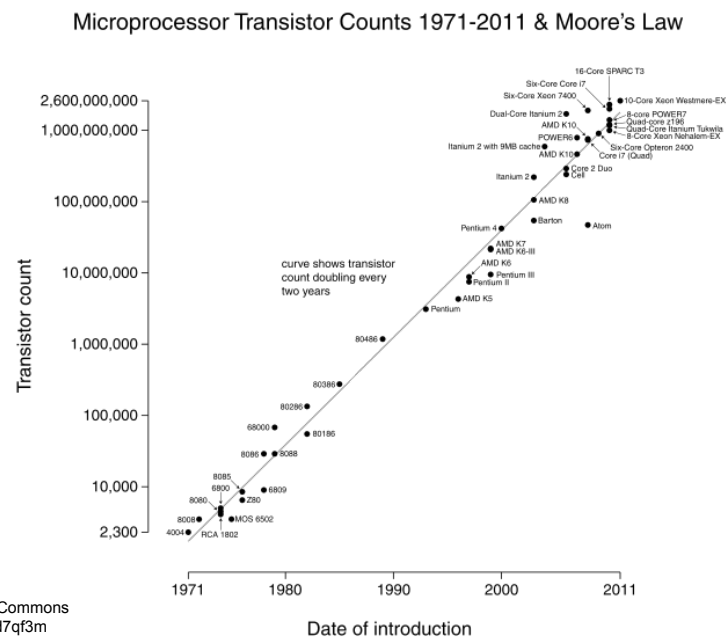


UNIT 11A

Concurrency: Multitasking & Distributed Processing

15110 Principles of Computing, Carnegie Mellon University - CORTINA

1



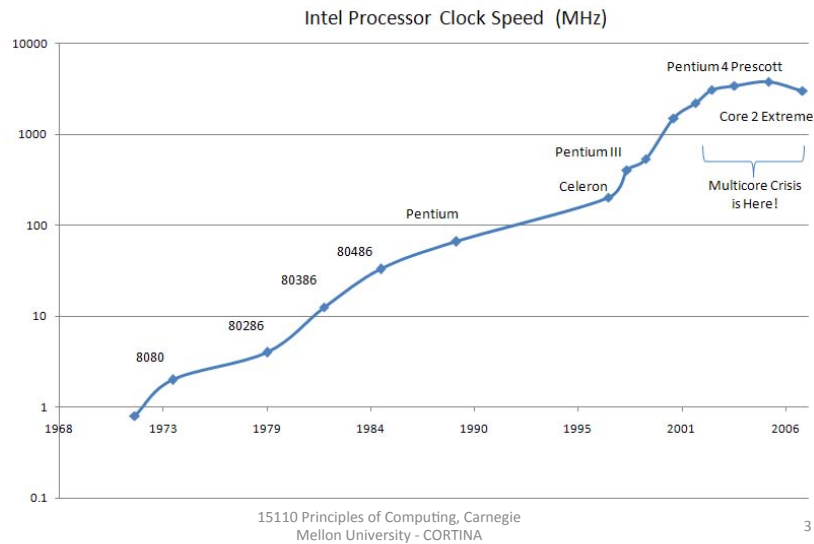
Source: Wikimedia Commons
<http://tinyurl.com/3d7qf3m>

15110 Principles of Computing, Carnegie Mellon University - CORTINA

2

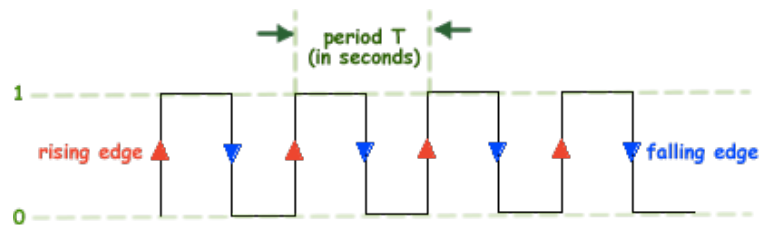
Clock Speed Is No Longer Increasing

Source: Bob Warfield <http://tinyurl.com/3pt6we9>



3

Clock Speed Is No Longer Increasing



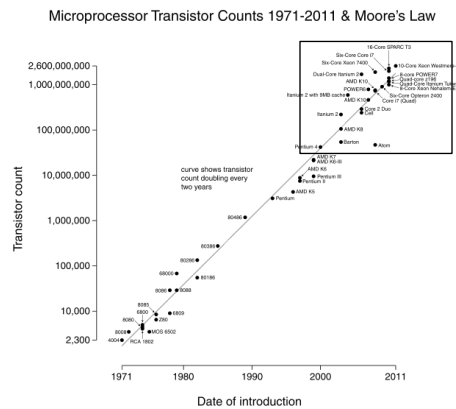
- 1 MHz is 1,000,000 cycles/second
- Computers run with a "clock" that synchronizes operations
- At each rising edge all signals go through one level of gates, etc.

15110 Principles of Computing, Carnegie Mellon University - CORTINA

4

Moore's Law

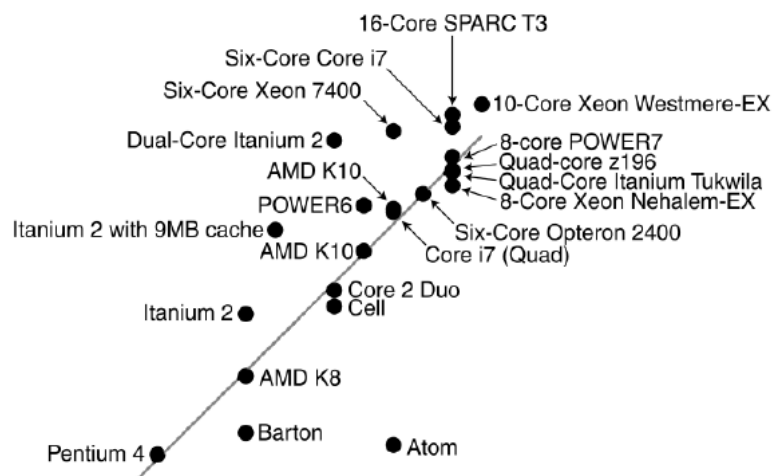
- multi-core
executing
simultaneously
- prediction:
thousands
of cores



15110 Principles of Computing, Carnegie
Mellon University - CORTINA

5

New Processors are Multi-Cores



15110 Principles of Computing, Carnegie
Mellon University - CORTINA

6

Multitasking & Operating Systems

- **Multitasking** - The coordination of several computational processes on one processor or several cores.
- An **operating system** (OS) is the system software responsible for the direct control and management of hardware and basic system operations.
- An OS provides a foundation upon which to run application software such as word processing programs, web browsers, etc.

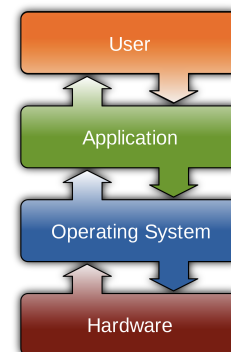
from Wikipedia

15110 Principles of Computing, Carnegie Mellon University - CORTINA

7

Operating Systems

- **UNIX**
 - System V, BSD, Linux
 - Proprietary: Solaris, Mac OS
- **Windows**
 - Windows 7
 - Windows 8
 - Windows 10



Wikipedia

15110 Principles of Computing, Carnegie Mellon University - CORTINA

8

Single-core Multitasking



- Each (un-blocked) application runs for a very short time on the computer's processor and then another process runs, then another...
- This is done at such a fast rate that it appears to the computer user that all applications are running at the same time.
 - How do actors appear to move in a motion picture?

15110 Principles of Computing, Carnegie Mellon University - CORTINA

9

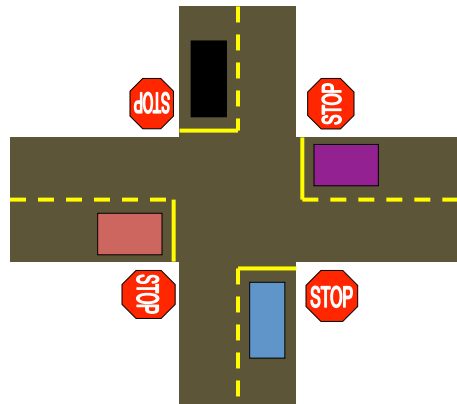
Critical Section

- A **critical section** is a section of computer code that must only be executed by one process or thread at a time.
- Examples:
 - A printer queue receiving a file to be printed
 - Code to set a seat as reserved
 - Web server that generates and returns a web page showing current registration in course

15110 Principles of Computing, Carnegie Mellon University - CORTINA

10

A Critical Section



Cars may not make turns through this intersection. They may only proceed straight ahead. When a car stops at this intersection, it can proceed straight ahead if there is no car to its left and no car to its right. Otherwise it must wait some random amount of seconds and then check again to see if it's safe to proceed. If not, it waits again, and so on.

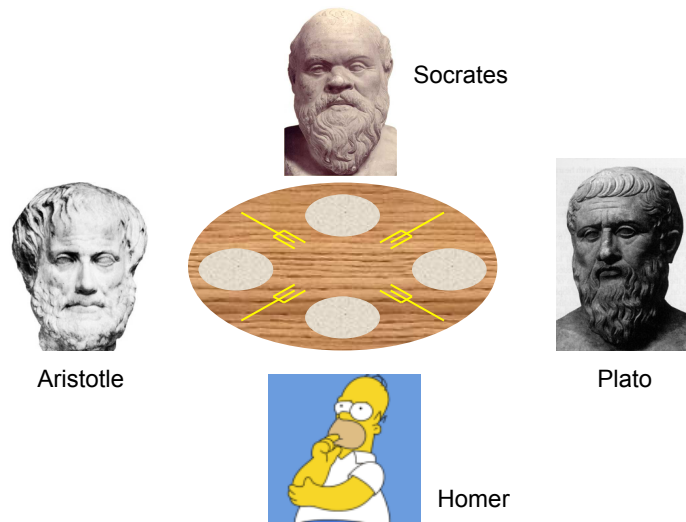
Shared Computing Resources

- memory
- tape drives
- disk drives
- printers
- communication ports
- input devices (keyboard, mouse)

Deadlock

- Deadlock is the condition when two or more processes are all waiting for some shared resource that other processes of the group hold, causing all processes to wait forever without proceeding.
- How can deadlock occur at the intersection with the 4-way stop?

Dining Philosopher's Problem



The Dining Philosopher's

- Each philosopher thinks for a while, then picks up his left fork, then picks up his right fork, then eats, then puts down his left fork, then puts down his right fork, thinks for a while...
 - We assume here that each philosopher thinks and eats for random times, and a philosopher cannot be interrupted while he picks up or puts down a single fork.
- Each fork models a "resource" on a computer controlled by an OS.
- How can the philosophers get deadlocked?

15110 Principles of Computing, Carnegie
Mellon University - CORTINA

15

Distributed Systems

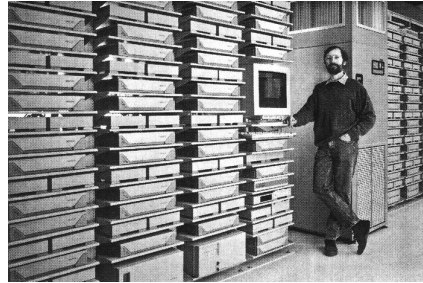
- A distributed system is an application that consists of processes that
 - execute on multiple computers connected through a network, and
 - cooperate to accomplish a task.
- Advantages
 - open
 - add new parts and interact with remote users
 - scalable
 - system can be altered to accommodate changes in numbers of users, resources, computer systems

15110 Principles of Computing, Carnegie
Mellon University - CORTINA

16

Render Farms

- Rendering
 - flatten a 3-d space to a 2-d
 - lighting (raytracing)
 - potential concurrency
 - frames
 - pixels within a frame
- Toy Story (1995)
 - about 4 hours for each frame
 - 80 SPARCstation 20 systems
 - 1 SPARCserver 1000 system

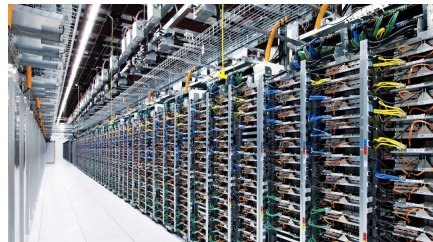


*Shades of
ENIAC!*

15110 Principles of Computing, Carnegie
Mellon University - CORTINA

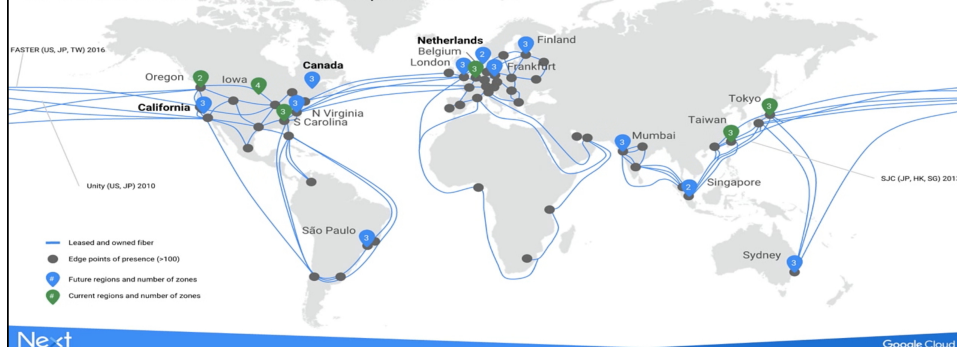
17

Google Data Centers



GCP Infrastructure

6 regions, 18 zones, over 100 points of presence, and a well-provisioned global network comprised of hundreds of thousands of miles of fiber optic cable.



Next

Google Cloud

15110 Principles of Computing, Carnegie
Mellon University - CORTINA

18

Challenge of Distributed Computing: Reliability in Context of Failure

*Failure is the defining difference between distributed and local programming, so you have to design distributed systems with the expectation of failure. Imagine asking people, "If the probability of something happening is one in 1013, how often would it happen?" Common sense would be to answer, "Never." ... But if you ask a physicist, she would say, "All the time. In a cubic foot of air, those things happen all the time." When you design distributed systems, you have to say, "Failure happens all the time." So when you design, **you design for failure**. It is your number one concern.*

— Ken Arnold

15110 Principles of Computing, Carnegie
Mellon University - CORTINA

19

Examples of Failures

- permanent network failures
- dropped messages between sender and receiver
- an individual computer breaks
- a process crashes or goes into an infinite loop



The "Blue Screen of Death"

15110 Principles of Computing, Carnegie
Mellon University - CORTINA

20