CS630
Operating Systems Design
Spring 2020

Jing Li

jingli@njit.edu

GITC 4106

Department of Computer Science

New Jersey Institute of Technology

Who am I?

- Jing Li
  - jingli@njit.edu
  - GITC 4106, Department of Computer Science

- Research areas:
  - Parallel computing
  - Real-time systems
  - Both theoretical analysis and system design

- Previously:
  - Ph.D. from Washington University in St. Louis, 2017

- Style of class: interactive – ask questions

NJIT

In This Lecture

- How does this class operate?
- What is an Operating System?
- Evolution of Operating Systems

Slides courtesy of Hung Daochuan, Chris Gill, David Ferry, Tarek Abdelzaher, Ion Stoica, John Kubiatowicz, David Culler, Anthony Joseph, Jonathan Ragan-Kelley.

Course Info

- Operating Systems Design:
  Advanced topics in OS for **graduate students**

- Prerequisites:
  Introductory courses of operating systems are prerequisite.
  Proficiency in C (and/or C++) is a firm prerequisite.
  Basic proficiency in using Linux is very beneficial.
  - CS252 Computer Organization
  - CS288 Intensive programming in Linux
  - CS332 Principles of Operating Systems
  - CS433 Intro to Linux Kernel Programming

- Have any of you taken?
  - CS680 Linux Kernel Programming

Course Info

- Course website in https://njit.instructure.com/
  - Announcements, lecture notes, assignments, etc.
  - Log in before next lecture:
    make sure it works & verification of presence

- Class time and location:
  - Wednesdays, 11:30 AM - 02:20 PM, in CKB 222

- Grader: Shaoze Fan, sf392@njit.edu
  - Office hour: GITC 4324, Wednesday 10:00AM to 11:30AM.

- Office hours:
  - GITC 4106, Fridays, 2:00 pm - 6:00 pm
  - Please make appointments by email for other times.

Textbook & Readings

- Textbook:
  - Operating Systems: Internals and Design Principles, William Stallings
- Optional textbook:
  - Operating System Concepts, Silberschatz, Galvin, and Gagne
- Readings & references:
  - Research Papers (will be posted on course website)
- Recommend books (with free online version):
  - Operating Systems: Three Easy Pieces by Arpaci-Dusseau and Arpaci-Dusseau is a good introduction to operating systems, without all the complexities of Linux.
  - Understanding the Linux Kernel by Bovet and Cesati, 2006, offers a more exhaustive reference than the course textbook, covering many data structures and code snippets in detail.

Class Structure

Typical Lecture Format

45 min. VBreak 45 min. Break 50 min. End

Time →

- 5-Minute Review
- 40-Minute Lecture
- 10-15 Minute Virtual Break: Administrative Matters / Comics
- 45-Minute Lecture
- 15 Minute Break (water, stretch)
- 50-Minute Lecture

Why Study Operating Systems?

- Learn how to build complex systems:
  - How can you manage complexity for future projects?
- Engineering issues:
  - Why is the web so slow sometimes? Can you fix it?
  - What features should be in the next mars Rover?
  - How do large distributed systems work?
- Buying and using a personal computer:
  - Why different PCs with same CPU behave differently
  - How to choose a processor?
  - Should you get Windows, Linux, Mac OS …?
  - Why does Microsoft have such a bad name?
- Business issues:
  - Should your division buy thin-clients vs PC?
- Security, viruses, and worms
  - What exposure do you have to worry about?

NJIT

Course Format and Grading

Point distribution subject to change

Activity	Points
➢ Class Participation	5
➢ Lab Assignments	30
➢ Midterm Exam	30
➢ Final Exam	35

(Extra points for project: 2)

Point distribution design motivation

Final Grade

Curved:

- A – top 1% to 24%
- B+ – top 25% to 49%
- B – top 50% to 75%
- C+/C/F/W the rest of the class

Class Participation – Option 1

- Answer question in class
  - One point per lecture

Class Participation – Option 2 Presentation

- Two students in a team to present one paper
  - Form your team
  - 20 to 30min presentation

- Paper to present will be provided

- Decision for choosing option 2: about first exam

- Prepare slides for the paper
  - Submit one week before the presentation slot
  - Will be given high-level feedback

NJIT

Labs – Learn by Doing

- There will be 6 to 8 lab assignments for this course.
- The purpose of these labs is to apply course concepts and to evaluate kernel mechanisms and behaviors.
- Each lab will be completed in a team of up to 3 students, and teams may be different for each lab.
- Labs submitted on time (as determined by Moodle's receipt time stamp) do not have penalty. Labs submitted up to 72 hours late will be given a 20% penalty. Labs submitted less than a week late will be given a 50% penalty. Labs submitted after a week late will not be given credit.

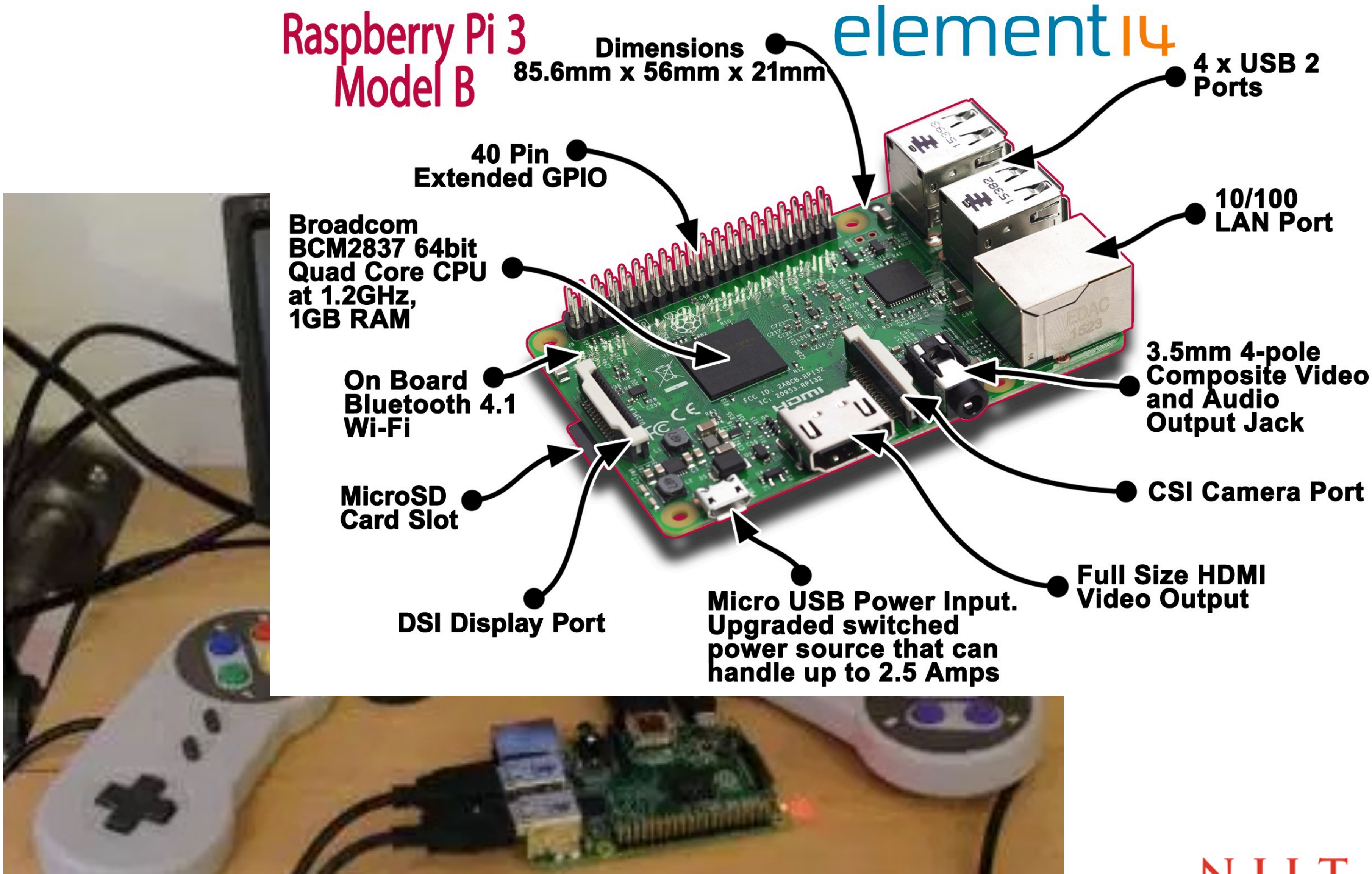
Teamwork – Simulates Industrial Environment

- Lab teams have 2 to 3 members
  - Must work in groups in "the real world"

- Communicate with colleagues (team members)
  - Communication problems are natural
  - What is each member's responsibility?
  - What have you done?
  - What answers you need from others?

- Communicate with supervisor (instructor & class)

Labs & Project: Raspberry Pi



https://projects.raspberrypi.org/en/projects/raspberry-pi-getting-started

Buy Raspberry Pi

- On Amazon

V-Kits Raspberry Pi 3 Model B+ (Plus) Basic Starter Kit [Latest Model 2018]
by Vilros
54^{99} prime
Get it **Wed, Sep 12 - Fri, Sep 14**
FREE Shipping

More Buying Choices
$49.99 (2 new offers)

★★★★☆ 137
75% off item with purchase of 1 items
See Details

V-Kits Raspberry Pi 3 Model B+ (Plus) Complete Starter Kit with Official Black Case [LATEST MODEL 2018]
by Vilros
79^{99} prime
Get it **Wed, Sep 12 - Fri, Sep 14**
FREE Shipping

★★★★☆ 43
75% off item with purchase of 1 items
See Details

Sponsored ⓘ
CanaKit Raspberry Pi 2.5A Power Supply
by CanaKit
54^{99} prime
Get it **Wed, Sep 12 - Fri, Sep 1**
FREE Shipping

Raspberry Pi 3 B+ (B Plus) Ultimate Kit – Complete Set Includes Raspberry pi Motherboard, 7" Touchscreen Display, Power Supply,...
by NeeGo
179^{99} prime
Get it **Wed, Sep 12 - Fri, Sep 14**
FREE Shipping

★★★½☆ 36

Amazon's Choice

ELEMENT Element14 Raspberry Pi 3 B+ Motherboard
by ELEMENT
39^{78} prime
Get it **Wed, Sep 12 - Fri, Sep 14**
FREE Shipping

More Buying Choices
$37.80 (23 new offers)

★★★★☆ 107

- Computer Memory Size: **1 GB**
- Form Factor: **mboard**
- System Ram Type: **ddr2 sdram**
- Cpu Model Socket: **Broadcom BCM2837BO 64 bit ARMv8 QUAD Core A53 64bt**
- Processor Count: **4**

Raspberry Pi

- Pi 3 (b+) Power Usage: 400~500mA;
- Power Pi in class:
  - (0) charger;
  - (2) 5V (portable) battery, 1A-2.5A;
  - (3) USB3 port on Mac provides 900mA;

	Zero /mA	Zero W /mA	A+ /mA	A /mA	B+ /mA	B /mA	Pi2B /mA	Pi3B /mA	Pi3B+ /mA
Idling	100	120	100	140	200	360	230	230	400
Loading LXDE	140	160	130	190	230	400	310	310	690
Watch 1080p Video	140	170	140	200	240	420	290	290	510
Shoot 1080p Video	240	230	230	320	330	480	350	350	520

https://projects.raspberrypi.org/en/projects/raspberry-pi-getting-started

Tentative Syllabus

- Overview;
- OS concepts;
- Scheduling (uniprocessor);
- Scheduling (real-time);
- Processes;
- Threads and parallel computing;
- Concurrency;
- Midterm Exam;
- Synchronization;
- Deadlock;
- Memory management;
- Disk management;
- File systems;
- Page Replacement; Final exam review;
- Distributed Systems and Virtualization (If time permits);

Academic Dishonesty Policy

- Copying all or part of another person's work, or using reference material not specifically allowed, are forms of cheating and will not be tolerated. A student involved in an incident of cheating will be notified by the instructor and the following policy will apply:
  - https://www.njit.edu/policies/sites/policies/files/academic-integrity-code.pdf

- The instructor will take actions, including:
  - require repetition of the subject work,
  - assign a 'zero' grade to the subject work,
  - for repeated or serious offenses, assign an F grade for the course.

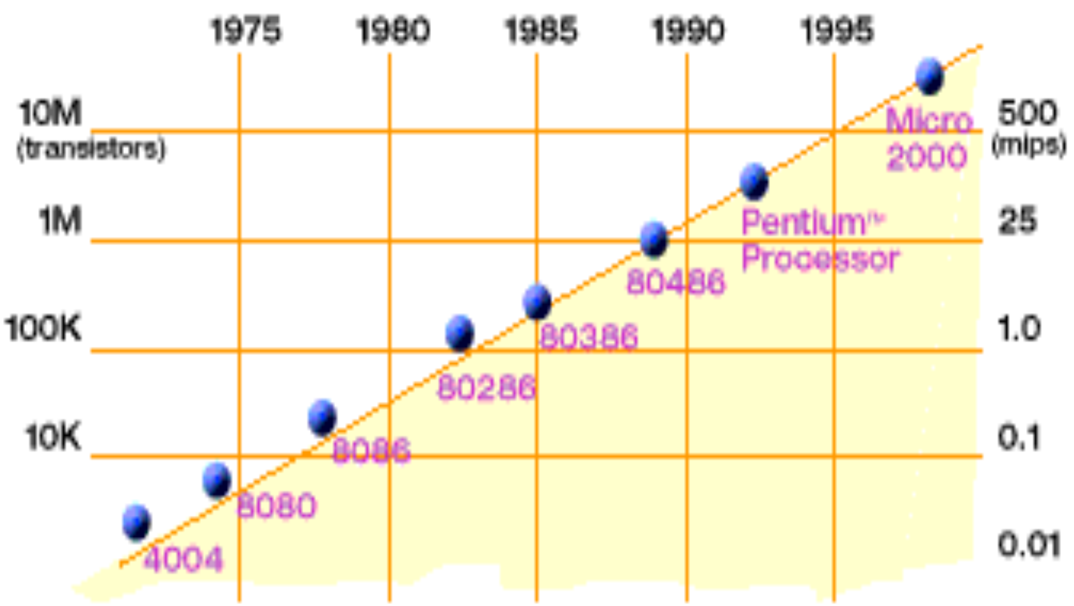
NJIT

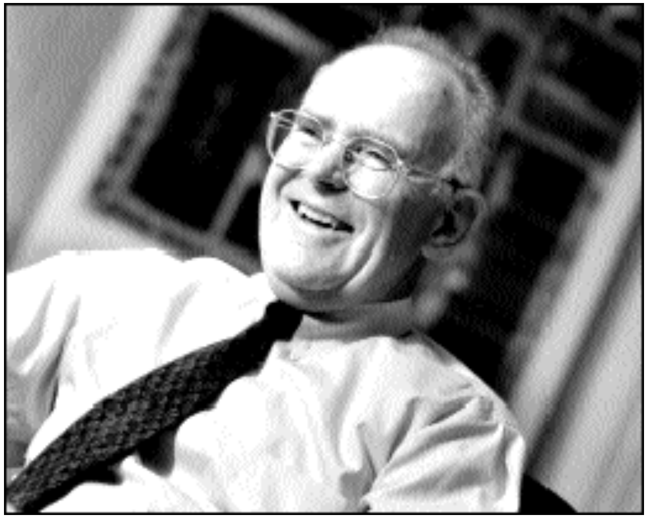
What is an Operating System?

New Jersey Institute of Technology

Technology Trends: Moore's Law

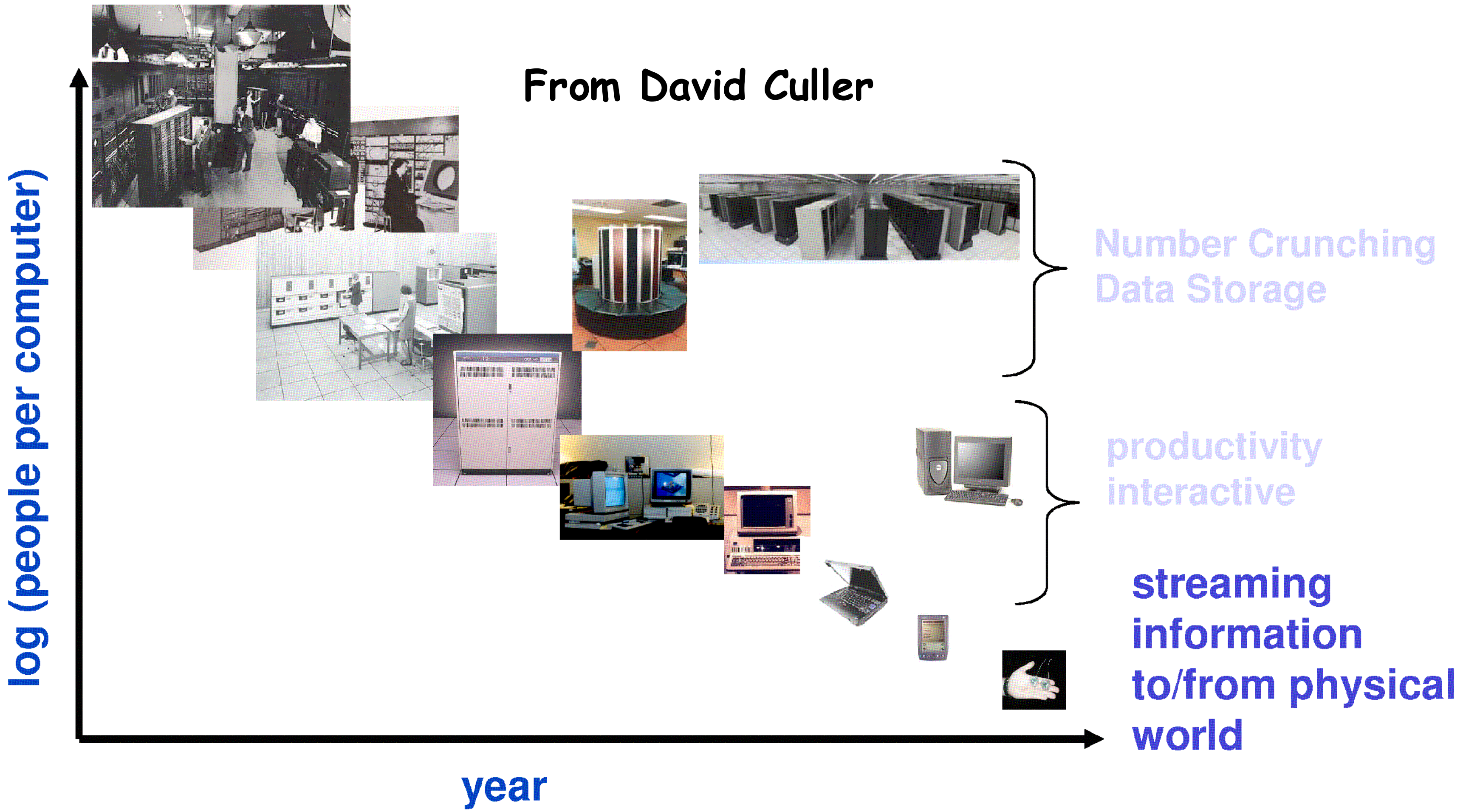
- Gordon Moore (co-founder of Intel)
- Predicted in 1965 that the transistor density of semiconductor chips would double roughly every 18 months.
- 2X transistors/Chip Every 1.5 years – called "Moore's Law"
- What does Moore's low give people?



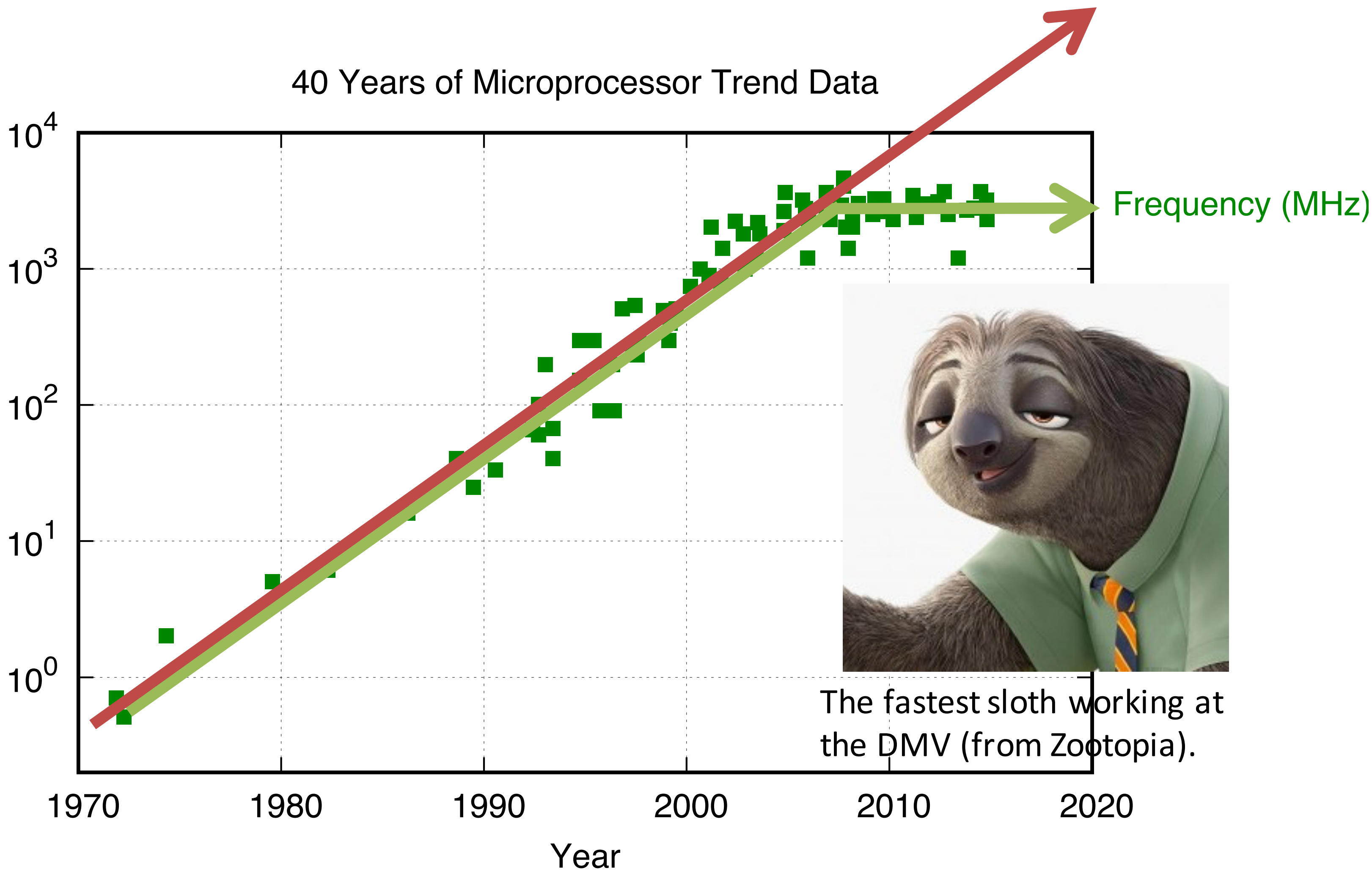


People-to-Computer Ratio Over Time

- Processors have become smaller, denser, and more powerful.



Trend of Computing Performance (Joy's Law)

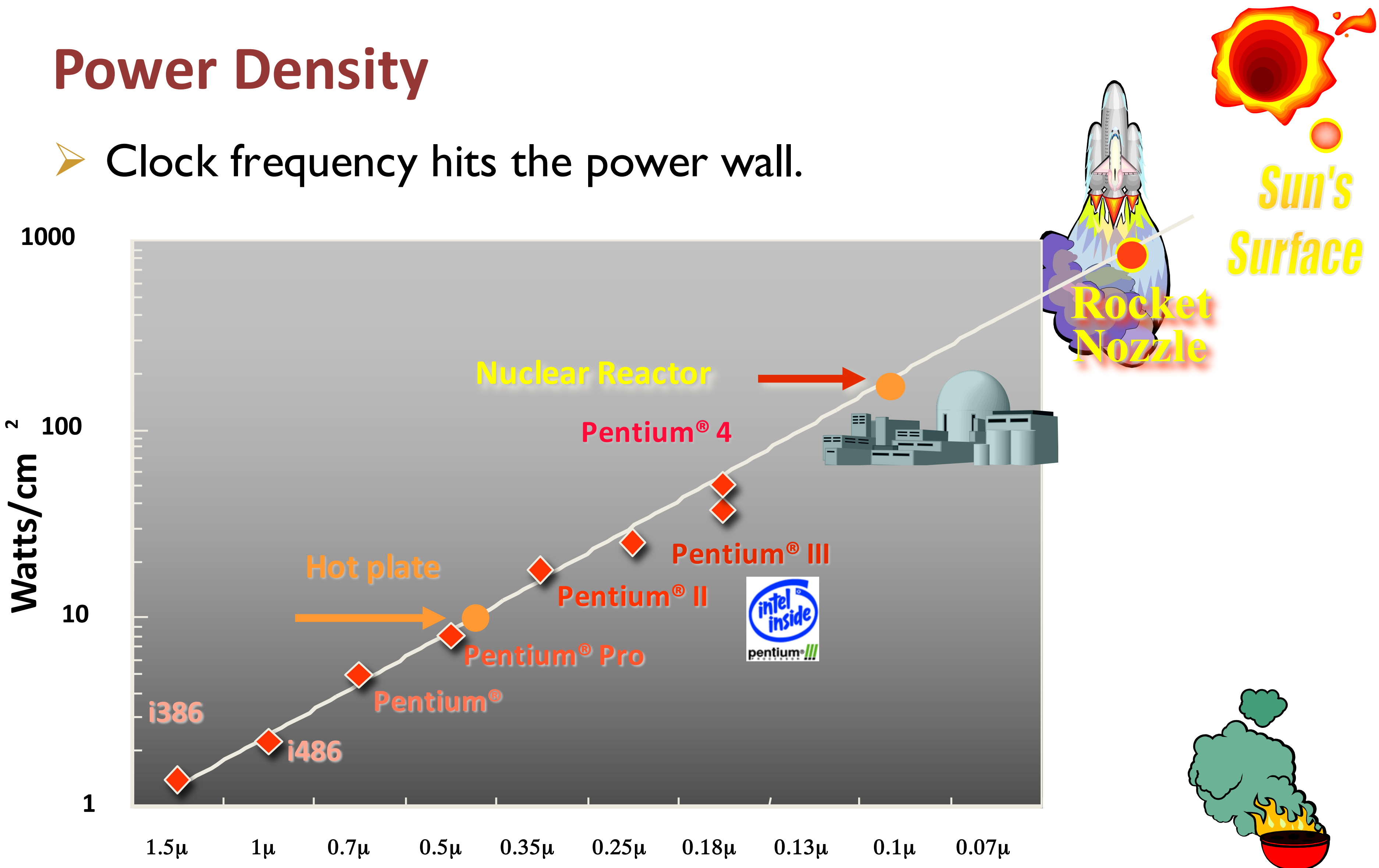


The fastest sloth working at the DMV (from Zootopia).

Original data up to the year 2010 collected and plotted by M. Horowitz, F. Labonte, O. Shacham, K. Olukotun, L. Hammond, and C. Batten. New plot and data collected for 2010-2015 by K. Rupp.

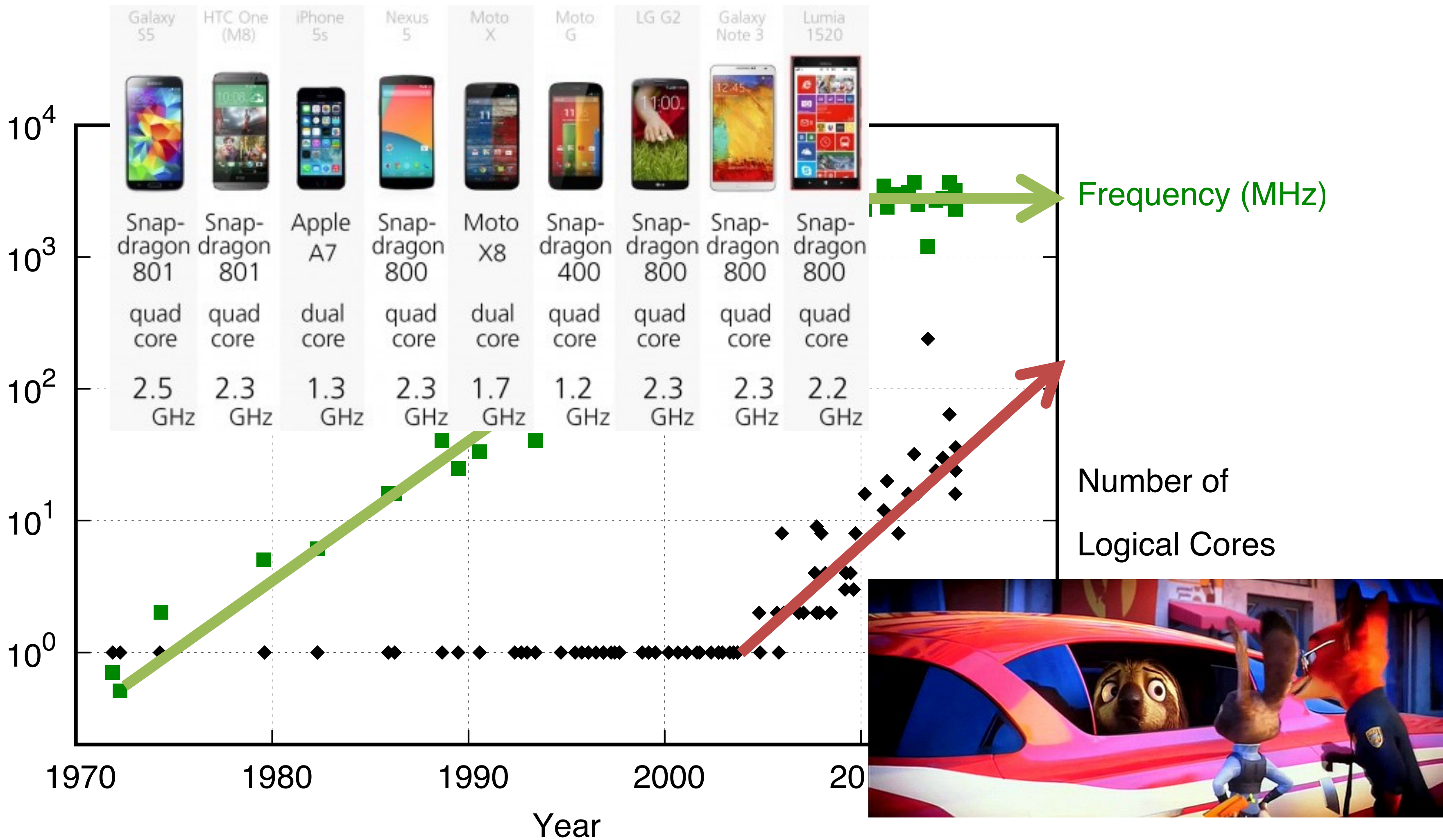
Power Density

- Clock frequency hits the power wall.



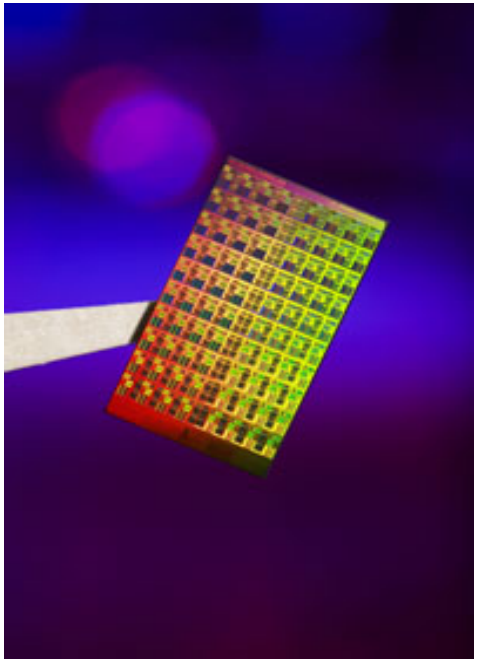
* "New Microarchitecture Challenges in the Coming Generations of CMOS Process Technologies" – Fred Pollack, Intel Corp. Micro32 conference key note - 1999.

Multi-Core Systems



Original data up to the year 2010 collected and plotted by M. Horowitz, F. Labonte, O. Shacham, K. Olukotun, L. Hammond, and C. Batten. New plot and data collected for 2010-2015 by K. Rupp.

Multi-Core Systems: The Future is Here



- Intel 80-core multicore chip (Feb 2007)
  - 80 simple cores
  - Two floating point engines /core
  - Mesh-like "network-on-a-chip"
  - 100 million transistors
  - 65nm feature size
- 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

NJIT

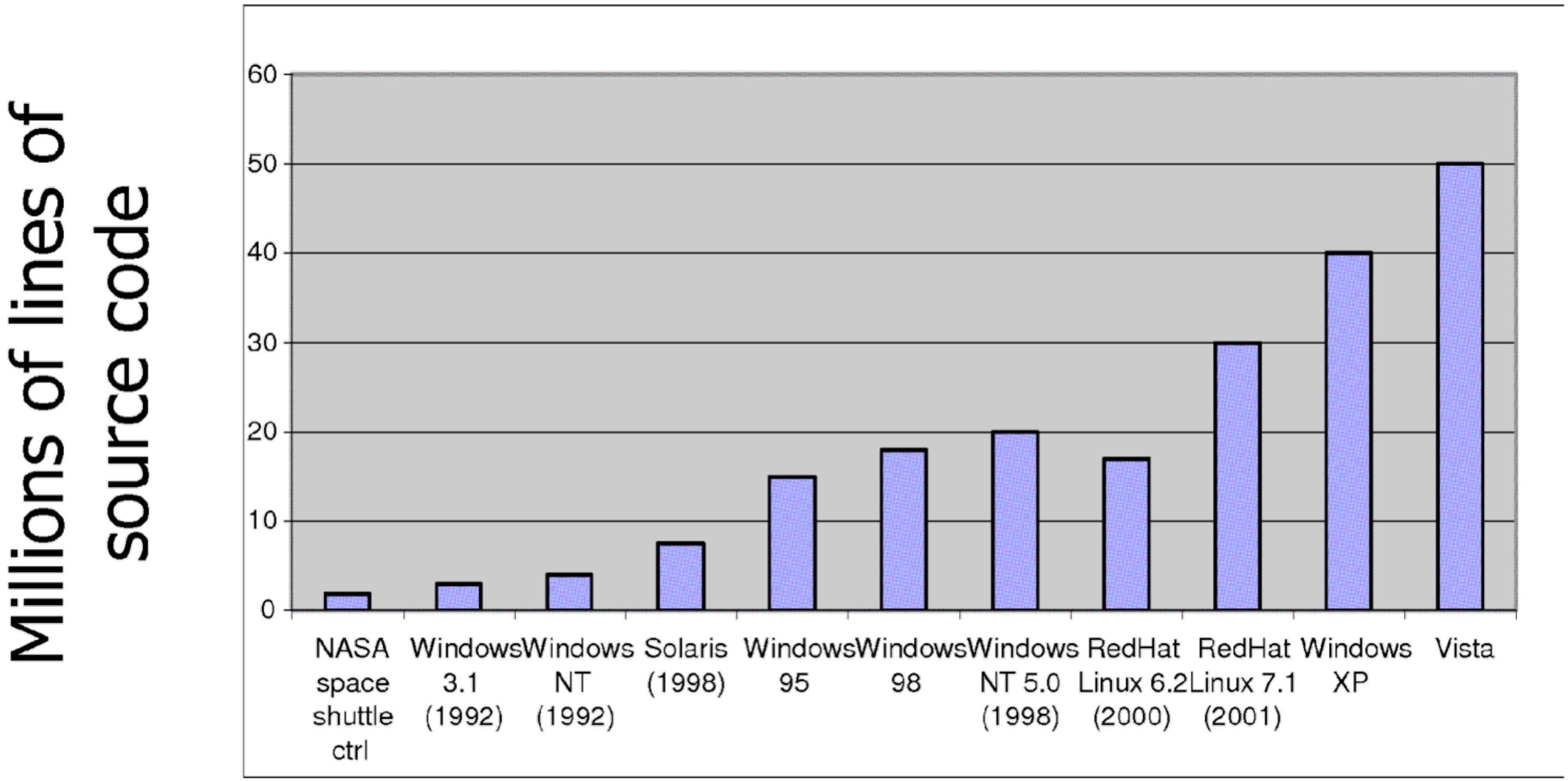
Societal Scale Systems

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

NJIT

Increasing System Complexity

- Millions of lines of source code

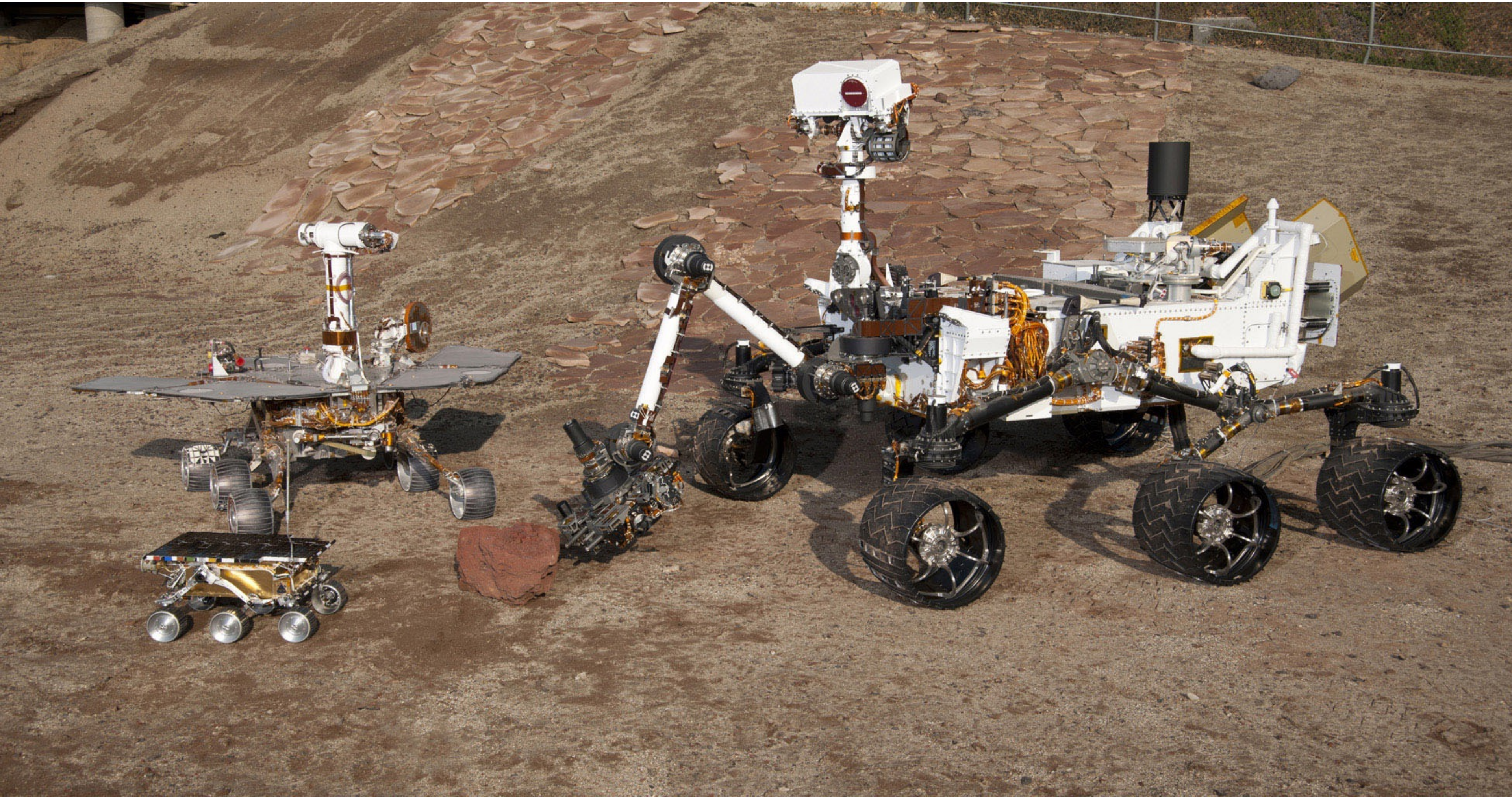


From MIT's 6.033 course

Exciting Times in Systems Research

- Moore's law ending → many challenges
- Many-cores machines
  - Amazon's X1 instances: 120 vcores and 2TB RAM
- Large scale distributed systems maturing, but many challenges remain
- Specialized hardware: FPGAs, GPUs, ASICs
- New memory technologies
- More and more requirements
  - Scale, Latency, Accuracy, Cost, Security

Example: Mars Rover (Pathfinder) Requirements



Protective Fabric Used in Mars Landing:

https://www.youtube.com/watch?v=7G75kQm7pSE

Example: Mars Rover (Pathfinder) Requirements

- Pathfinder hardware limitations/complexity:
  - 20Mhz processor, 128MB 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

How Do We Tame Complexity?

- Every piece of computer hardware can be different
  - Different CPU: Pentium, PowerPC, 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?

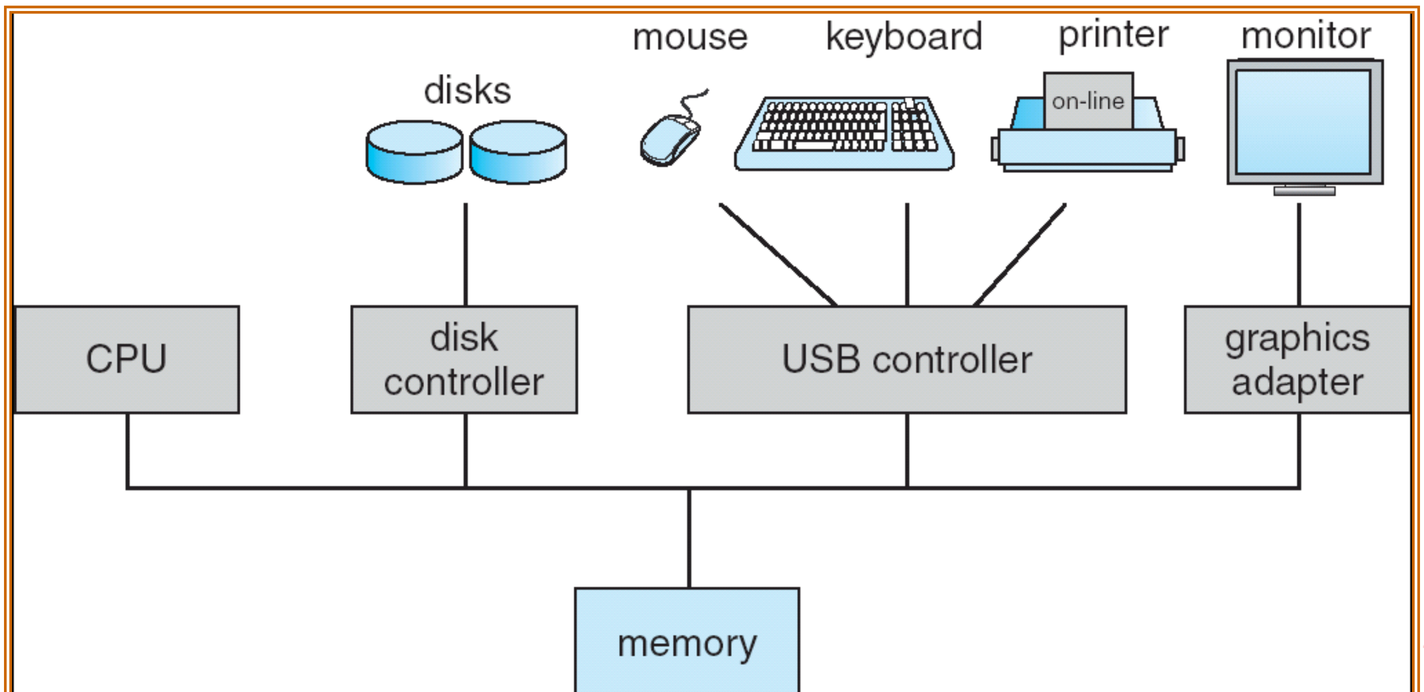
Abstraction, Arbitration & Protection

- OS offers abstractions, arbitration and protection

Application

Operating System — Virtual Machine Interface (above), Physical Machine Interface (below)

Hardware



Operating System

- A program that controls the execution of application programs
- An interface between applications and hardware

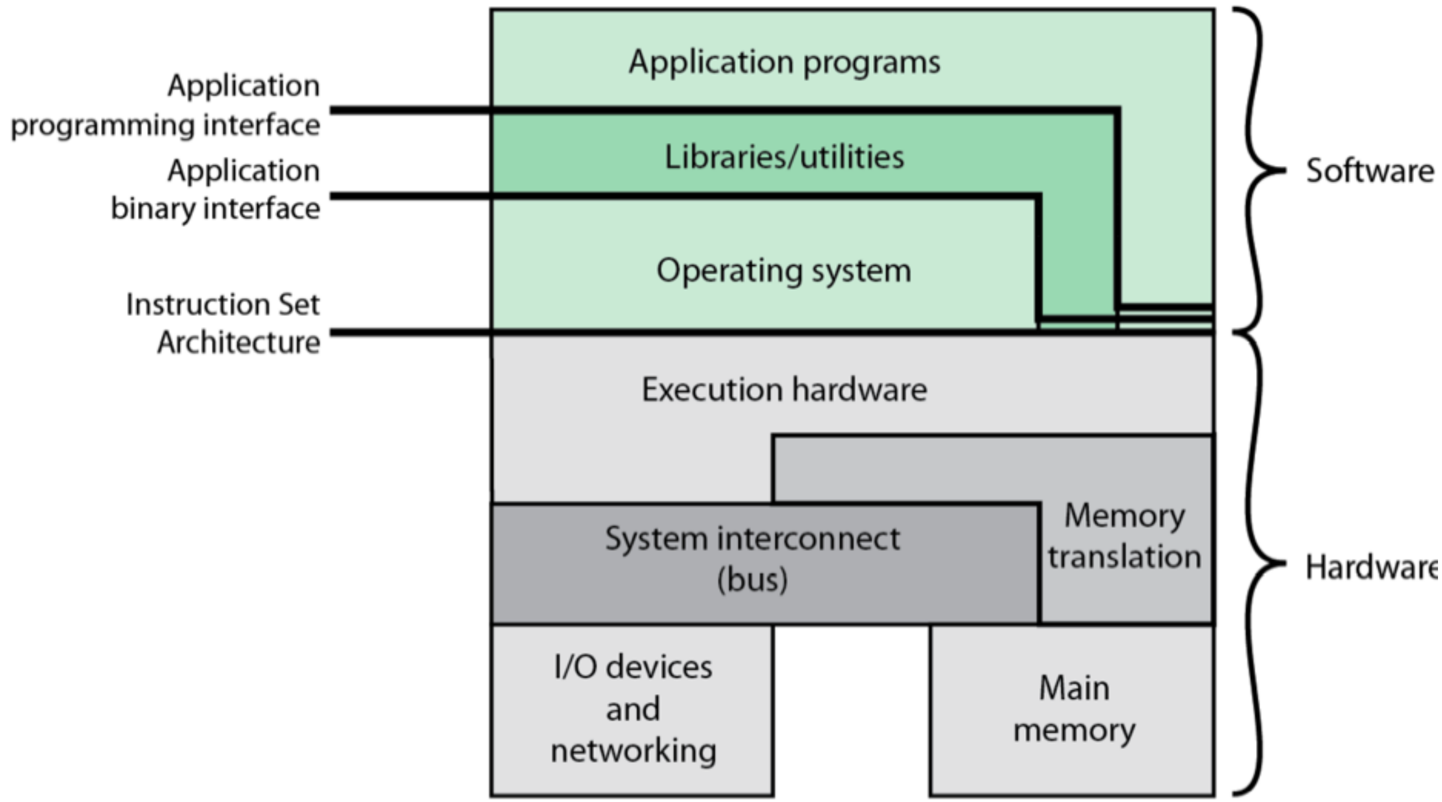


Figure 2.1 Computer Hardware and Software Infrastructure

What does an Operating System do?

- Silerschatz and Gavin:
  "An OS is Similar to a government"
  - Begs the question: does a government do anything useful by itself?
- Coordinator and Traffic Cop:
  - Manages all resources
  - Settles conflicting requests for resources
  - Prevent errors and improper use of the computer
- Facilitator:
  - Provides facilities that everyone needs
  - Standard Libraries, Windowing systems
  - Make application programming easier, faster, less error-prone
- Some features reflect both tasks:
  - E.g. File system is needed by everyone (Facilitator)
  - But File system must be Protected (Traffic Cop)

Operating System Services

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

What is an Operating System,... Really?

- Components:
  - Memory Management
  - I/O Management
  - CPU Scheduling
  - File System?
  - Communications? (Does Email belong in OS?)
  - Multitasking/multiprogramming?
- What about?
  - Multimedia Support?
  - User Interface?
  - Internet Browser?
- No universally accepted definition

What if we didn't have an Operating System?

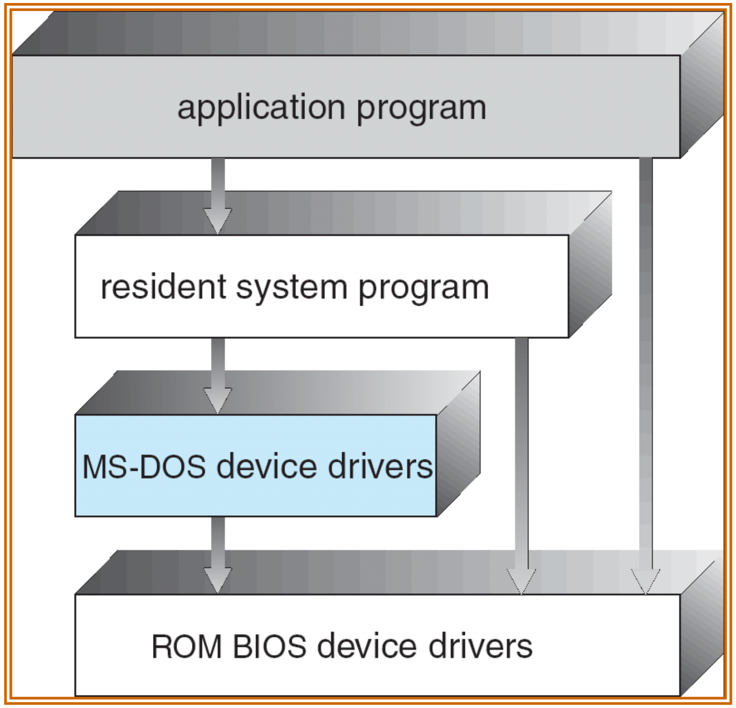
- Source Code ⇒ Compiler ⇒ Object Code ⇒ Hardware
- How do you get object code onto the hardware?
- How do you print out the answer?
  - Once upon a time, had to Toggle in program in binary and read out answer from LED's!

Altair 8080 computer

Simple System: What if only one application?

- What OS functionalities are not needed?

- Examples:
  - Very early computers, e.g., MS-DOS
  - Early PCs
  - Embedded controllers (elevators, etc.)



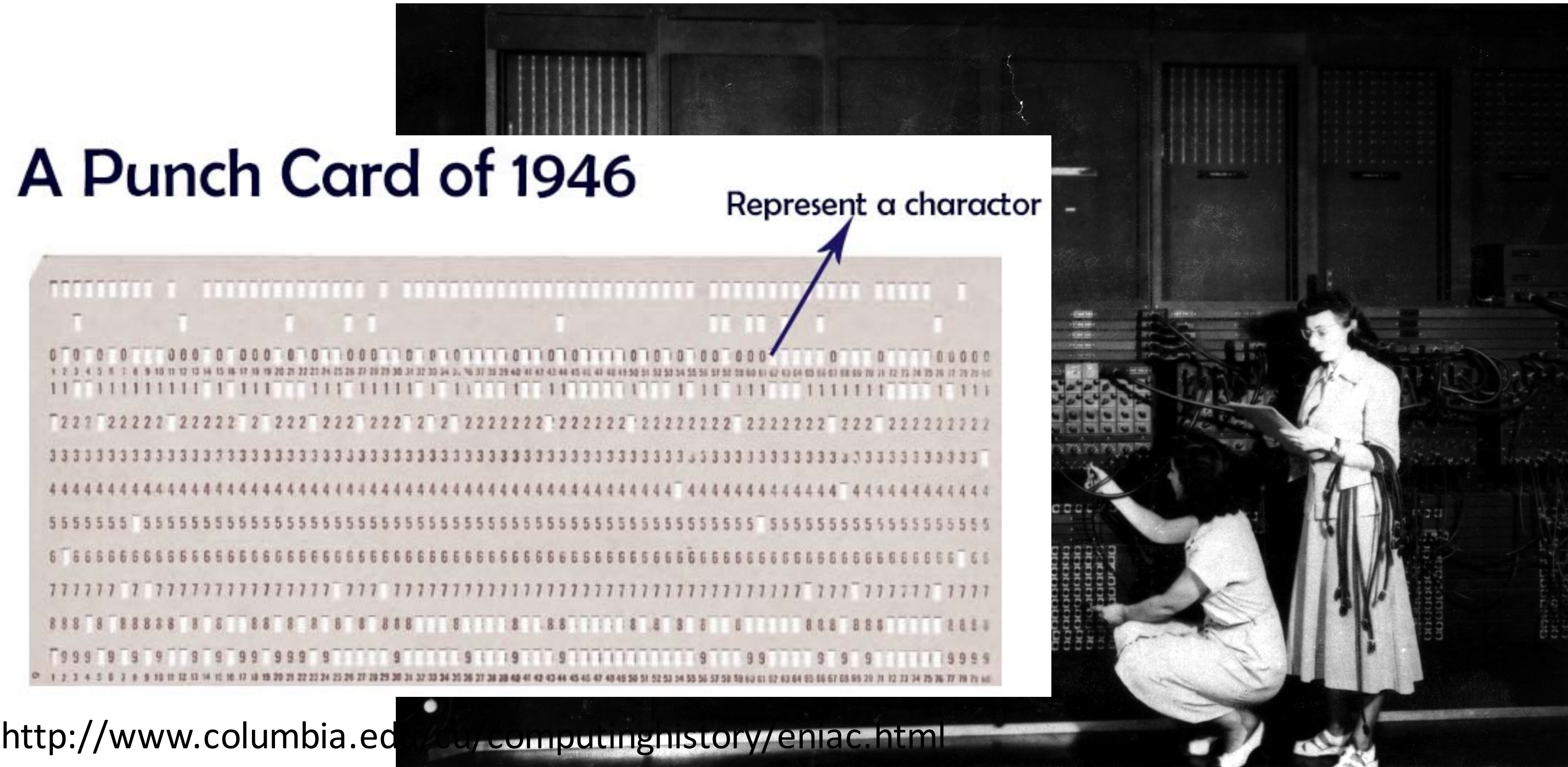
- OS becomes just a library of standard services
  - Standard device drivers
  - Interrupt handlers
  - Math libraries

Evolution of Operating Systems

New Jersey Institute of Technology

Eras of Computer Systems

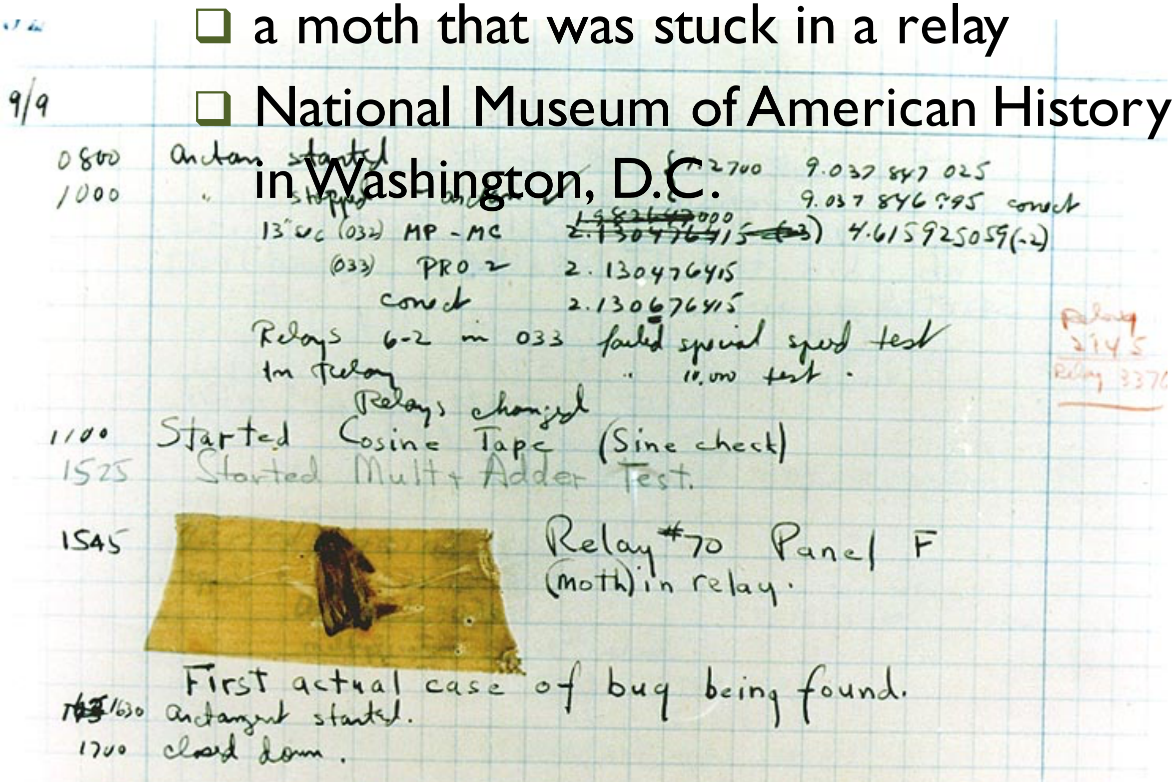
- First fully electronic computer in the U.S.
  - Electronic Numerical Integrator And Computer (ENIAC) in 1946
  - No program memory, re-wiring for each program



http://www.columbia.edu/cu/computinghistory/eniac.html

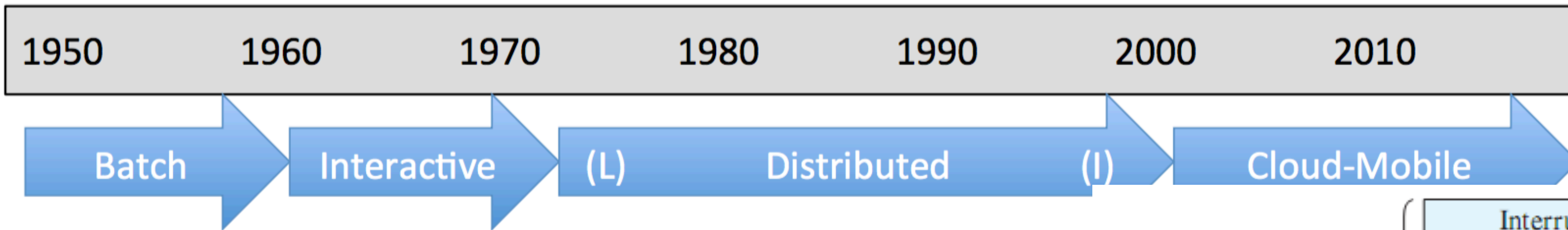
Female Computer Scientists and Programmers

- Grace Hopper
  - Part of the development team that designed the UNIVAC I computer in 1944
  - Grace Hopper Celebration of Women in Computing
- Anecdote about debugging
  - a moth that was stuck in a relay
  - National Museum of American History in Washington, D.C.



Eras of Operating Systems

- Simple Batch Systems



- Batch processing for multiple job input cards
- Operator loads monitor to run batched jobs from a prepared input tape
- Programs are constructed to branch back to the monitor after termination (early Fortran language)

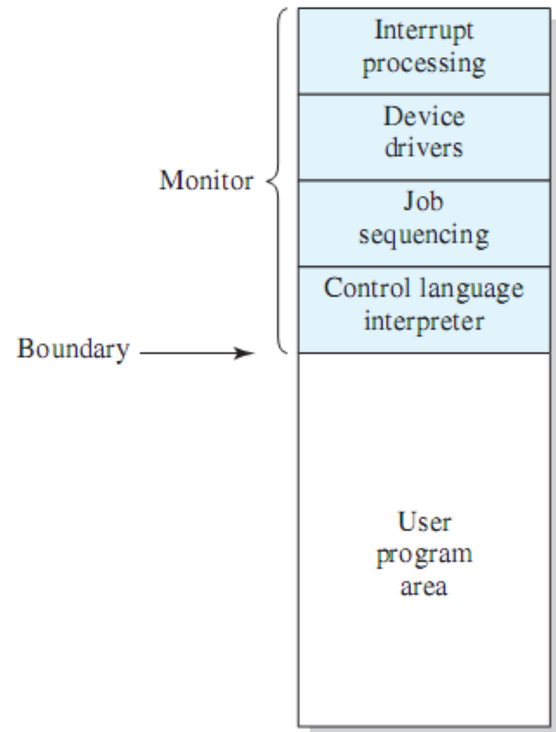
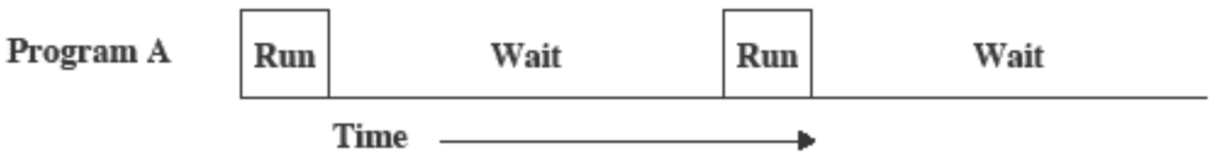


Figure 2.3 Memory Layout for a Resident Monitor

Uniprogramming

- The processor spends a certain amount of time executing, until it reaches an I/O instruction; it must then wait until that I/O instruction concludes before proceeding



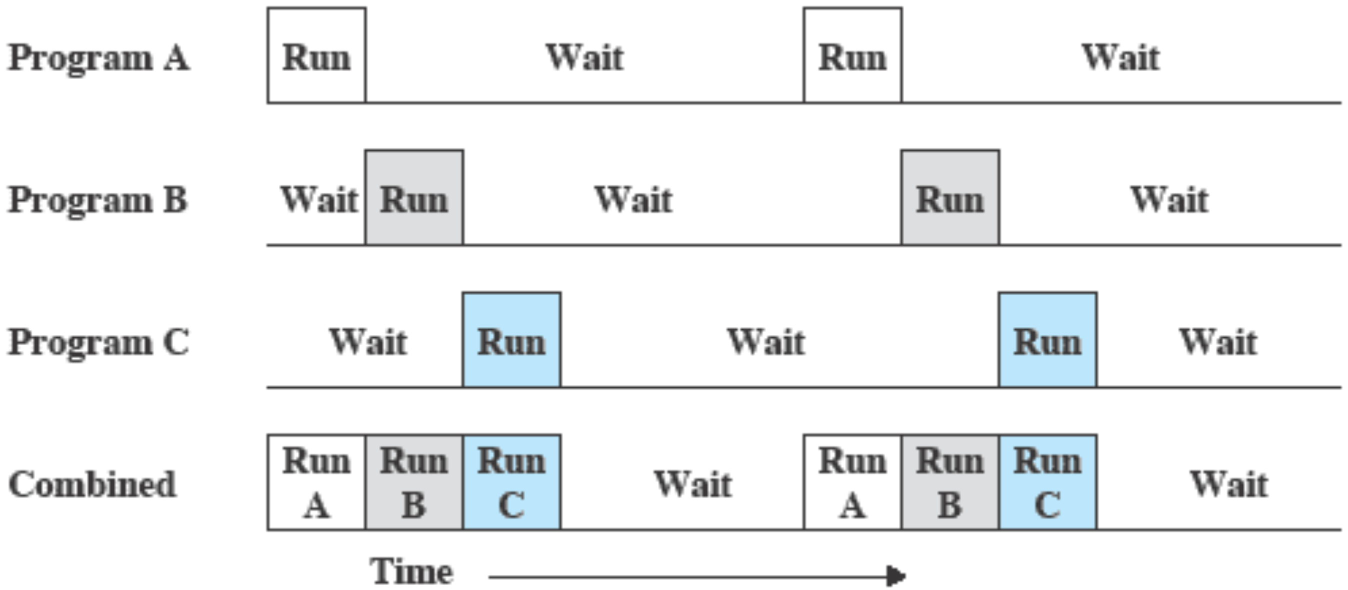
(a) Uniprogramming

Read one record from file	15 μs
Execute 100 instructions	1 μs
Write one record to file	15 μs
TOTAL	31 μs

$$\text{Percent CPU Utilization} = \frac{1}{31} = 0.032 = 3.2\%$$

Multiprogramming

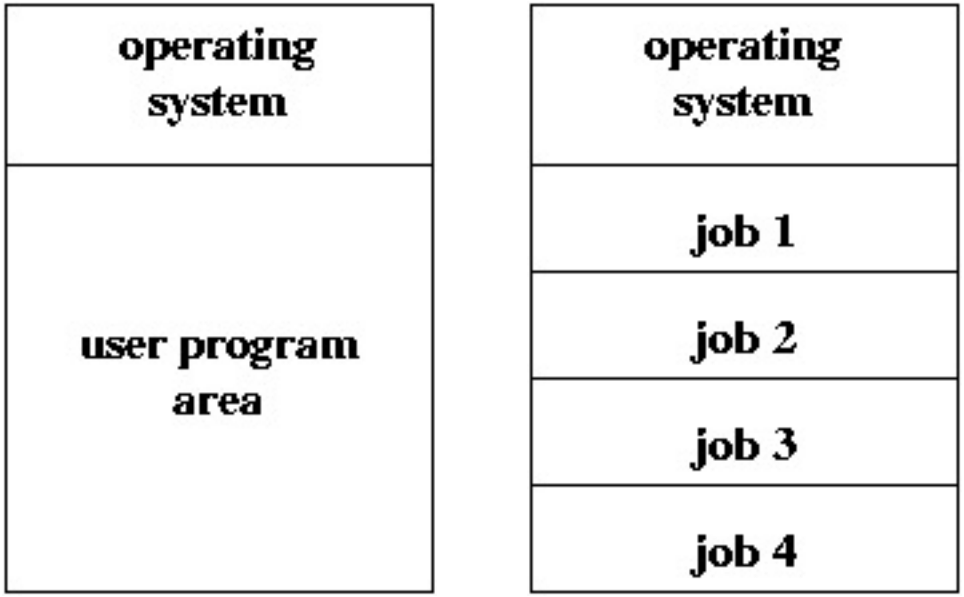
- There must be enough memory to hold the OS (resident monitor) and one user program
- When one job needs to wait for I/O, the processor can switch to the other job, which is likely not waiting for I/O



(c) Multiprogramming with three programs

From Uni- to Multi-programming

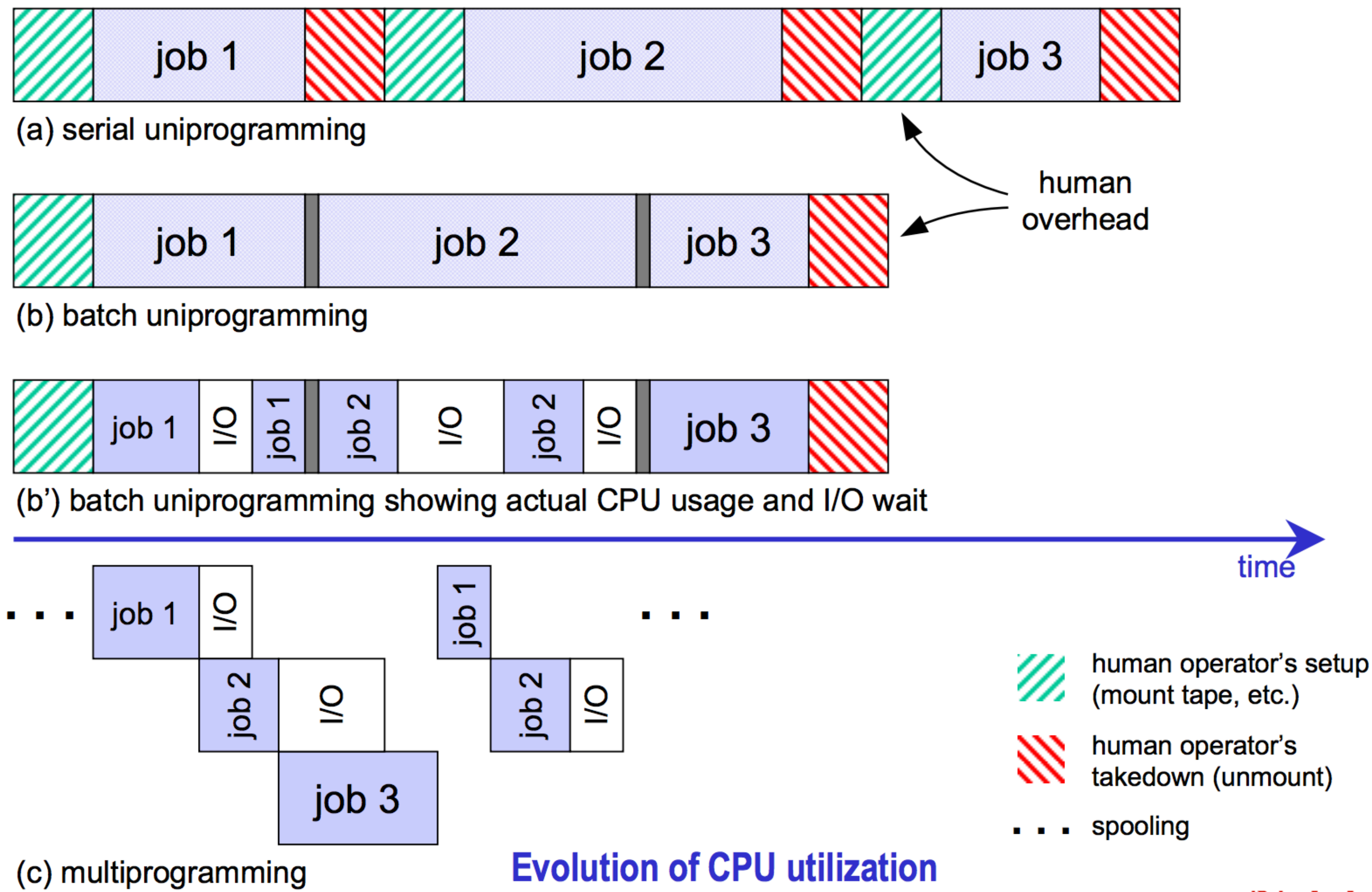
- System memory:



Uniprogramming   Multiprogramming

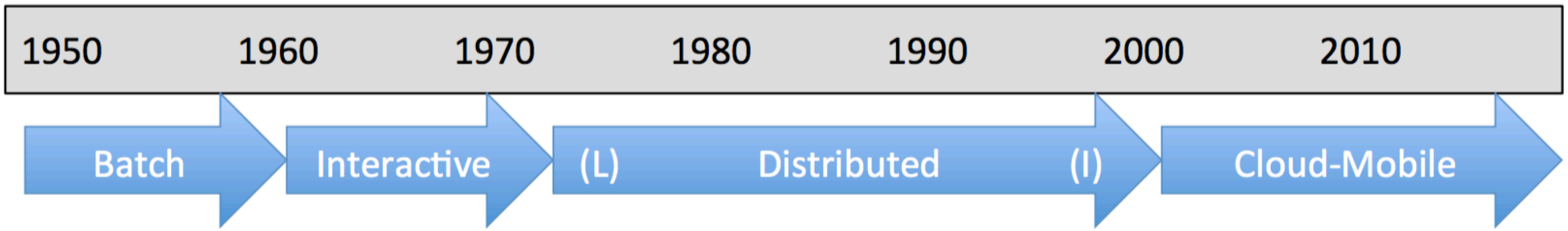
- What OS service required?
  - CPU scheduling - Need to choose a job to run next when one job makes an I/O request or terminates.
  - Memory management and protection - Need to make sure that job 1 can't read or interfere with job 2's memory.
  - I/O request must be made through system calls - System calls have access to the hardware, while the user processes should not.

Evolution of CPU utilizations



Eras of Operating Systems

- Time sharing systems (multitasking)



	Batch Multiprogramming	**Time Sharing**
Principal objective	Maximize processor use	Minimize response time
Source of directives to operating system	Job control language commands provided with the job	Commands entered at the terminal

- This is done by switching user tasks or processes transparently. We want each interactive user to get a turn on the CPU quickly - good response time. Need to switch among processes quickly - context switching.

Timesharing explained by MIT (Videos)

https://www.youtube.com/watch?v=Anxxe8SdX78

https://www.youtube.com/watch?v=Jc6jrhycDsA

New Jersey Institute of Technology

Eras of Operating Systems

- Internet