OPERATING SYSTEMS

RONG ZHENG

Disclaimer: Many materials used in the slides are adopted from those of other colleagues

GOAL OF THIS COURSE

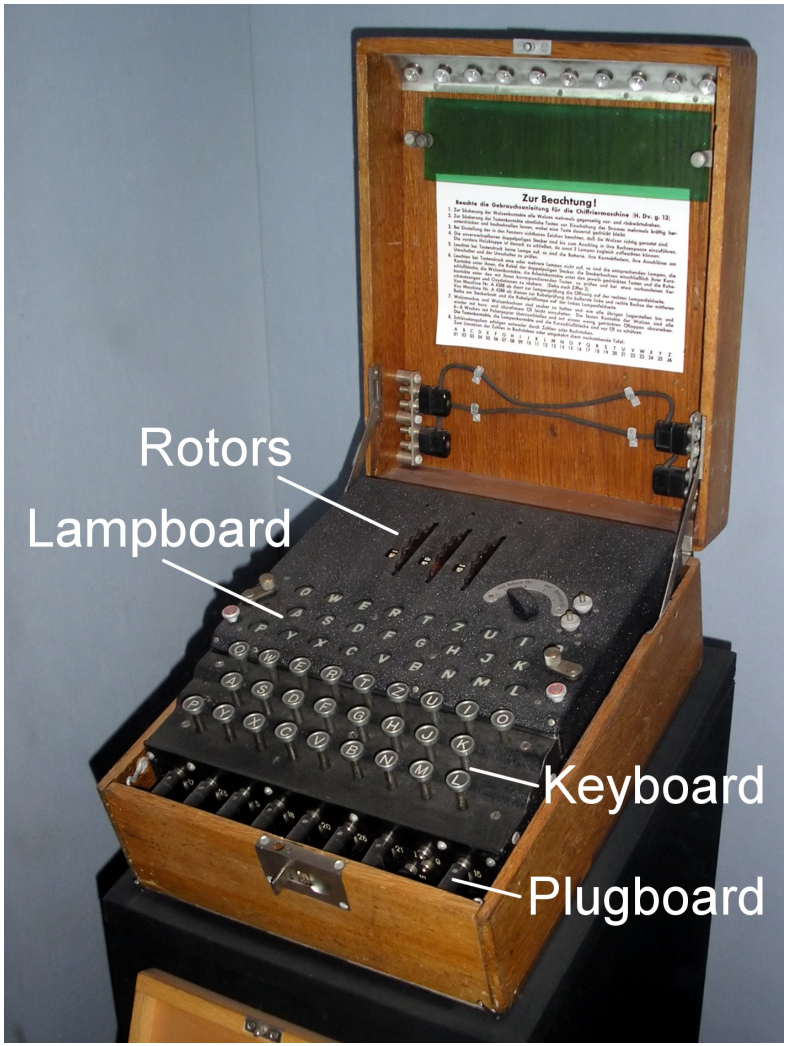
Learn how “systems” work

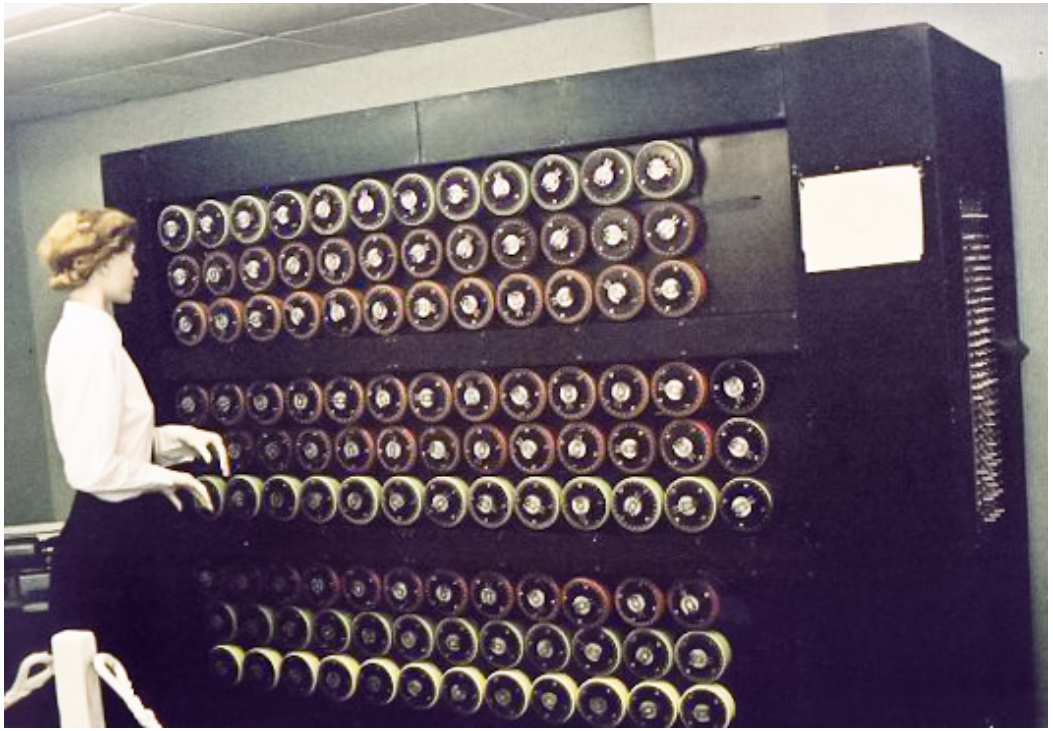
Main challenges in building systems

Principles of system design, i.e., how to address these challenges

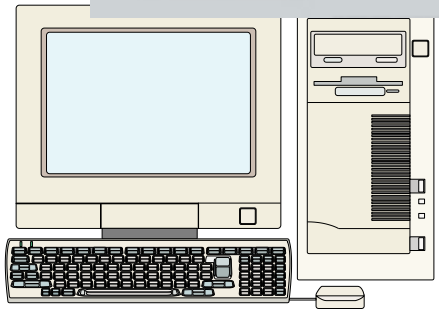
Learn how to apply these principles to building systems

WHAT DO TURING'S BOMBE MACHINE AND TODAY'S COMPUTER HAVE IN COMMON?

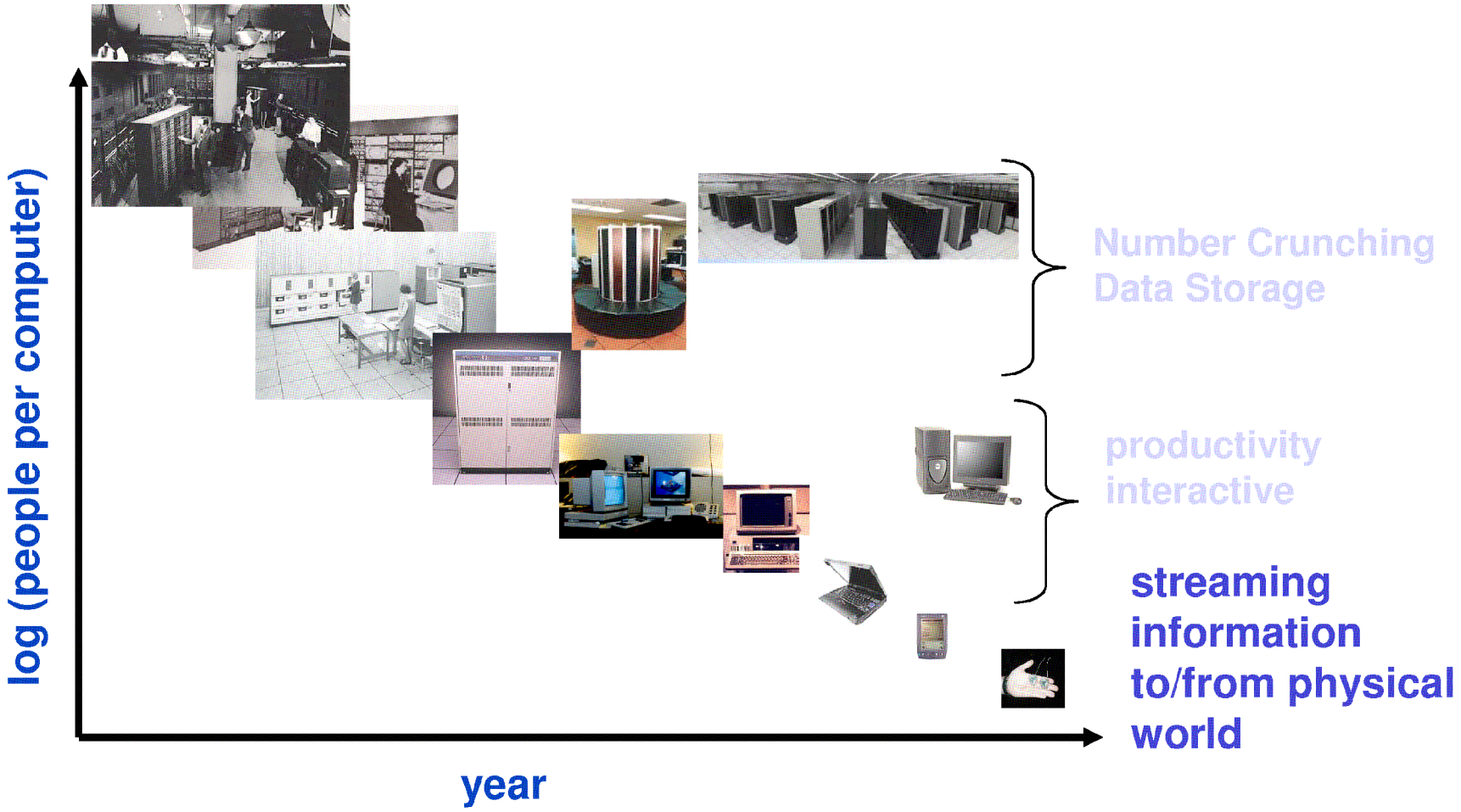




COMPUTING DEVICES EVERYWHERE



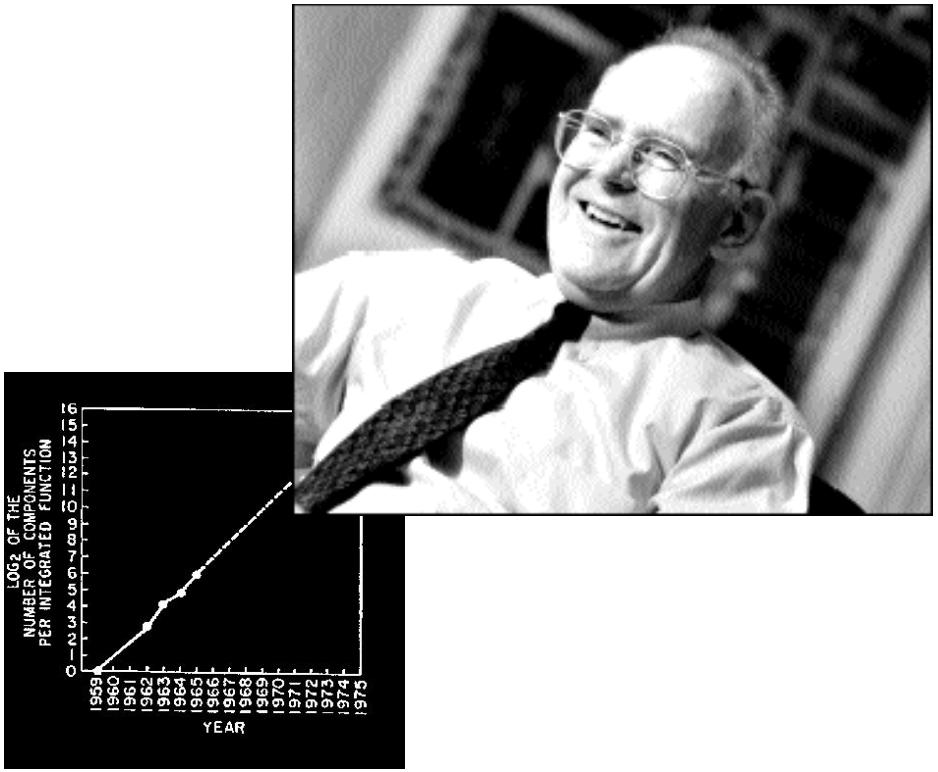
PEOPLE-TO-CPU RATIO OVER TIME



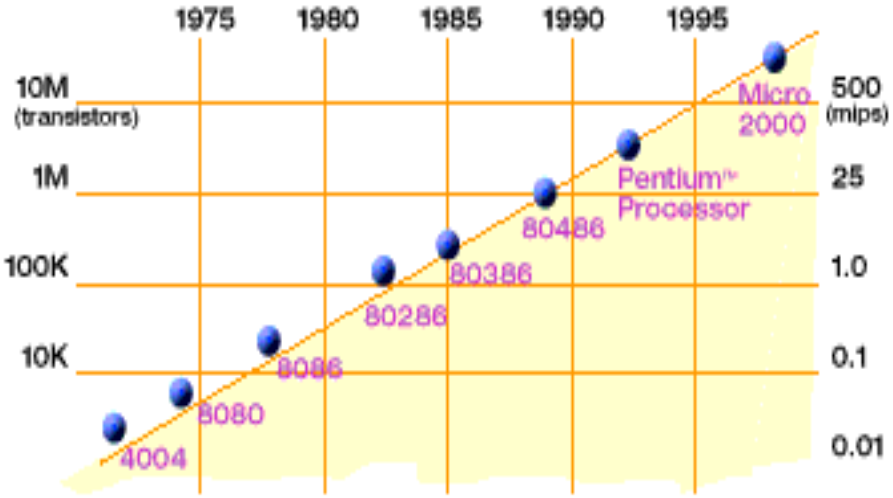
Today: Multiple CPUs/person!

- Approaching 100s?

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.



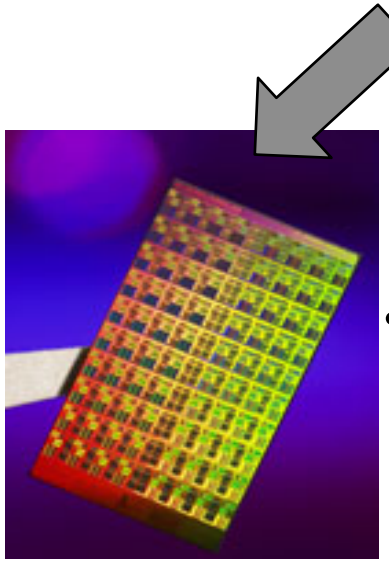
2X transistors/Chip Every 1.5 years

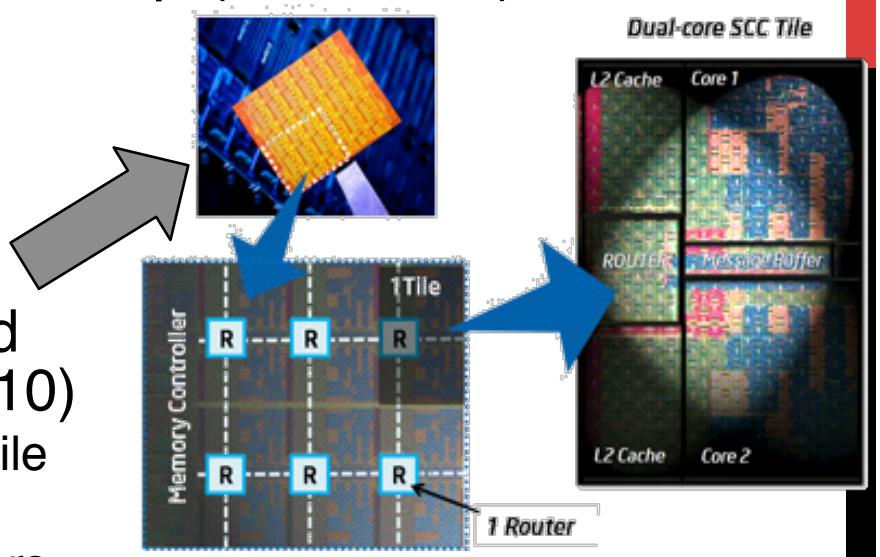
Called "Moore's Law"

Microprocessors have become smaller, denser, and more powerful.

MANYCORE CHIPS

- 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

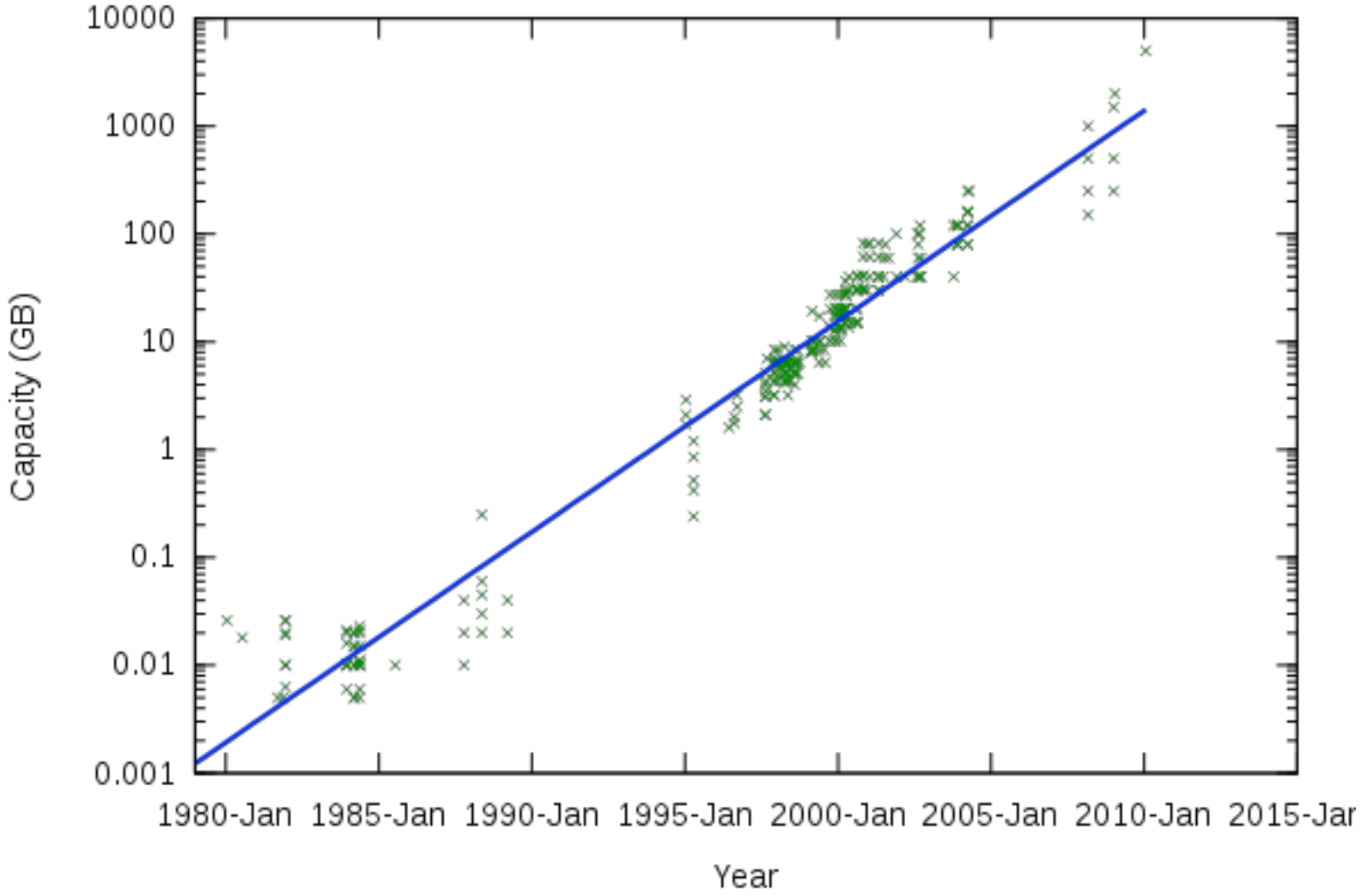
- 64? 128? Hard to say exact boundary

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

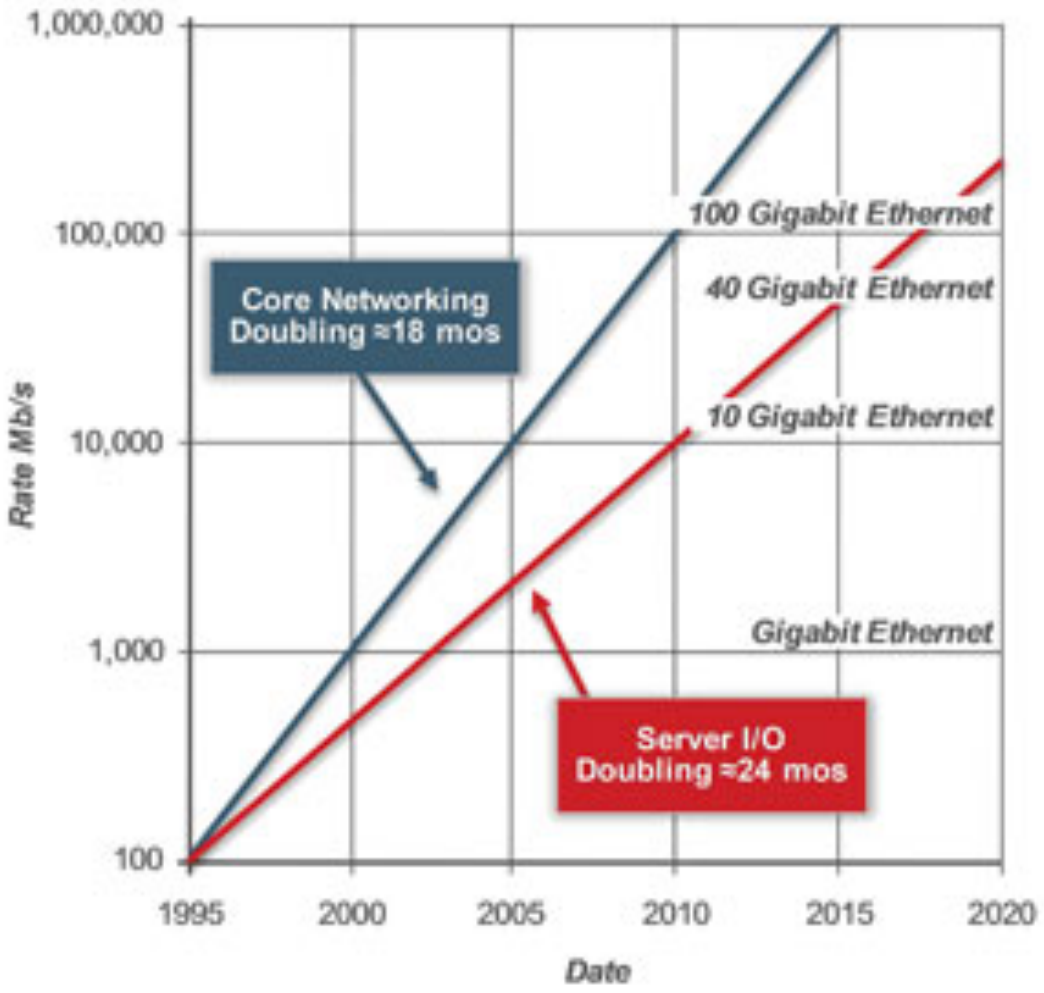
STORAGE CAPACITY



***Retail* hard disk capacity in GB**

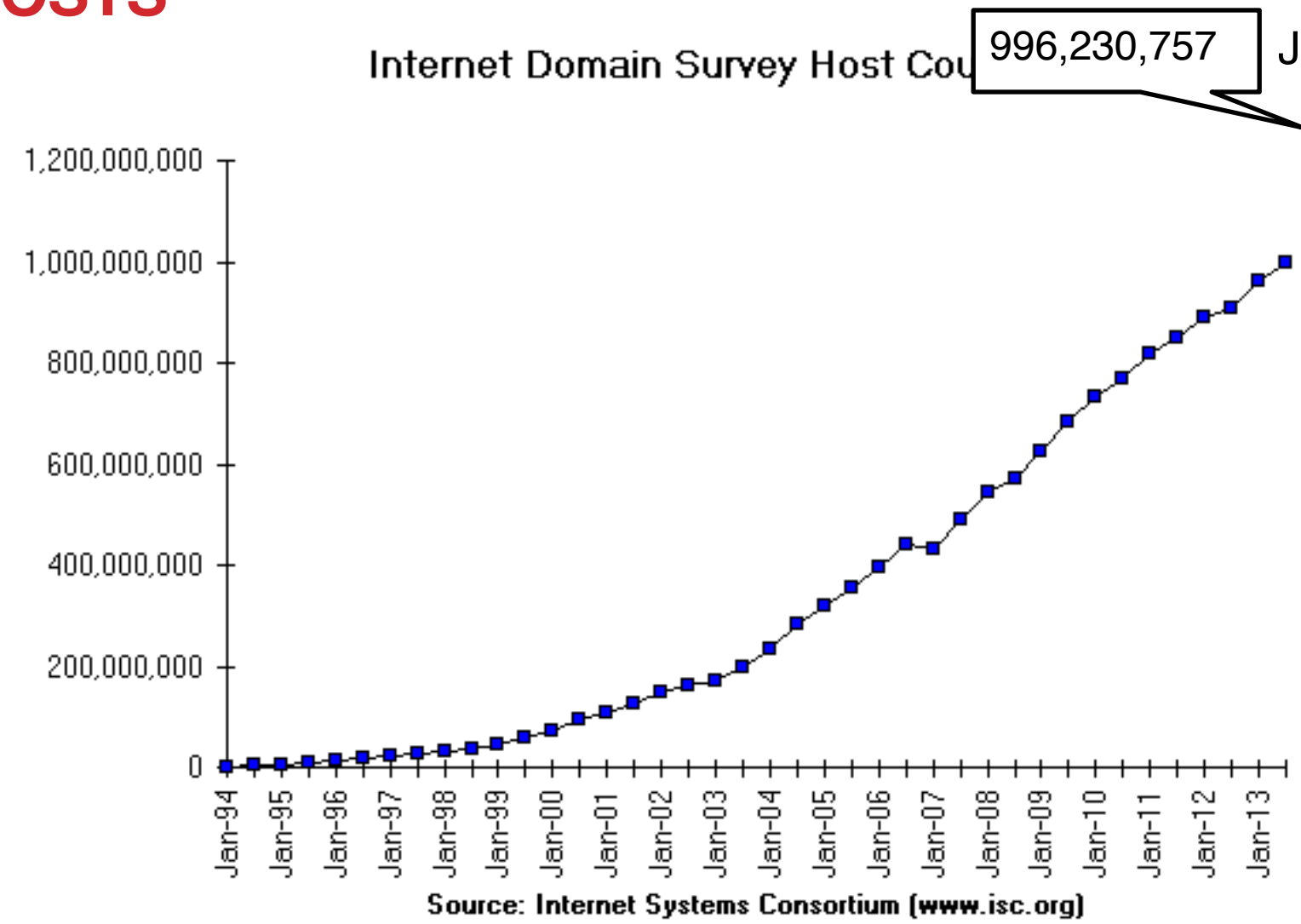
(source: http://www.digitaltonto.com/2011/our-emergent-digital-future/)

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: ~2.5 BILLION USERS!

WORLD INTERNET USAGE AND POPULATION STATISTICS June 30, 2012						
World Regions	Population (2012 Est.)	Internet Users Dec. 31, 2000	Internet Users Latest Data	Penetration (% Population)	Growth 2000-2012	Users % of Table
Africa	1,073,380,925	4,514,400	**167,335,676**	15.6 %	3,606.7 %	7.0 %
Asia	3,922,066,987	114,304,000	**1,076,681,059**	27.5 %	841.9 %	44.8 %
Europe	820,918,446	105,096,093	**518,512,109**	63.2 %	393.4 %	21.5 %
Middle East	223,608,203	3,284,800	**90,000,455**	40.2 %	2,639.9 %	3.7 %
North America	348,280,154	108,096,800	**273,785,413**	78.6 %	153.3 %	11.4 %
Latin America / Caribbean	593,688,638	18,068,919	**254,915,745**	42.9 %	1,310.8 %	10.6 %
Oceania / Australia	35,903,569	7,620,480	**24,287,919**	67.6 %	218.7 %	1.0 %
WORLD TOTAL	**7,017,846,922**	**360,985,492**	**2,405,518,376**	**34.3 %**	**566.4 %**	**100.0 %**

NOTES: (1) Internet Usage and World Population Statistics are for June 30, 2012. (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:

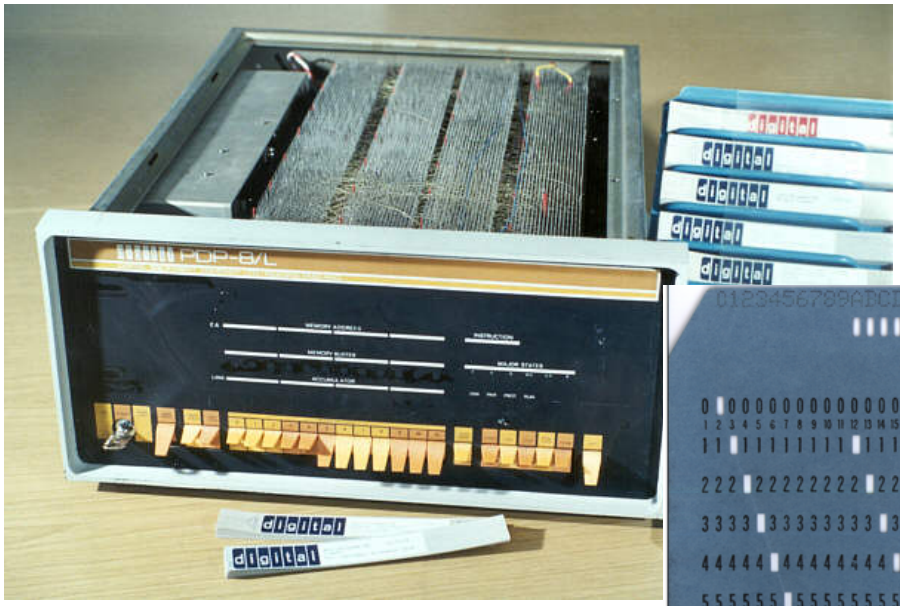
- 487M smartphones
- 414M PC clients
  - 210M notebooks
  - 112M desktops
  - 63M tablets
- 25M smart TVs

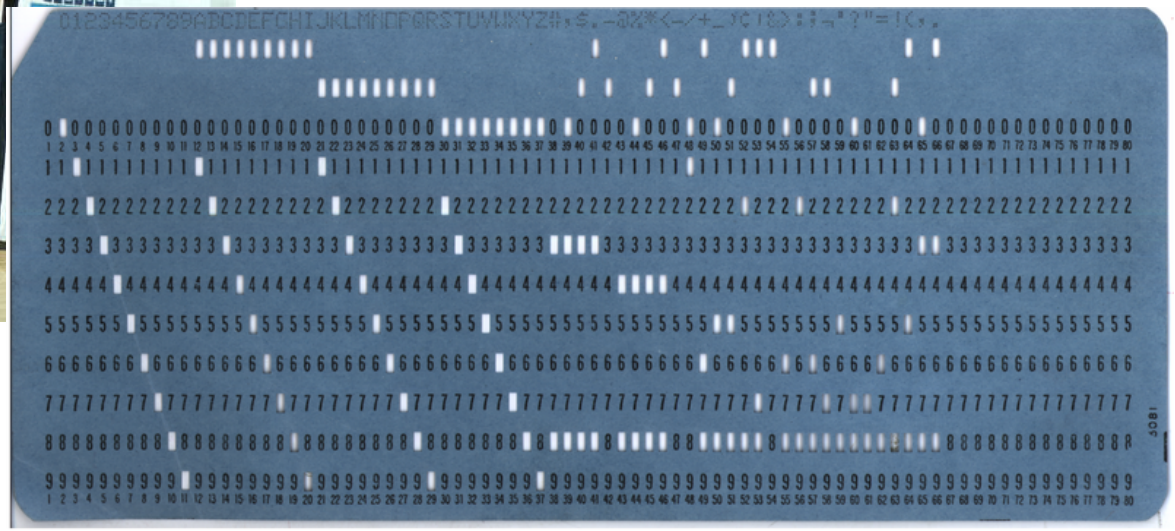
4 billion phones in the world → smartphone over next decade

QUESTION

How to program/manage such complexity?

- **Abstractions!**





THE INSTRUCTOR

Rong Zheng

- **Office: ITB 121**
- **Office hr: Tue. 4:30 – 6:30pm**
- **Research areas:**
  - Mobile & pervasive computing
  - Wireless networking

THE TAS

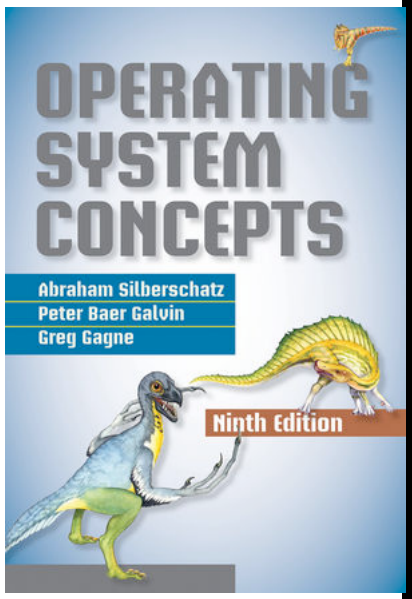
- **Ala Shaabana (shaabaa@mcmaster.ca)**
  - January
- **Yuanhao Yu (yhyu.mail@gmail.com)**
  - Feburary
- **Qiang Xu (xuq22@mcmaster.ca)**
  - March

TAs will be present during all lab sessions (starting next week)

Attendance of lab sessions NOT required

COURSE MATERIALS

Textbook: A. Silberschatz, P. Gavin and G. Gagne, *Operating Systems Concepts*, 9th edition, Wiley & Sons



Nachos tutorial:

http://www.cas.mcmaster.ca/~rzheng/Nachos_Tutorial/

Course webpage
http://www.cas.mcmaster.ca/~rzheng/course/CAS3SH3w15/

- All course materials (lecture notes, tutorials, projects)
- Announcement

Avenue to Learn

- Exercise
- Group enrollment
- Announcement

SCOPE OF THE COURSE

- **Process management**
- **Synchronization**
- **Memory management**
- **File system & I/O**
- **Advanced topics**

COURSE ORGANIZATION

Lectures

Tutorials

- Online materials

Lab sessions

- Not mandatory
- TAs will be around to answer questions

MARKINGS

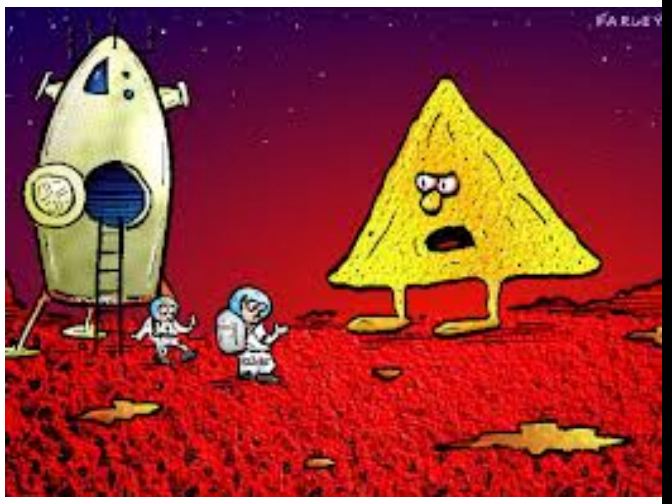
- **4 programming assignments (10, 10, 10, 15%)**
  - One bonus programming assignment 10%
- **Exercise (10%)**
  - Avenue
- **Midterm (20%)**
  - In class
- **Final (25%, not-comphrensive)**

Curved. To get an A in the class (>=85%)

PROGRAMMING ASSIGNMENTS

Nachos (Not Another Complete Heuristic Operating System) 5.0j

- **An instructional OS ported to Java**
- **To understand abstract "concepts" introduced in the lectures**
- **To implement key building blocks of OS**



Yes, you do need to know how to program in Java!

PROGRAMMING ASSIGNMENTS

- **Project 1 individually**
- **Project 2 – 4 in groups (maximum 4 students)**
  - Group sign up via Avenue
  - Self-enrollment via avenue finalized by the end of the 2nd week
  - Changes discouraged and should be approved by the instructor
- **Use SVN set up by dept sysadmin**
- **Two phrases: (design) questions and code submission**
- **Autograder codes will be provided for testing**
- **In some projects, you may need to develop your own test cases/codes**
- **More details will be provided by the TAs during lab sessions**

WORK WITH YOUR TEAM MATES

- **Regular code review meetings**
- **Pair programming**
- **Alert the instructor problems earlier on**

THE HARE AND THE TURTLE

BEHAVING IN THE CLASSROOM

Non course-related activities such as answering their phones, browsing the web or playing solitaire are discouraged

QUESTIONS?

Email rzheng@mcmaster.ca with subject title “3SH3”

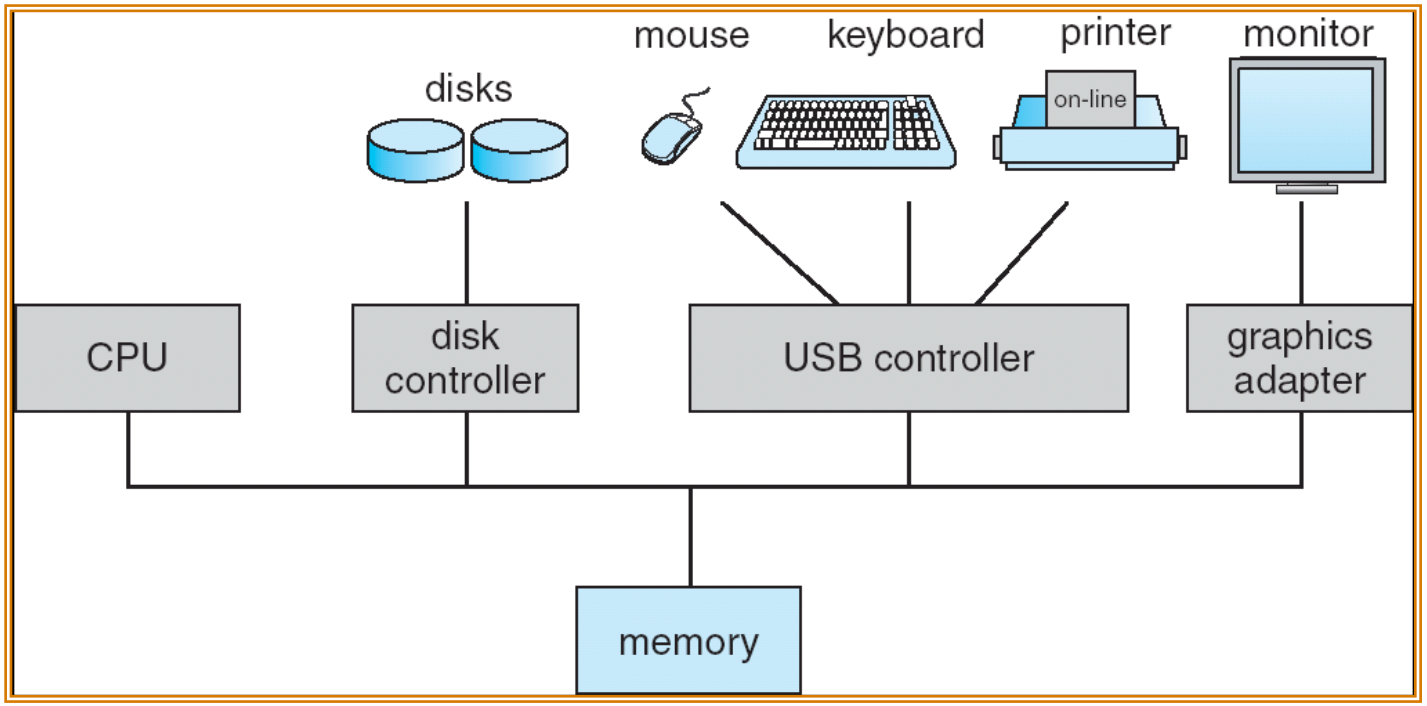
THE REAL INTRODUCTION TO OS

- WHAT IS OPERATING SYSTEM?
- TYPES OF OS
- HISTORY OF OS

COMPUTER SYSTEM ORGANIZATION

One or more CPUs, device controllers connect through common bus providing access to shared memory

Concurrent execution of CPUs and devices competing for memory cycles



CHALLENGE: SCALE AND DYNAMIC RANGE

Enormous scale, heterogeneity, and dynamic range:

- CPU: sensor motes → GPUs
  - Cores: one → 100s [2-orders of magnitude variation]
  - Clusters: few machines → 10,000s machines [4 orders of mag.]
- Network: Inter-core networks → Internet
  - Latency: nanosecs → secs (satellite) [9 orders of mag.]
  - Bandwidth: Kbps → Gbps [6 orders of mag.]
  - …
- Storage: caches → disks
  - Size: MB → TB [6 orders of mag.]
  - Access time: few nanosecs → millisecs [6 orders of mag.]

HOW DO WE TAME COMPLEXITY?

Every piece of computer hardware different

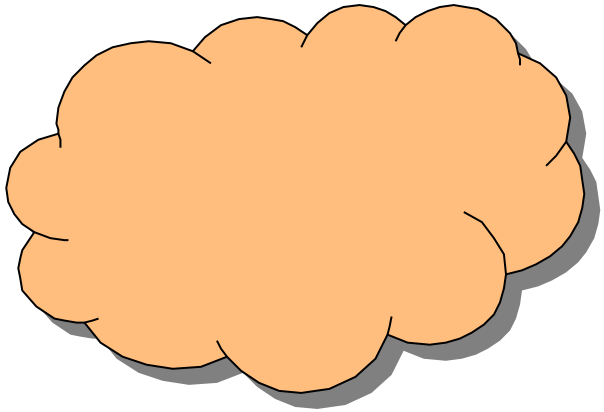
- Different CPU
  - Pentium, ARM, PowerPC, ColdFire
- Different amounts of memory, disk, …
- Different types of devices
  - Mice, keyboards, sensors, cameras, fingerprint readers, touch screen
- Different networking environment
  - Cable, DSL, Wireless, …

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?

WHAT IS AN OPERATING SYSTEM?

“What stands between the user and the bare machine”



WHAT IS AN OPERATING SYSTEM?

The basic software required to operate a computer.

Silberschatz and Gavin: "An OS is Similar to a government"

- Begs the question: does a government do anything useful by itself?

OS as a Traffic Cop:

- Manages all resources
- Settles conflicting requests for resources
- Prevent errors and improper use of the computer

OS as a facilitator ("useful" abstractions):

- Provides facilities/services that everyone needs
- Standard libraries, windowing systems
- Make application programming easier, faster, less error-prone

WHAT BELONGS TO OS

Loose definition: what shipped by the vendor

More strictly,

Kernel

- Part that stays in main memory
- Controls the execution of all other programs

Other programs interact with it through system calls

WHAT DO NOT BELONG TO OS

All user programs

- Compilers, spreadsheets, word processors, and so forth

Most utility programs

- mkdir is a user program calling mkdir()

The command language interpreter

- Anyone can write his/her UNIX shell

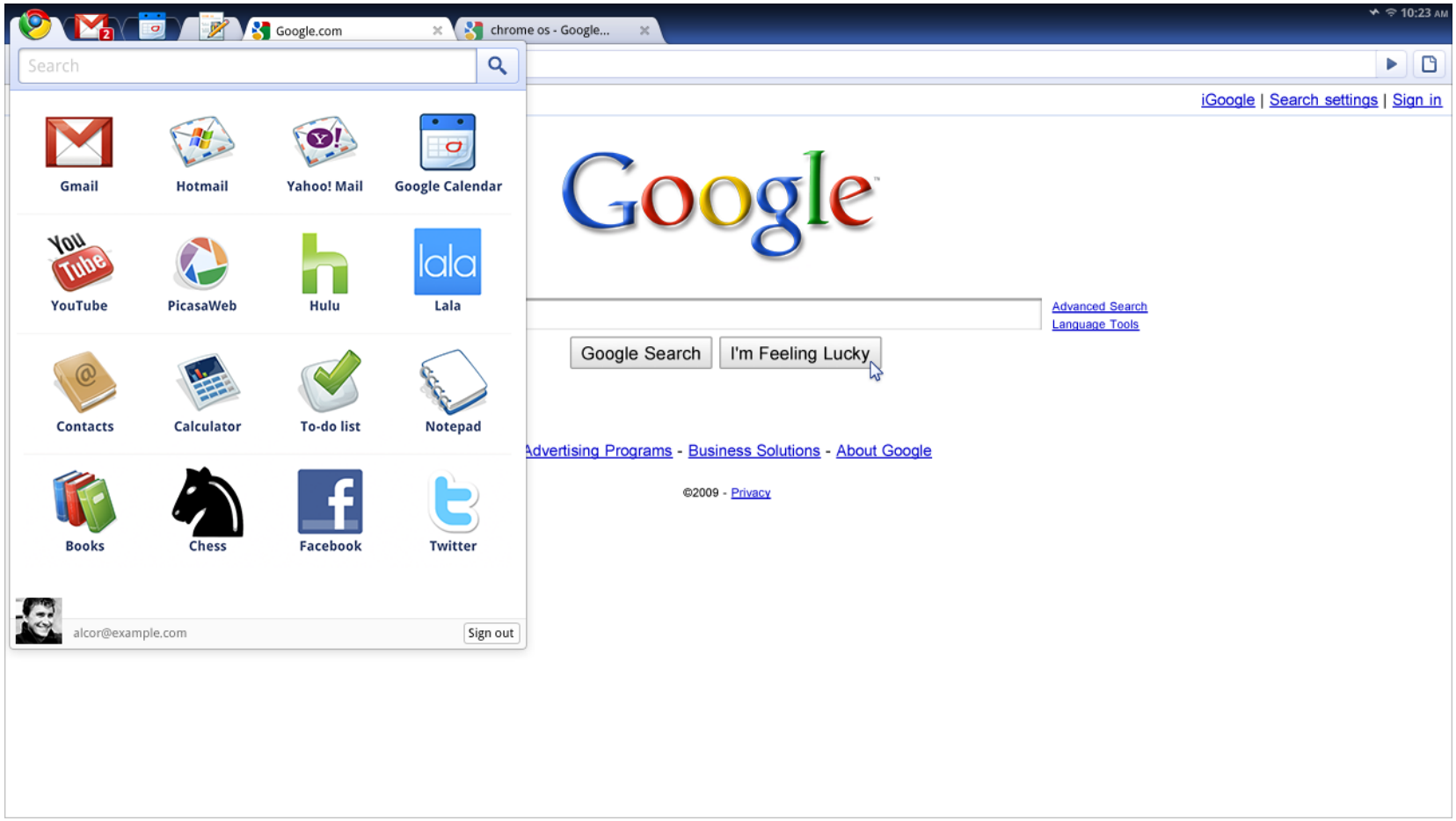
IS WEB BROWSER PART OF OS?

In 1998, the United States Federal Government charged Microsoft with anti-trust violations. One of the the complaints was that Microsoft had "bundled" its Internet Explorer (IE) web browser with its Windows operating system (OS).

In its defense, Microsoft asserted that the web browser was, in fact, part of the operating system. Therefore, it made no sense to talk of bundling the browser; it was in the natural order of things that an OS vendor would include a browser in their product.

In the end, Microsoft was found to be a monopoly, but received only minimal sanctions…

WAIT! WHAT ABOUT CHROME OS?



WHAT BELONGS TO OS

Loose definition: what shipped by the vendor

More strictly,

Kernel

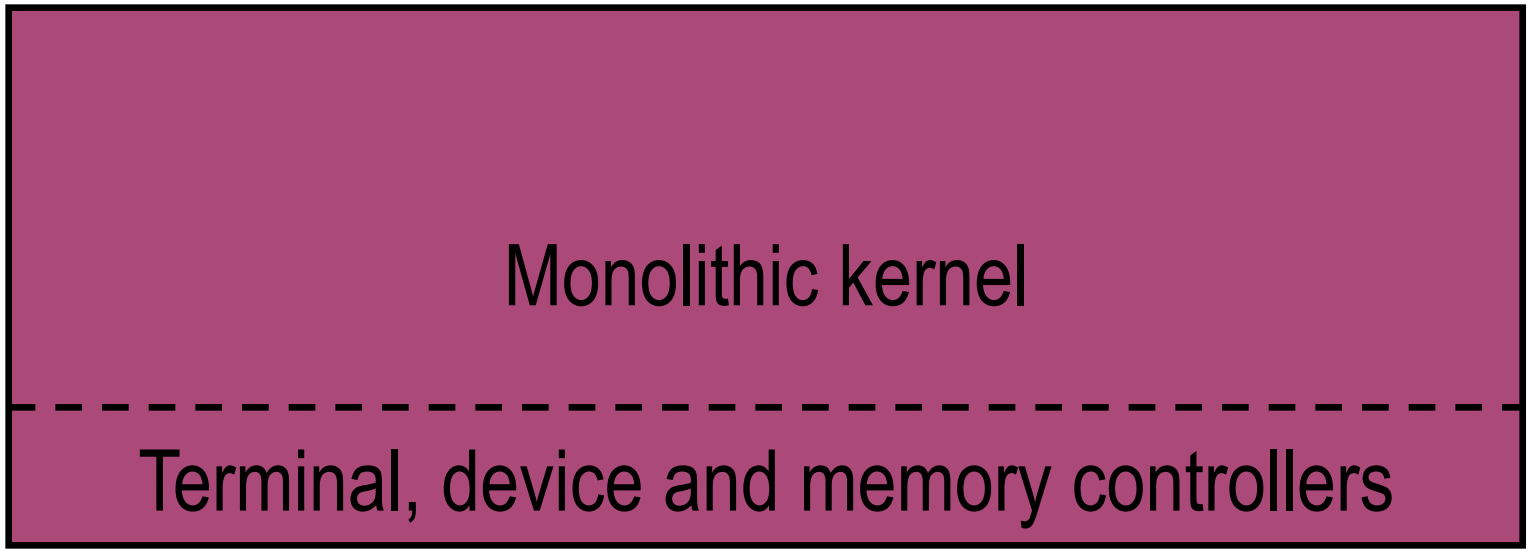
- Part that stays in main memory
- Controls the execution of all other programs

Other programs interact with it through system calls

DIFFERENT TYPES OF KERNELS

Monolithic kernel

- All kernel functions share the same address space
- This includes devices drivers and other kernel extensions



MICRO-KERNEL (MACH)

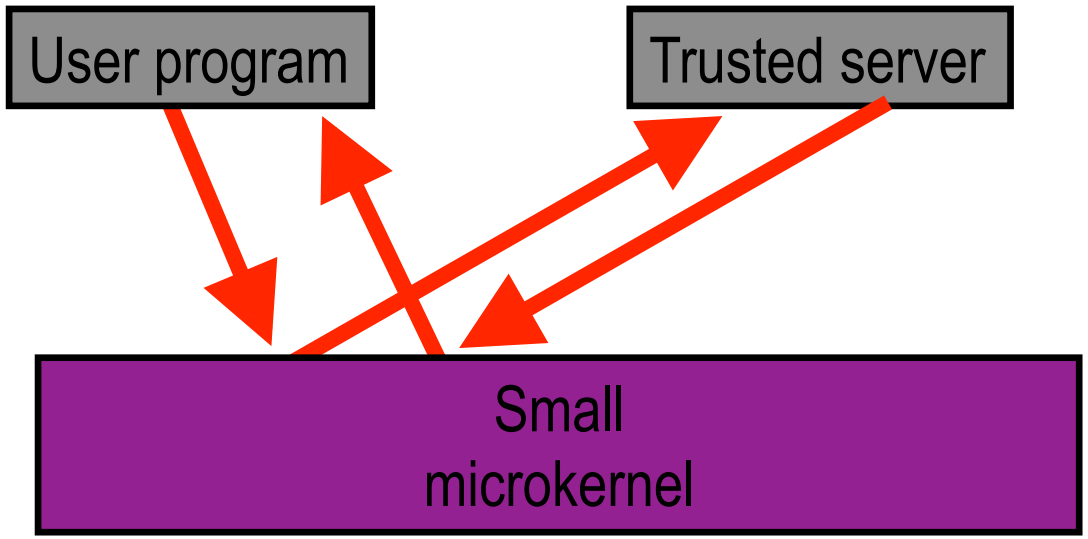
A reaction against "bloated" monolithic kernels

- Hard to manage, extend, debug and secure

Key idea is making kernel smaller by delegating non-essential tasks to trusted user-level servers

- Same idea as subcontracting

Microkernel keeps doing what cannot be delegated



MODULAR KERNELS (LINUX, WINDOWS)

Modules are object files whose contents can be linked to—and unlinked from—the kernel at any time

- Run inside the kernel address space
- Used to add to the kernel device drivers for new devices

Extensibility:

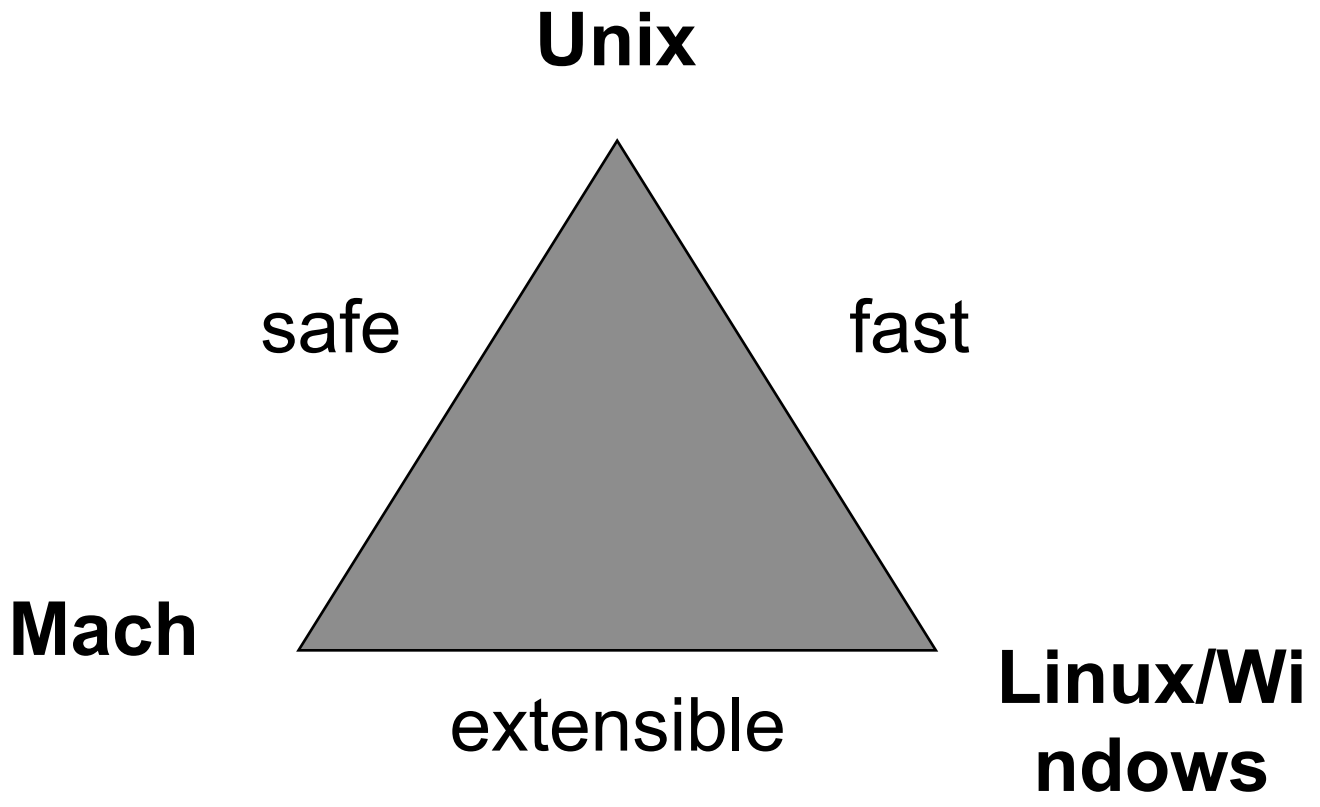
- Can add new features the kernel
- In many cases, the process is completely transparent to the user

Lack of performance penalty:

- Modules run in the kernel address space

Low reliability

A COMPARISON



KEY FUNCTIONS OF OS

1. To provide a better user interface
2. To manage the system resources
3. To protect users' programs and data

PHASE1 : HARDWARE EXPENSIVE, HUMAN CHEAP

Hardware: main frames

Representative OS:

- Eniac, … Multics

Focus: how to get the computation job done

- Uniprogramming

"I think there is a world market for maybe five computers." -- *Thomas Watson, chairman of IBM, 1943*

PHASE1 : HARDWARE EXPENSIVE, HUMAN CHEAP

Bach systems: Allow users to submit a batches of requests to be processed in sequence

Include a command language specifying what to do with the inputs

- Compile
- Execute and so forth

PHASE 2: HARDWARE CHEAPER, HUMAN EXPENSIVE

Hardware: PCs, Workstations

Representative OS:

- MSDOS, Windows, UNIX, Linux etc.

Focus: Increase the utilization of resources and user friendliness

- From interactive systems through consoles to graphical user interfaces (GUI)
- Multiprogramming, time sharing for managing resources
- Dual (kernel/user) mode separation, memory management unit

THE XEROX ALTO

COINCIDENTALLY, XEROX IS ALSO INVENTOR OF

Laser printing

Ethernet

The GUI paradigm/mouse

Object-oriented programming (Smalltalk)

Ubiquitous computing

GAP BETWEEN ACCESS OF MEMORY/STORAGE HIERARCHY

Level	Device	Access Time
1	Fastest registers (2 GHz CPU)	**0.5 ns**
2	Main memory	**10-70 ns**
3	SSD	**0.1ms**
4	Secondary storage (disk)	**7 ms**
5	Mass storage (CD-ROM)	**a few s**

AN ANALOGY

Consider a busy restaurant with only one stove

dish 1

dish 2

dish 3

Ingredients

HOW TO MAXIMIZE THE PROFIT

Maximize the utilization of the stove

- Take multiple orders at a time – multiple processes
- Buy the ingredients in bulk from store (save time from multiple trips to the store (remote storage)) – block read, prefetching
- Prepare the ingredients ahead of time – caching
- Interleaving orders that take time to cook and that take time to prepare – pipelining
- Serve VIP's order first – scheduling
- Bring out multiple dishes together – delayed write

Buy more stoves – multiprocessor, multicore

MULTIPROGRAMMING IS AT THE CORE OF MODERN OS DESIGN

Advantages

- Many applications use the peripherals much more than the CPU
- Multiprogramming was invented to keep the CPU busy
- User may initiate multiple tasks and expect short response time for GUI related processes

Multiprogramming allows:

- Time sharing: CPU divides its time among multiple processes, e.g., one tenth of a second on a program, then another tenth of a second on another one and so forth (this does not work with our kitchen analogy!)
- When waiting for I/O operations, other programs can utilize the CPU

Both are accomplished through interrupts

INTERRUPTS

Request to interrupt the flow of execution the CPU

Detected by the CPU hardware

- After it has executed the current instruction
- Before it starts the next instruction

Types of interrupts:

- I/O completion interrupts: Notify the OS that an I/O operation has completed,
- Timer interrupts: Notify the OS that a task has exceeded its quantum of CPU time
- Traps: Notify the OS of a program error (division by zero, illegal op code, illegal operand address, ...) or a hardware failure
- System calls: Notify OS that the running task wants to submit a request to the OS

DISABLING INTERRUPTS

We can disable interrupts

OS does it before performing short critical tasks that cannot be interrupted

- Works only for single-threaded kernels

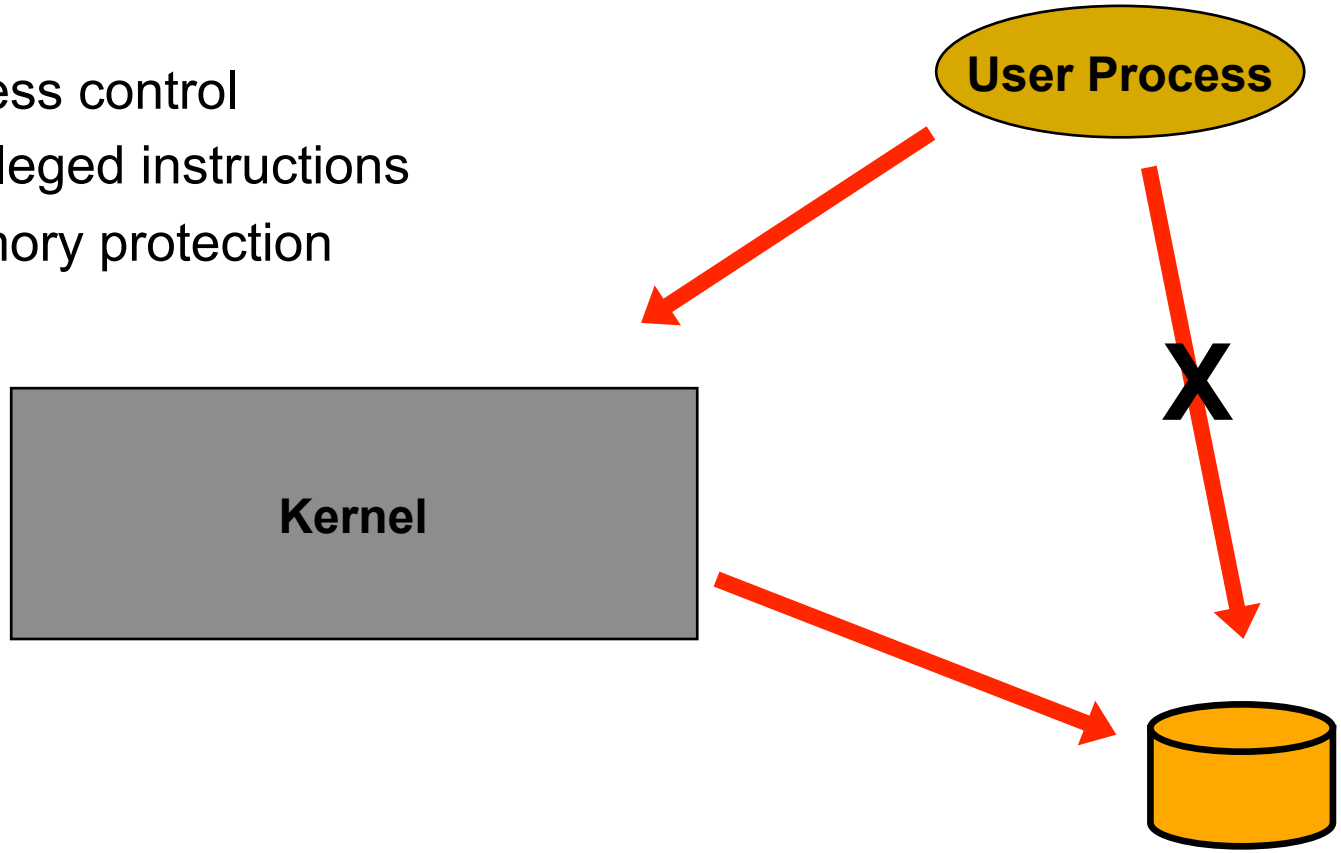
User tasks should be prevented from doing it

- Too dangerous

PROTECTION

In supporting multiprogramming, we need to prevent programs from interfering one another and protect users' data

- Access control
- Privileged instructions
- Memory protection

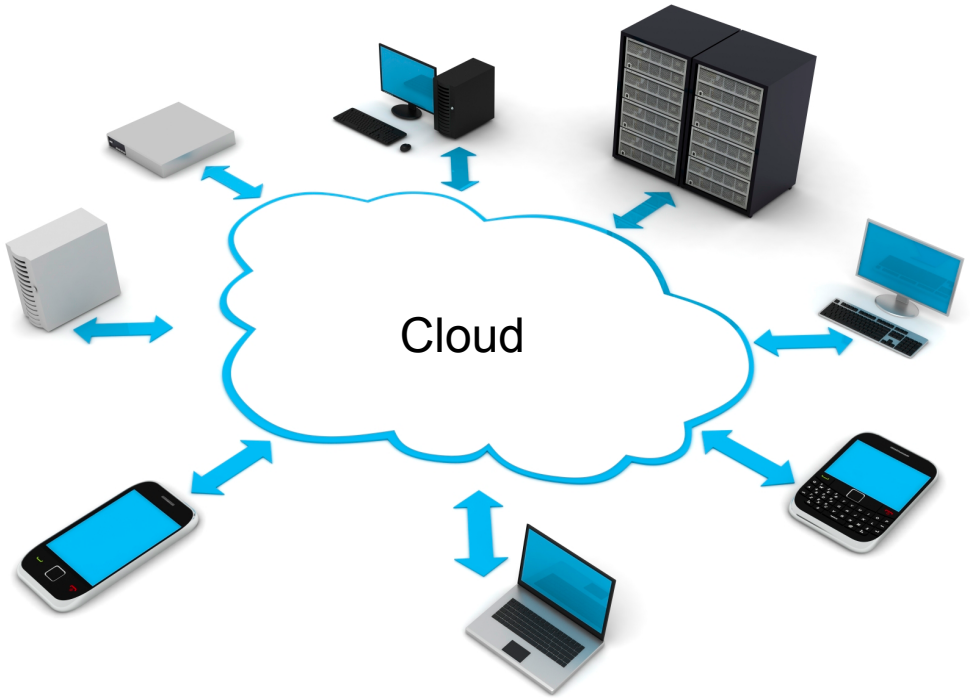


PHASE 3: HARDWARE REALLY CHEAP, HUMAN VERY EXPENSE

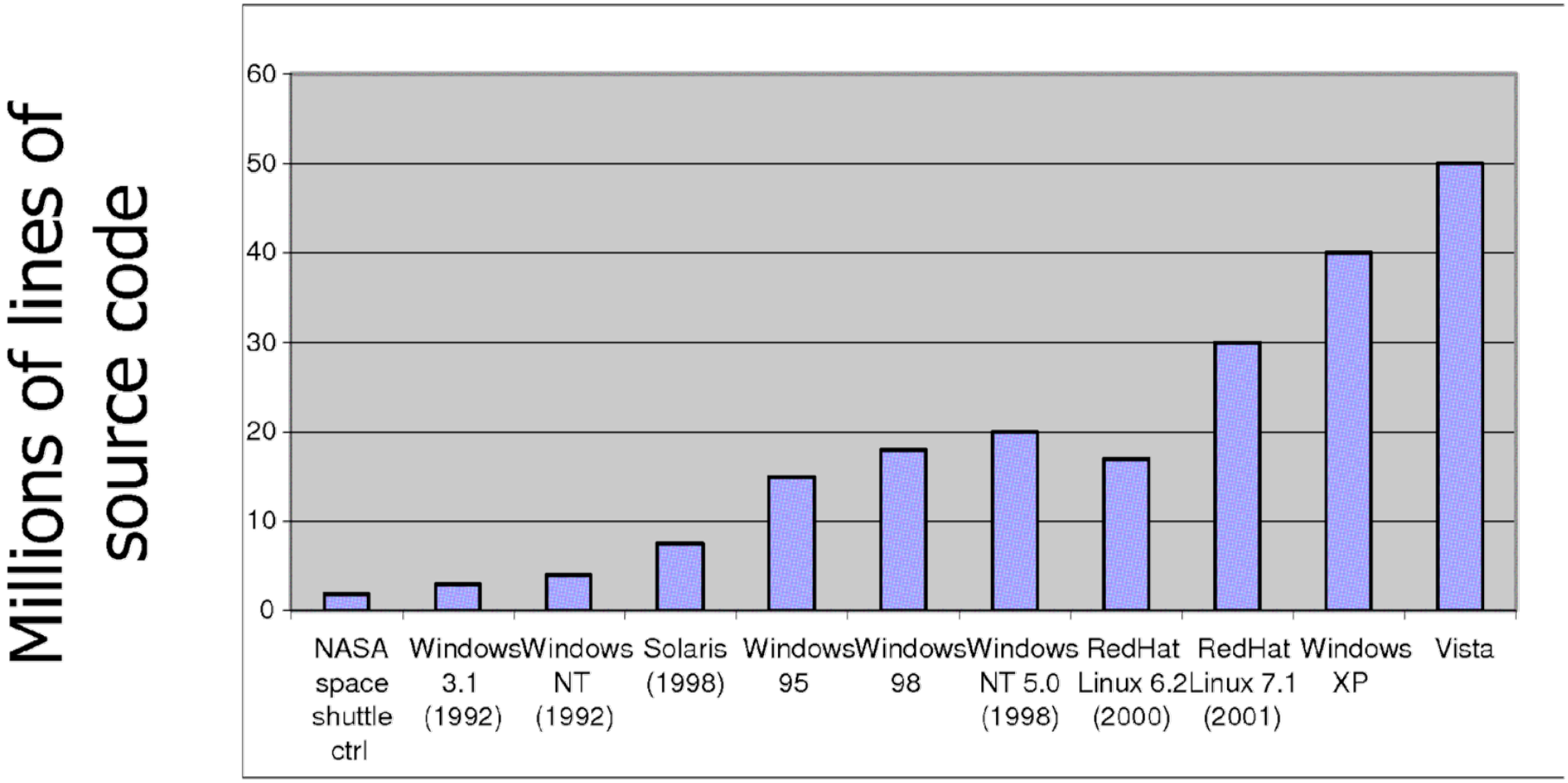
Hardware: tablets, phones, smart sensors etc.

Representative OS: iOS, android, window phone, (VMs)

Focus: energy consumption, manageability, reliability, user experiences, ubiquitous connectivity



ROME WAS NOT BUILT IN ONE DAY



OS ARCHAEOLOGY

Because of the cost of developing an OS from scratch, most modern OSes have a long lineage:

Multics → AT&T Unix → BSD Unix → Ultrix, SunOS, NetBSD,…

Mach (micro-kernel) + BSD → NextStep → XNU →
Apple OSX, iphone iOS

Linux → Android OS

CP/M → QDOS → MS-DOS → Windows 3.1 → NT → 95 → 98 → 2000 → XP → Vista → 7 → 8 → phone → …

Linux → RedHat, Ubuntu, Fedora, Debian, Suse,…

http://www.antipope.org/charlie/blog-static/2013/12/metaphor-for-the-day.html