Lecture 1

Overview

CPSC 275

Introduction to Computer Systems

What’s happening when running a program?

Java

```
public class Hello {
   public static void main(String[] args) {
      System.out.println("Hello, CPSC275!");
   }
}
```

C

```
void main(int argc, char *argv[])
{
  printf(“Hello, CPSC275!”);
}
```

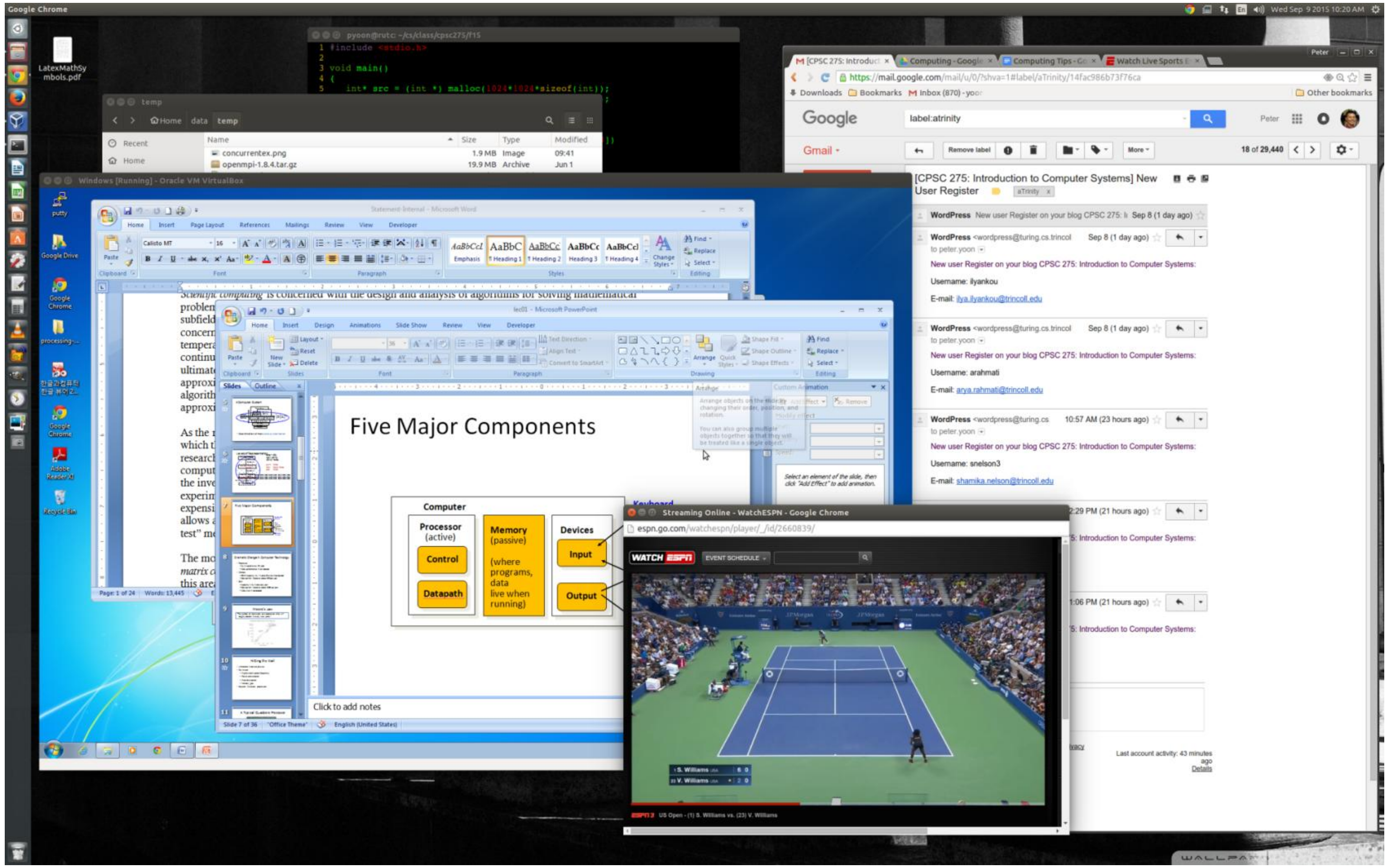
Why does this program

```
void copyij(int src[2048][2048],
            int dst[2048][2048])
{
  int i,j;
  for (i = 0; i < 2048; i++)
    for (j = 0; j < 2048; j++)
      dst[i][j] = src[i][j];
}
```

run faster than this one?

```
void copyji(int src[2048][2048],
            int dst[2048][2048])
{
  int i,j;
  for (j = 0; j < 2048; j++)
    for (i = 0; i < 2048; i++)
      dst[i][j] = src[i][j];
}
```

What makes multiple programs run on the same system?



pyoon@rutc: ~/cs/class/cpsc275/f15

```
#include <stdio.h>

void main()
{
   int* src = (int *) malloc(1024*1024*sizeof(int));
```

Name	Size	Type	Modified
concurrentex.png	1.9 MB	Image	09:41
openmpi-1.8.4.tar.gz	19.9 MB	Archive	Jun 1

Five Major Components

Computer

Processor (active)

Control

Datapath

Memory (passive)

(where programs, data live when running)

Devices

Input

Output

Click to add notes

label:atrinity

[CPSC 275: Introduction to Computer Systems] New User Register

WordPress <wordpress@turing.cs.trincol Sep 8 (1 day ago)
to peter.yoon

New user Register on your blog CPSC 275: Introduction to Computer Systems:

Username: ilyankou

E-mail: ilya.ilyankou@trincoll.edu

WordPress <wordpress@turing.cs.trincol Sep 8 (1 day ago)
to peter.yoon

New user Register on your blog CPSC 275: Introduction to Computer Systems:

Username: arahmati

E-mail: arya.rahmati@trincoll.edu

WordPress <wordpress@turing.cs 10:57 AM (23 hours ago)
to peter.yoon

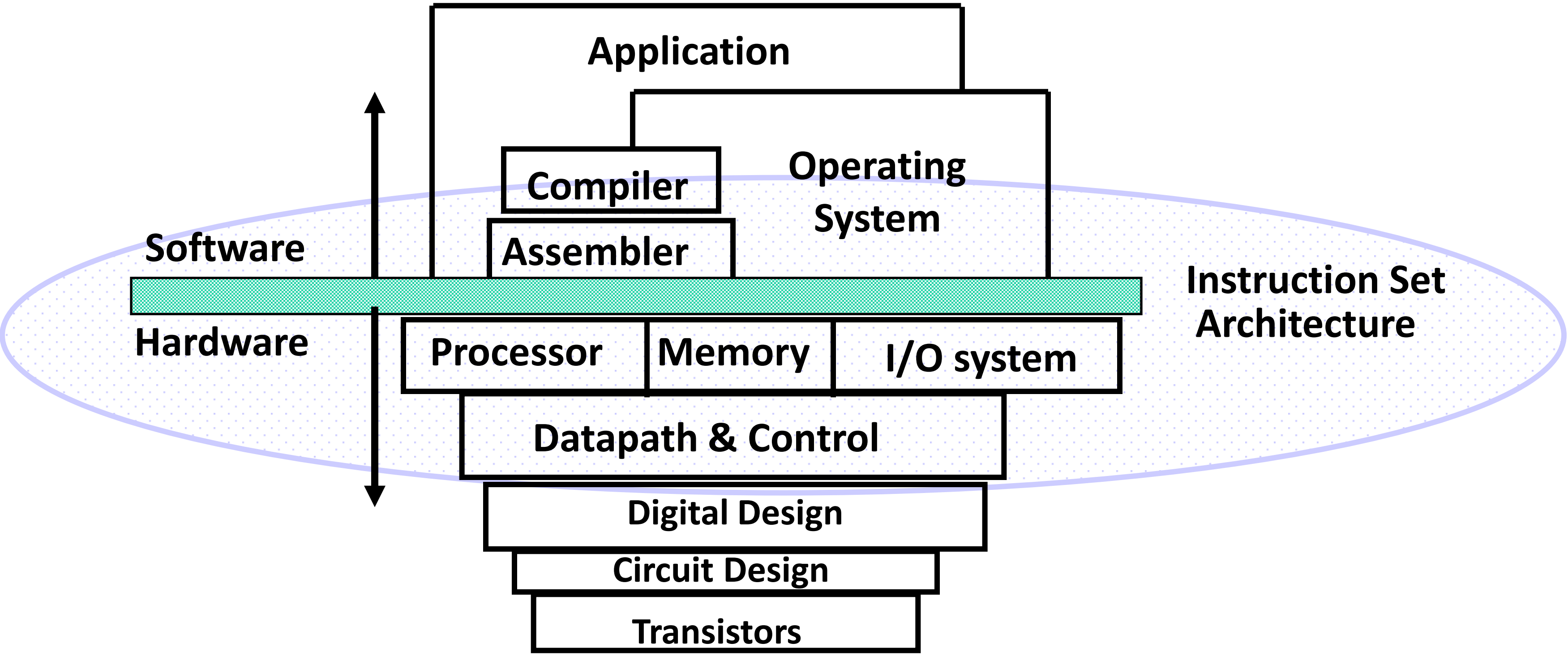
New user Register on your blog CPSC 275: Introduction to Computer Systems:

Username: snelson3

E-mail: shamika.nelson@trincoll.edu

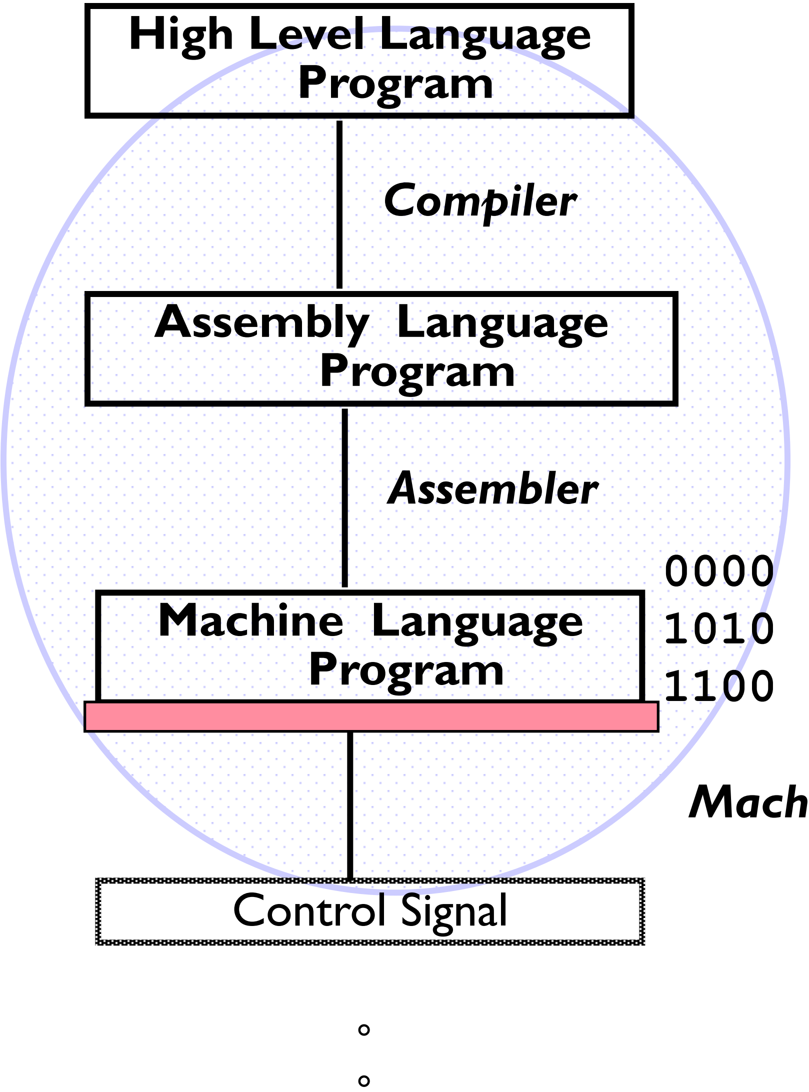
Google Chrome
Wed Sep 9 2015 10:20 AM
LatexMathSymbols.pdf
temp
Recent
Home
Windows [Running] - Oracle VM VirtualBox
Statement Internal - Microsoft Word
test1 - Microsoft PowerPoint
Remove label
More
18 of 29,440
Streaming Online - WatchESPN - Google Chrome
espn.go.com/watchespn/player/_/id/2660839/
WATCH ESPN
EVENT SCHEDULE
S. Williams
V. Williams
US Open - (1) S. Williams vs. (23) V. Williams

A typical computer system



- Coordination of many levels of *abstraction*

Levels of representation



High Level Language Program

Compiler

Assembly Language Program

Assembler

Machine Language Program

Machine Interpretation

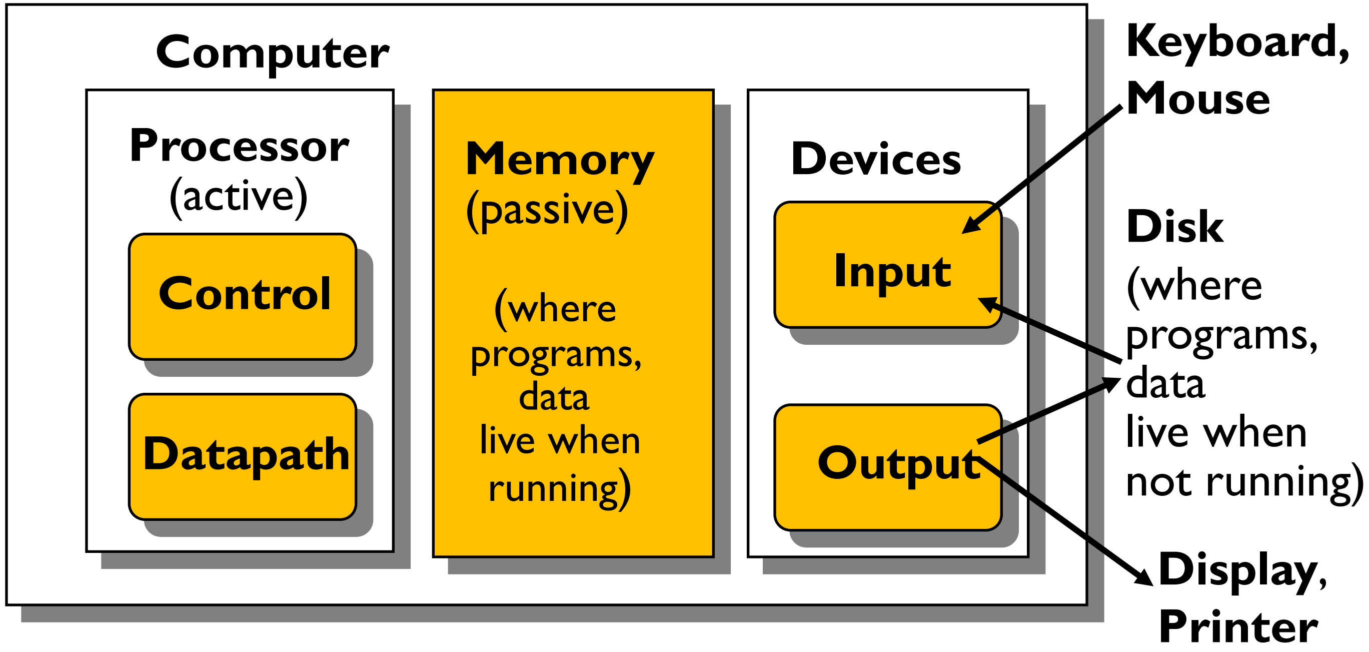
Control Signal

```
temp = v[k];
v[k] = v[k+1];
v[k+1] = temp;
```

```
pushl  %ebp
movl   %esp, %ebp
popl   %ebp
```

```
0000 1001 1100 0110 1010 1111 0101 1000
1010 1111 0101 1000 0000 1001 1100 0110
1100 0110 1010 1111 0101 1000 0000 1001
```

Major components



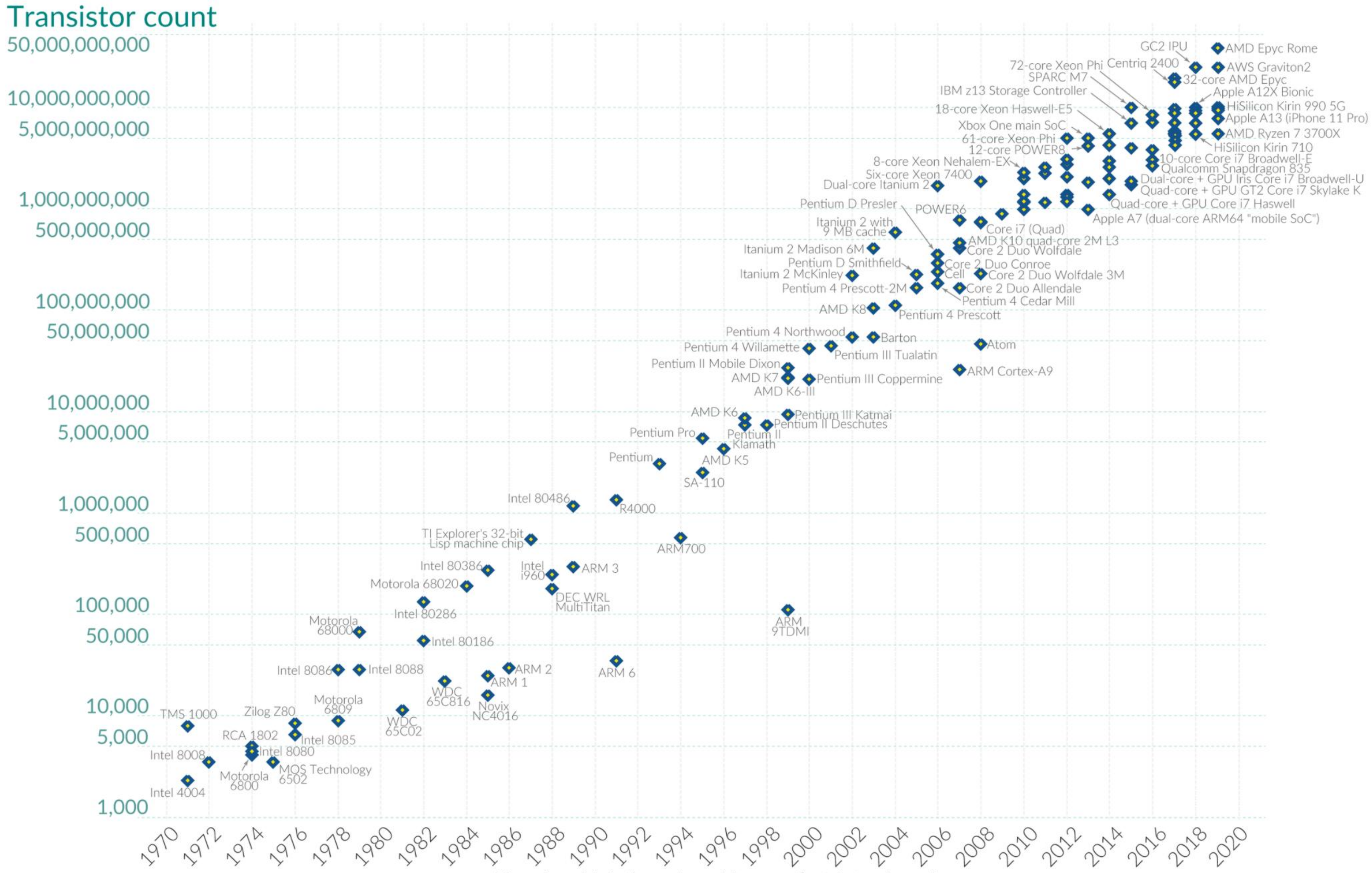
Dramatic change in computer technology

- Processor
  - 2x in speed every 1.5 years
  - 100x performance in last decade
- Memory
  - DRAM capacity: 2x / 2 years; 64x size in last decade
  - Cost per bit: improves about 25% per year
- Disk
  - capacity: > 2x in size every year
  - Cost per bit: improves about 100% per year
  - 120x size in last decade

Moore’s Law

“The number of transistors on a processor chip will roughly double in every two years.” (1965)

Moore’s Law

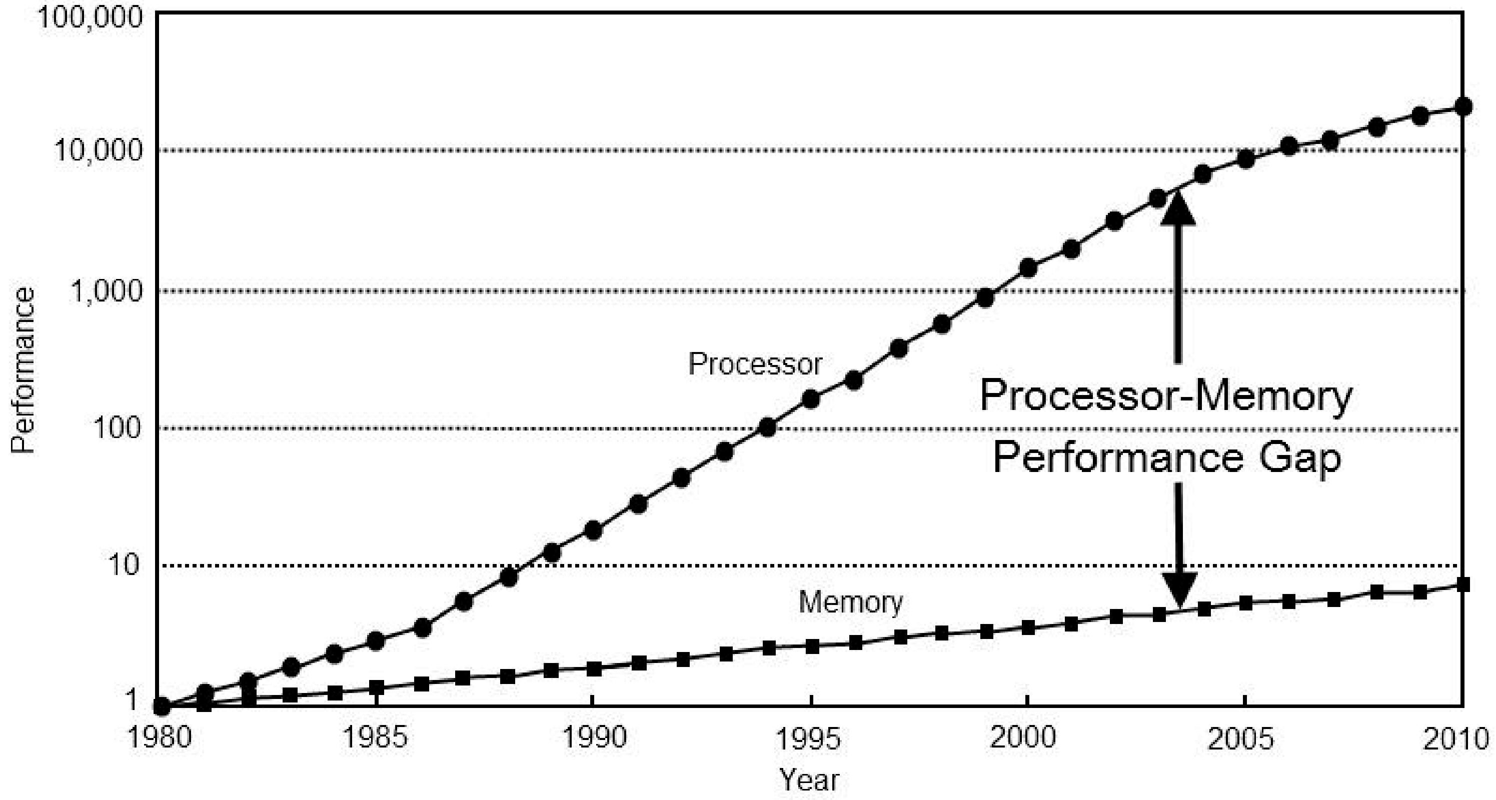


Data source: Wikipedia (wikipedia.org/wiki/Transistor_count)

Hitting the performance wall

- Limitations in device physics
- Key Issues
  - Higher *clock* speed (frequency)
  - Power consumption
  - Heat dissipation
  - Processor-memory performance gap

Processor-memory performance gap



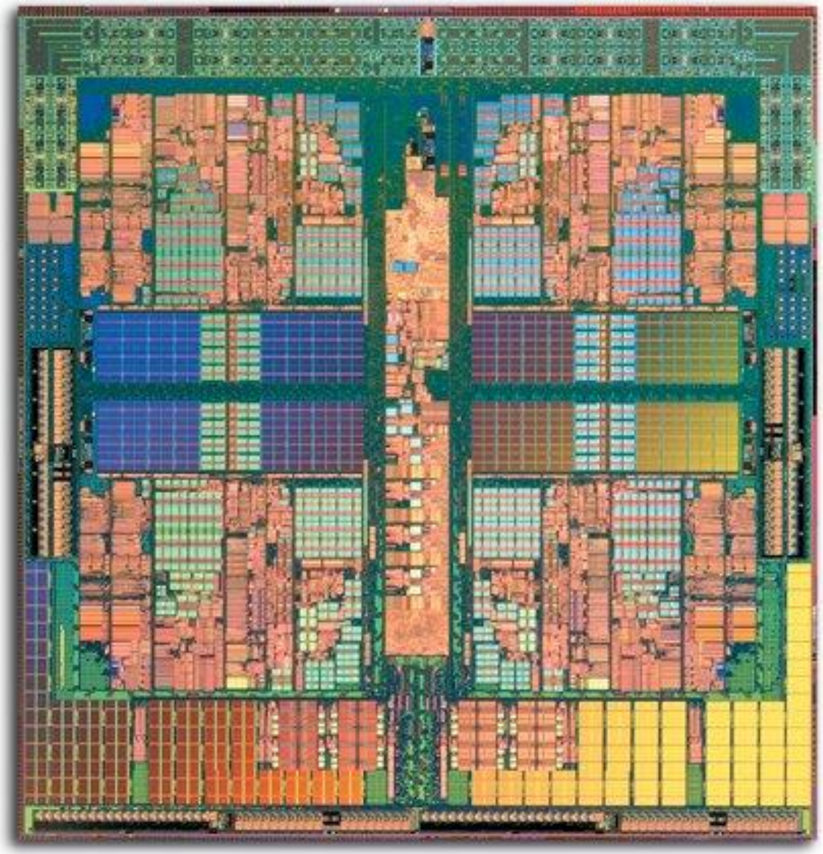
Source: David Patterson, UC Berkeley

Hitting the performance wall

- Limitations in device physics
- Key Issues
  - Higher *clock* speed (frequency)
  - Power consumption
  - Heat dissipation
  - Processor-memory performance gap

- Solution: *Multicore* architecture

A typical quad-core processor



From a programmer’s perspective

- What the programmer writes
- How it is converted to something the computers can understand
- How the computer interprets/runs the program
- What makes programs go slow

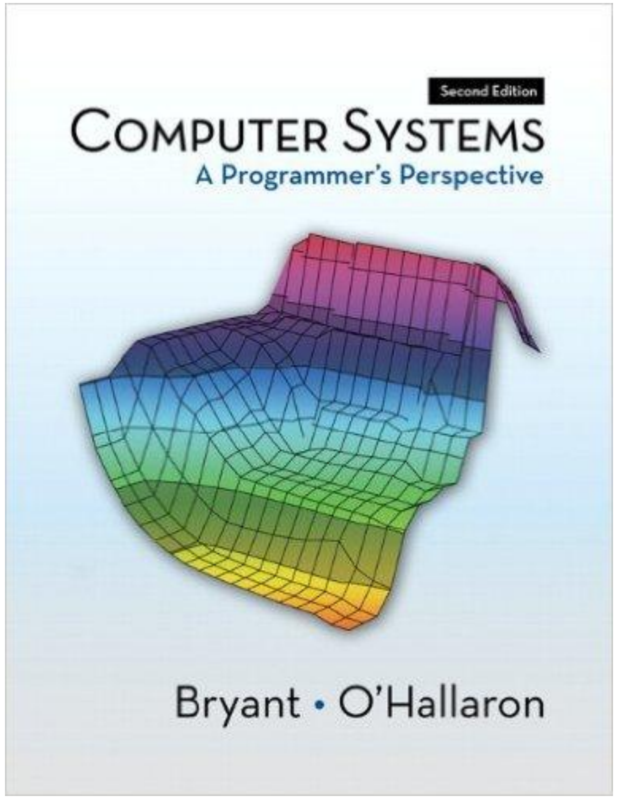
Big ideas in CPSC 275

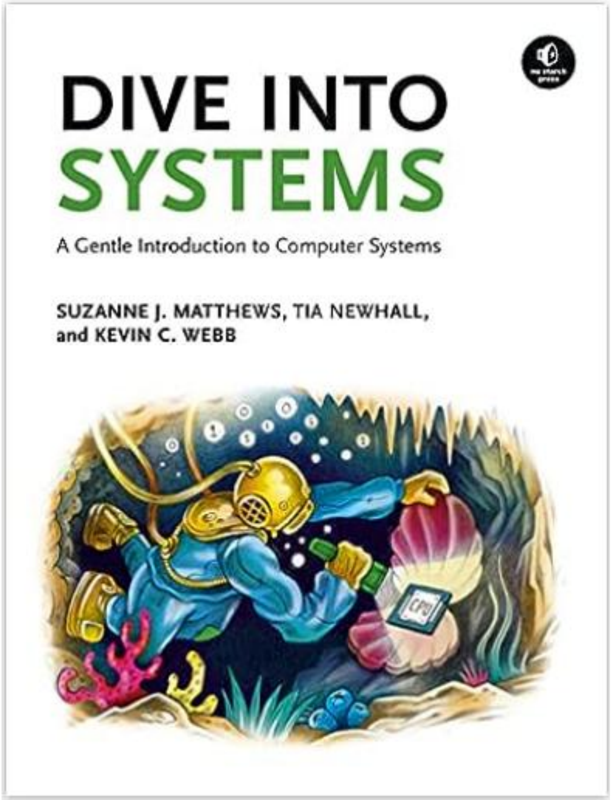
- **Data** can be anything (integers, floating point, characters, ...): a program determines what it is
- **Stored program** concept: instructions treated like data
- Principle of **abstraction:** used to build systems as layers
- Principle of **locality**: exploited via a memory hierarchy (*cache*)
- Greater performance by exploiting **parallelism**
- Principles/pitfalls of **performance measurement**

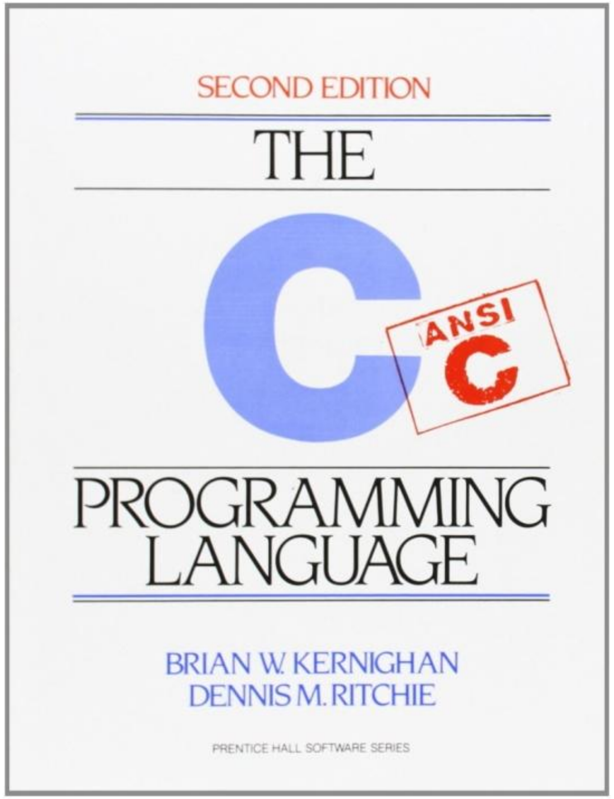
Teaching staff

- Prof. Peter Yoon
  - Office: MECC 127
  - Phone: 297-2461
  - Email: peter.yoon@trincoll.edu
  - Office Hours: MWF 10:00 a.m. – 10:50 p.m. or by appointment
- Teaching Assistants:
  - Shivanshu Dwivedi
  - Anupam Khargharia
  - Chi Le
  - Kamilla Volkova
- Course website: http://www.cs.trincoll.edu/~pyoon/cpsc275

Recommended Text




DIVE INTO
SYSTEMS
A Gentle Introduction to Computer Systems
SUZANNE J. MATTHEWS, TIA NEWHALL,
and KEVIN C. WEBB
CPU



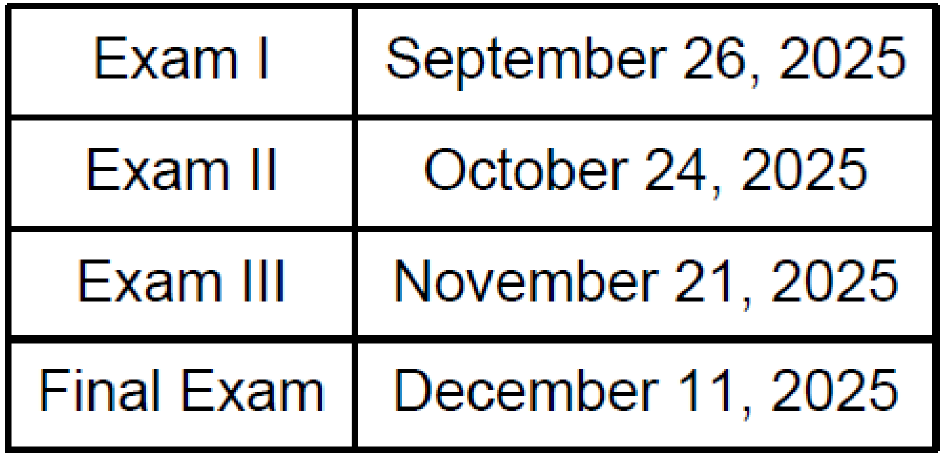
Lab

- Dates/Time
  - W 1:30 p.m. – 4:10 p.m. in MECC 124 (Yoon)
  - R 1:30 p.m. – 4:10 p.m. in MECC 124 (Chakraborttii)
- Topics
  - Linux OS
  - Shell programming
  - Assembly programming
  - Systems programming
  - Debugging strategies
  - Code optimization
  - Code review
  - And much more

Requirements/Grading

Attendance	5%
Quiz	10%
Lab	5%
Graded Labs (3)	20%
Assignments	10%
Exams (3)	30%
Final Exam	20%

Exam schedule



Exam I	September 26, 2025
Exam II	October 24, 2025
Exam III	November 21, 2025
Final Exam	December 11, 2025

Course Policy

- Attendance very important!
- Late work will *not* be accepted
- Learning disability
- Academic honesty
- Classroom rules
  - Laptops: permitted
  - Electronic communications: forbidden
    - No Web surfing, email, texting, cell phone calls, etc.

Academic Honesty

- Strictly prohibited
  - Giving and receiving help in the actual development of code or writing of an assignment
  - Looking at another person’s code
  - Showing your code to another person
  - Sharing a copy of all or part of your code, regardless of whether that copy is on paper or in a computer file
  - Turning in the work of any other person(s) (former students, friends, people on the Internet, etc.) or code generated by a machine and representing it as your own work
  - Fabricating compilation or execution results
- Penalty for cheating
  - A failing grade (F) for the course
  - Will be asked to appear at an academic dishonesty hearing.