Operating Systems
Syllabus

Isfahan University of Technology
Electrical and Computer Engineering Department
1400-1 semester

Zeinab Zali

Instructor info

Zeinab Zali

zali@iut.ac.ir

RESEARCH INTERESTS

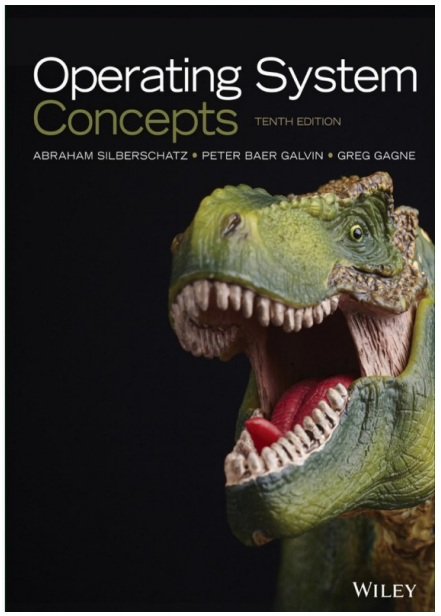
- Computer Networks (NFV, SDN, 5G orchestration)
- Content distribution and Computing on networks
- Operating Systems, Distributed Systems
- Cloud Computing Infrastructures and Big Data processing
- Machine learning applications in computer networks
- NLP, MT

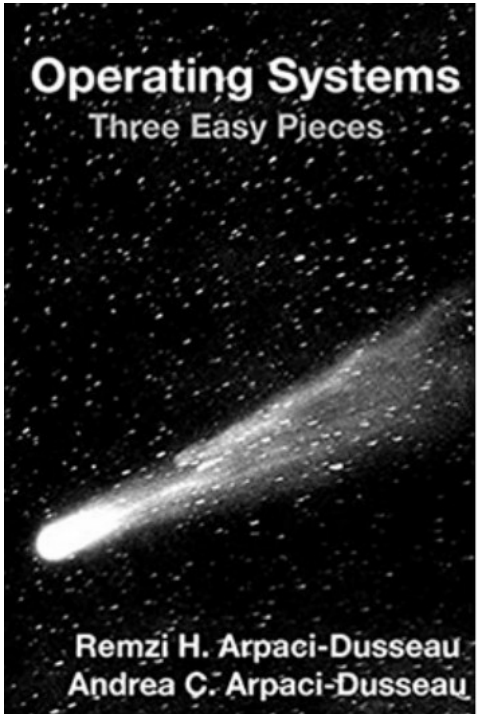
Text books

(Main) Operating Systems Concepts,

A. Silberschatz, P.B. Galvin, G. Gange,

10th edition, 2018





Operating systems: Three easy pieces,

Arpaci-Dusseau, Remzi H and

ArpaciDusseau, Andrea C, 2018

Plan

	Subject	References	HW/ Project
1	OS Introduction	Easy ch2 Dinosaur ch1	
2	OS Structure	Dinosaur ch2	
3	Process	Dinosaur ch3: concepts Dinosaur ch3: API	Project1
4	process	Dinosaur ch3: API, IPC Dinosaur ch3: IPC	HW1
5	thread	Dinosaur ch4	
6	CPU scheduling	Dinosaur ch5: concepts Dinosaur ch5: basic algorithms	
7	CPU scheduling	Dinosaur ch5: basic algorithms Dinosaur ch5: Linux scheduling	HW2
8	synchronization	Dinosaur ch6: problem statement Dinosaur ch6, Easy ch28: basic methods	

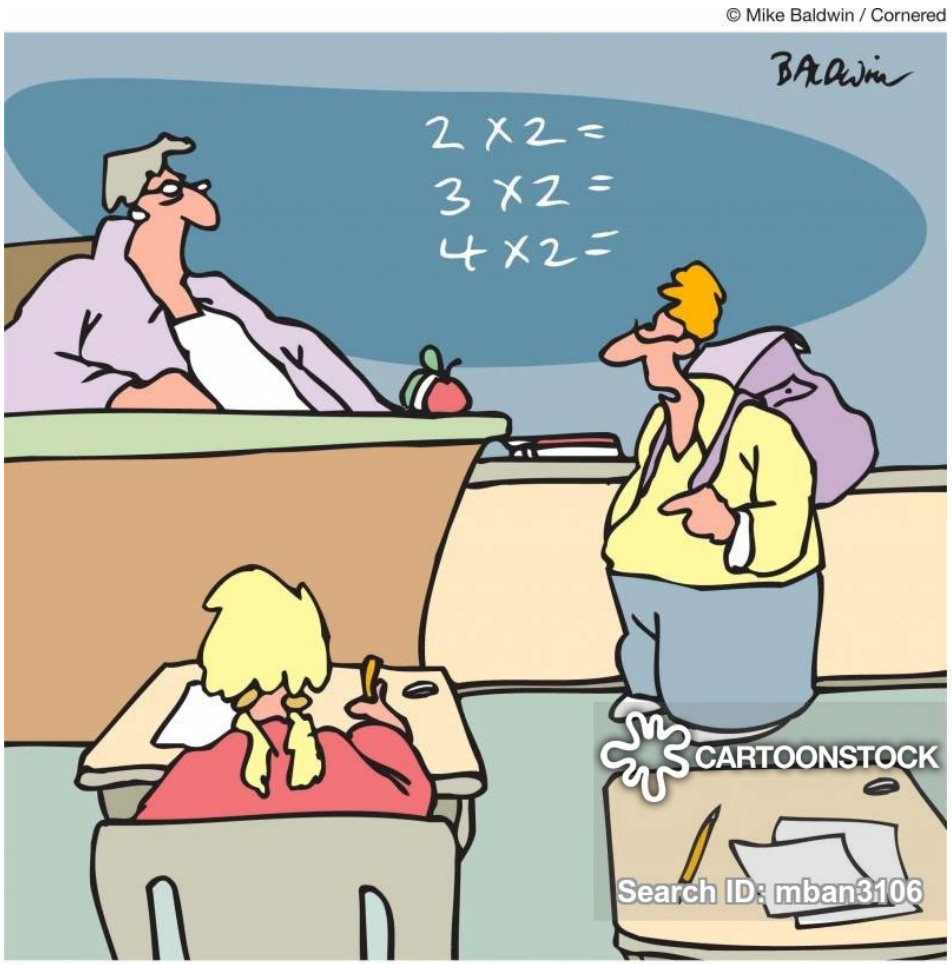
Plan

	Subject	References	HW/ projetcs
1	synchronization	Dinosaur ch6, Easy ch28-29: locks Dinosaur ch6, Easy ch31: semaphore	
2	synchronization	Dinosaur ch6,Easy ch30: condition variables, monitors Dinosaur ch7,Easy ch30: Bounded Buffer	
3	synchronization	Dinosaur ch7: reader/writer Dinosaur ch7, Easy ch31: dining philosopher	HW3
4	Deadlocks	Dinosaur ch8: problem statement and concepts Dinosaur ch8: prevention	
5	Deadlocks Main Memory	Dinosaur ch8: Avoidance and detection Dinosaur ch9: concepts and Address space	project2
6	Main Memory	Dinosaur ch9: paging, segmentation Dinosaur ch9: swapping, Examples	
7	Virtual Memory	Dinosaur ch10: basics, demand paging Dinosaur ch10: Page replacement	HW4
8	Storage Management		

Grading

Class activities	1 (5%)
Homework	3 (15%)
project	3 (15%)
Quiz	3 (15%)
Midterm	5 (25%)
Final	6 (30%)

Deadlines and online sessions



"I'm not late. Everyone learns at their own speed."

How do we enjoy this course

Deadlines

- For each homework/project, **late submission for one week** is accepted but with a **penalty of 5% per day**.
- For all of homeworks/projects, totally late submission of 10 days is accepted without penalty.
- Please don't ask for deadline extension:) pay attention to course plan and figure out your own plan for your all courses
- Please keep in touch with the course concepts and instructor!
  - Listening , Studying, programming, Searching, Sending email to me, asking from the TAs

How do we enjoy this course

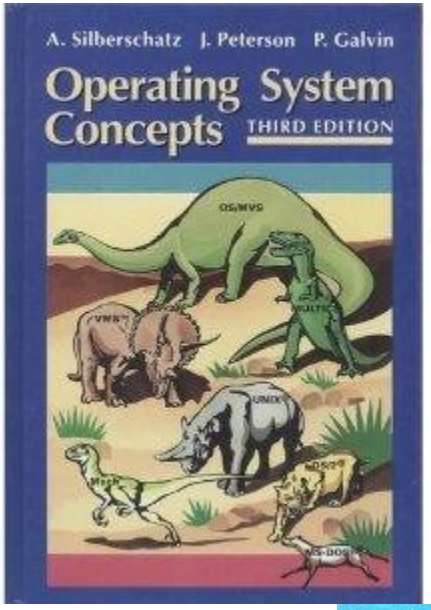
- **Class model**
  - both Online and Offline (Considering students' opinions!)
- **Interactive online classes:**
  - You can talk about the course concepts in new applications and OSs (sometimes proposed by myself)
  - some breaks during session time for your talk (about anything related to university!)

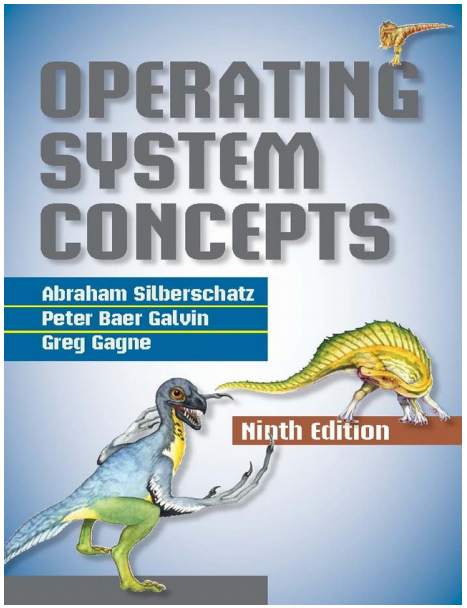
What your friends says about the course!:)

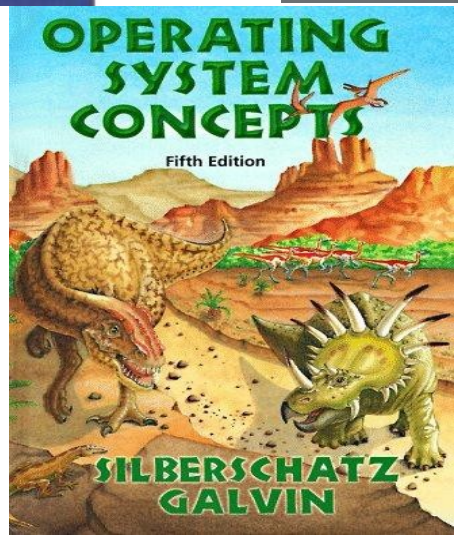
What are we looking for?

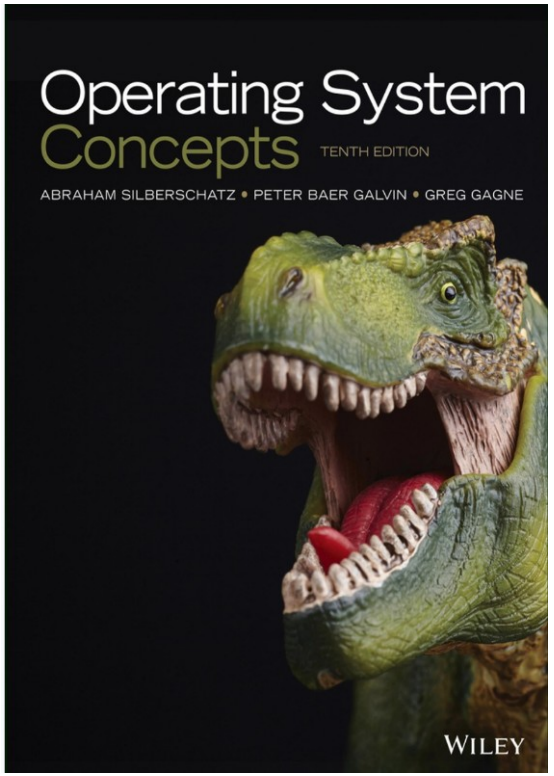
What do you expect from this course?

Why dinosaurs?

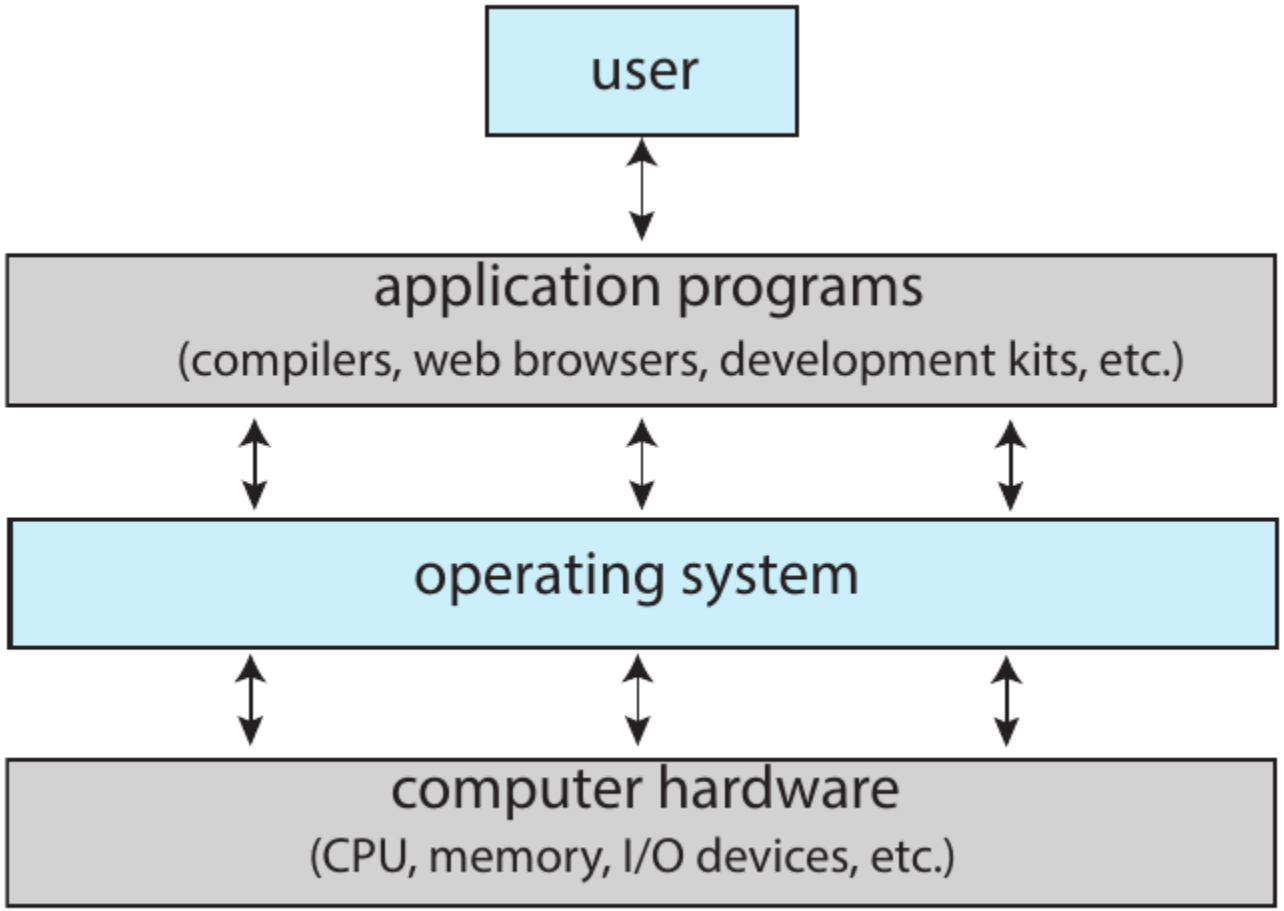


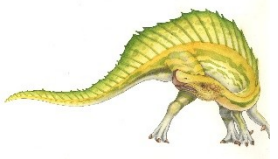






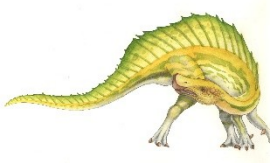
Four Components of a Computer System



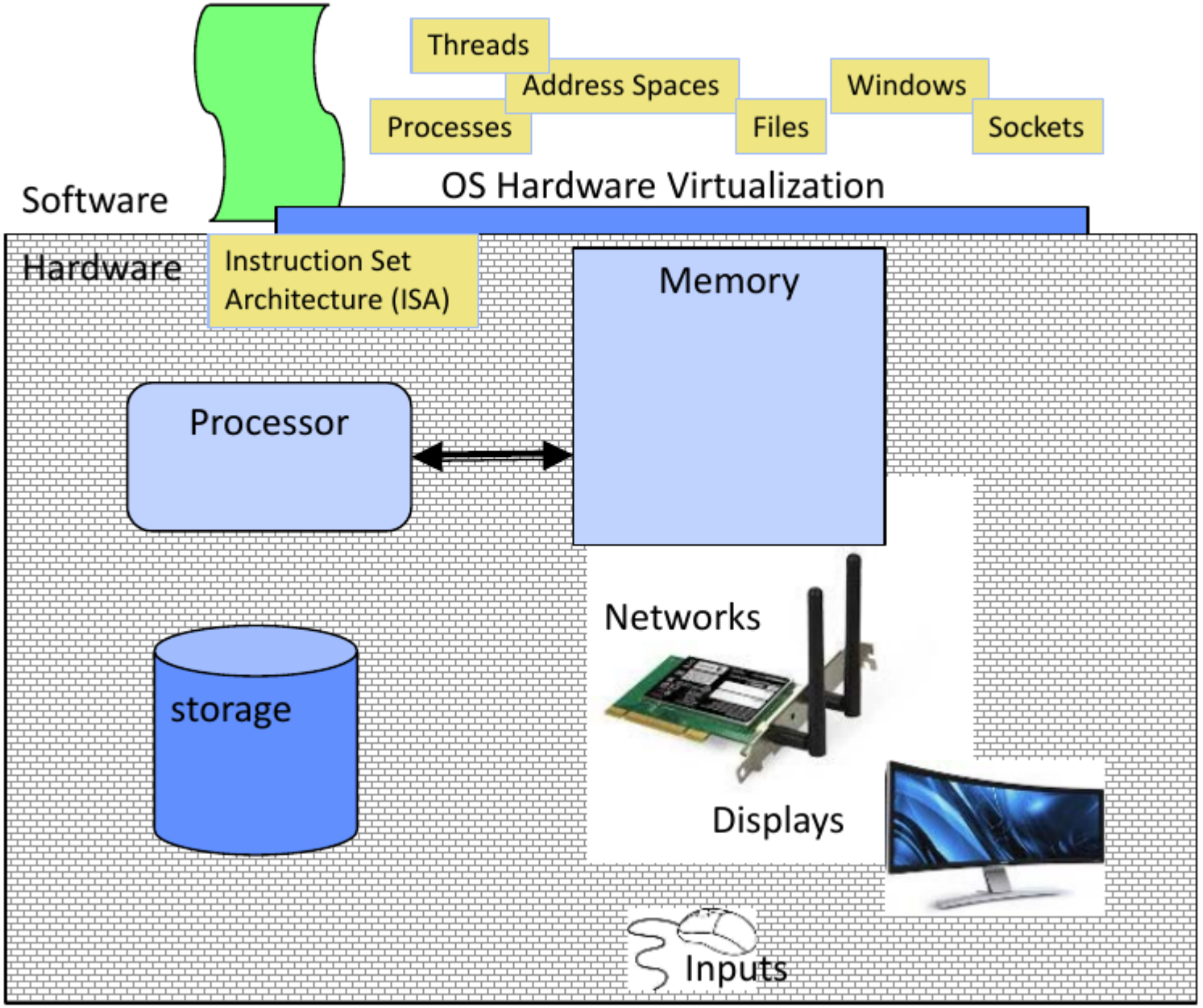


Computer System Structure

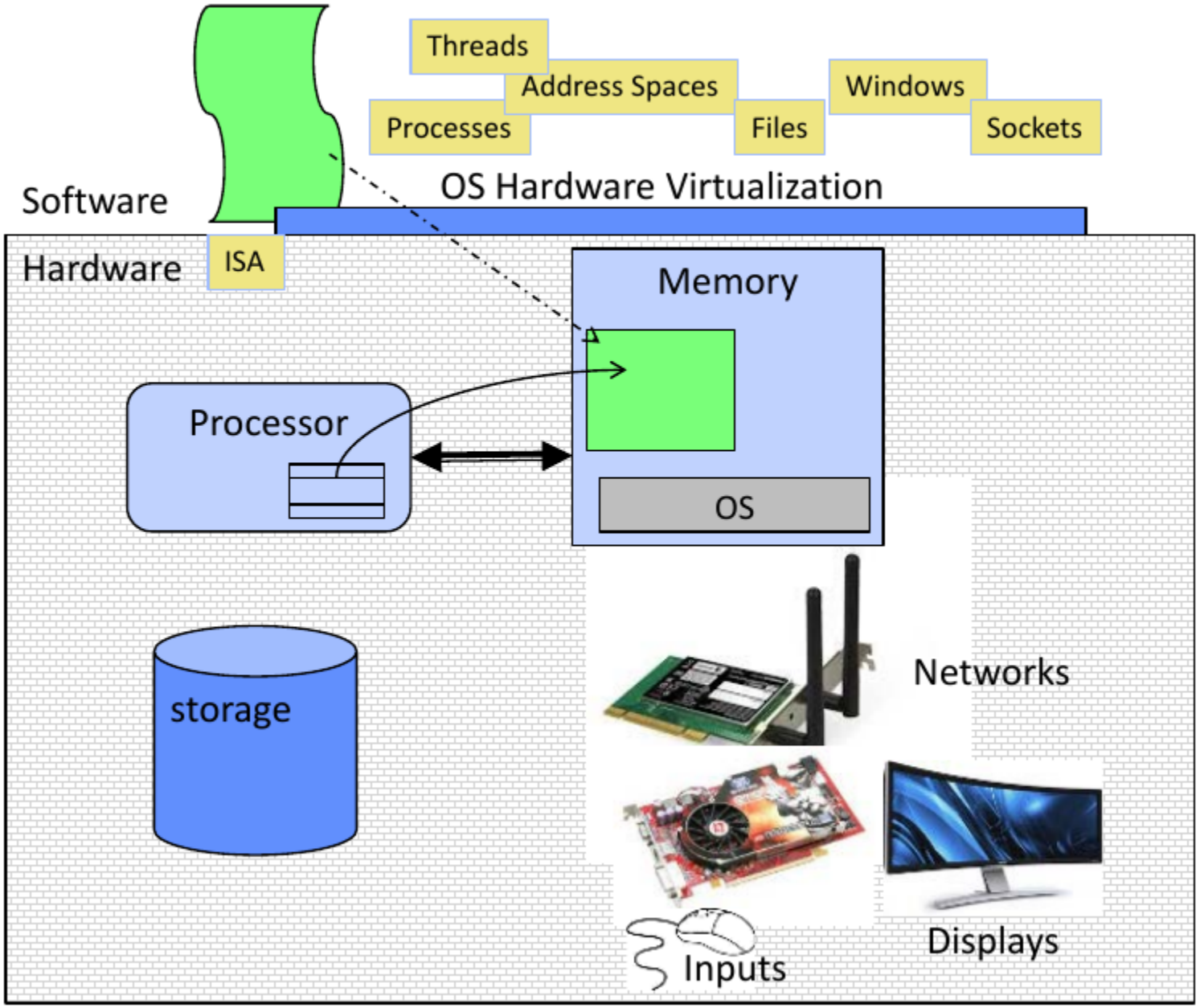
- Computer system can be divided into four components
  - Hardware – provides basic computing resources
    - CPU, memory, I/O devices
  - Operating system
    - Controls and coordinates use of hardware among various applications and users
  - Application programs – define the ways in which the system resources are used to solve the computing problems of the users
    - Word processors, compilers, web browsers, database systems, video games
  - Users
    - People, machines, other computers



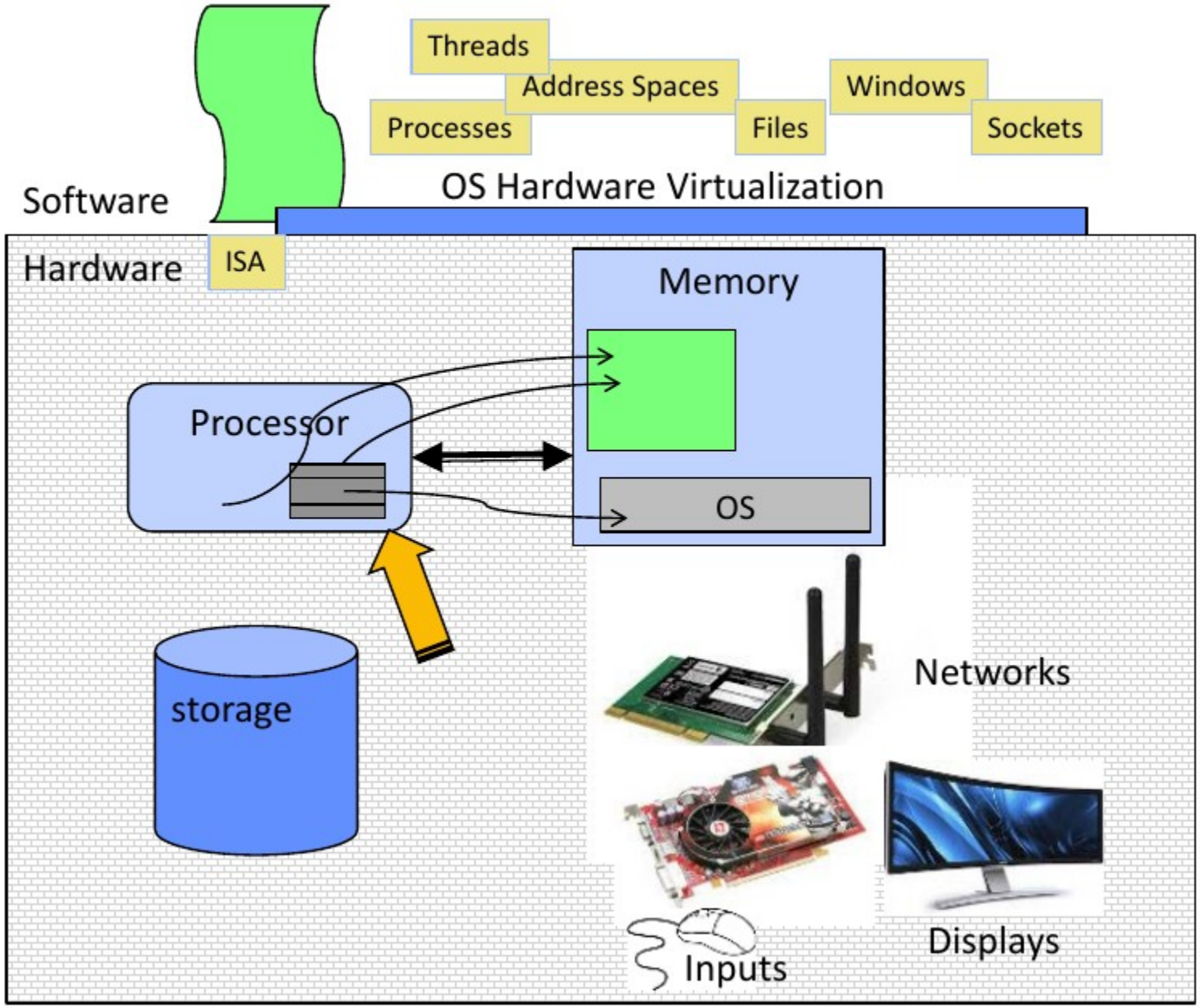
OS Basics: "Virtual Machine" Boundary



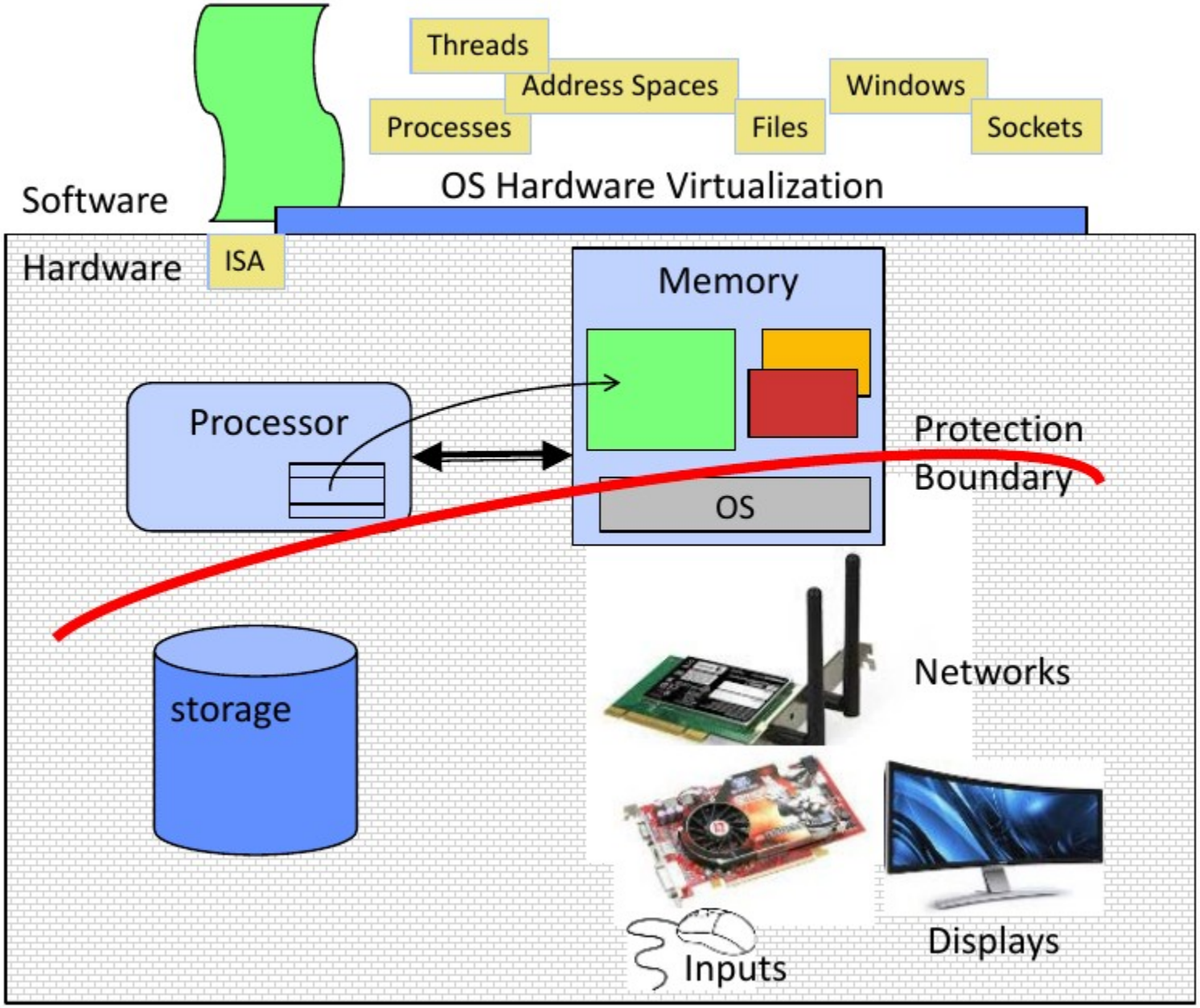
OS Basics: Program => Process



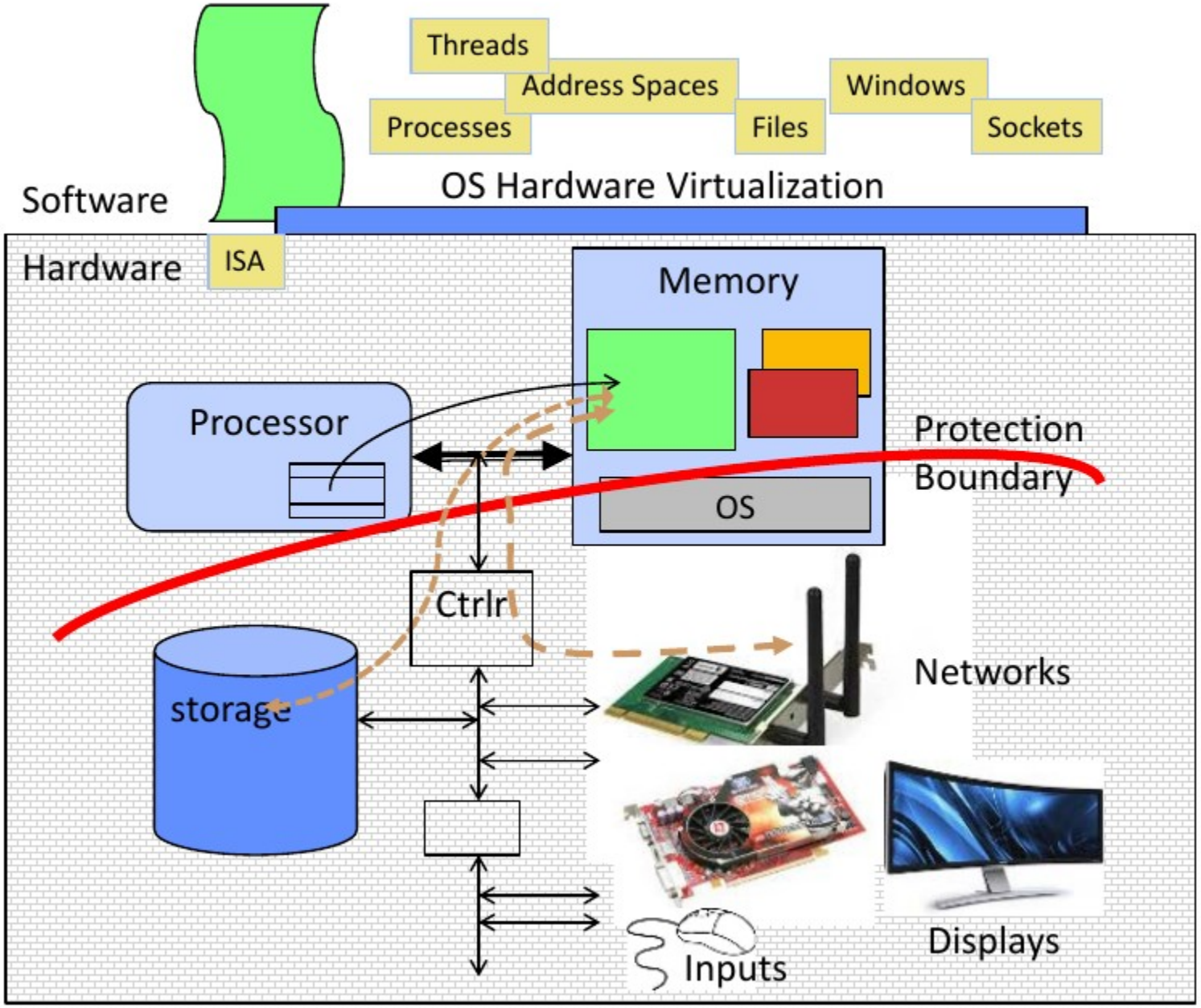
OS Basics: Context Switch



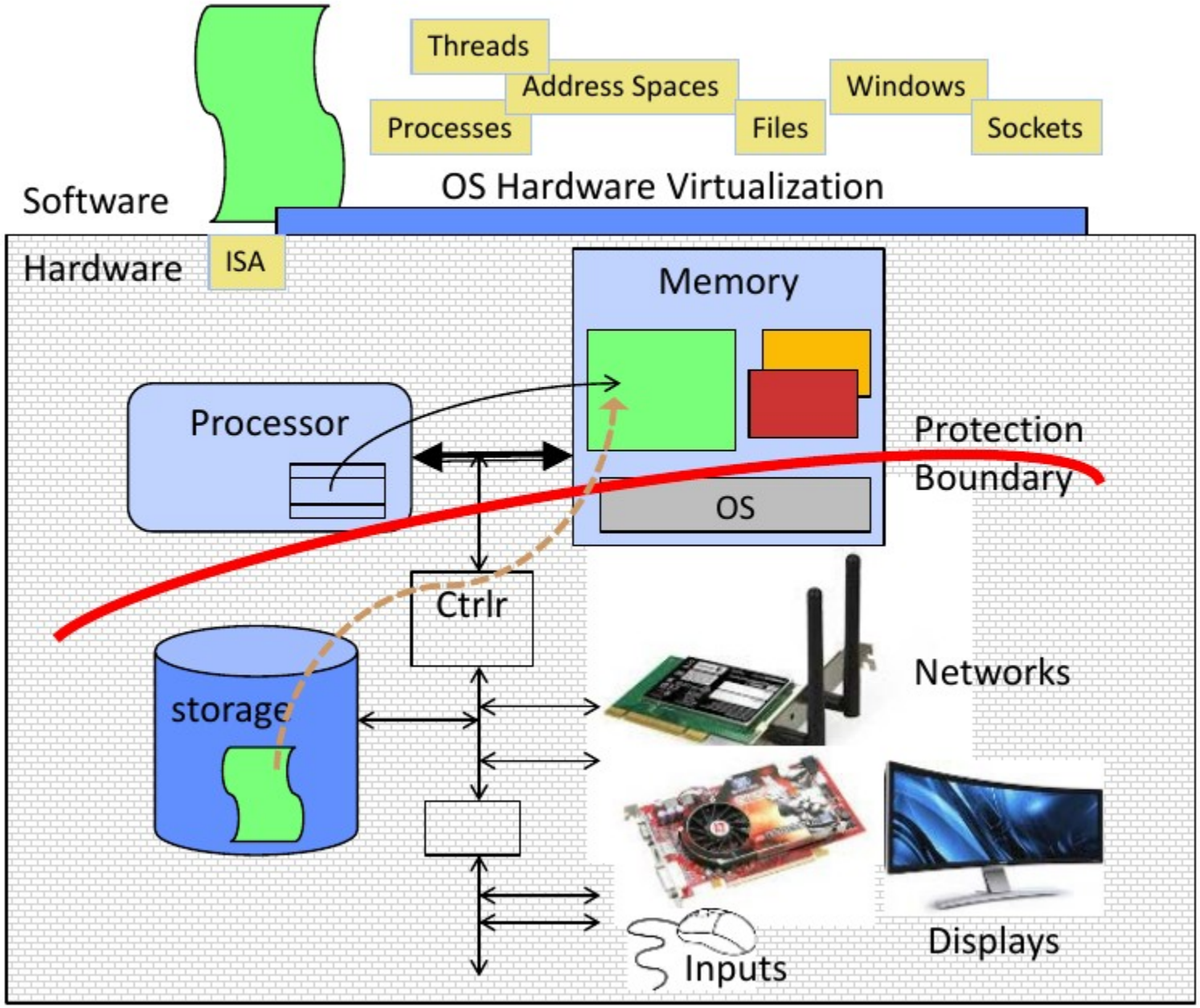
OS Basics: Scheduling, Protection



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
Displays
Inputs

OS Basics: Loading



Whats is an OS?

- There is a body of software, in fact, that is responsible for
  - making it easy to run programs (even allowing you to seemingly run many at the same time)
  - allowing programs to share memory and CPU
  - enabling programs to interact with devices
  - and other fun stuff
- That body of software is called the operating system (OS)
- OS is in charge of making sure the system operates correctly and efficiently in an easy-to-use manner

OS three pieces

- Virtualization
  - A primary way the OS does its roles
    - Specially its key role as a resource manager for managing two main resource
      - CPU
      - Memory

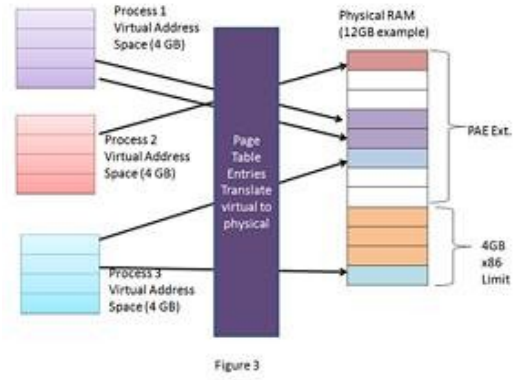
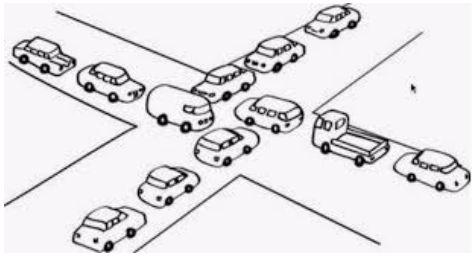
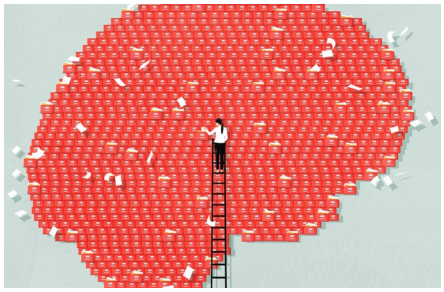


Figure 3

- Concurrency
  - A conceptual term to refer to a host of problems that arise, and must be addressed, when working on many things at once in the same program
    - OS itself
    - Multi-thread or multi-process programs



- Persistence
  - storing any files the user creates in a reliable and efficient manner on the disks of the system



OS design goals

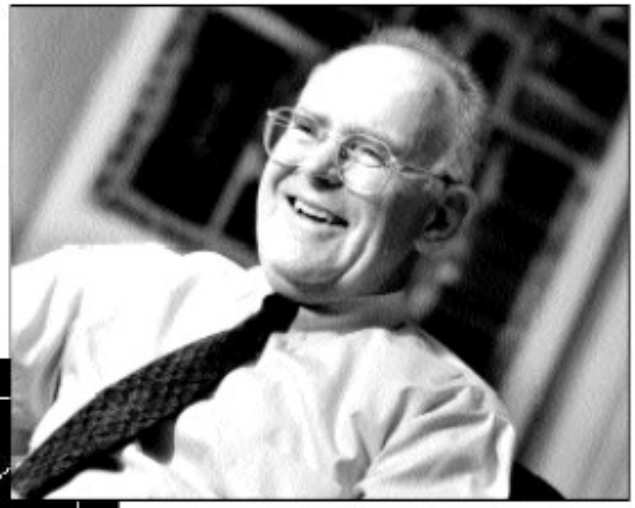
- build up some abstractions in order to make the system convenient and **easy to use.**
  - **Abstractions are fundamental to everything we do in computer science**
- **high performance**; minimizing the overheads of the OS
- **Protection** between applications, as well as between the OS and applications
  - making sure that the malicious or accidental bad behavior of one program does not harm others;
  - **isolating processes**
- **Reliability**; OS must run non-stop; when it fails, all applications running on the system fail as well
- **Security**; an extension of protection, really against malicious applications especially in these highly-networked times
- Others: mobility, energy efficiency, ...

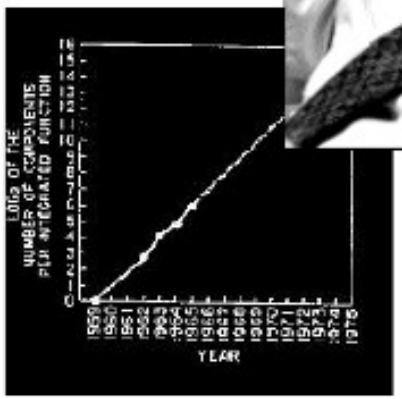
Sum up: What is an Operating System?

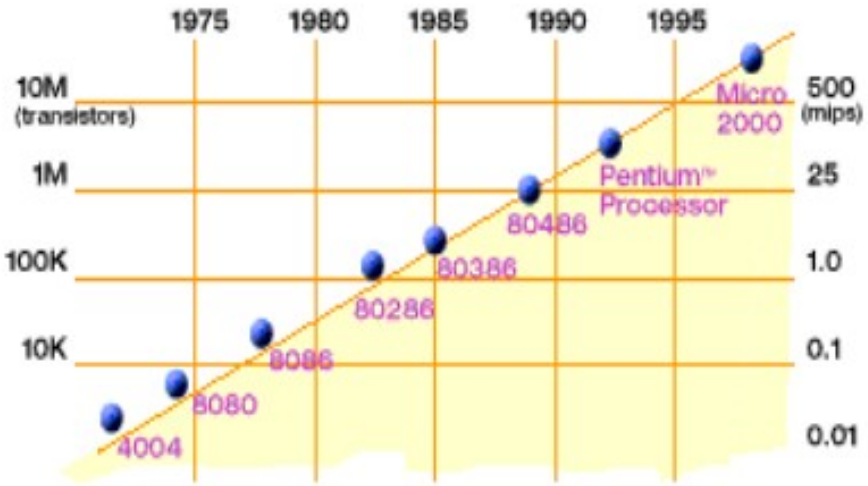
- ✓ A program that acts as an intermediary between a user of a computer and the computer hardware.
- ✓ Operating system operations
  - Process Management
  - Memory Management
  - Storage Management
  - I/O handling
  - Protection and Security
    - user-ID, Group-ID, permission
    - Viruses, attacks, intrusion

What makes OS exciting and challenging

Technology Trends: Moore's Law







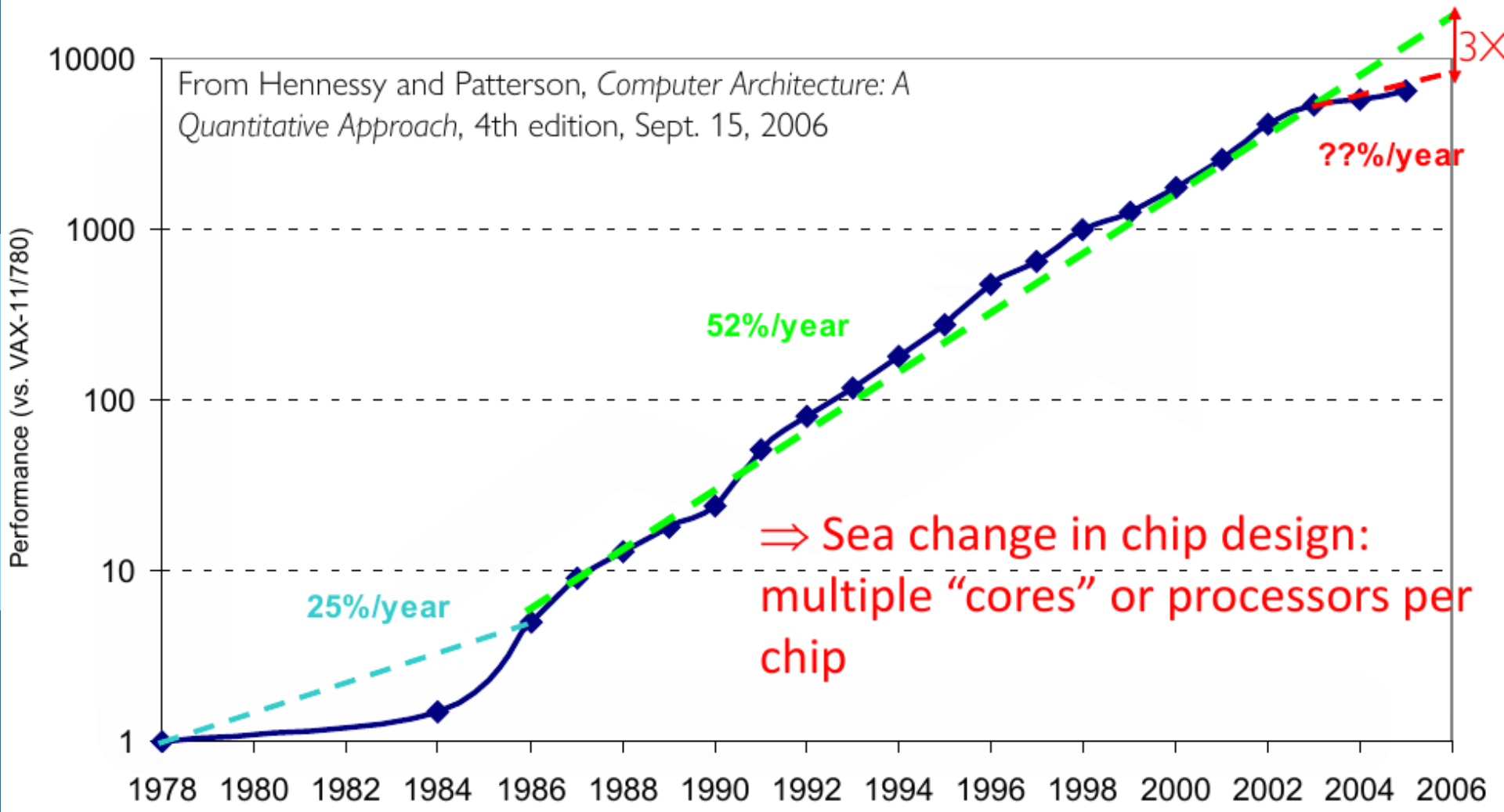
2X transistors/Chip Every 1.5 years

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

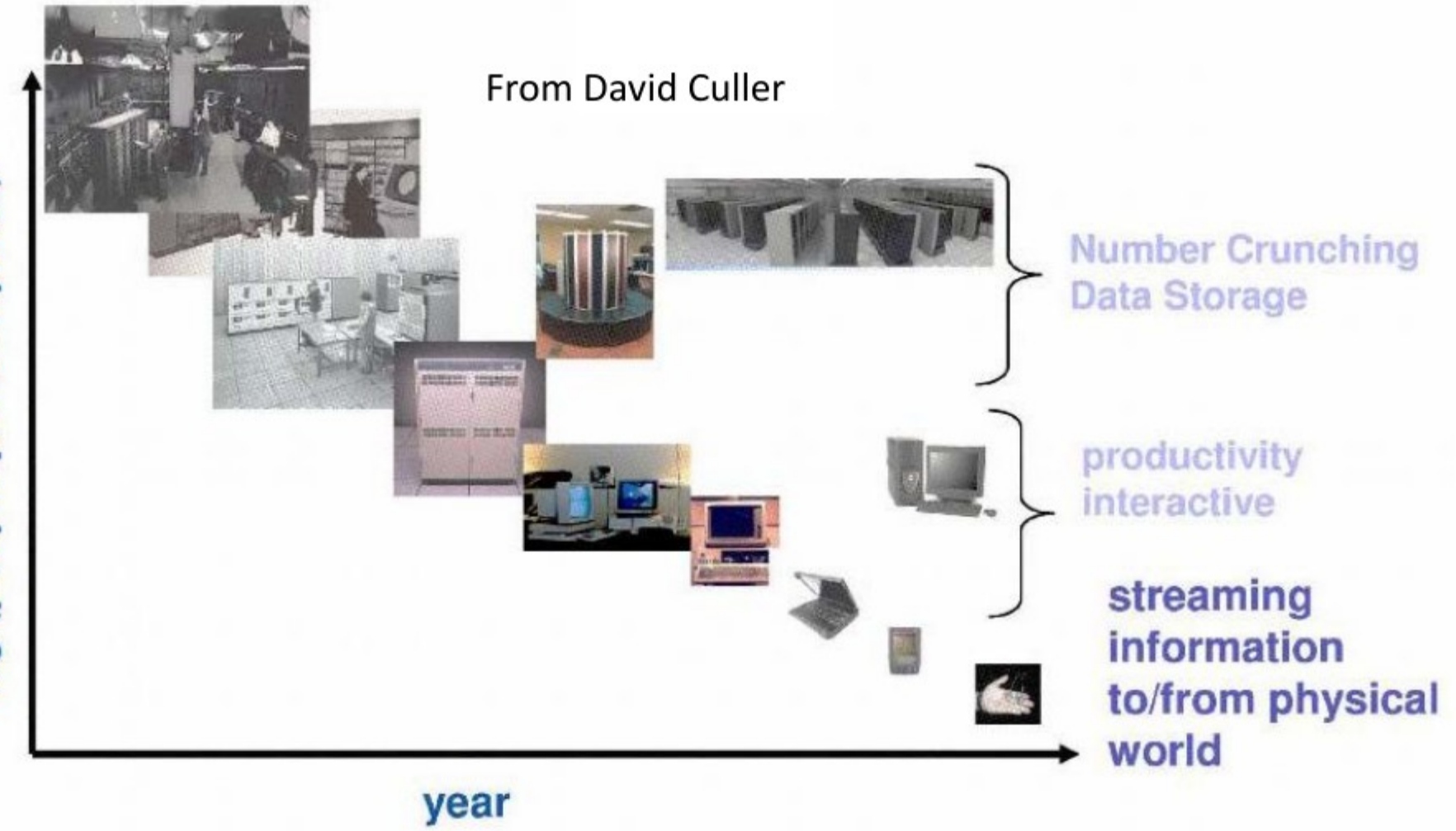
Microprocessors have become smaller, denser, and more powerful

New Challenge: Slowdown in Joy's law of Performance



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

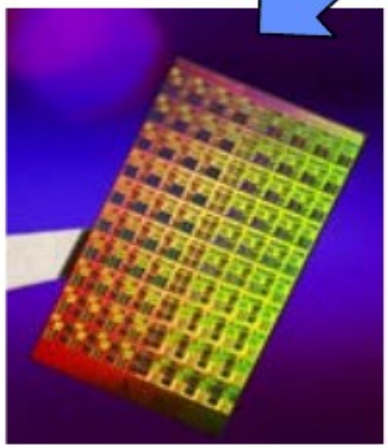
People-to-Computer Ratio Over Time

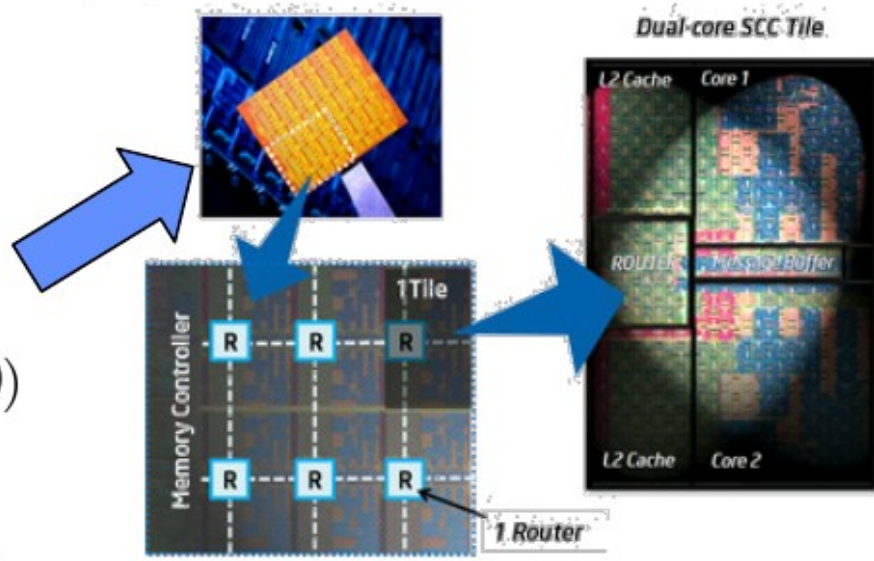


- Today: multiple CPUs/person!
  - Approaching 100s?

ManyCore Chips: The future is here

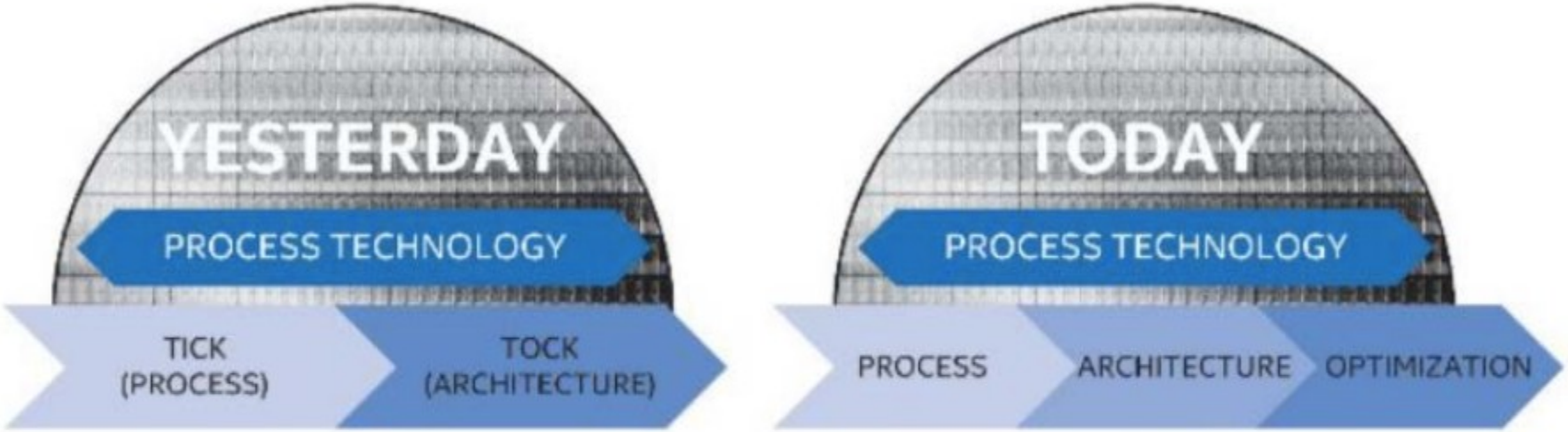
- Intel 80-core multicore chip (Feb 2007)
  - 80 simple cores
  - Two FP-engines / core
  - Mesh-like network
  - 100 million transistors
  - 65nm feature size
- Intel Single-Chip Cloud Computer (August 2010)
  - 24 "tiles" with two cores/tile
  - 24-router mesh network
  - 4 DDR3 memory controllers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



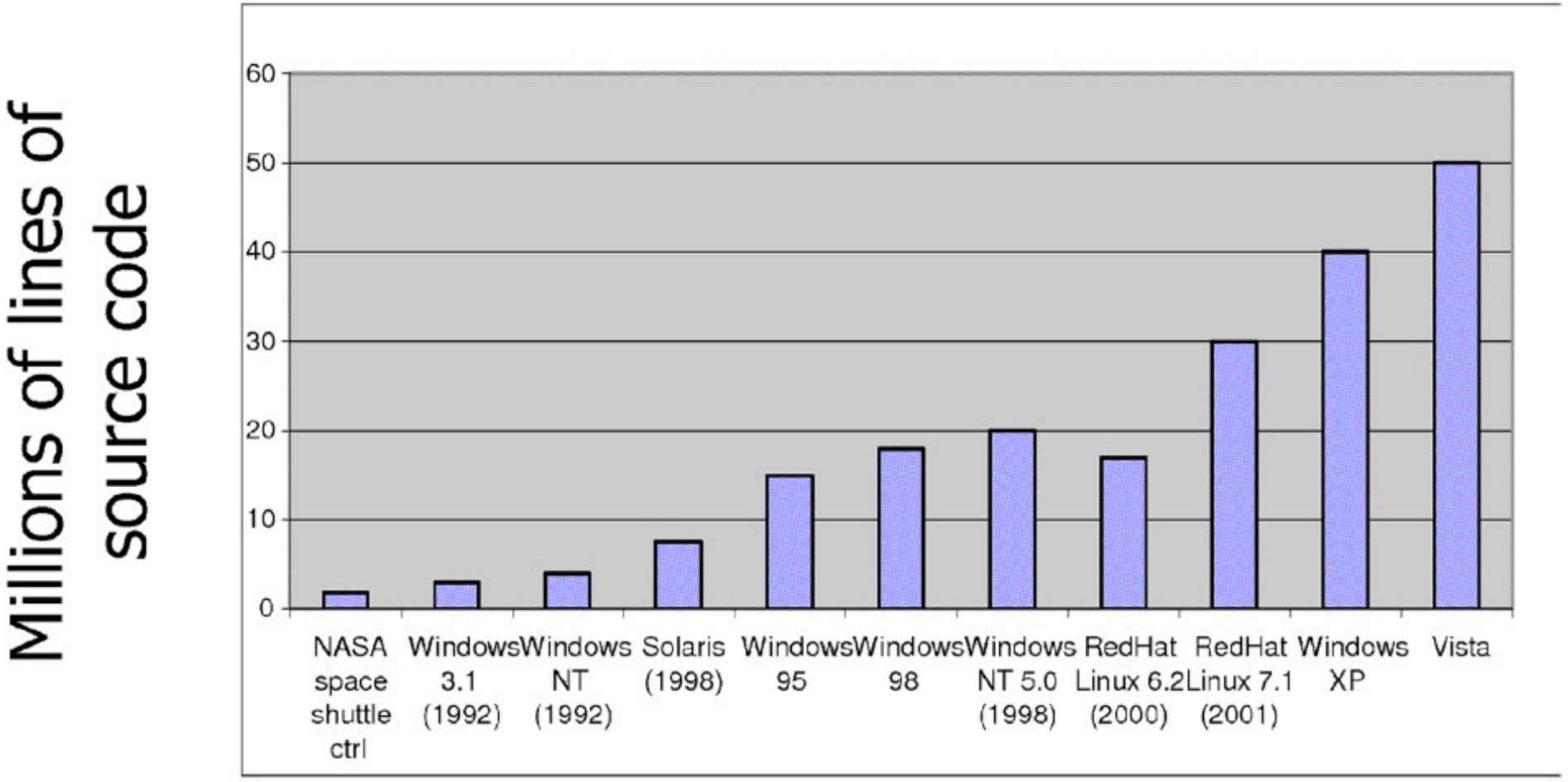
- Moore's Law has (officially) ended -- Feb 2016
  - No longer getting 2 x transistors/chip every 18 months…
  - or even every 24 months

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

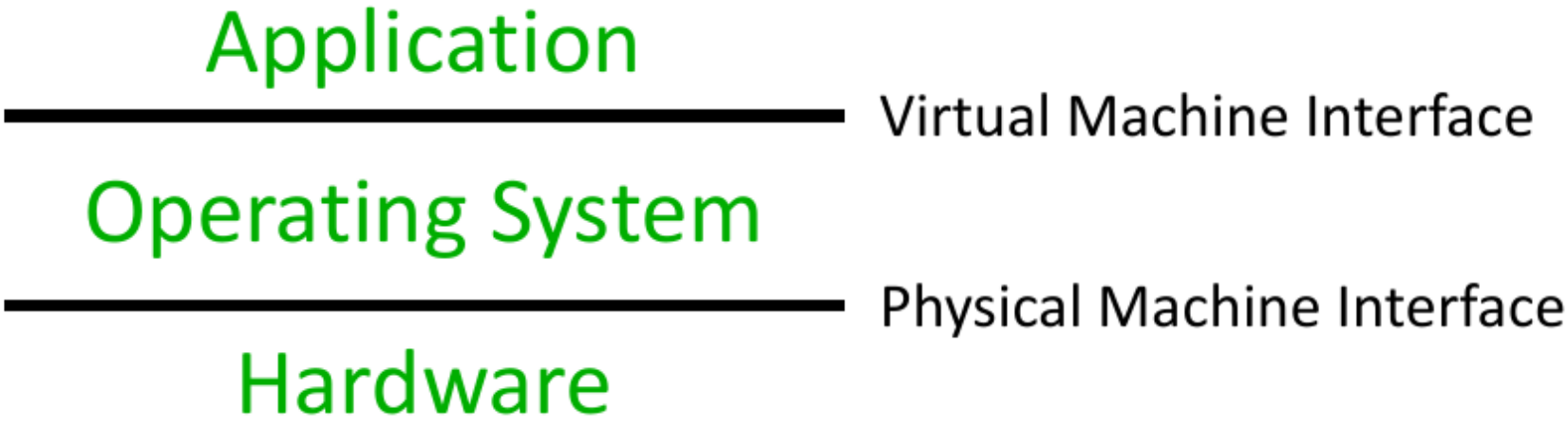


From MIT's 6.033 course

How do we tame complexity?

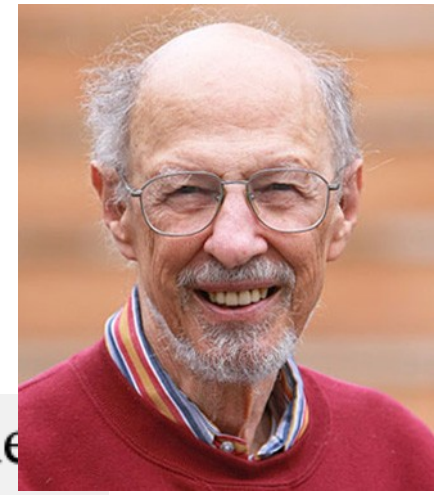
- Every piece of computer hardware different
  - Different CPU
    - » Pentium, PowerPC, ColdFire, ARM, MIPS
  - Different amounts of memory, disk, …
  - Different types of devices
    - » Mice, Keyboards, Sensors, Cameras, Fingerprint readers
  - Different networking environment
    - » Cable, DSL, 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)

What do you think about future? Like people 50 years before



- In 1965, MIT's Fernando Corbató and the other designers of the Multics operating system envisioned a computer facility operating "like a power company or water company."

- Plug your thin client into the computing utility and play your favorite Intensive Compute & Communicate Application
  - Have today's clouds brought us closer to this reality? Think about it.