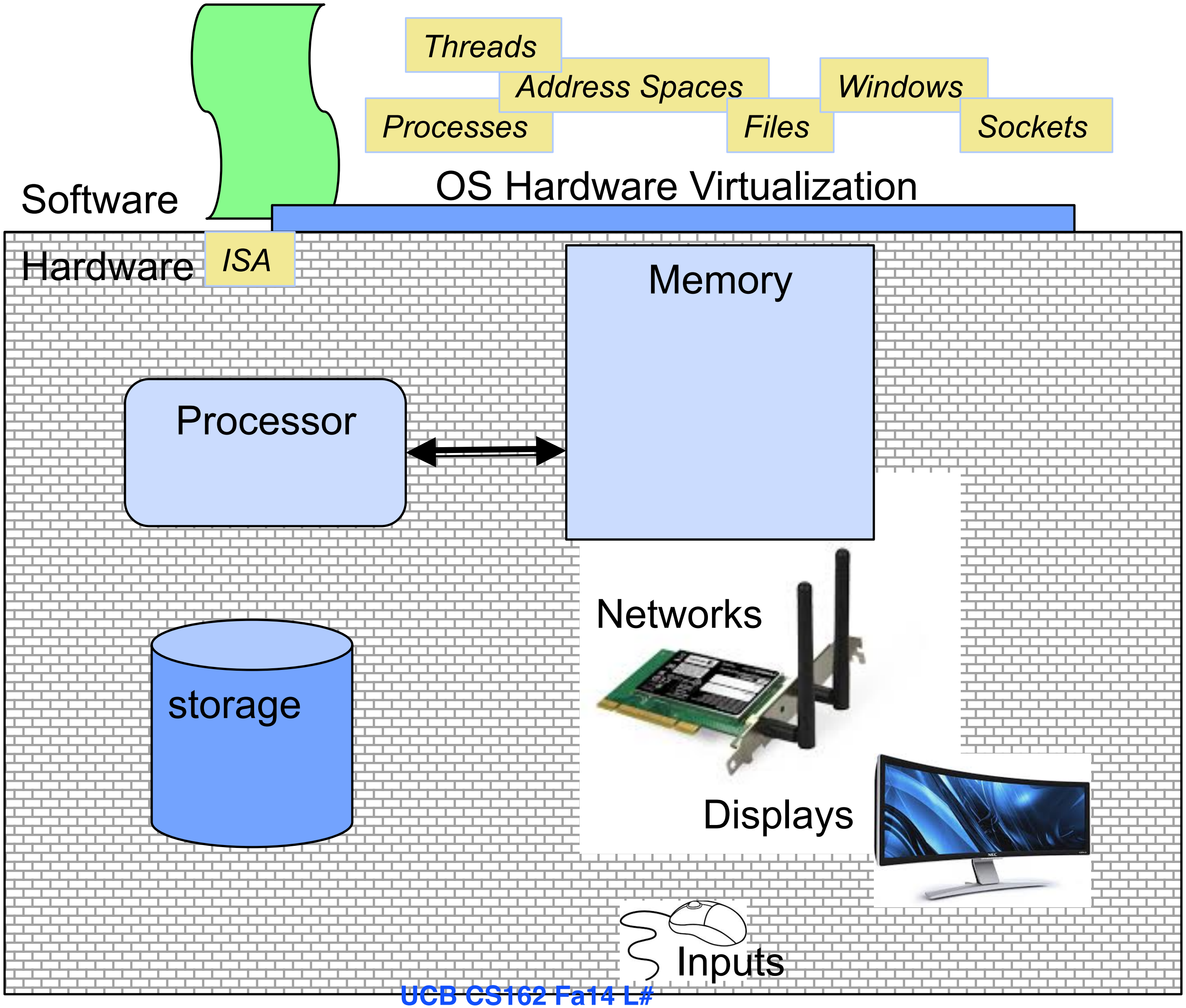
Switchboard Operator

Computer Operators

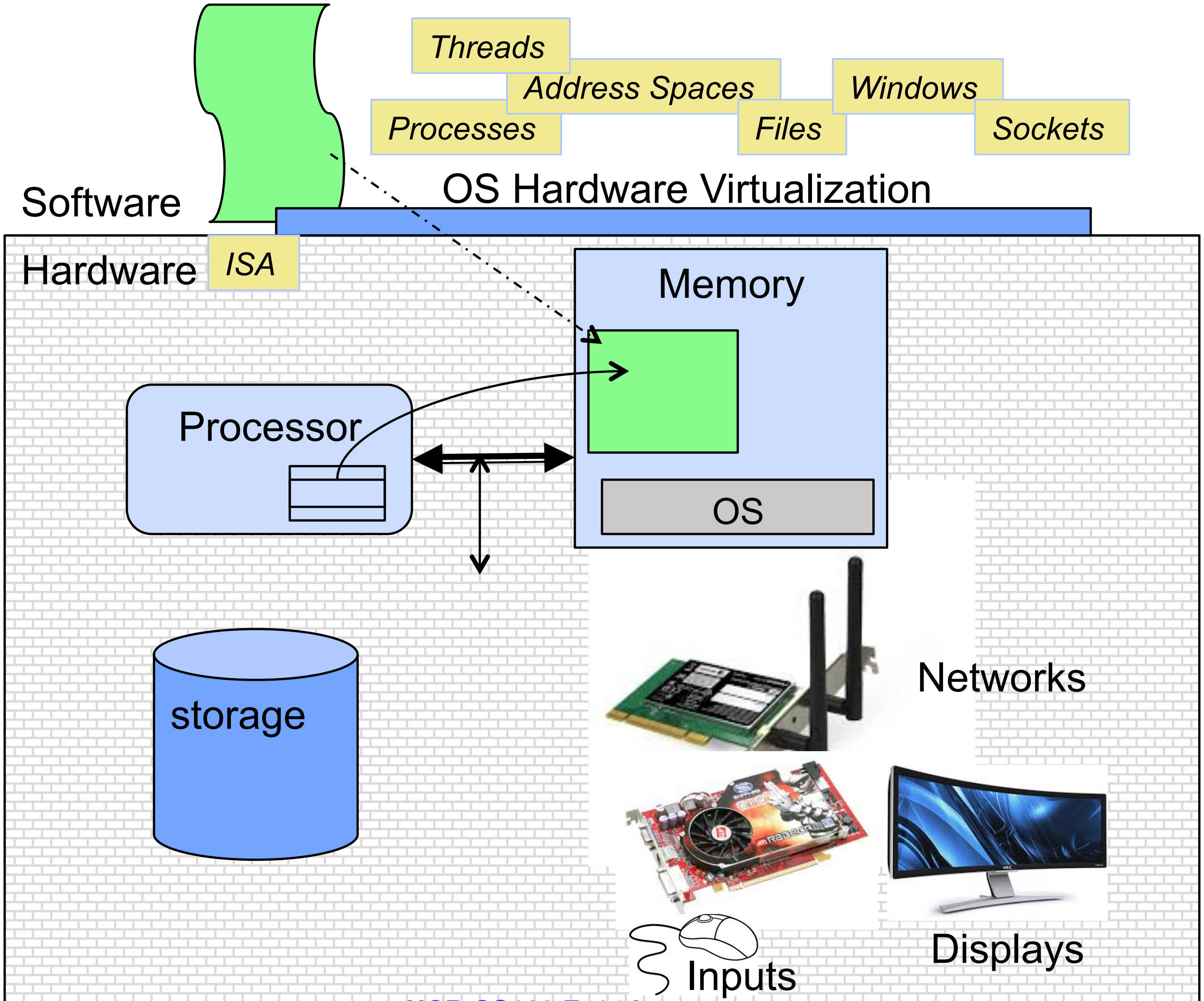


What make something a system?

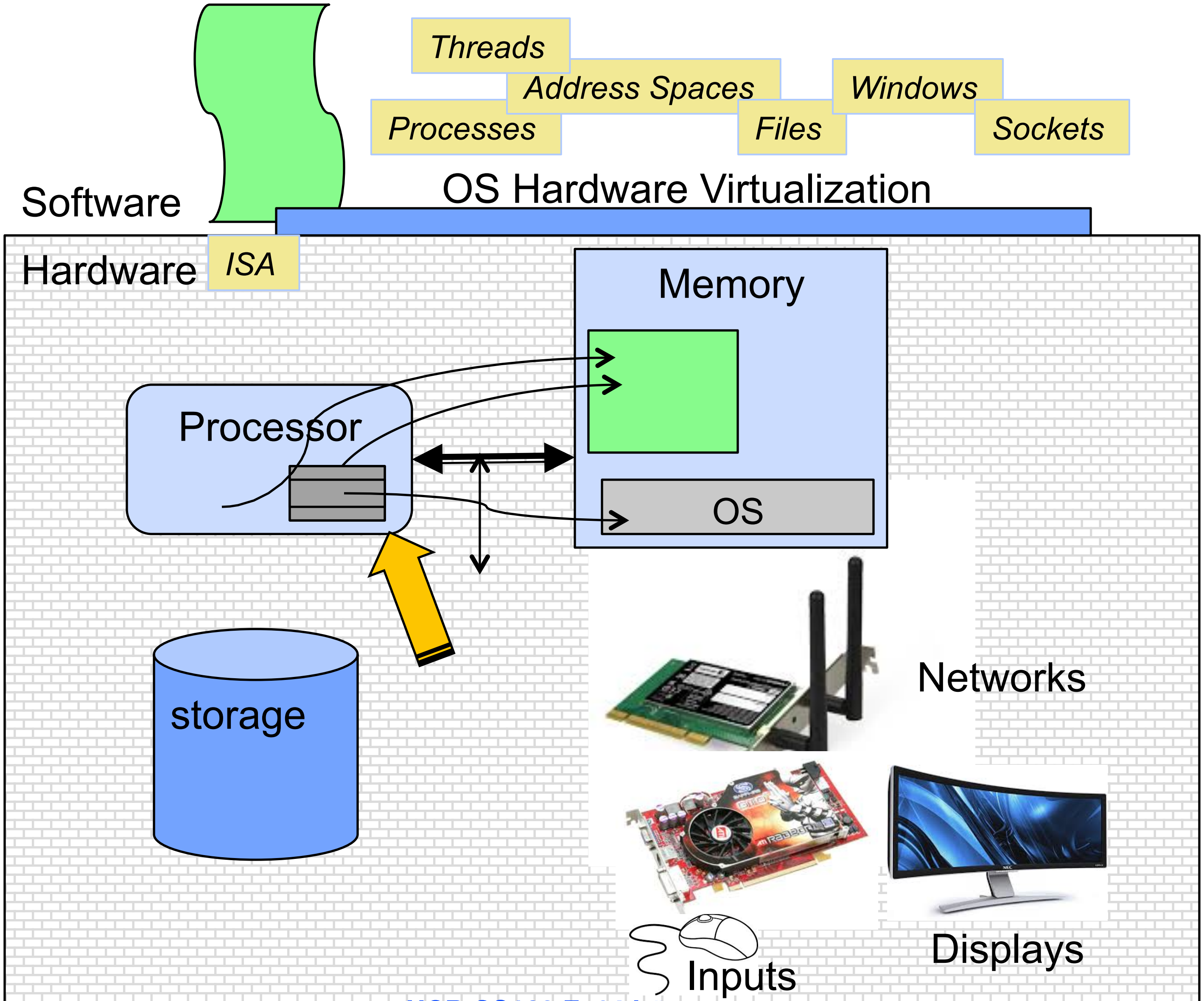
OS Basics: “Virtual Machine” Boundary



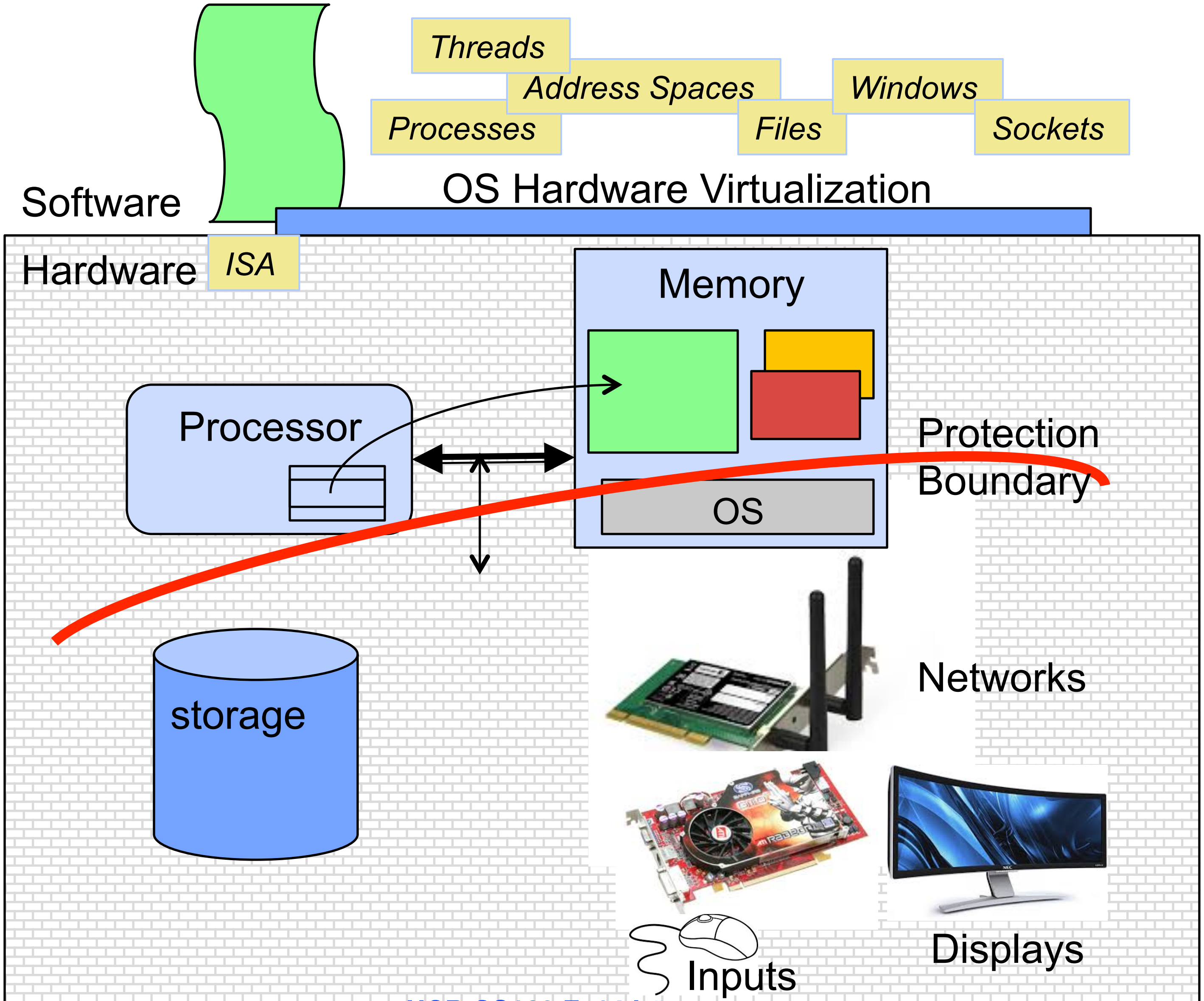
OS Basics: Program => Process



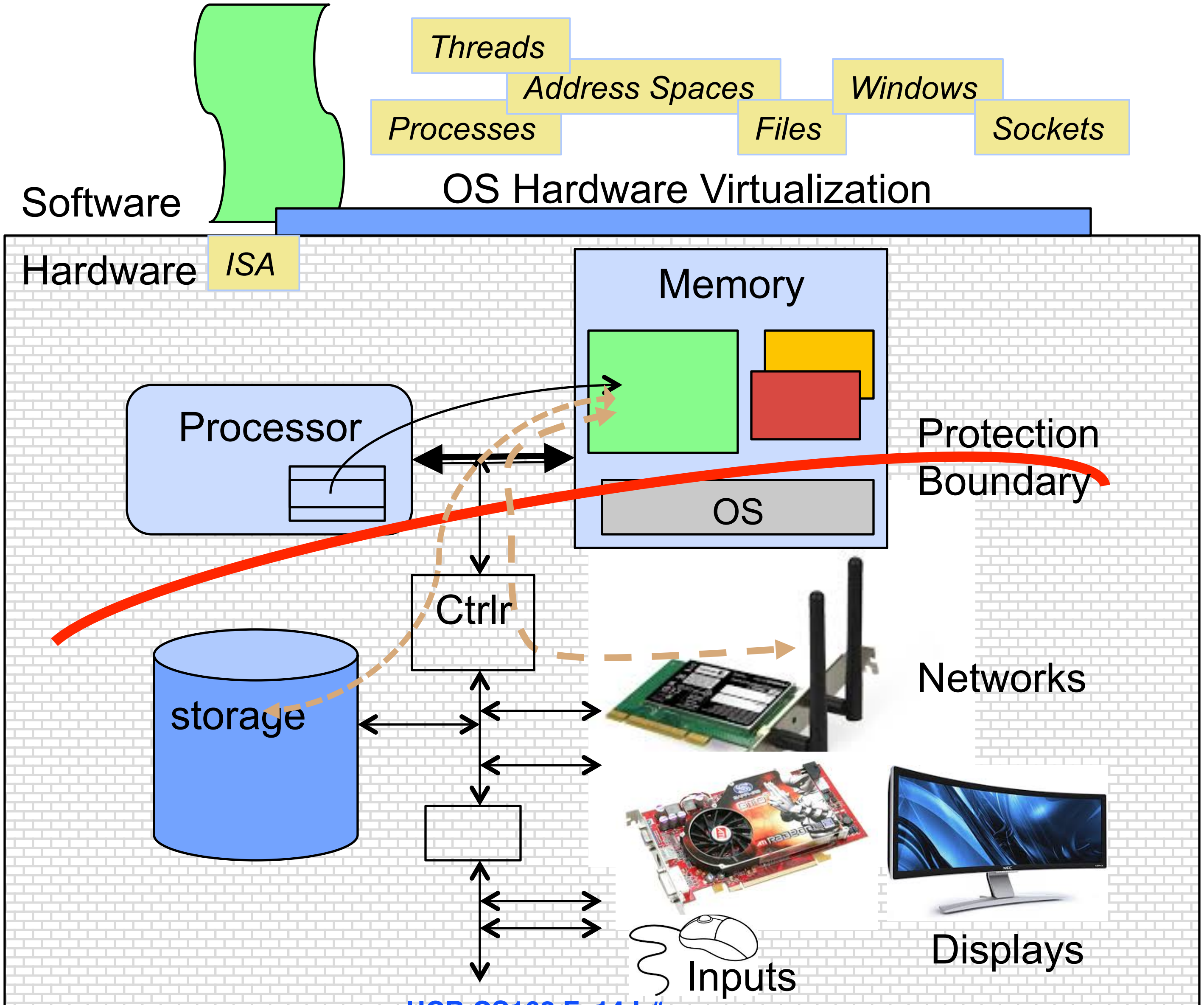
OS Basics: Context Switch



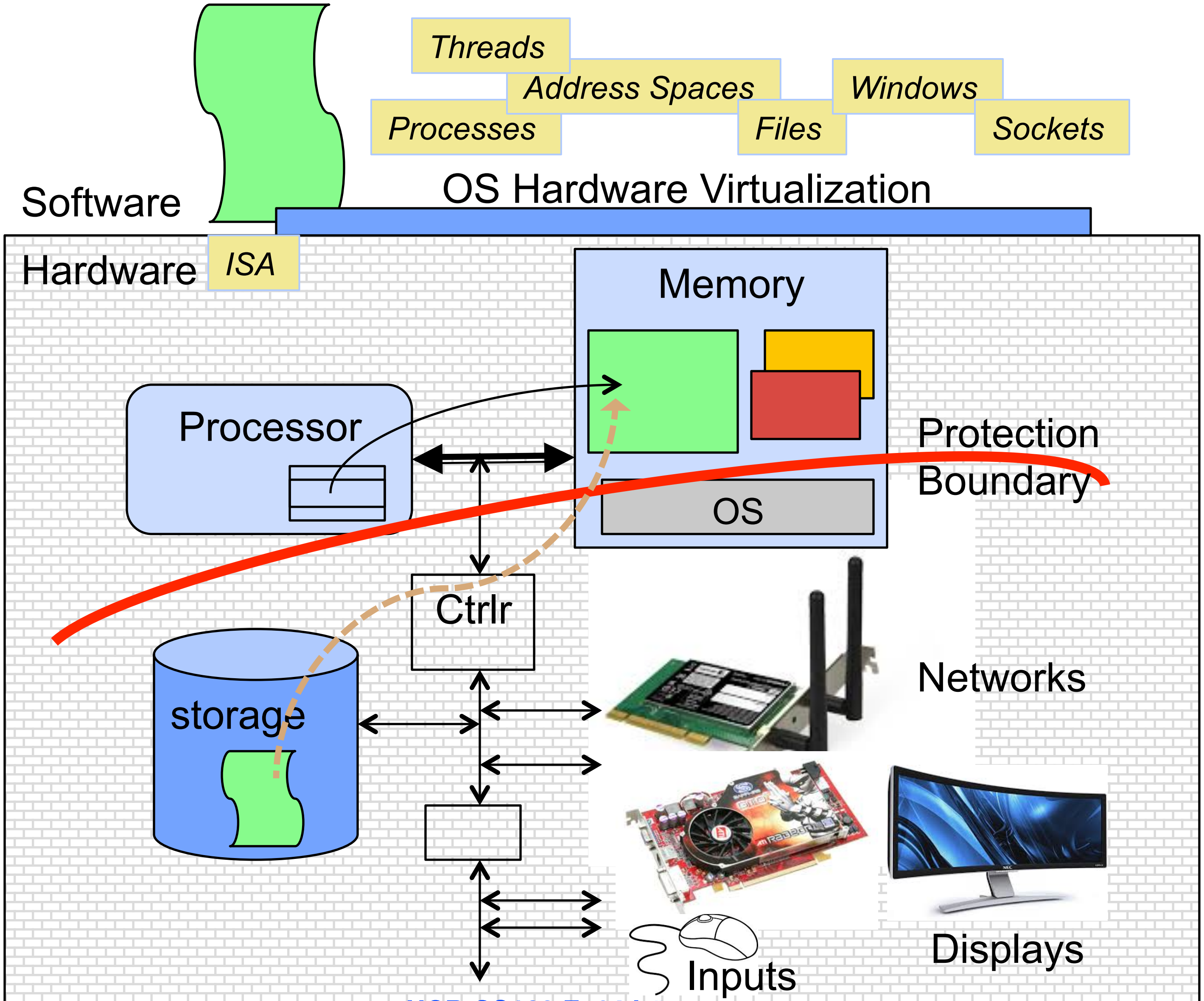
OS Basics: Scheduling, Protection



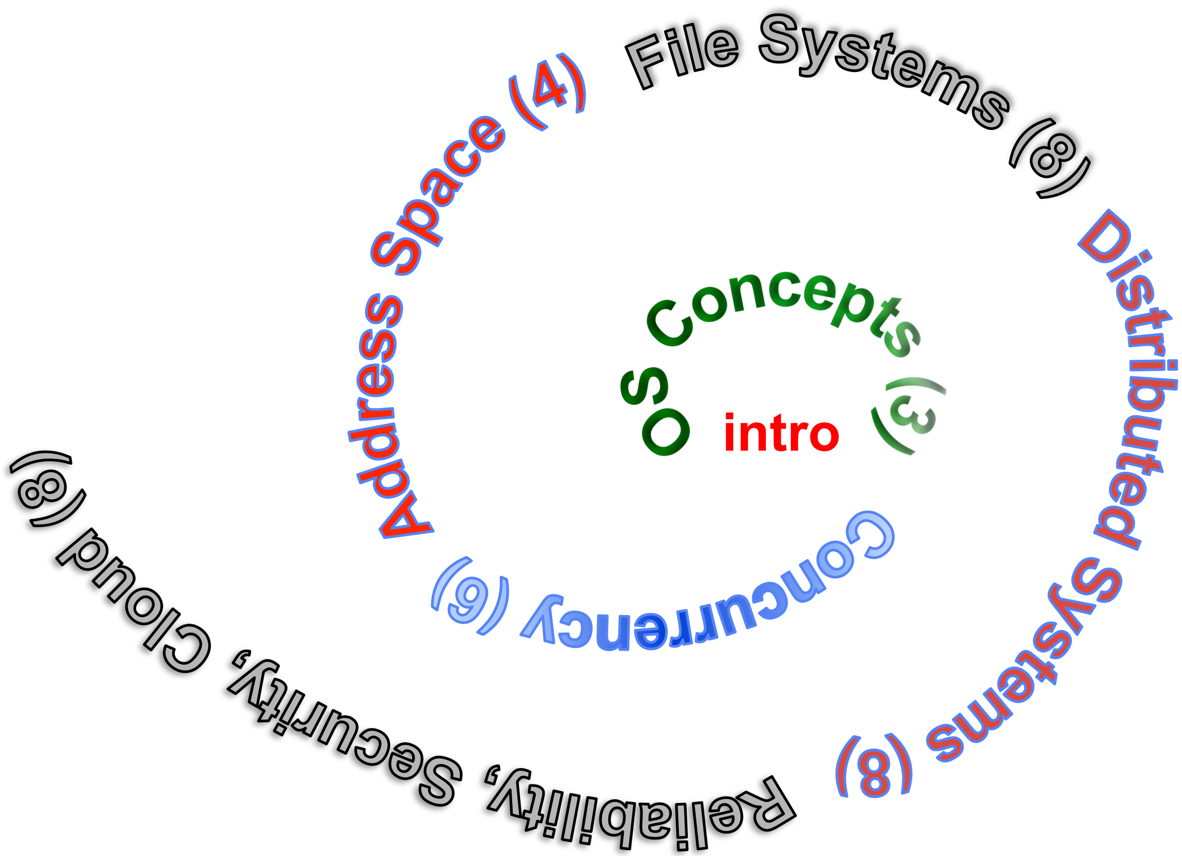
OS Basics: I/O


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

OS Basics: Loading



Course Structure: Spiral

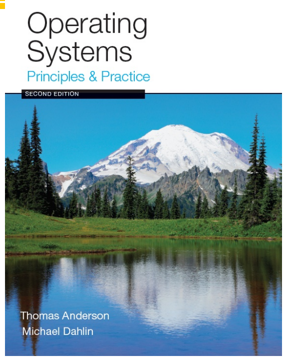


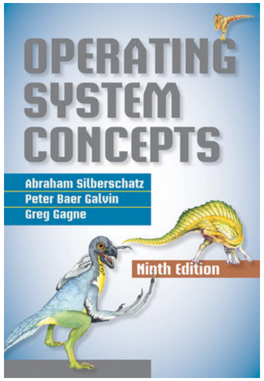
Syllabus

- OS Concepts (3)
  - Process, I/O, Networks and VM
- Concurrency (6)
  - Threads, scheduling, locks, deadlock, scalability, fairness
- Address Space (4)
  - Virtual memory, address translation, protection, sharing
- File Systems (8)
  - i/o devices, file objects, storage, naming, caching, performance, paging, transactions, databases
- Distributed Systems (8)
  - Protocols, N-Tiers, RPC, NFS, DHTs, Consistency, Scalability, multicast
- Reliability & Security
  - Fault tolerance, protection, security
- Cloud Infrastructure

Textbook & Readings

- 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





Learning by Doing

- Three Group Projects
  - 1. Threads & Scheduling (Pintos in C)
  - 2. User-programs (Pintos in C)
  - 3. Key-value store (Java)
- Individual Homework (1-2 weeks)
  - 0. Tools, Autograding, recall C, executable
  - 1. Simple Shell
  - 2. Web server
  - …

Group Project Simulates Industrial Environment

- **Project teams have 3-4 members**
  - Must work in groups in “the real world”
  - Same section much perferred
- **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

Grading

- 40% midterms (~13% each)
- 40% projects (~13% each)
- 15% homework
- 5% participation
- Project grading
  - [10 pts] Initial design
  - [10 pts] Design review
  - [50 pts] Code (3 checkpoints)
  - [30 pts] Final design
  - [0 pts] Peer Evaluation
- Submission via *git push* to release branch
  - Triggers autograder
- Regular *git push* so TA sees your progress

Getting started

- **Start homework 0 immediately**
  - Gets cs162-xx@cory.eecs.berkeley.edu (and other inst m/c)
  - Github account
  - Registration survey
  - Vagrant virtualbox – VM environment for the course
    - Consistent, managed environment on your machine
  - icluster24.eecs.berkeley.edu is same
  - Get familiar with all the cs162 tools
  - Submit to autograder via git
- **Go to section next week (and afterwards)**
- **Waitlist ???**
  -

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

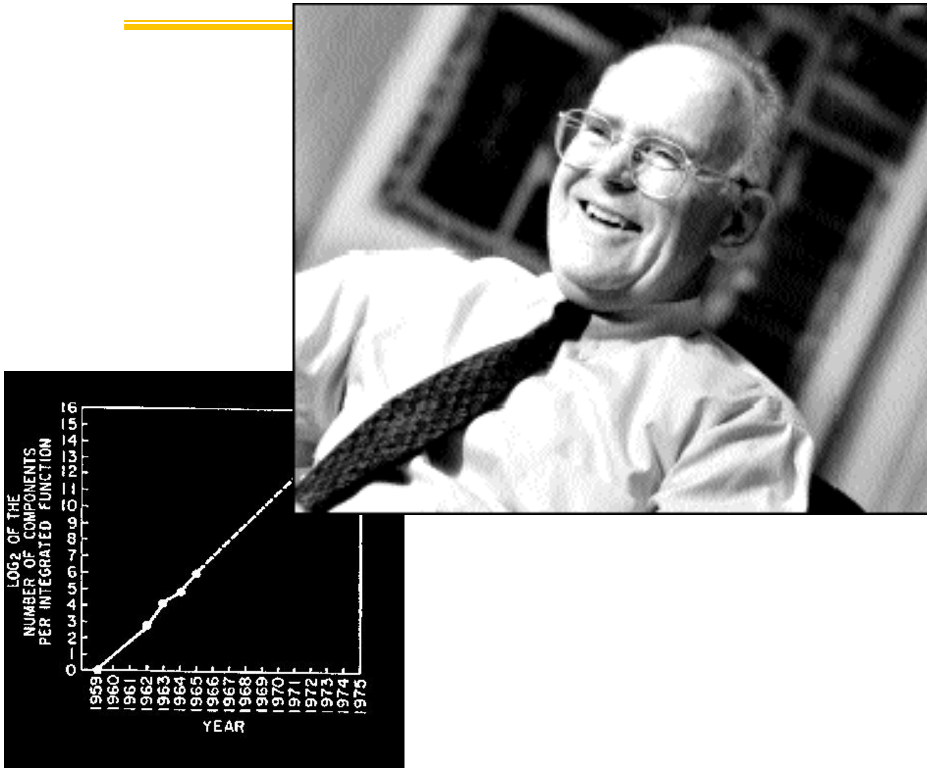
Copying OR reading online code or test cases from from prior years

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

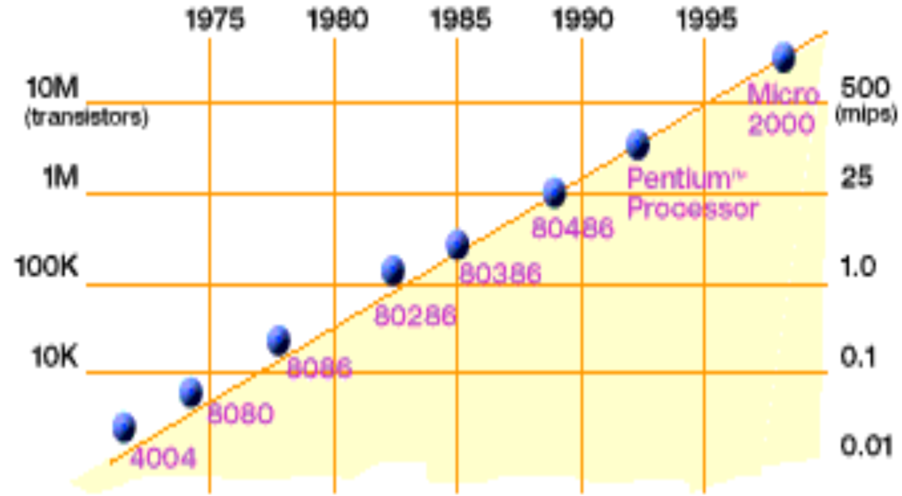
Questions

What make Operating Systems exciting and Challenging

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.

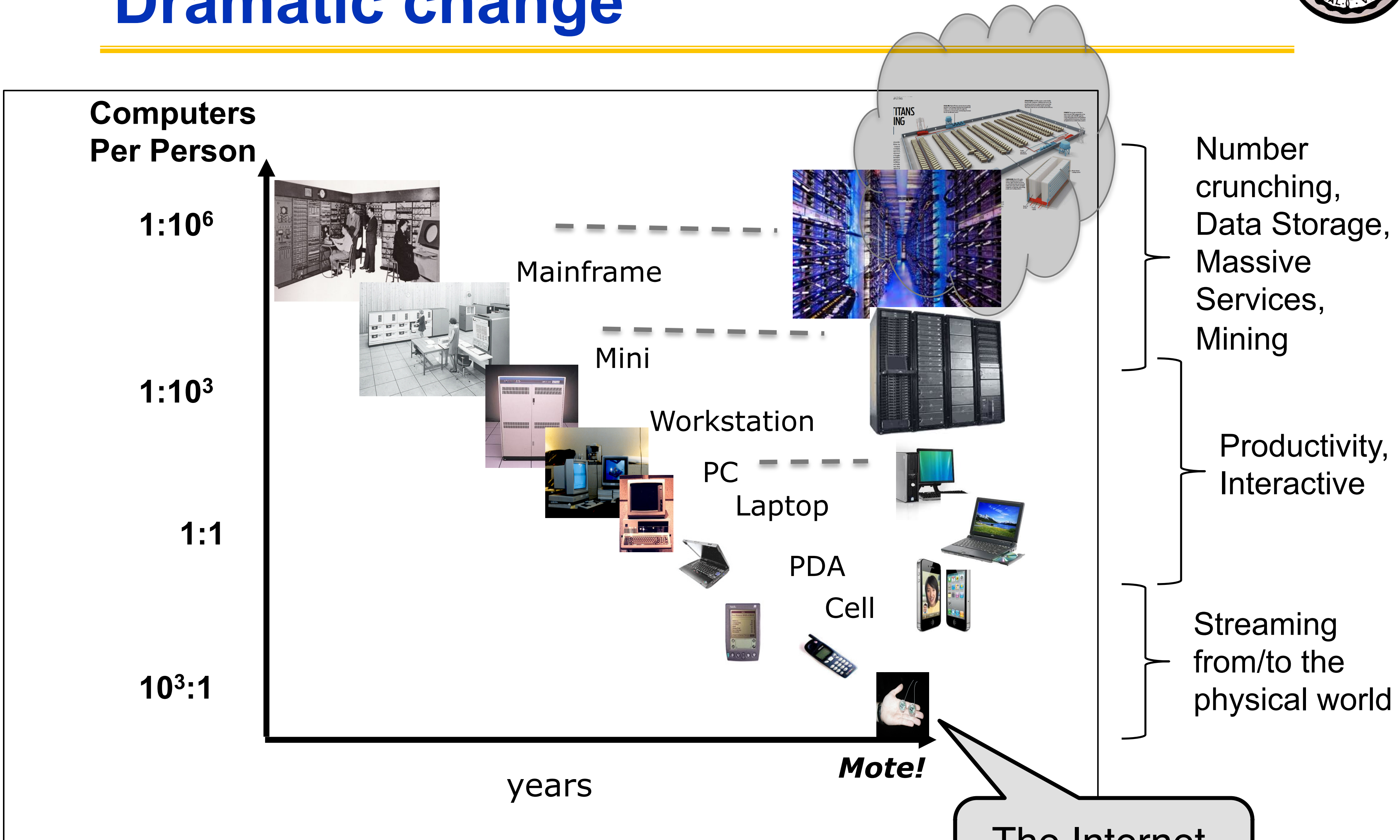


2X transistors/Chip Every 1.5 years

Called “Moore’s Law”

Microprocessors have become smaller, denser, and more powerful.

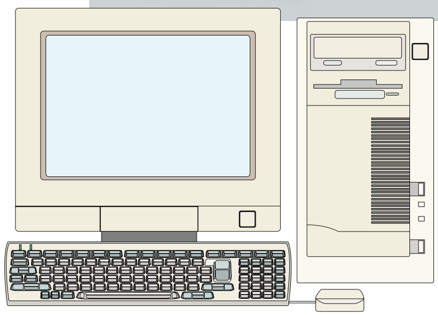
Dramatic change



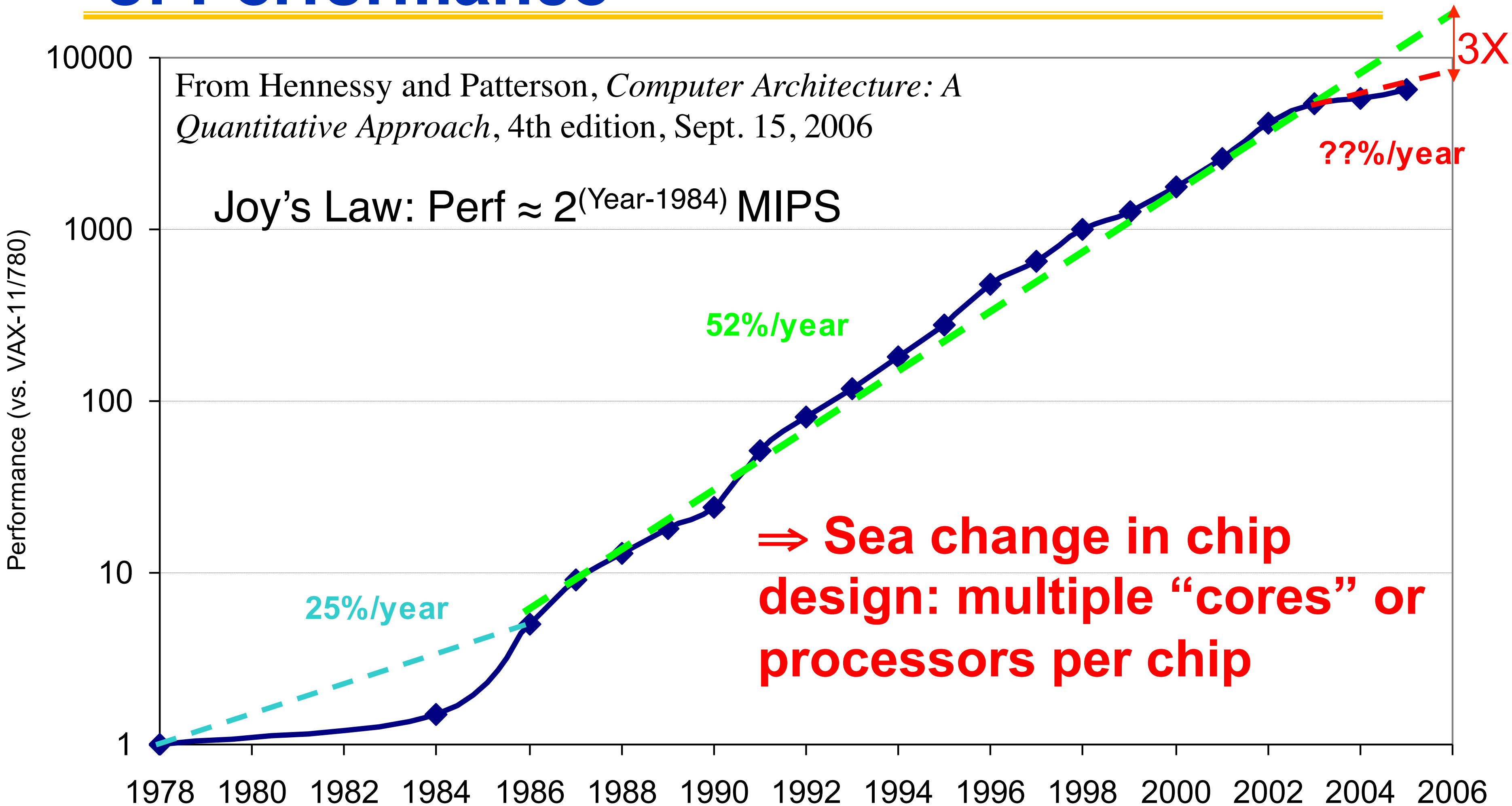
The Internet of Things!

Bell's Law: new computer class per 10 years

Computing Devices Everywhere



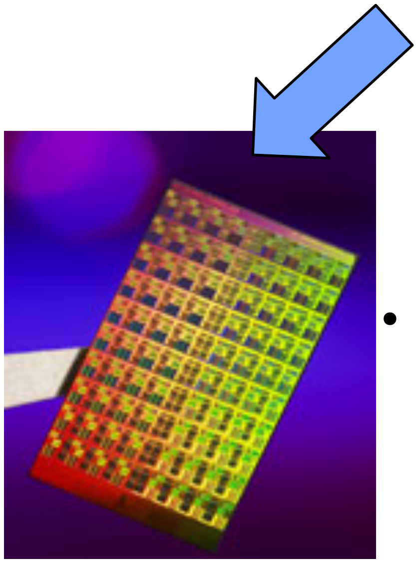
New Challenge: Slowdown in Joy's law of Performance

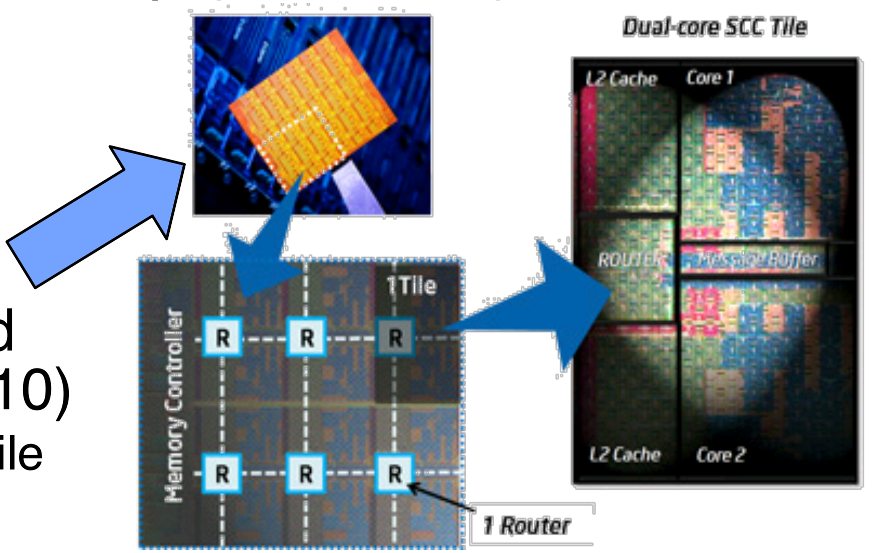


- VAX : 25%/year 1978 to 1986
- RISC + x86: 52%/year 1986 to 2002
- RISC + x86: ??%/year 2002 to present

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

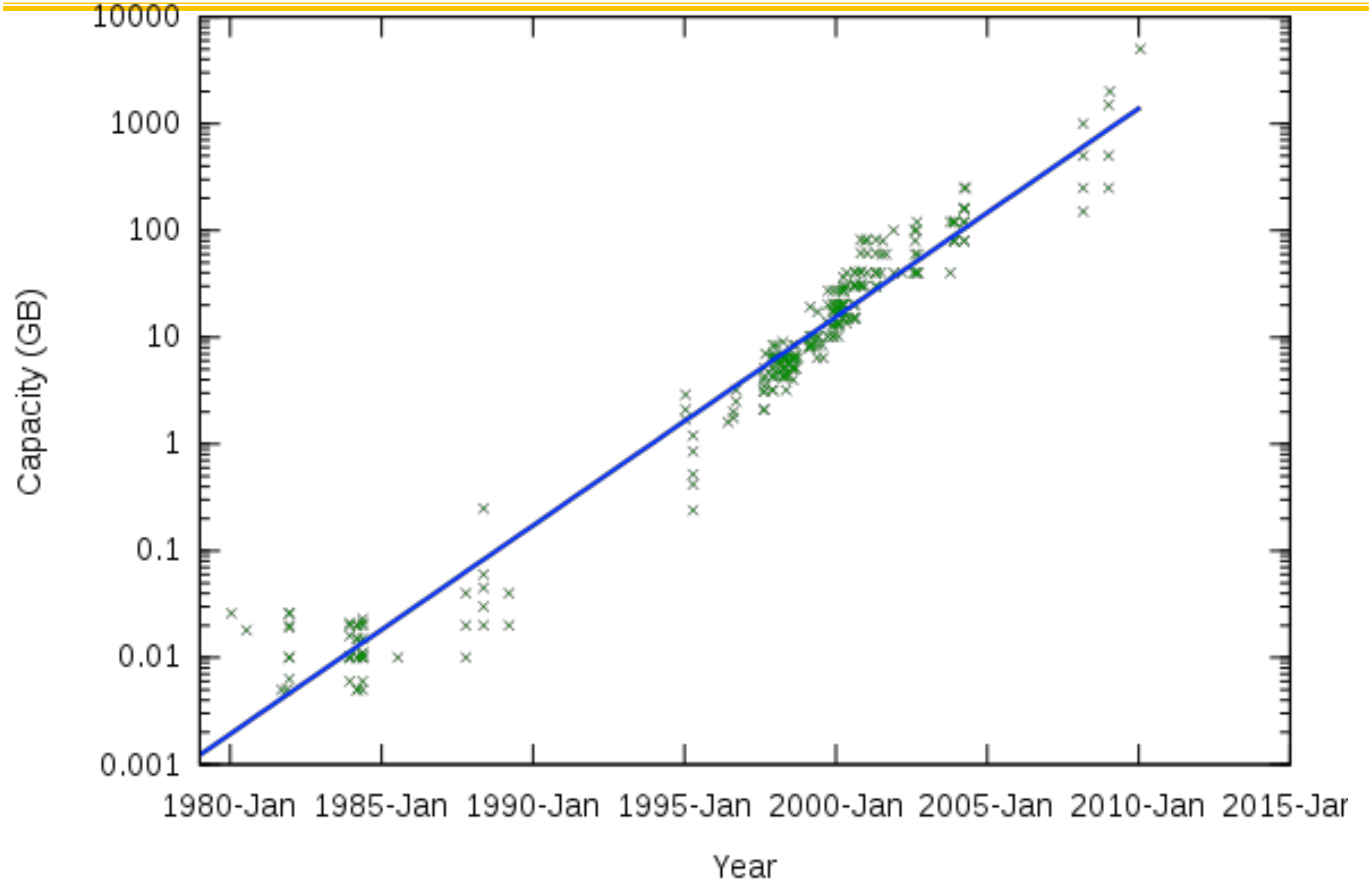




- Intel Single-Chip Cloud Computer (August 2010)
  - 24 “tiles” with two cores/tile
  - 24-router mesh network
  - 4 DDR3 memory controllers
  - Hardware support for message-passing

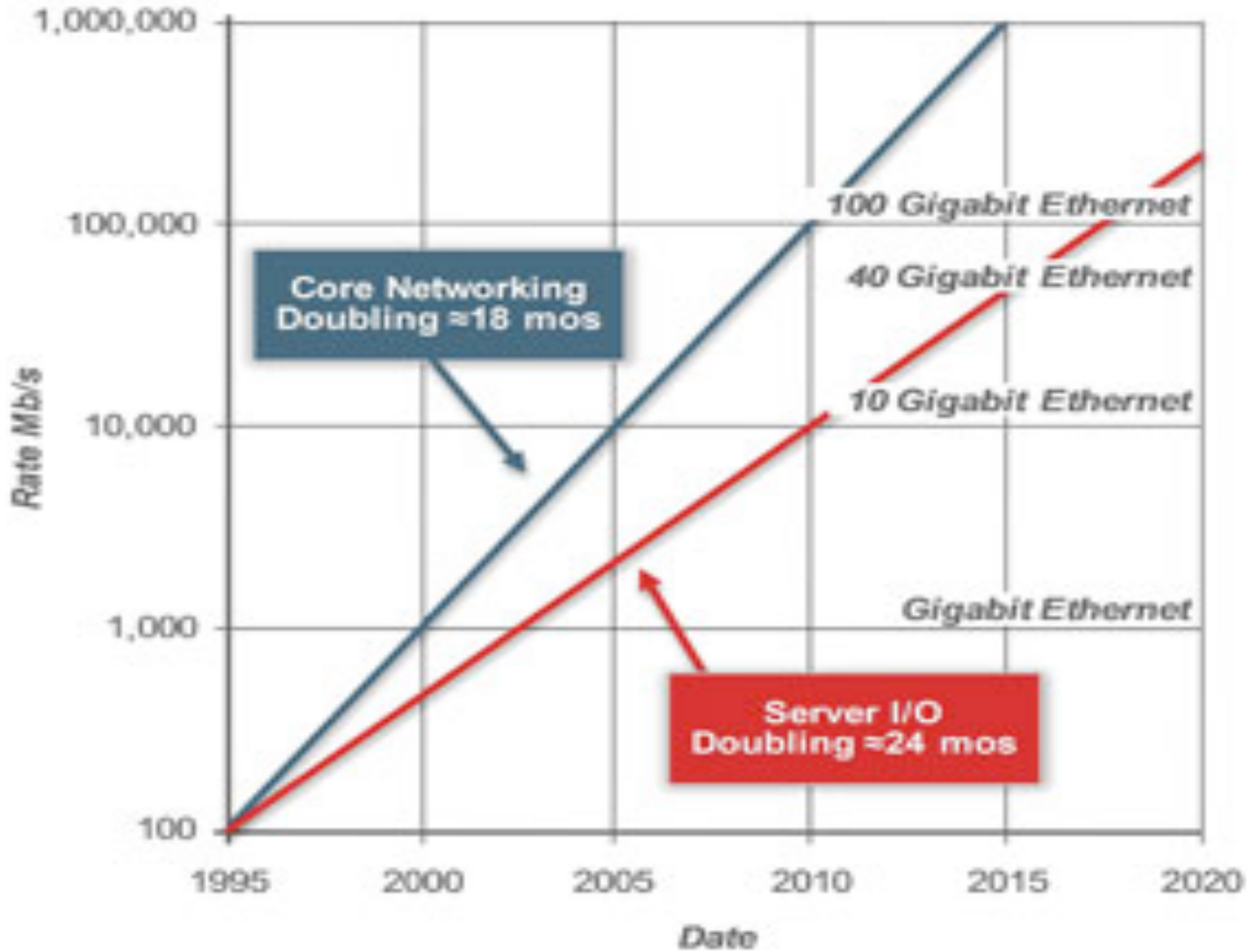
- **“ManyCore” refers to many processors/chip**
  - **64? 128? Hard to say exact boundary**
- **How to program these?**
  - **Use 2 CPUs for video/audio**
  - **Use 1 for word processor, 1 for browser**
  - **76 for virus checking???**
- **Parallelism must be exploited at all levels**

Storage Capacity



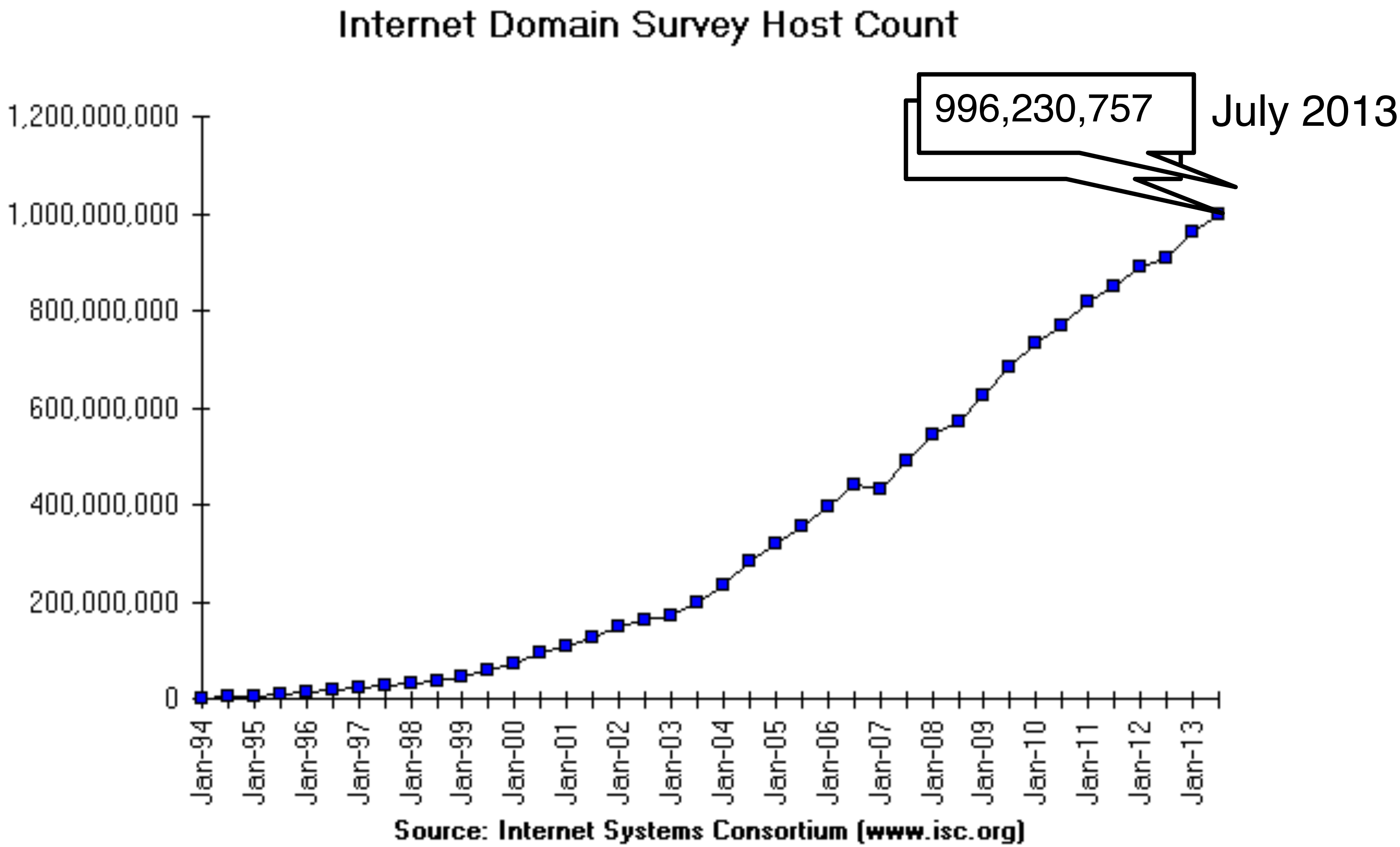
- ***Retail*** **hard disk capacity in GB**
  (source: http://www.digitaltonto.com/2011/our-emergent-digital-future/)

Network Capacity



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

Internet Scale: .96 Billion Hosts



https://www.isc.org/solutions/survey

Internet Scale: Almost 2.5 Billion Users!

WORLD INTERNET USAGE AND POPULATION STATISTICS December 31, 2013						
World Regions	Population (2014 Est.)	Internet Users Dec. 31, 2000	Internet Users Latest Data	Penetration (% Population)	Growth 2000-2014	Users % of Table
Africa	1,125,721,038	4,514,400	**240,146,482**	21.3 %	5,219.6 %	8.6 %
Asia	3,996,408,007	114,304,000	**1,265,143,702**	31.7 %	1,006.8 %	45.1 %
Europe	825,802,657	105,096,093	**566,261,317**	68.6 %	438.8 %	20.2 %
Middle East	231,062,860	3,284,800	**103,829,614**	44.9 %	3,060.9 %	3.7 %
North America	353,860,227	108,096,800	**300,287,577**	84.9 %	177.8 %	10.7 %
Latin America / Caribbean	612,279,181	18,068,919	**302,006,016**	49.3 %	1,571.4 %	10.8 %
Oceania / Australia	36,724,649	7,620,480	**24,804,226**	67.5 %	225.5 %	0.9 %
WORLD TOTAL	**7,181,858,619**	**360,985,492**	**2,802,478,934**	**39.0 %**	**676.3 %**	**100.0 %**

NOTES: (1) Internet Usage and World Population Statistics are for December 31, 2013. (2) CLICK on each world region name for detailed regional usage information. (3) Demographic (Population) numbers are based on data from the US Census Bureau and local census agencies. (4) Internet usage information comes from data published by Nielsen Online, by the International Telecommunications Union, by GfK, local ICT Regulators and other reliable sources. (5) For definitions, disclaimers, navigation help and methodology, please refer to the Site Surfing Guide. (6) Information in this site may be cited, giving the due credit to www.internetworldstats.com.

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

Not Only PCs connected to the Internet

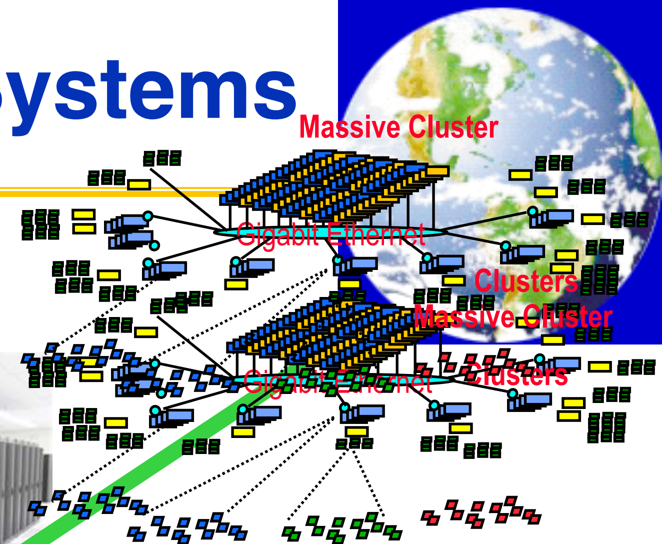
- **Smartphone shipments now exceed PC shipments!**

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

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

Societal Scale Information Systems

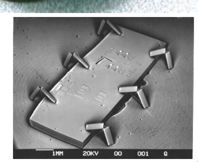
- **The world is a large distributed system**
  - **Microprocessors in everything**
  - **Vast infrastructure behind them**



Scalable, Reliable, Secure Services

Internet Connectivity

Databases
Information Collection
Remote Storage
Online Games
Commerce
…



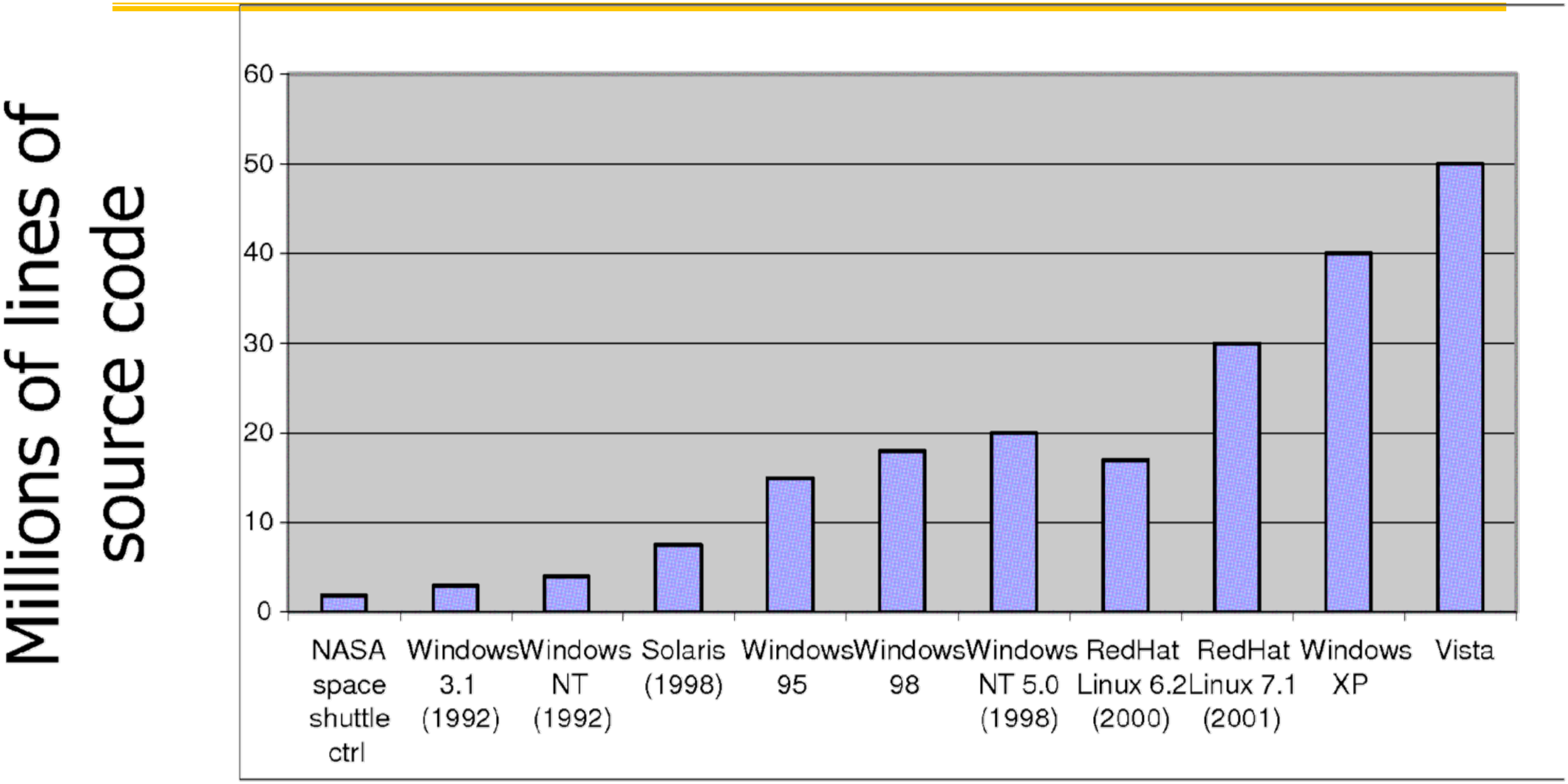
MEMS for Sensor Nets

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!

Increasing Software Complexity



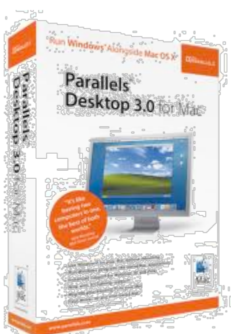
From MIT's 6.033 course

How do We Tame Complexity?

- Every piece of computer hardware different
  - Different CPU
    - Pentium, ARM, PowerPC, ColdFire
  - Different amounts of memory, disk, …
  - Different types of devices
    - Mice, keyboards, sensors, cameras, fingerprint readers, touch screen
  - Different networking environment
    - Cable, DSL, Wireless, …
- 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?

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)

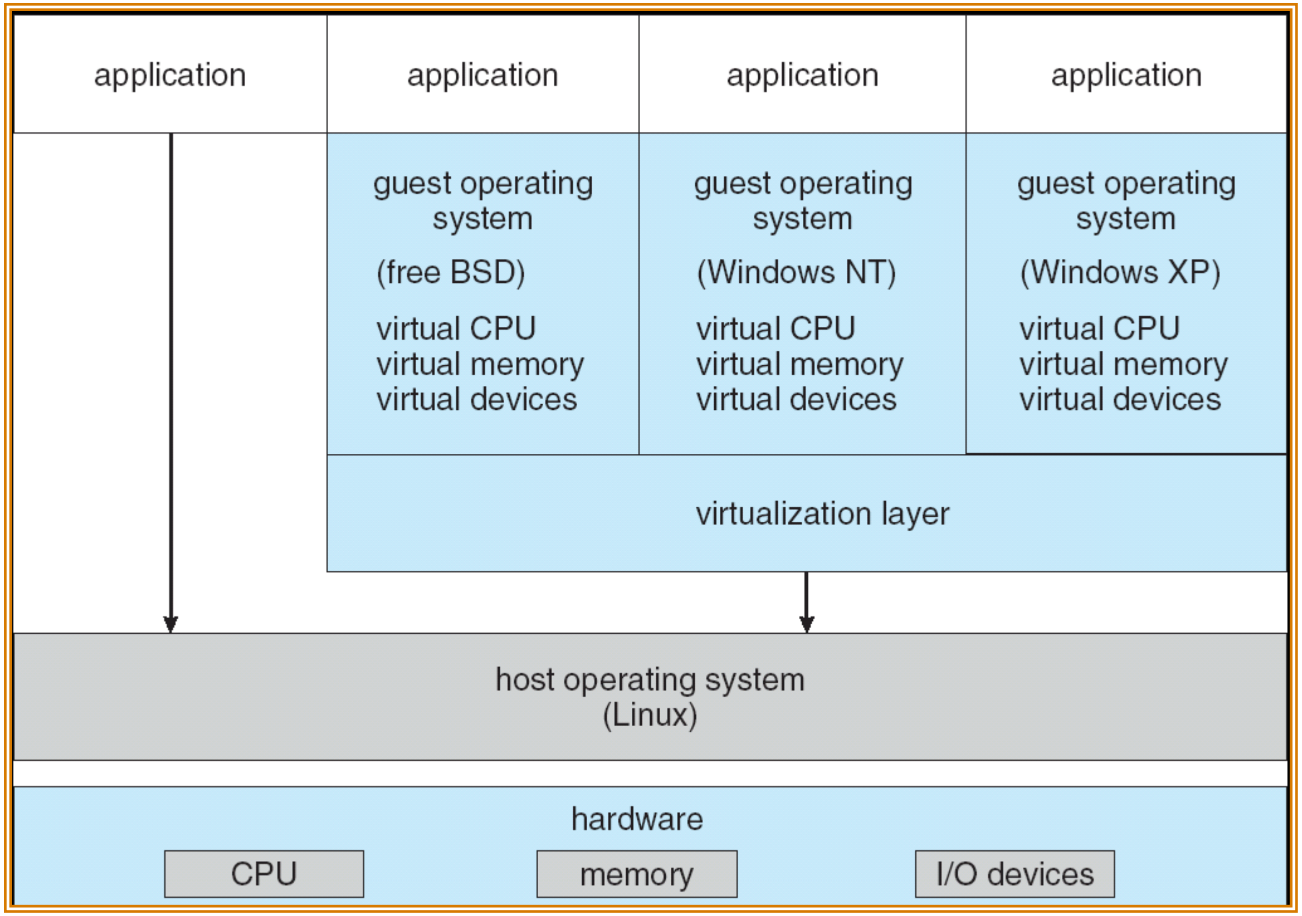


Process VMs

- **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

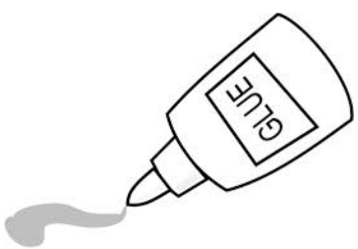
- **Useful for OS development**
  - When OS crashes, restricted to one VM
  - Can aid testing programs on other OSs



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