CS162
Operating Systems and Systems Programming
Lecture 1

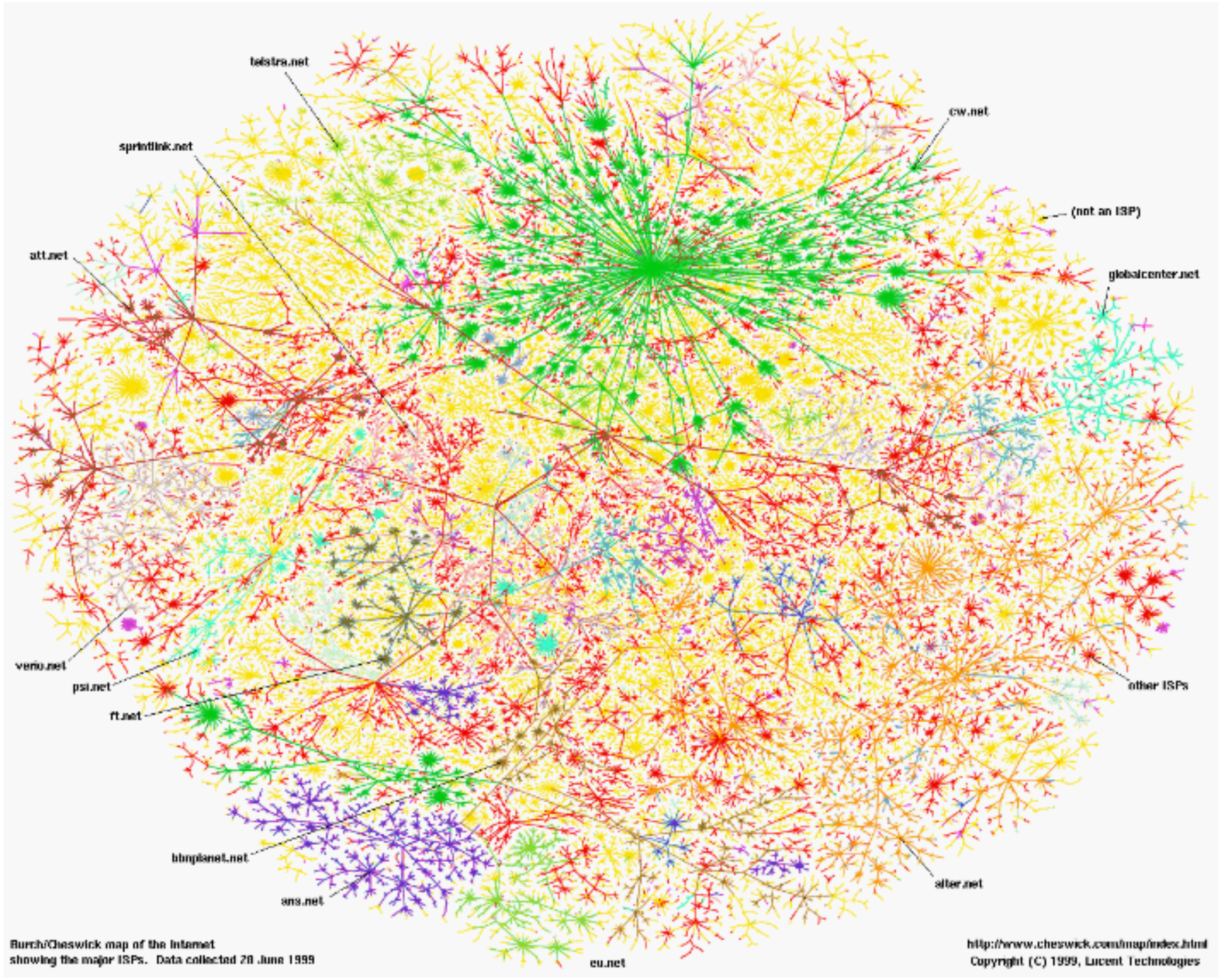
What is an Operating System?

January 21st, 2020

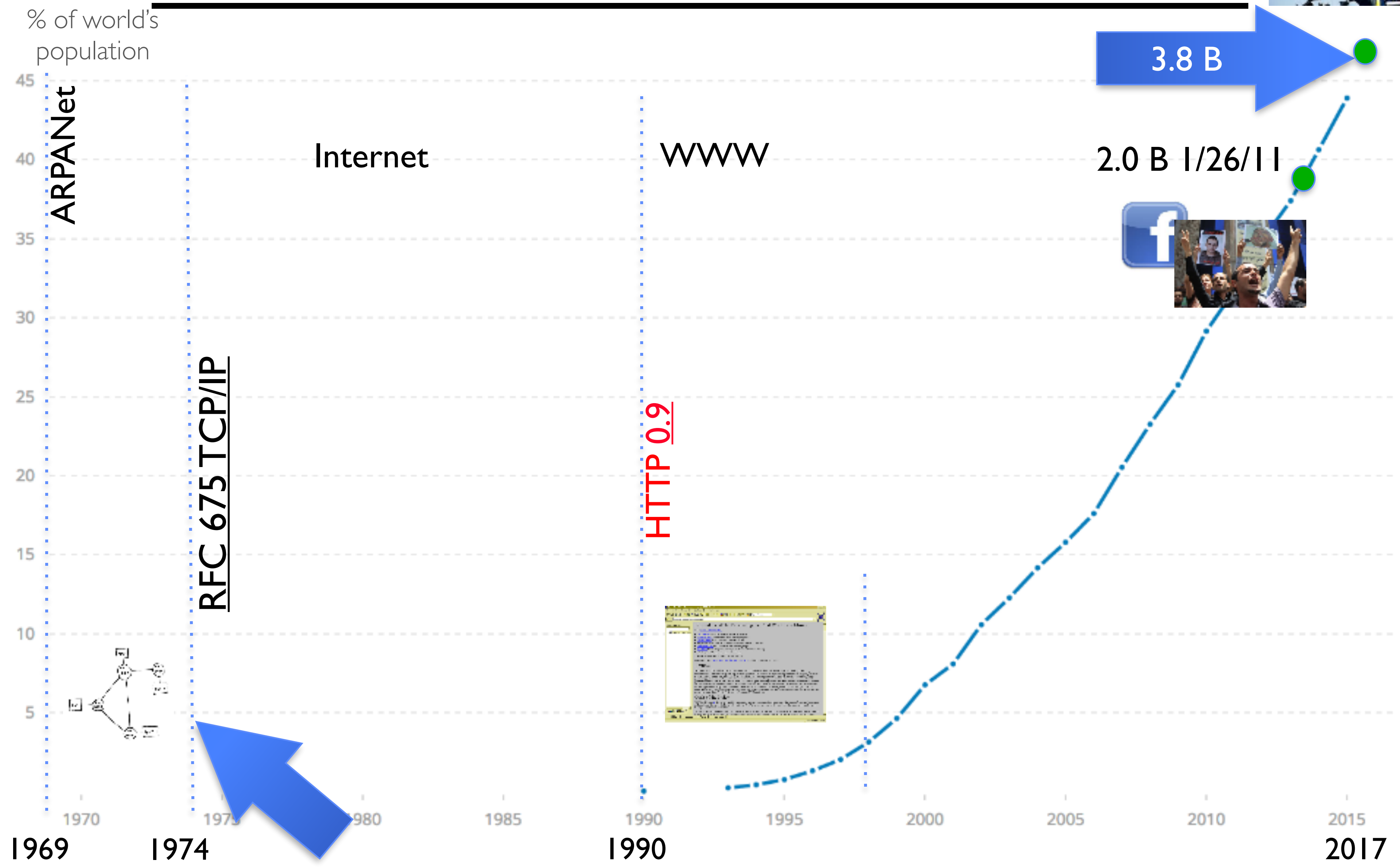
Prof. John Kubiatowicz
http://cs162.eecs.Berkeley.edu

Acknowledgments: Lecture slides are from the Operating Systems course taught by John Kubiatowicz at Berkeley, with few minor updates/changes*. When slides are obtained from other sources, a a reference will be noted on the bottom of that slide, in which case a full list of references is provided on the last slide.*

Greatest Artifact of Human Civilization…

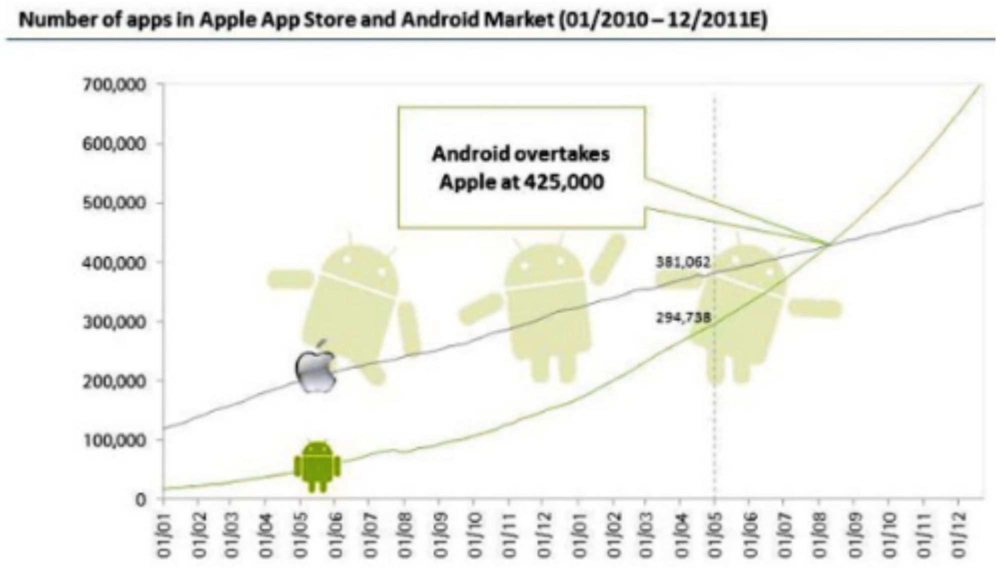

telstra.net
cw.net
sprintlink.net
(not an ISP)
att.net
globalcenter.net
verio.net
psi.net
ft.net
other ISPs
bbnplanet.net
alter.net
ans.net
Burch/Cheswick map of the Internet
showing the major ISPs. Data collected 28 June 1999
eu.net
http://www.cheswick.com/map/index.html
Copyright (C) 1999, Lucent Technologies

Internet Scale: Over 3.8 Billion Users!

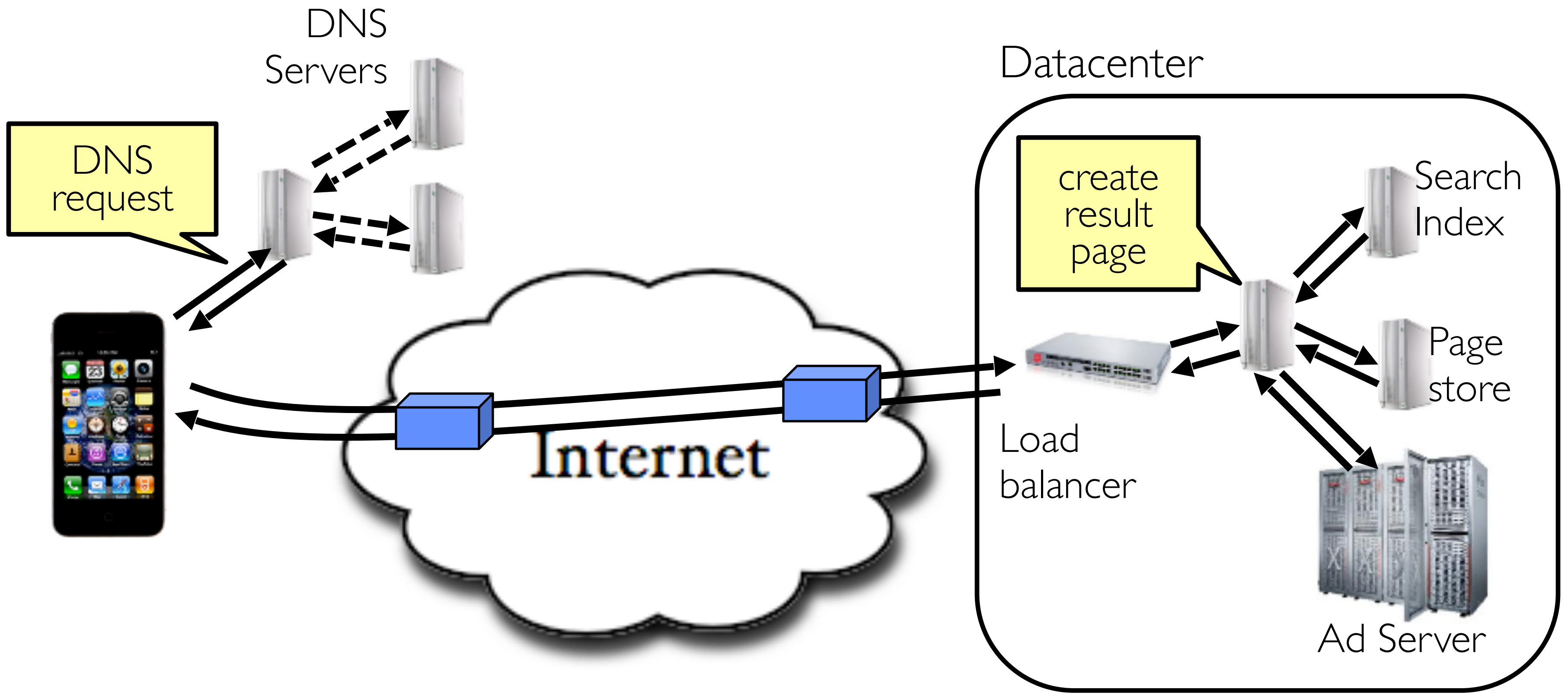


Operating Systems are at the Heart of it All!

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



Example: What's in a Search Query?



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

Why take CE424?

- 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.

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?"

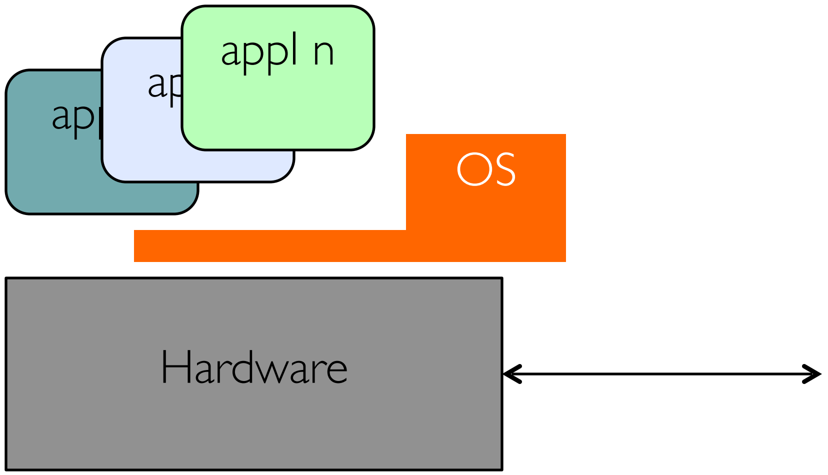
Interactive is important!

Ask Questions!

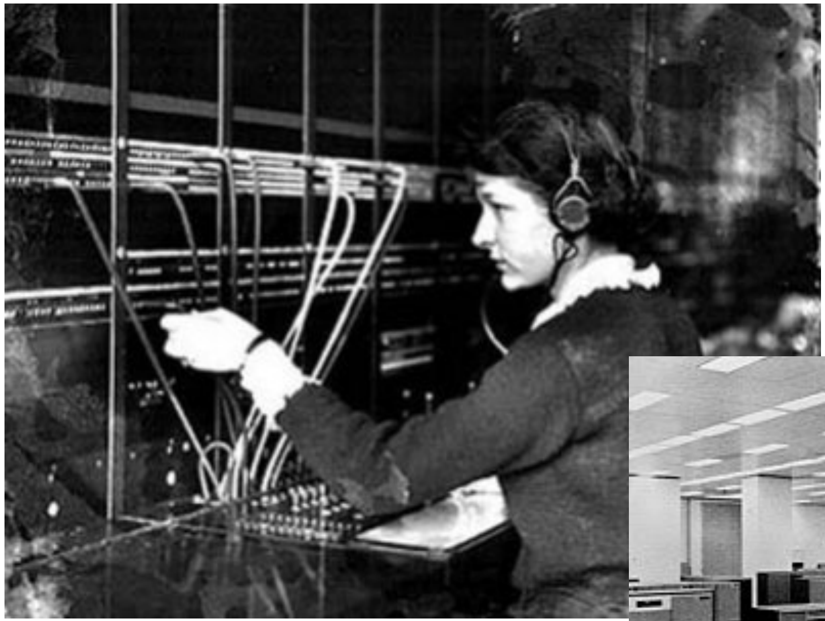
Slides courtesy of David Culler, Anthony D. Joseph, John Kubiatowicz, AJ Shankar, George Necula, Alex Aiken, Eric Brewer, Ras Bodik, Ion Stoica, Doug Tygar, and David Wagner.

What is an operating system?

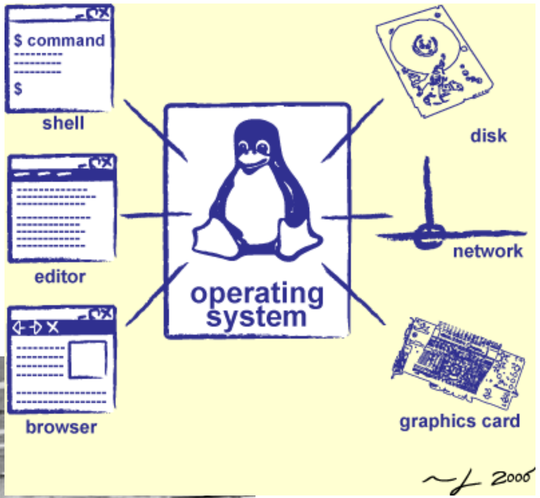
- Special layer of software that provides application software access to hardware resources
  - Convenient abstraction of complex hardware devices
  - Protected access to shared resources
  - Security and authentication
  - Communication amongst logical entities



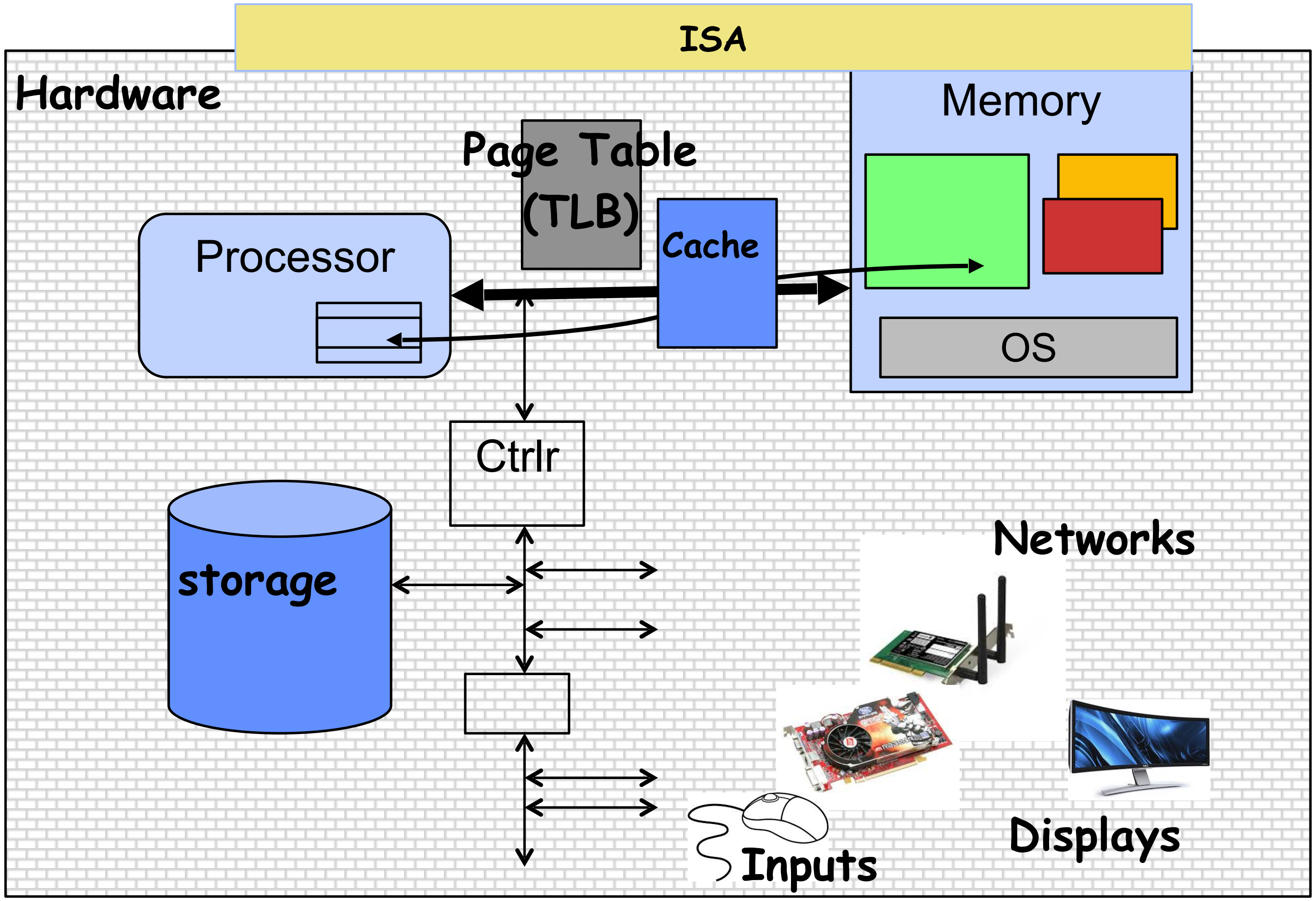
Operator …


Switchboard Operator

Computer Operators



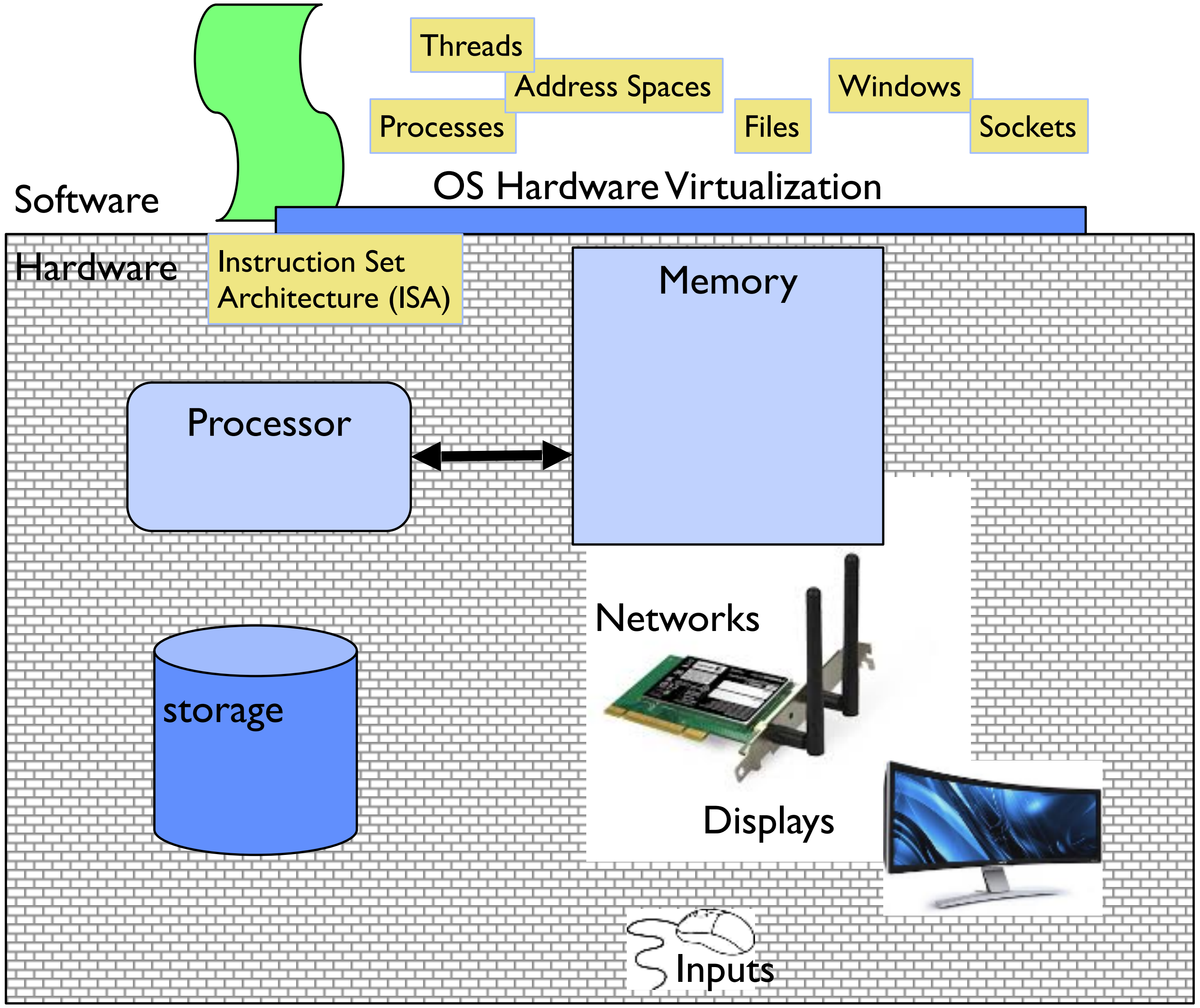
CE323 – Machine Structures



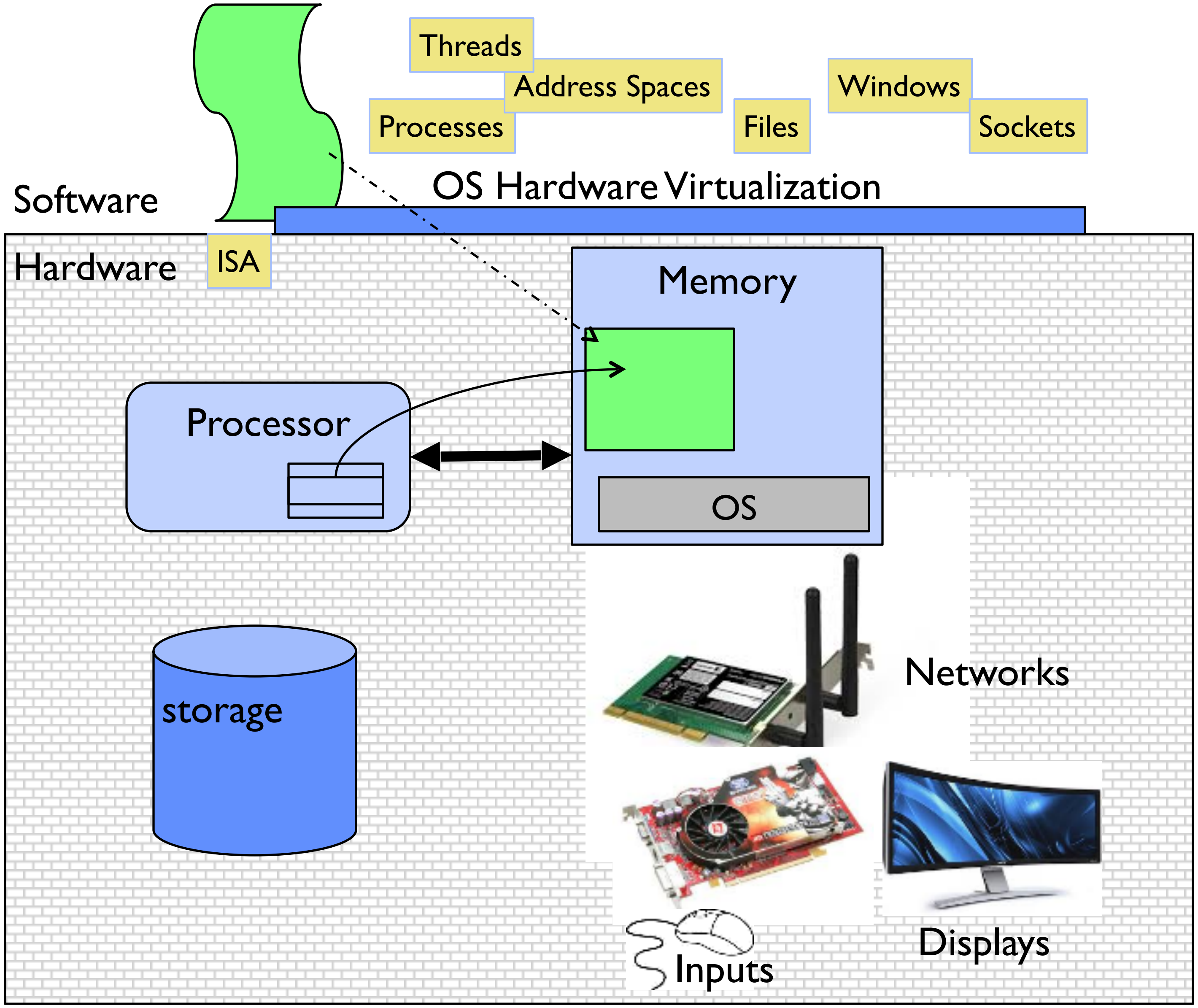
What is an Operating System?

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

OS Basics: "Virtual Machine" Boundary



OS Basics: Program ⇒ Process

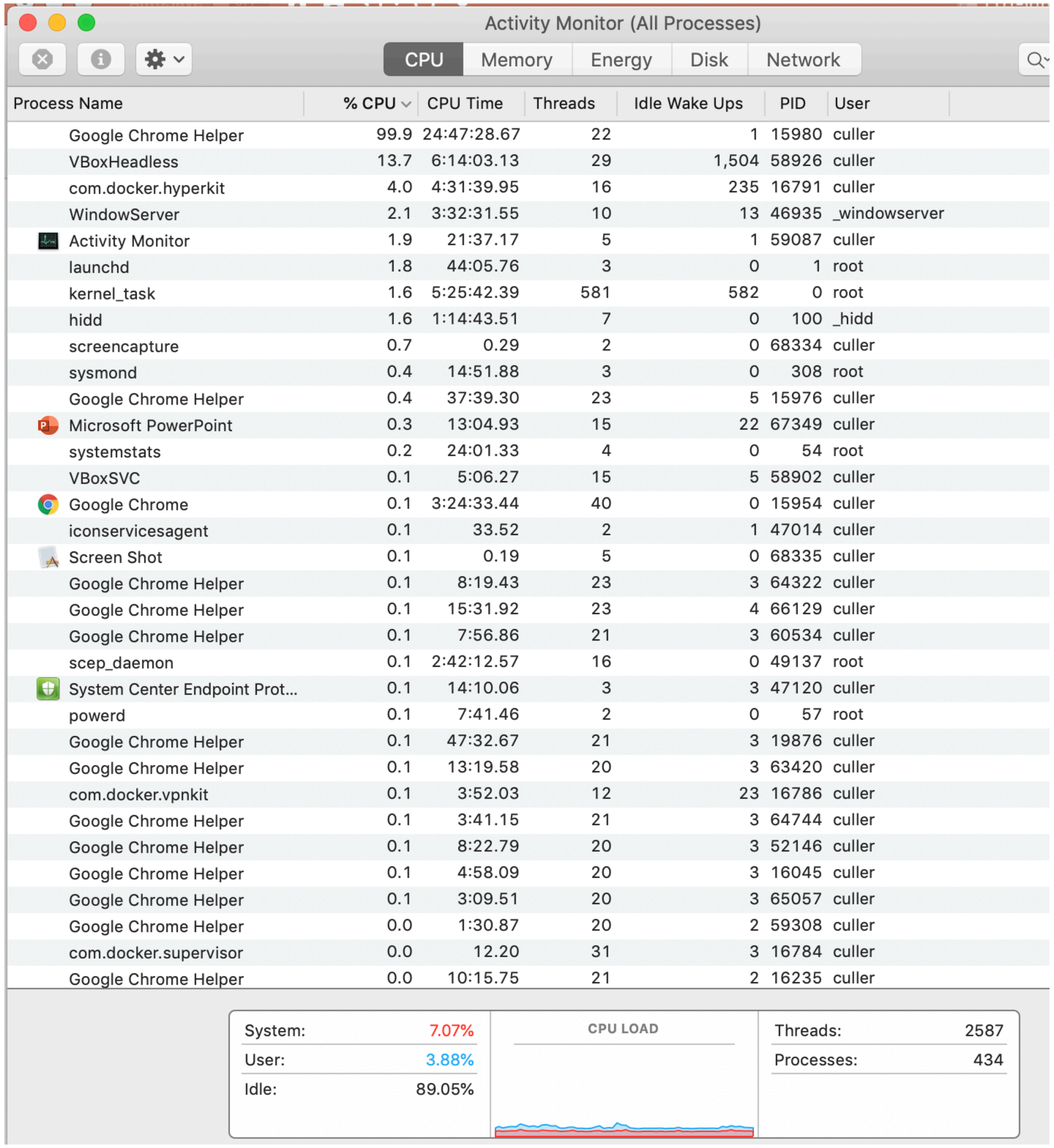


Defn: Process

- Address Space
- One or more *threads* of control
- Additional system state associated with it

- Thread:
  – locus of control (PC)
  – Its registers (processor state when running)
  – And its "stack" (SP)
    » As required by programming language runtime

For Example …


Activity Monitor (All Processes)

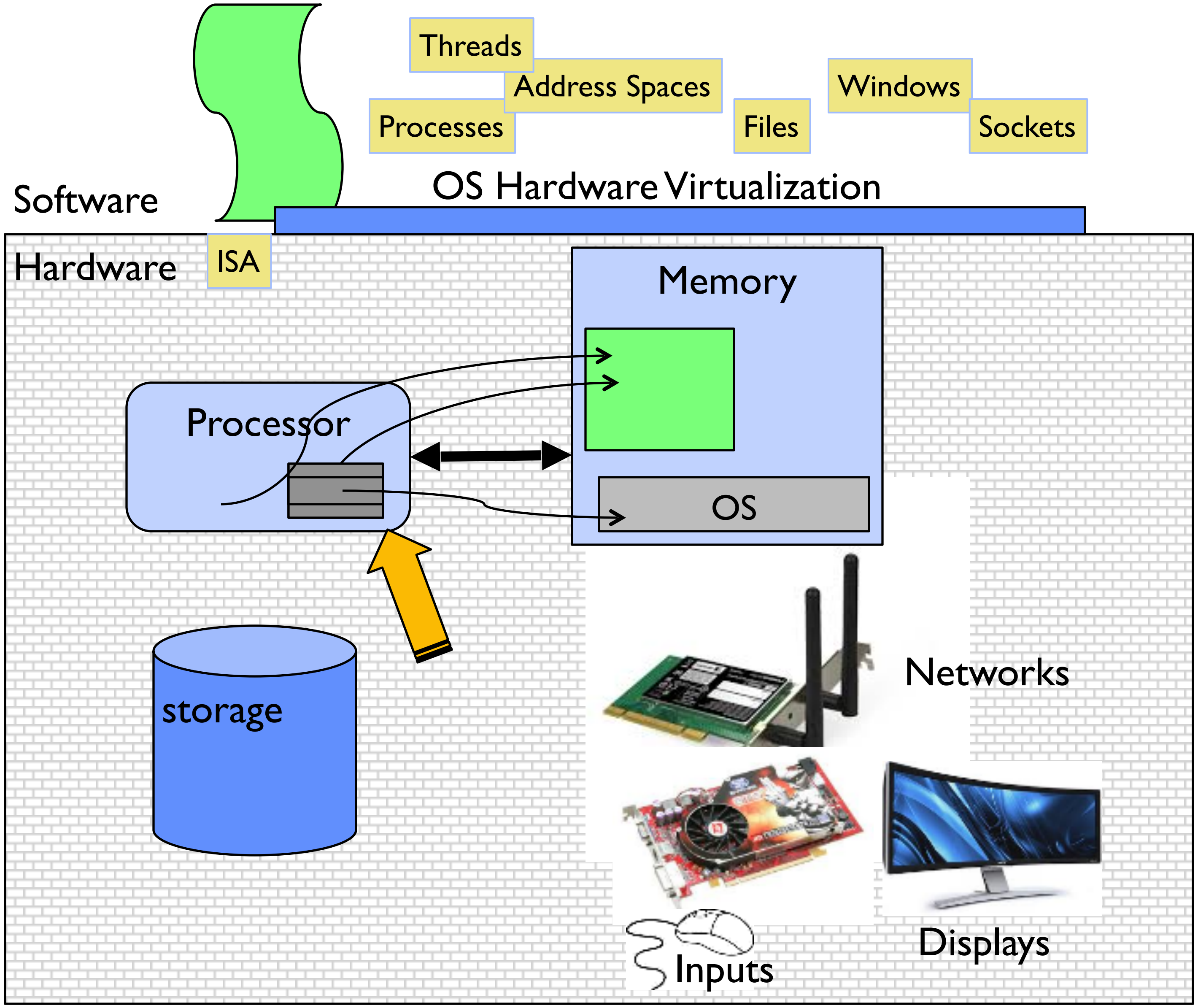
Process Name	% CPU	CPU Time	Threads	Idle Wake Ups	PID	User
Google Chrome Helper	99.9	24:47:28.67	22	1	15980	culler
VBoxHeadless	13.7	6:14:03.13	29	1,504	58926	culler
com.docker.hyperkit	4.0	4:31:39.95	16	235	16791	culler
WindowServer	2.1	3:32:31.55	10	13	46935	_windowserver
Activity Monitor	1.9	21:37.17	5	1	59087	culler
launchd	1.8	44:05.76	3	0	1	root
kernel_task	1.6	5:25:42.39	581	582	0	root
hidd	1.6	1:14:43.51	7	0	100	_hidd
screencapture	0.7	0.29	2	0	68334	culler
sysmond	0.4	14:51.88	3	0	308	root
Google Chrome Helper	0.4	37:39.30	23	5	15976	culler
Microsoft PowerPoint	0.3	13:04.93	15	22	67349	culler
systemstats	0.2	24:01.33	4	0	54	root
VBoxSVC	0.1	5:06.27	15	5	58902	culler
Google Chrome	0.1	3:24:33.44	40	0	15954	culler
iconservicesagent	0.1	33.52	2	1	47014	culler
Screen Shot	0.1	0.19	5	0	68335	culler
Google Chrome Helper	0.1	8:19.43	23	3	64322	culler
Google Chrome Helper	0.1	15:31.92	23	4	66129	culler
Google Chrome Helper	0.1	7:56.86	21	3	60534	culler
scep_daemon	0.1	2:42:12.57	16	0	49137	root
System Center Endpoint Prot...	0.1	14:10.06	3	3	47120	culler
powerd	0.1	7:41.46	2	0	57	root
Google Chrome Helper	0.1	47:32.67	21	3	19876	culler
Google Chrome Helper	0.1	13:19.58	20	3	63420	culler
com.docker.vpnkit	0.1	3:52.03	12	23	16786	culler
Google Chrome Helper	0.1	3:41.15	21	3	64744	culler
Google Chrome Helper	0.1	8:22.79	20	3	52146	culler
Google Chrome Helper	0.1	4:58.09	20	3	16045	culler
Google Chrome Helper	0.1	3:09.51	20	3	65057	culler
Google Chrome Helper	0.0	1:30.87	20	2	59308	culler
com.docker.supervisor	0.0	12.20	31	3	16784	culler
Google Chrome Helper	0.0	10:15.75	21	2	16235	culler

System:	7.07%
User:	3.88%
Idle:	89.05%

CPU LOAD

Threads:	2587
Processes:	434

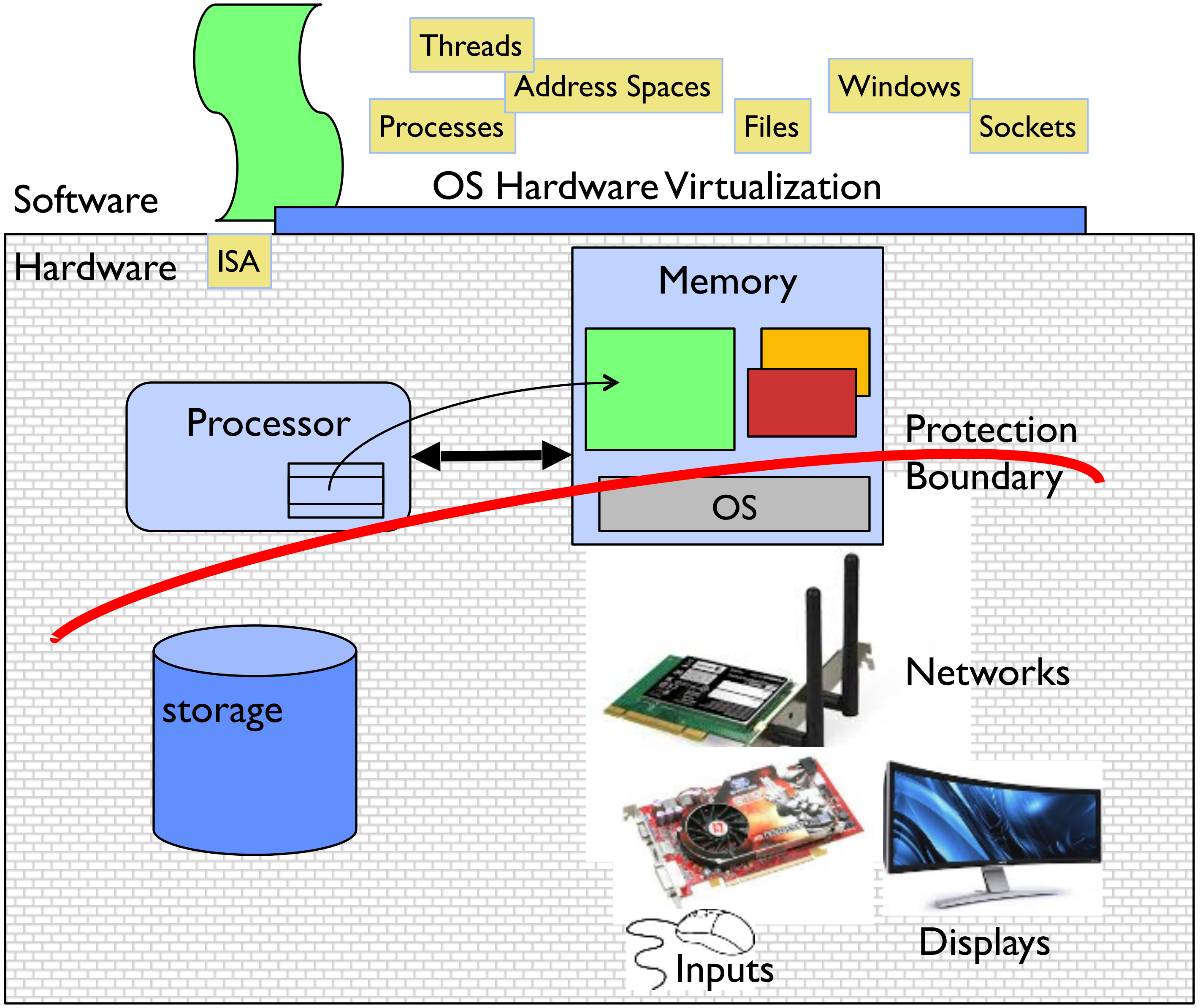
OS Basics: Context Switch



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

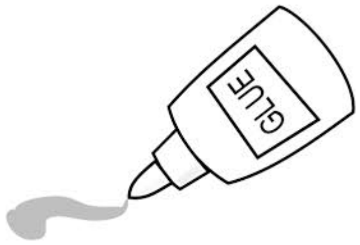
OS Basics: Scheduling, Protection



What is an Operating System?

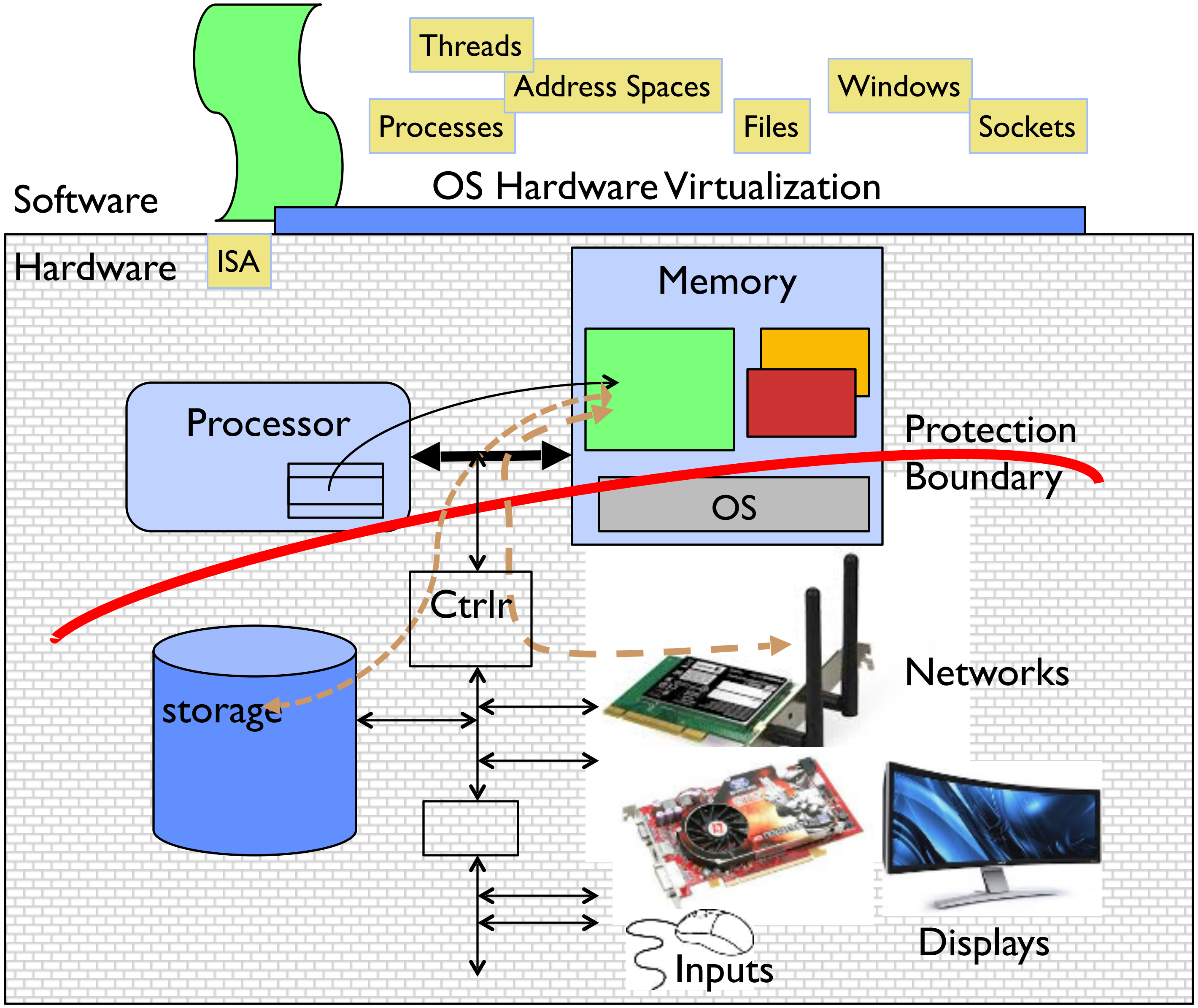
- Referee
  - Manage sharing of resources, Protection, Isolation
    - » Resource allocation, isolation, communication

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

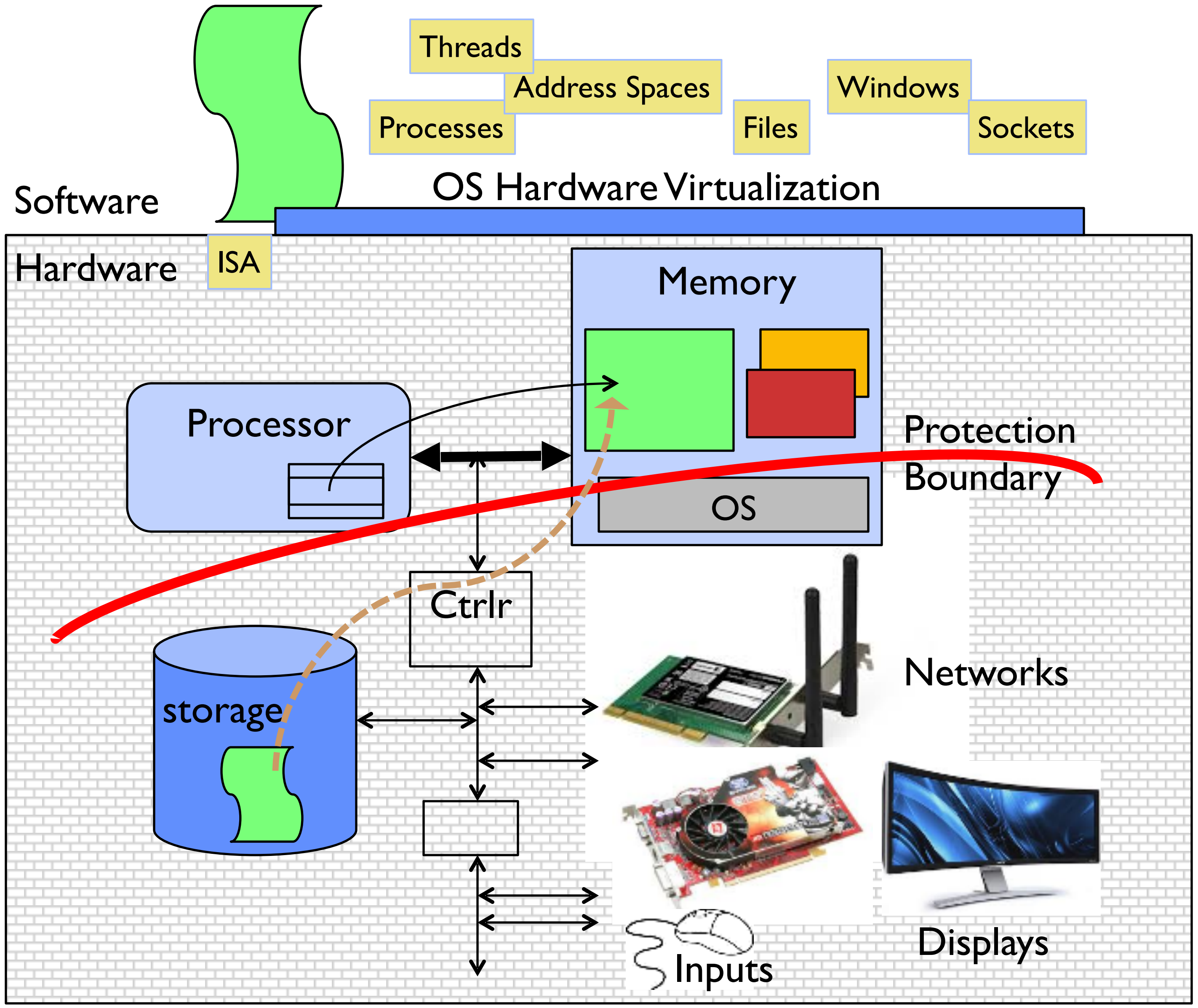


- Glue
  - Common services
    - » Storage, Window system, Networking
    - » Sharing, Authorization
    - » Look and feel

OS Basics: I/O

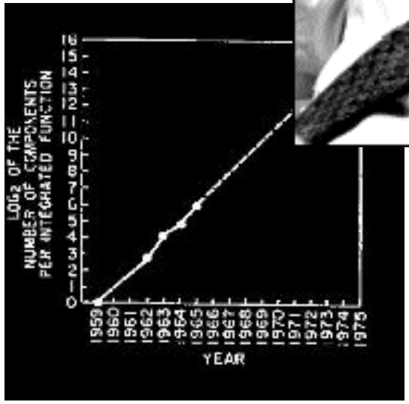

Threads
Address Spaces
Windows
Processes
Files
Sockets
Software
OS Hardware Virtualization
Hardware
ISA
Memory
Processor
Protection
Boundary
OS
Ctrlr
storage
Networks
Displays
Inputs

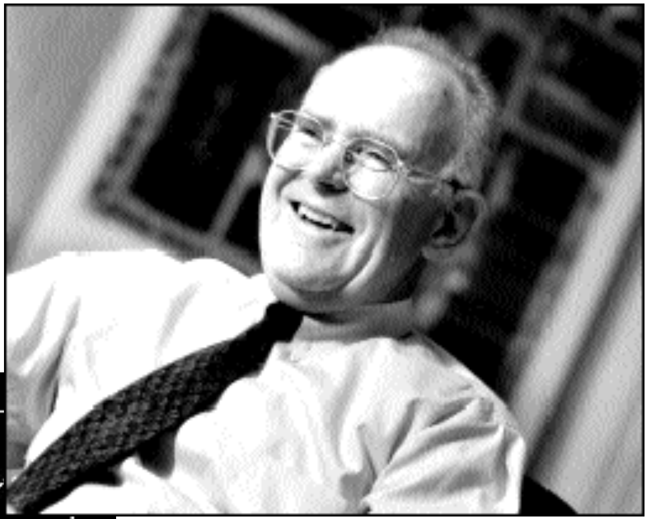
OS Basics: Creating Process/Loading Program

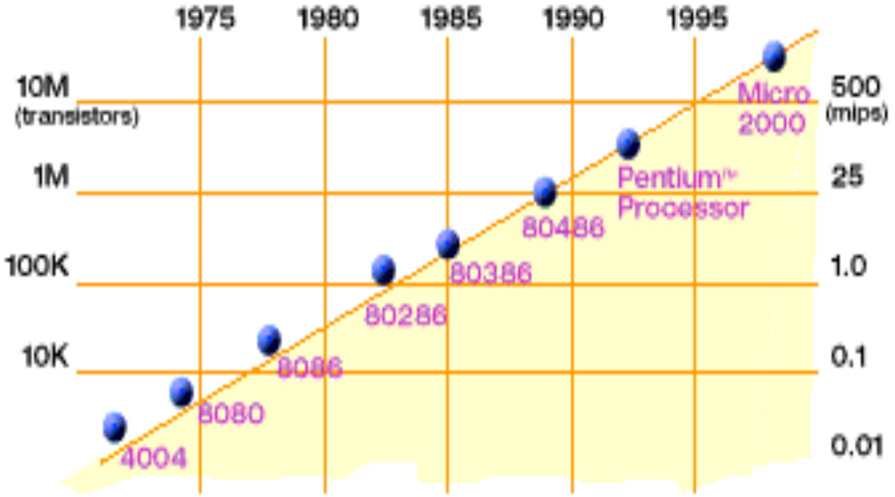


What makes Operating Systems Exciting and Challenging?

Technology Trends: Moore's Law







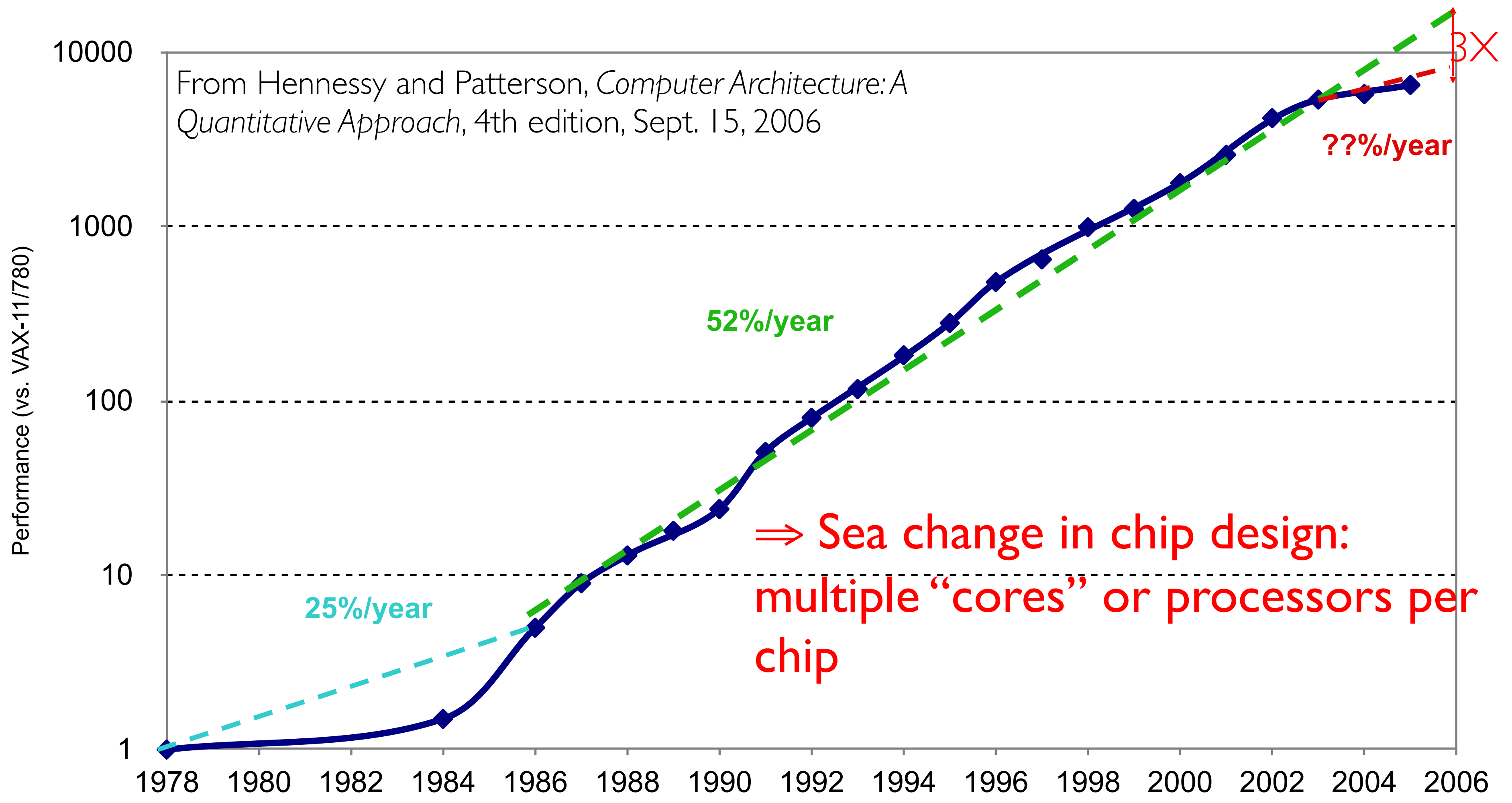
2X transistors/Chip Every 1.5 years

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

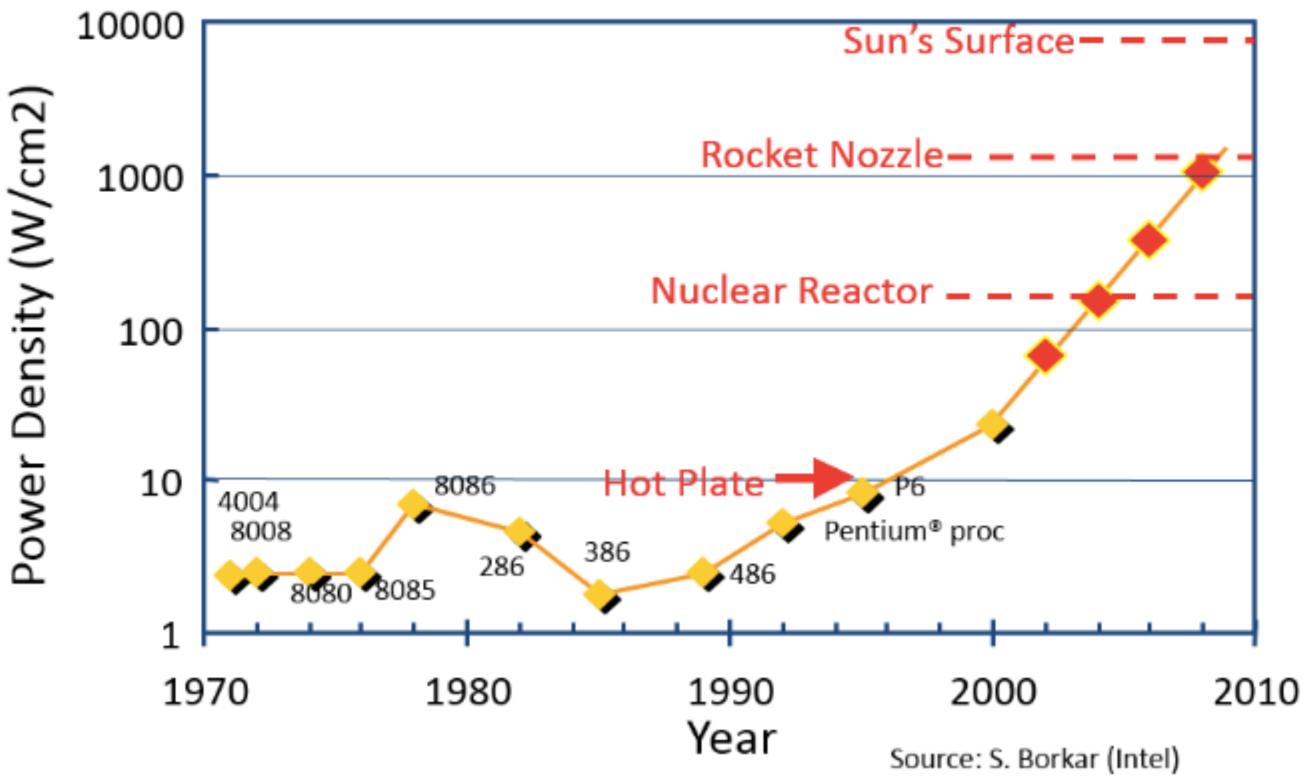
Microprocessors have become smaller, denser, and more powerful

Big Challenge: Slowdown in Joy's law of Performance



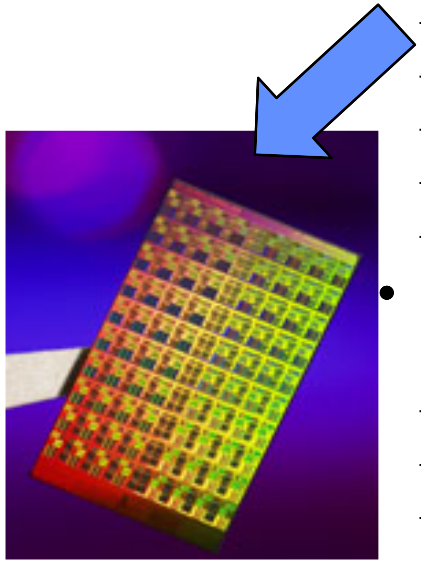
- VAX : 25%/year 1978 to 1986
- RISC + x86 : 52%/year 1986 to 2002
- RISC + x86 : ??%/year 2002 to present

Another Challenge: Power Density

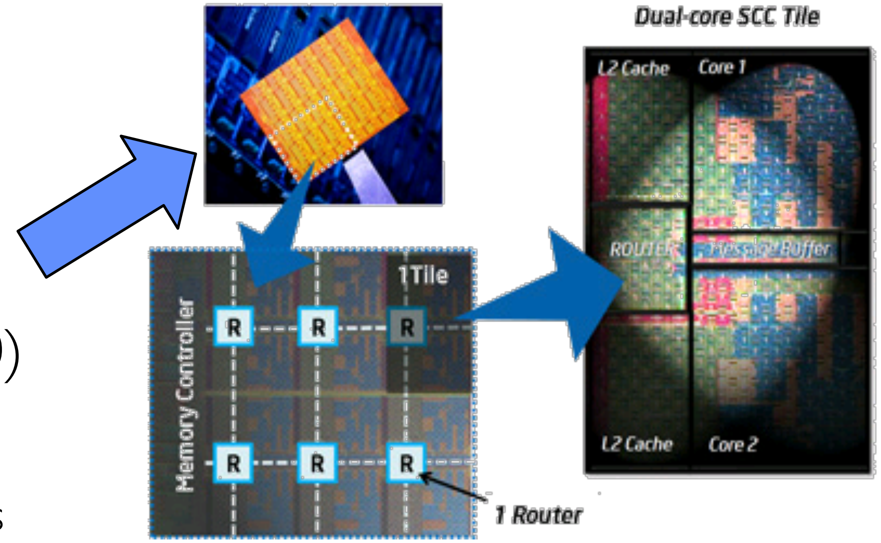


- Moore's law extrapolation
  - Potential power density reaching amazing levels!
- Flip side: battery life very important
  - Moore's law yielded more functionality at equivalent (or less) total energy consumption

ManyCore Chips: The future arrived in 2007

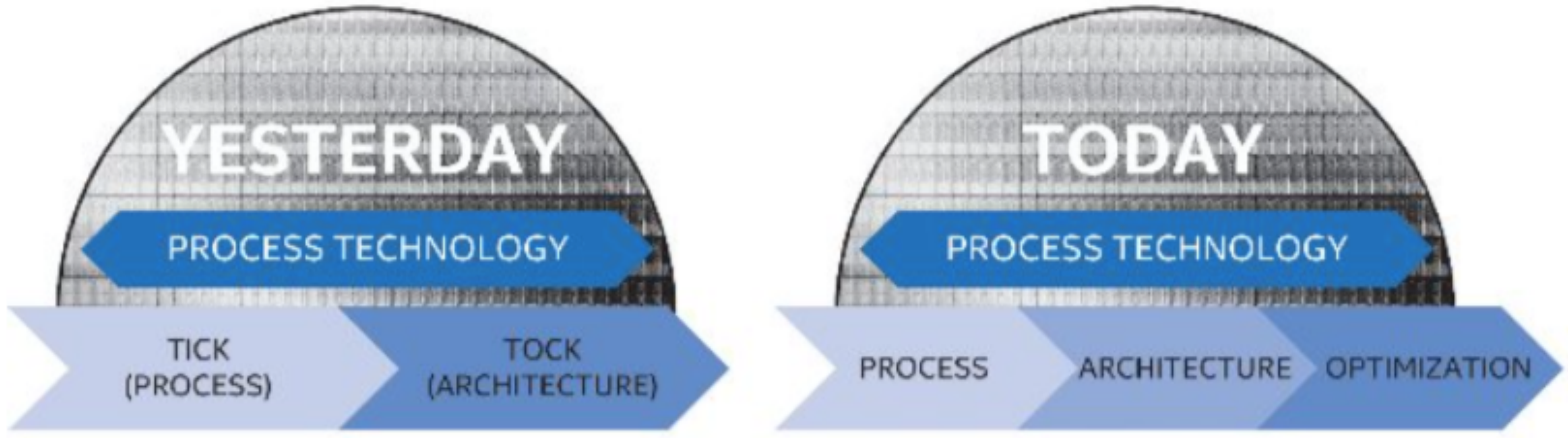


- Intel 80-core multicore chip (Feb 2007)
  - 80 simple cores
  - Two FP-engines / core
  - Mesh-like network
  - 100 million transistors
  - 65nm feature size
- Intel Single-Chip Cloud Computer (August 2010)
  - 24 "tiles" with two cores/tile
  - 24-router mesh network
  - 4 DDR3 memory controllers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



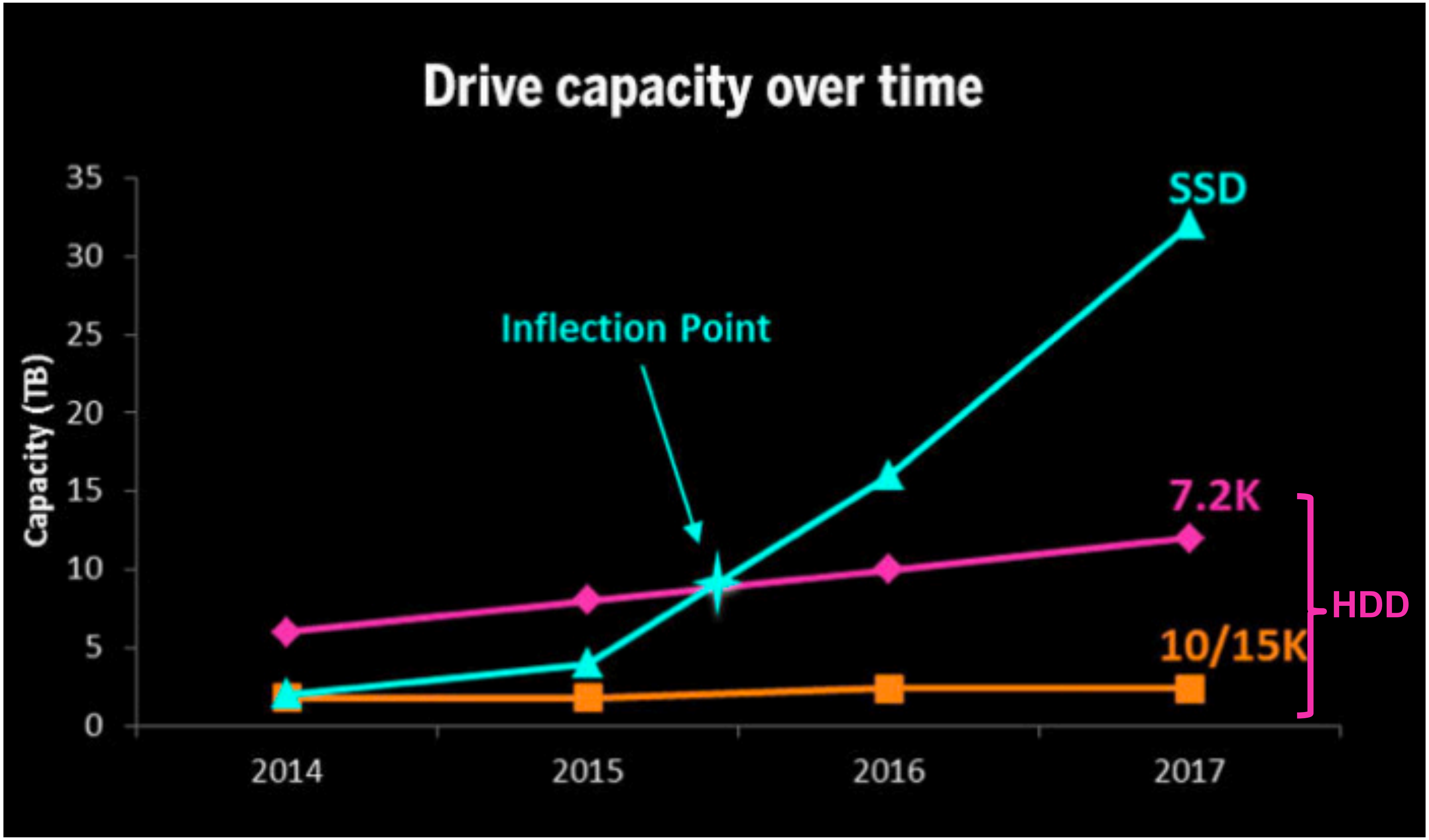
- How to program these?
  - Use 2 CPUs for video/audio
  - Use 1 for word processor, 1 for browser
  - 76 for virus checking???
- Parallelism must be exploited at all levels
- Amazon X1 instances (2016)
  - 128 virtual cores, 2 TB RAM

But then Moore's Law Ended…



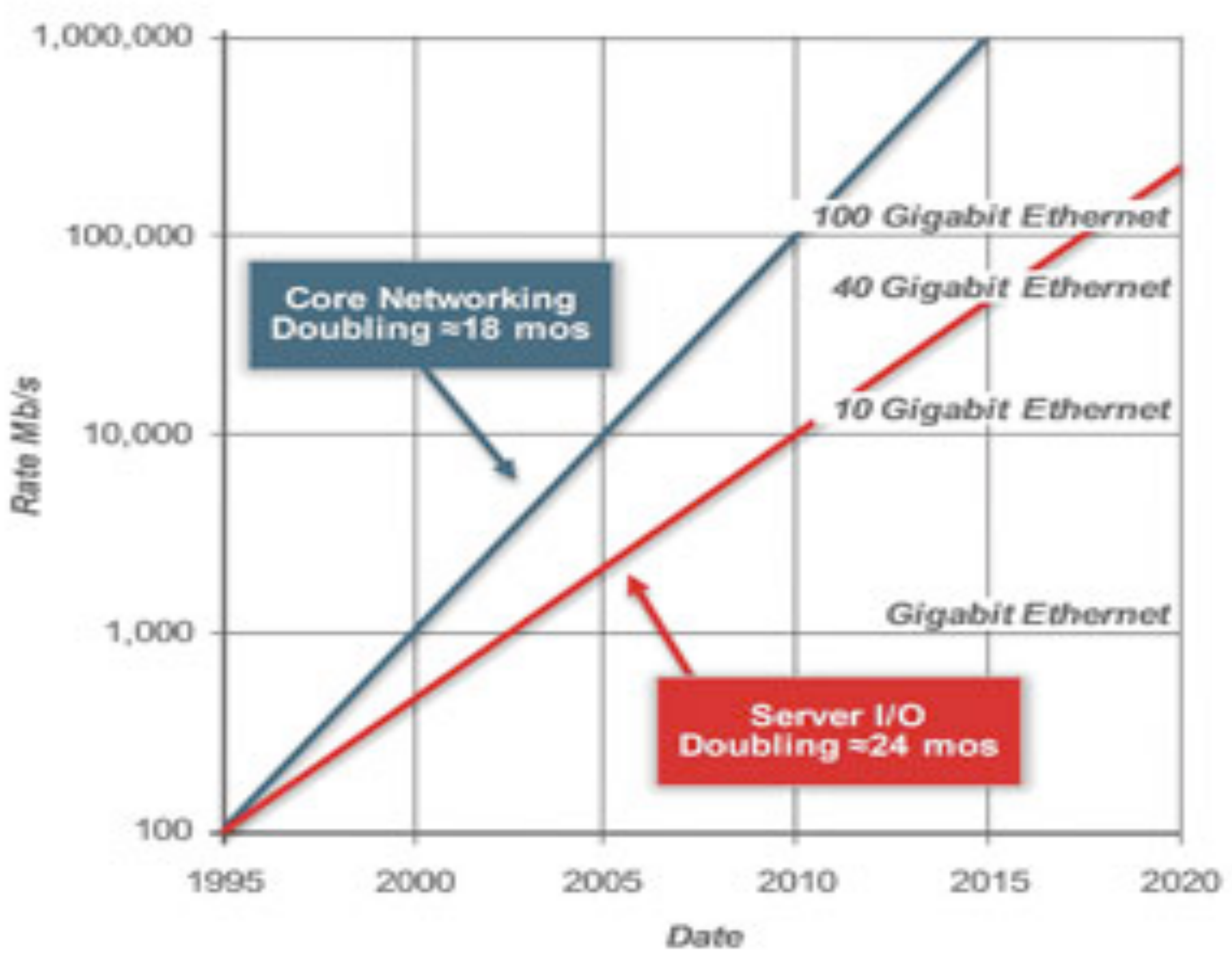
- Moore's Law has (officially) ended -- Feb 2016
  - No longer getting 2 x transistors/chip every 18 months…
  - or even every 24 months
- May have only 2-3 smallest geometry fabrication plants left:
  - Intel and Samsung and/or TSMC
  - Vendors moving to 3D stacked chips
  - More layers in old geometries

Storage Capacity Still Growing



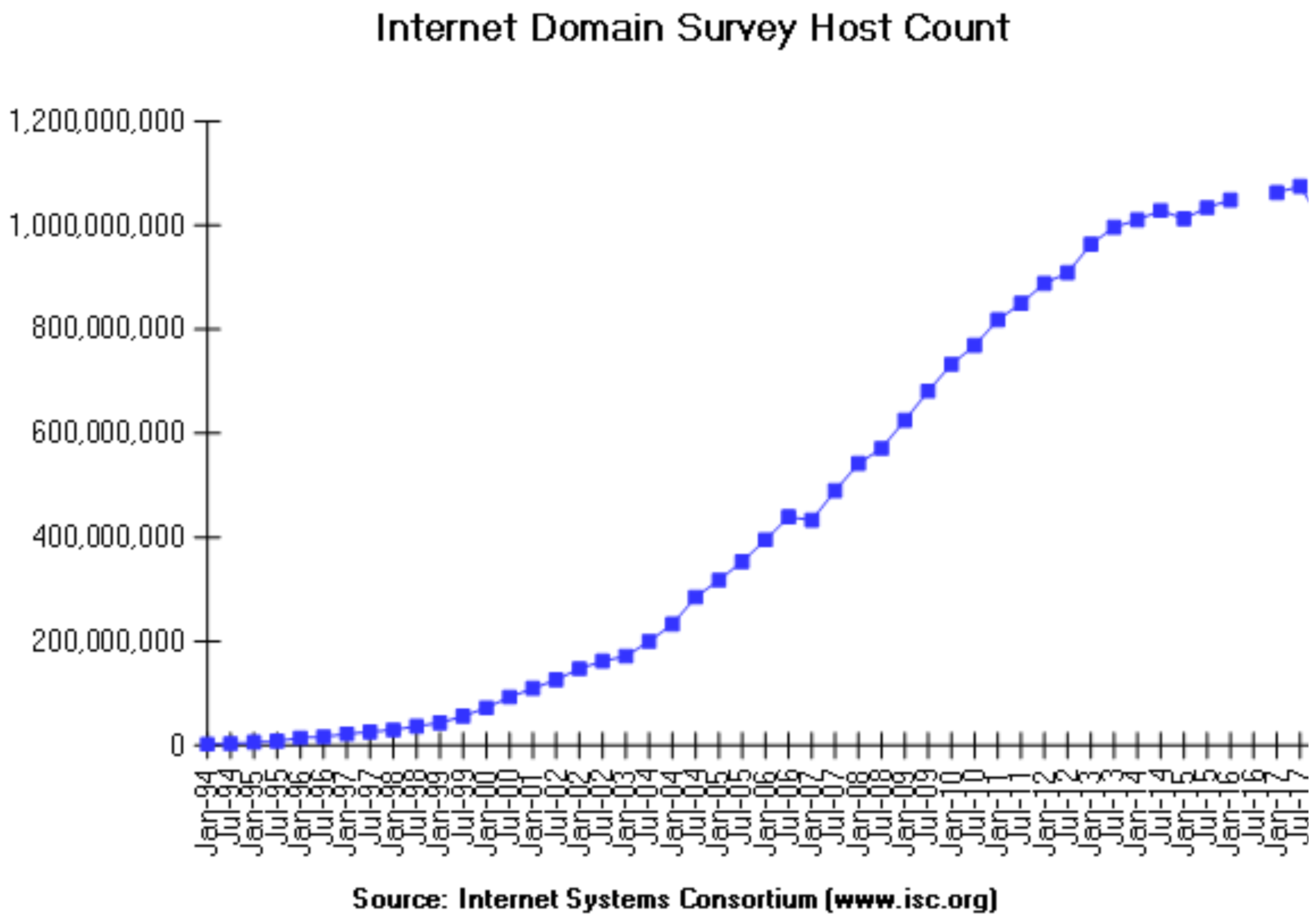
(source: https://www.networkworld.com/article/3153244/data-center/solid-state-drives-are-now-larger-than-hard-disk-drives-the-impact-for-your-data-center.html)

Network Capacity Still Increasing

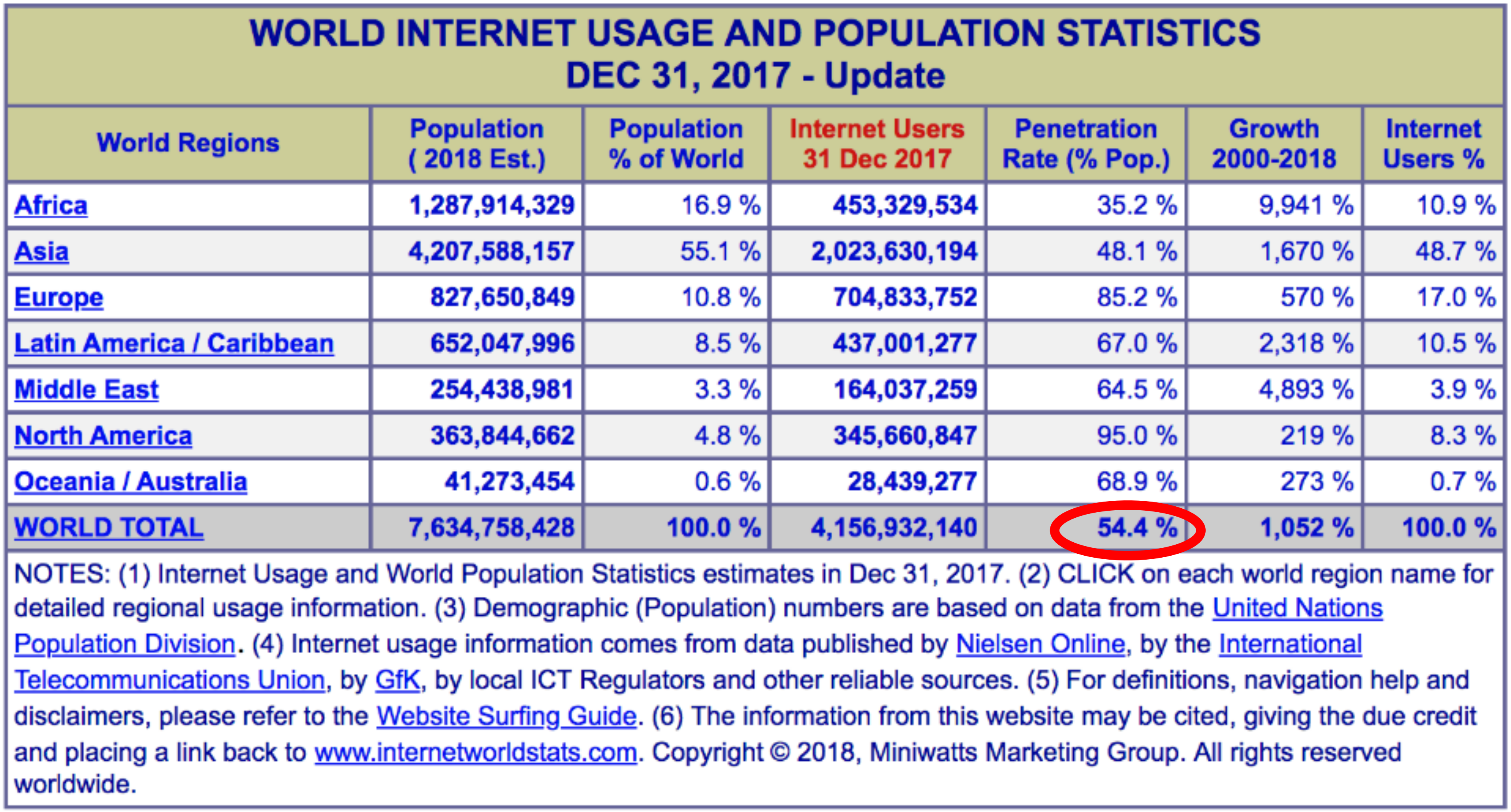


(source: http://www.ospmag.com/issue/article/Time-Is-Not-Always-On-Our-Side)

Internet Scale: 1.06 Billion Hosts (Jan 2017)



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

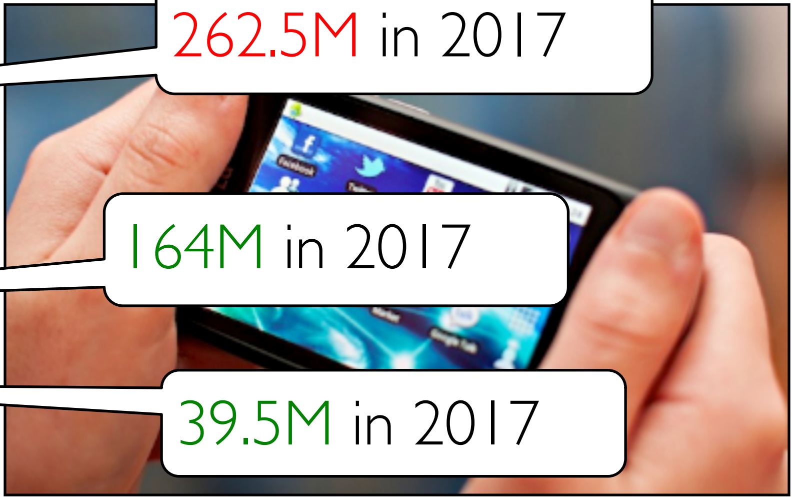
- In 2011, smartphone shipments exceeded PC shipments!
- 2011 shipments:
  - 487M smartphones
  - 414M PC clients
    - » 210M notebooks
    - » 112M desktops
    - » 63M tablets
  - 25M smart TVs

1.53B in 2017

262.5M in 2017

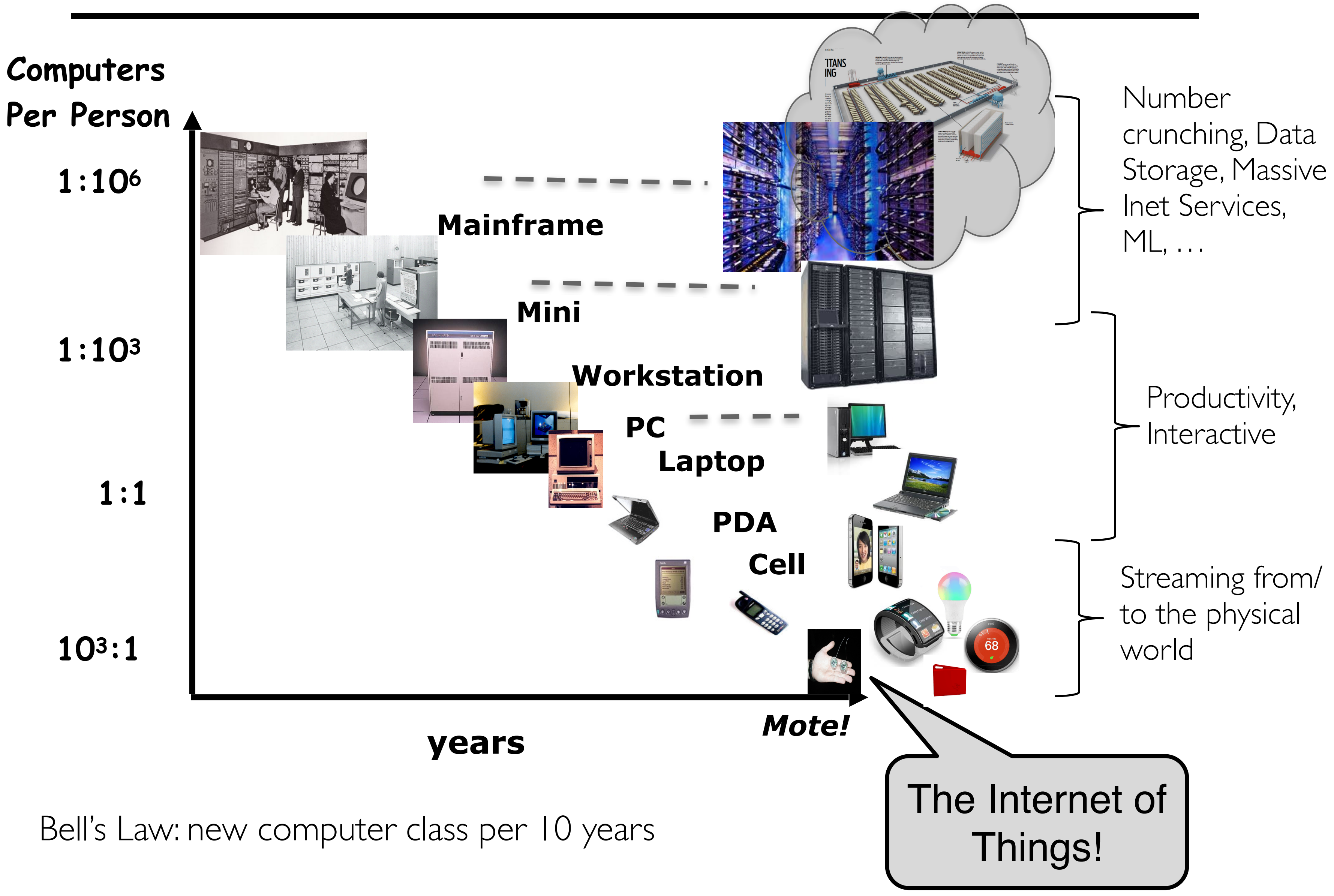
164M in 2017

39.5M in 2017



- 4 billion phones in the world → smartphones over next few years
- Then…

People-to-Computer Ratio Over Time


Computers Per Person
1:10⁶
1:10³
1:1
10³:1
years
Mainframe
Mini
Workstation
PC
Laptop
PDA
Cell
Mote!
Number crunching, Data Storage, Massive Inet Services, ML, …
Productivity, Interactive
Streaming from/ to the physical world
The Internet of Things!

Bell's Law: new computer class per 10 years

Vast Range of Timescales

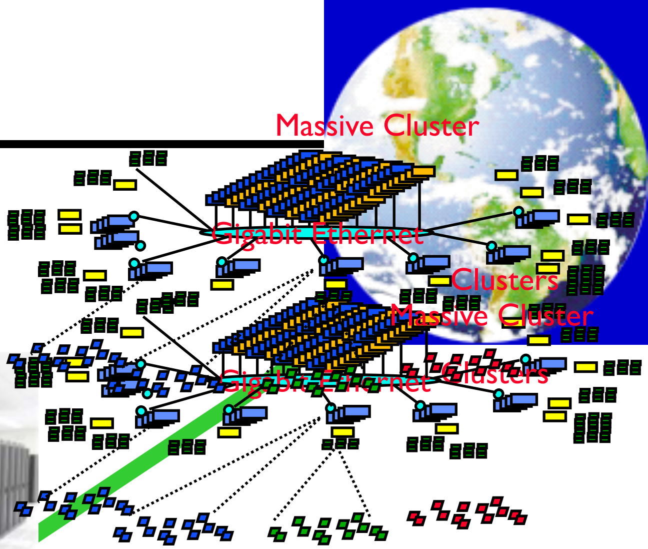
Jeff Dean: "Numbers Everyone Should Know"

L1 cache reference	0.5 ns
Branch mispredict	5 ns
L2 cache reference	7 ns
Mutex lock/unlock	25 ns
Main memory reference	100 ns
Compress 1K bytes with Zippy	3,000 ns
Send 2K bytes over 1 Gbps network	20,000 ns
Read 1 MB sequentially from memory	250,000 ns
Round trip within same datacenter	500,000 ns
Disk seek	10,000,000 ns
Read 1 MB sequentially from disk	20,000,000 ns
Send packet CA->Netherlands->CA	150,000,000 ns

Key Stroke / Click
100 ms

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

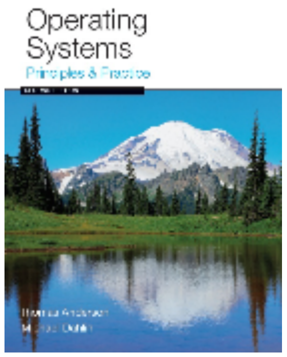
Scalable, Reliable, Secure Services

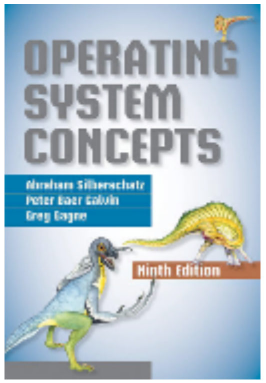
Databases
Information Collection
Remote Storage
Online Games
Commerce
…

MEMS for Sensor Nets

Infrastructure, Textbook & Readings

- Infrastructure
  - Website: http://sharif.edu/~kharrazi/courses/40424-002
  - Discord
- Textbook: Operating Systems: Principles and Practice (2nd Edition) Anderson and Dahlin
- Recommend: Operating Systems Concepts, 9th Edition Silbershatz, Galvin, Gagne
- 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 Homework: Learn Systems Programming
  1. Tools, Autograding, recall C, executable
  2. Simple Shell
  3. Web server
  4. Memory Management
- Three Group Projects (Pintos in C)
  1. Threads & Scheduling
  2. User-programs
  3. File Systems

Group Projects

- Project teams have 4 members
  - Must work in groups in "the real world"
- Communicate with colleagues (team members)
  - Communication problems are natural
  - What have you done?
  - What answers you need from others?
  - Dividing up by Task is the worst approach. Work as a team.
  - 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 right away (hopefully Today!)
  - Vagrant virtualbox – VM environment for the course
    - » Consistent, managed environment on your machine
  - Get familiar with all the tools
  - Submit to autograder via git
- Early Drop
  - Given the assignments, this is a highly rewarding but time consuming course
  - If you are not serious about taking, please drop early

Preparing Yourself for this Class

- The projects will require you to be very comfortable with programming and debugging C
  - Pointers (including function pointers, void*)
  - Memory Management (malloc, free, stack vs heap)
  - Debugging with GDB
- You will be working on a larger, more sophisticated code base than anything you've likely seen in previous classes
- C programming reference (still in beta):
  - https://cs162.eecs.berkeley.edu/ladder/

Grading

- 15% Midterms
- 20% Final
- 25% Homework
- 40% Group HWs
- Group HWs grading
    - [10 pts] Initial design
    - [10 pts] Design review
    - [10 pts] Design document
    - [60 pts] Code (3 checkpoints)
    - [10 pts] Final design
- Submission via git push to release branch
- Regular git push so TA sees your progress

CE424 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

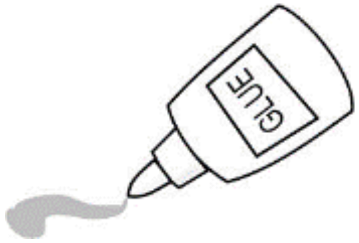
We compare all project submissions against prior year submissions and online solutions and will take actions (described on the course overview page) against offenders

More rules on the course website.

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

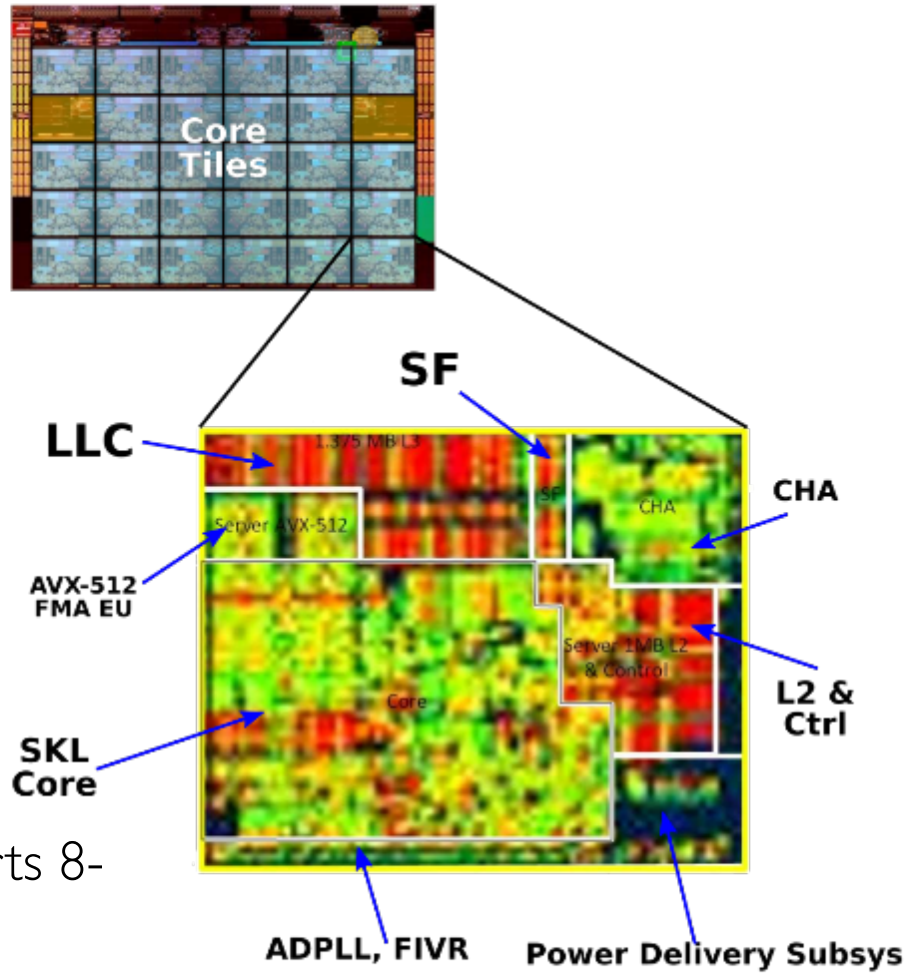
Challenge: Complexity

- 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!

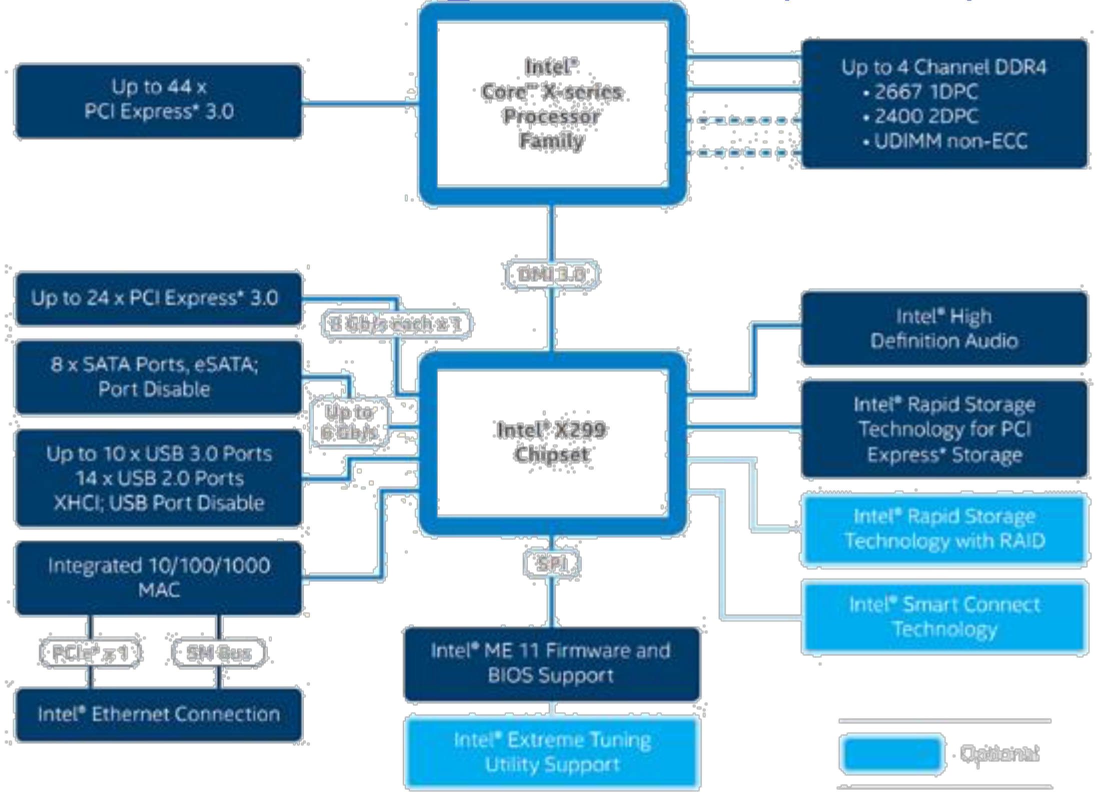
The World Is Parallel: Intel SkyLake (2017)

- Up to 28 Cores, 56 Threads
  - 694 mm^2 die size (estimated)
- Many different instructions
  - Security, Graphics
- Caches on chip:
  - L2: 28 MiB
  - Shared L3: 38.5 MiB (non-inclusive)
  - Directory-based cache coherence
- Network:
  - On-chip Mesh Interconnect
  - Fast off-chip network directly supports 8-chips connected
- DRAM/chips
  - Up to 1.5 TiB
  - DDR4 memory

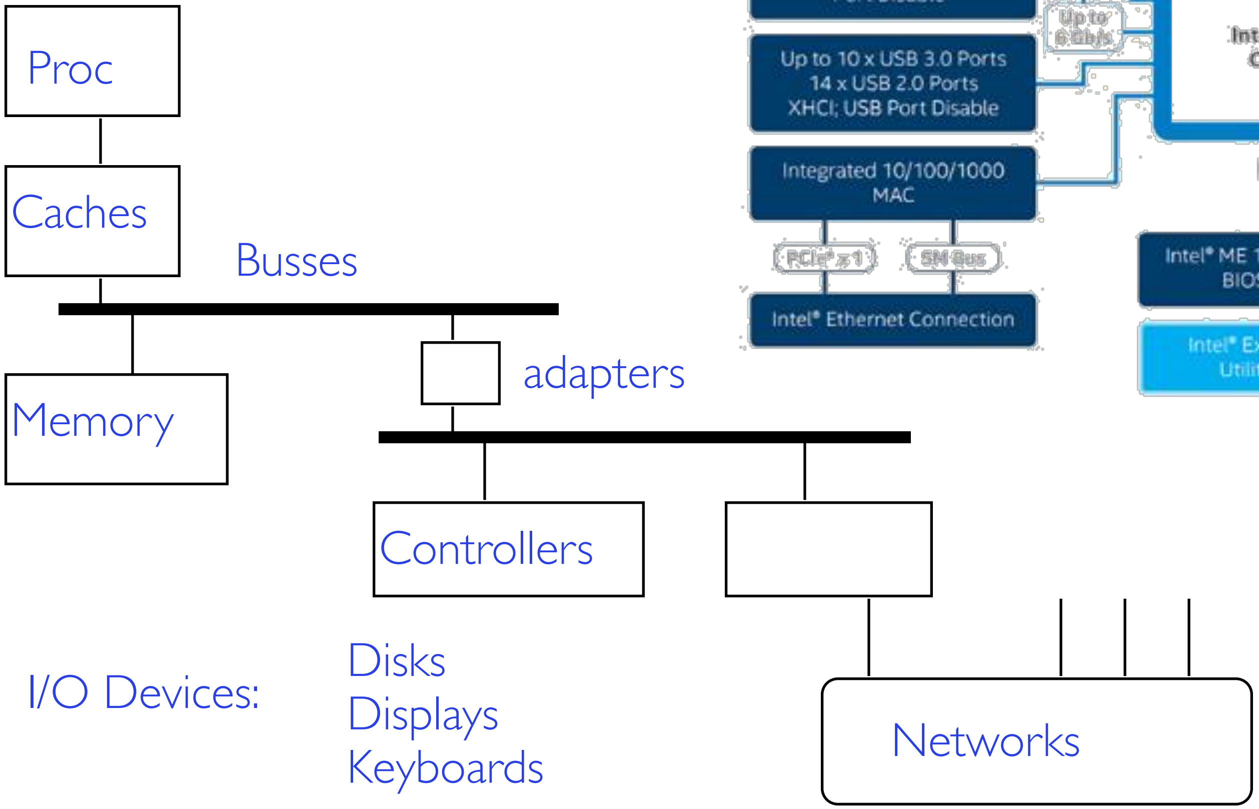


HW Functionality comes with great complexity!

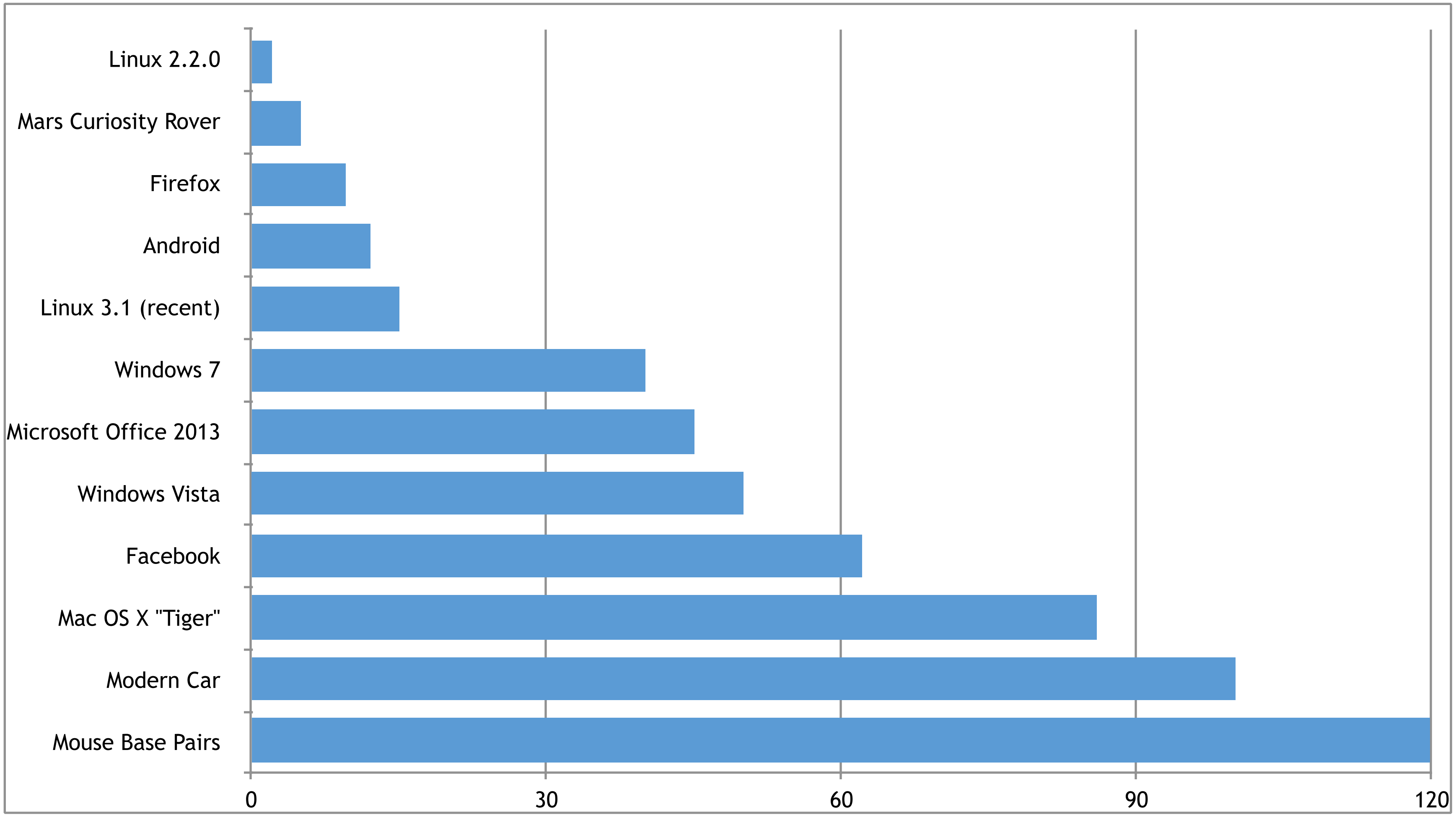
Intel Skylake-X I/O Configuration



Up to 44 x PCI Express* 3.0
Intel® Core™ X-series Processor Family
Up to 4 Channel DDR4
• 2667 1DPC
• 2400 2DPC
• UDIMM non-ECC
DMI 3.0
Up to 24 x PCI Express* 3.0
8 Gb/s each x1
8 x SATA Ports, eSATA; Port Disable
Up to 6 Gb/s
Up to 10 x USB 3.0 Ports 14 x USB 2.0 Ports XHCI; USB Port Disable
Intel® X299 Chipset
Intel® High Definition Audio
Intel® Rapid Storage Technology for PCI Express* Storage
Intel® Rapid Storage Technology with RAID
Intel® Smart Connect Technology
Integrated 10/100/1000 MAC
PCIe* x1
SM Bus
Intel® Ethernet Connection
SPI
Intel® ME 11 Firmware and BIOS Support
Intel® Extreme Tuning Utility Support
Optional



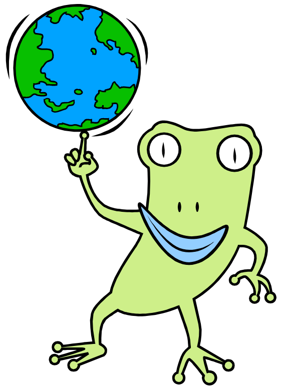
Increasing Software Complexity



Millions of Lines of Code

(source https://informationisbeautiful.net/visualizations/million-lines-of-code/)

Example: Some Mars Rover ("Pathfinder") Requirements

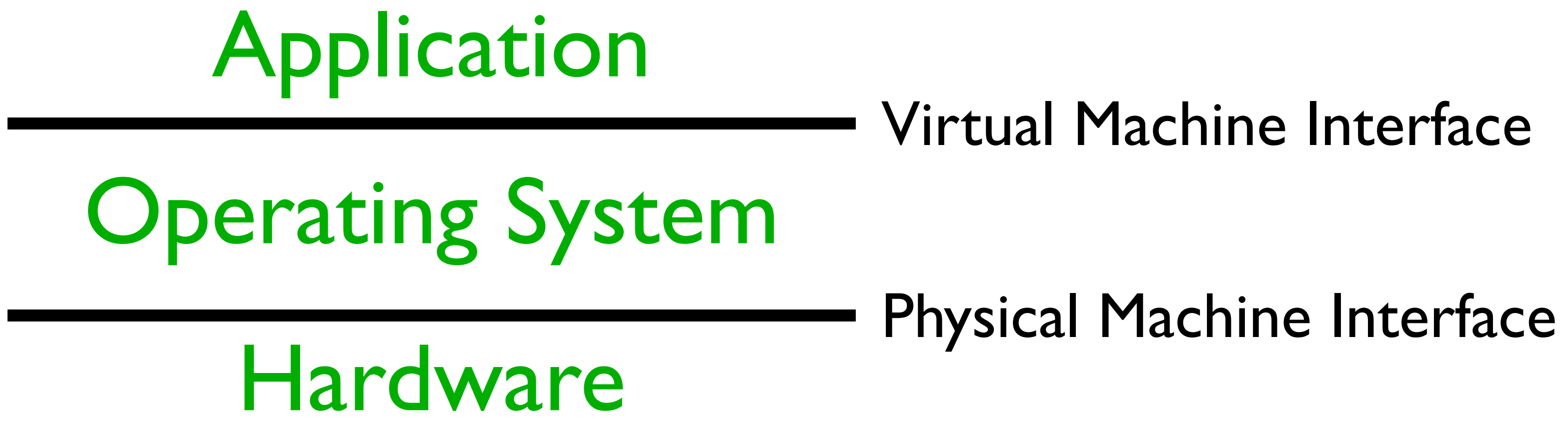


- Pathfinder hardware limitations/complexity:
  - 20Mhz processor, 128MB of DRAM, VxWorks OS
  - cameras, scientific instruments, batteries, solar panels, and locomotion equipment
  - Many independent processes work together
- Can't hit reset button very easily!
  - Must reboot itself if necessary
  - Must always be able to receive commands from Earth
- Individual Programs must not interfere
  - Suppose the MUT (Martian Universal Translator Module) buggy
  - Better not crash antenna positioning software!
- Further, all software may crash occasionally
  - Automatic restart with diagnostics sent to Earth
  - Periodic checkpoint of results saved?
- Certain functions time critical:
  - Need to stop before hitting something
  - Must track orbit of Earth for communication
- A lot of similarity with the Internet of Things?
  - *Complexity,* QoS, Inaccessbility, Power limitations … ?

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

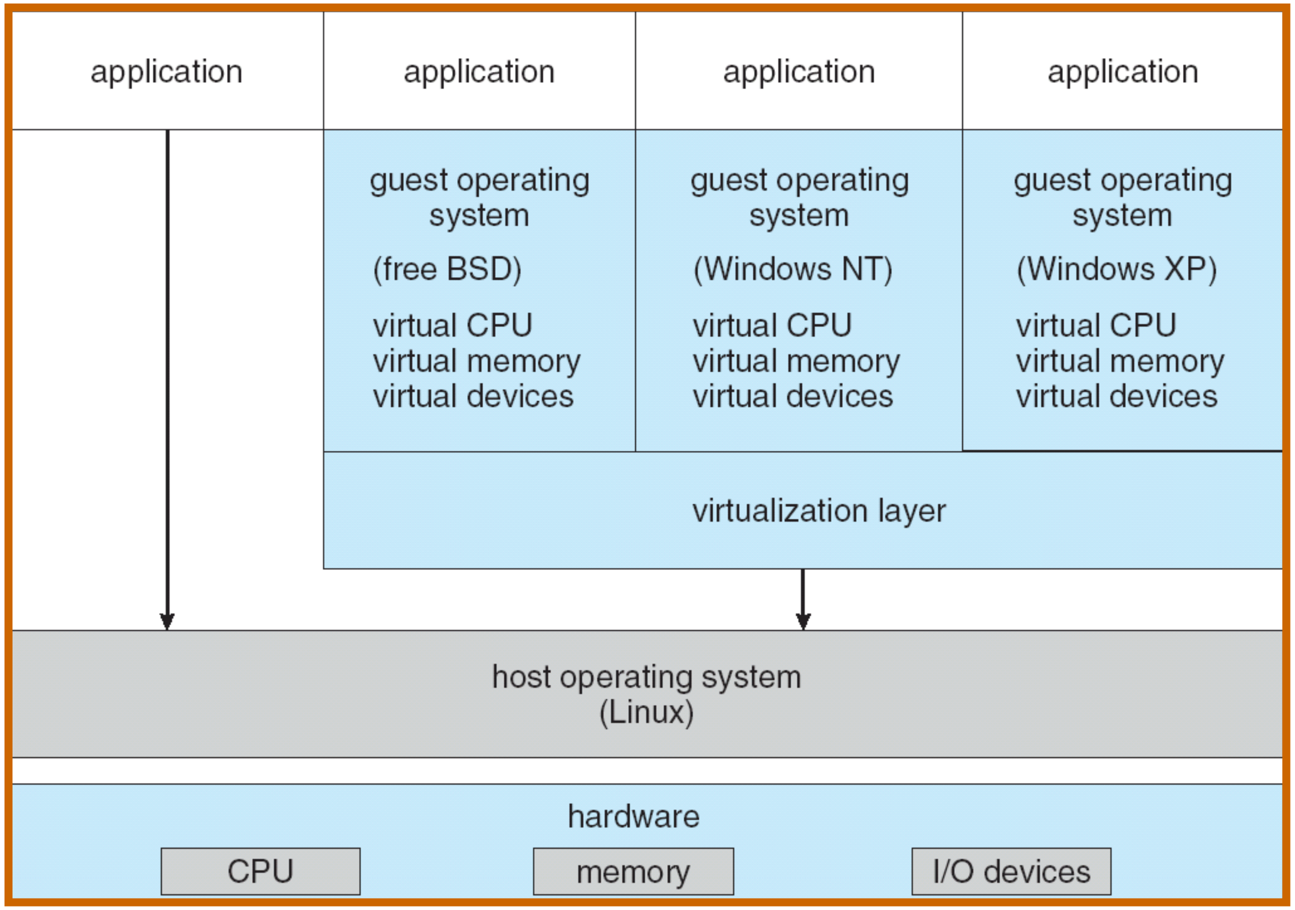
- Software emulation of an abstract machine
  - Give programs illusion they own the machine
  - Make it look like hardware has features you want
- Two types of "Virtual Machine"s
  - Process VM: supports the execution of a single program; this functionality typically provided by OS
  - System VM: supports 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



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
- CE424 combines things from many other areas of computer science:
  - Languages, data structures, hardware, and algorithms