



The Plan

<https://cs.wellesley.edu/~cs240/>

1

Welcome to
CS 240:
Foundations of
**Computer
Systems!**

Program, Application

Programming Language

Compiler/Interpreter

Operating System

Instruction Set Architecture

Microarchitecture

Digital Logic

Devices (transistors, etc.)

Solid-State Physics

2

Today

- 1 What is CS 240?
- 2 Why take CS 240?
- 3 How does CS 240 work?
- 4 Start diving into digital logic

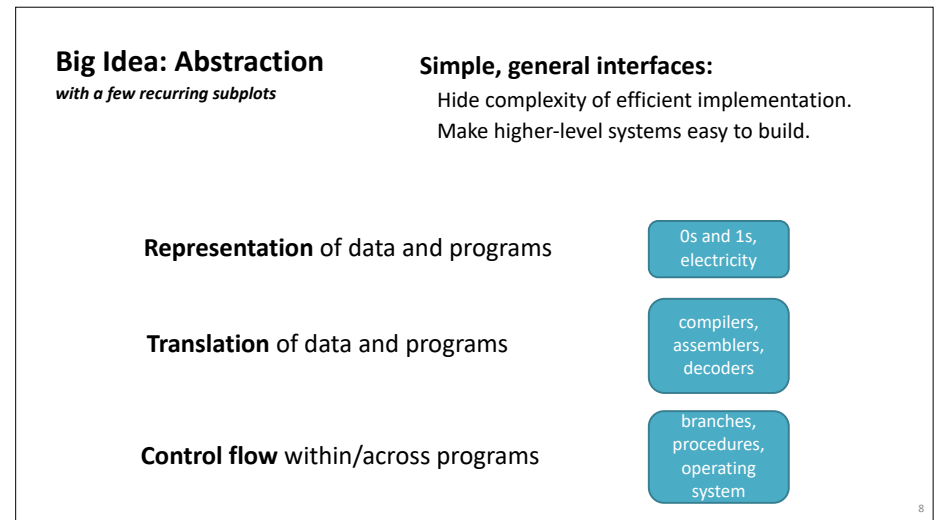
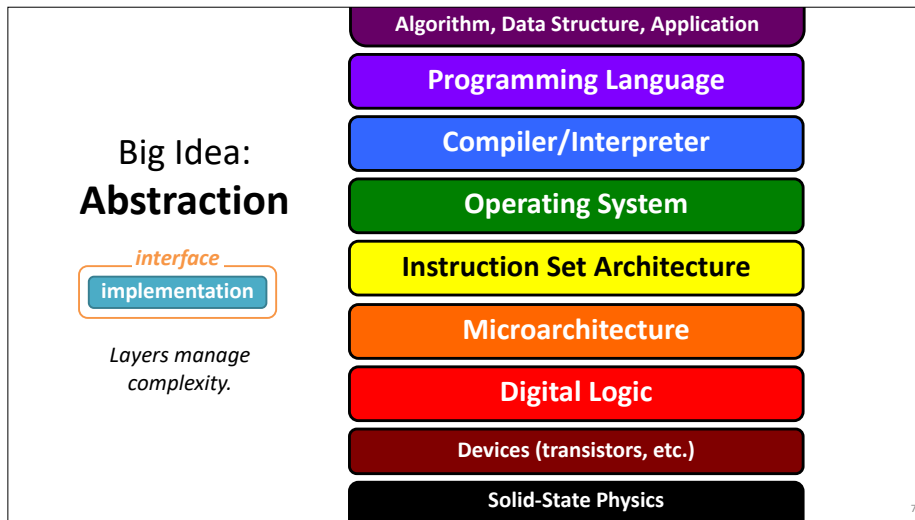
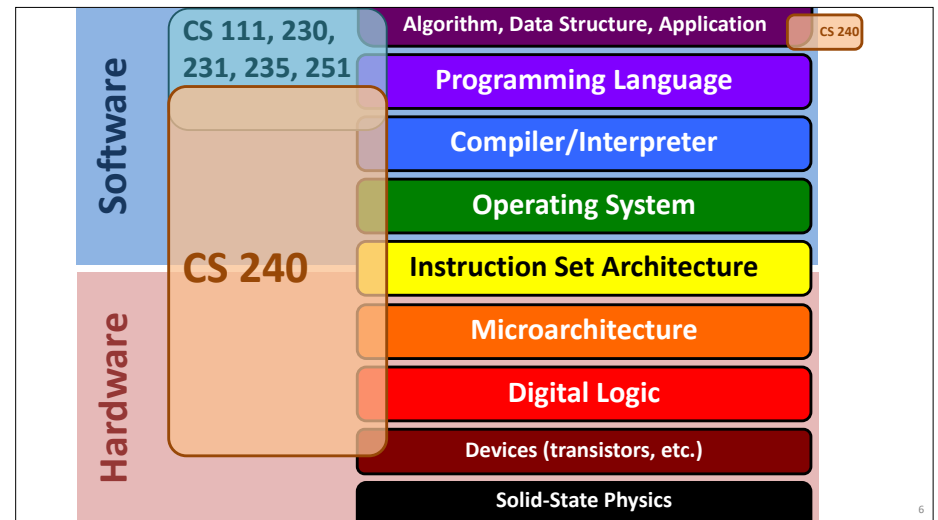
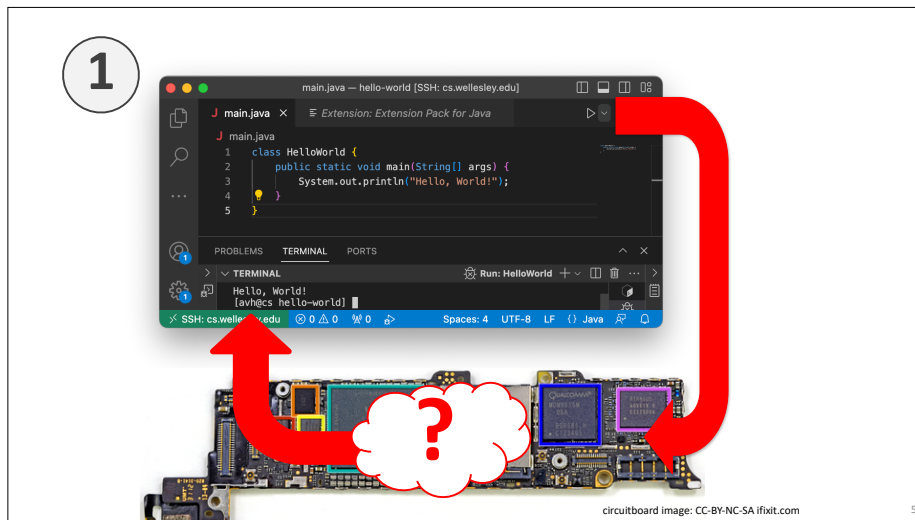
3

CS 111, 230, 231, 235, 251:

- How do you use programming to solve a problem?
- How do you structure a program?
- How do you know it is correct or efficient?
- How hard is it to solve a problem?
- How is computation expressed?
- What does a program mean?
- ...

A BIG question is missing...

4



CS 240 in 3 acts (4-5 weeks each)

1. Hardware implementation

From transistors to a simple computer



2. Hardware-software interface

From instruction set architecture to programming in C

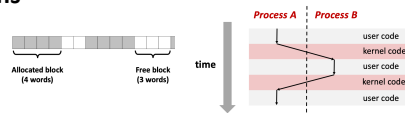
```
MOV x9, x10
ADD x12, x12, #1
*x = malloc (...);
```

3. Abstraction for practical systems

Memory hierarchy

Operating system basics

Higher-level languages and tools



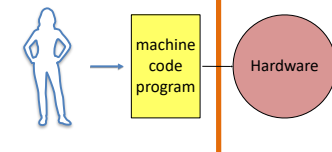
9

Machine Instructions

(adds two values and stores the result)

00000010100010101100100000010000

Instruction Set Architecture specification

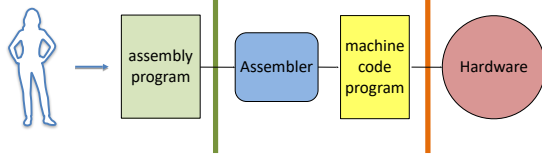


10

Assemblers and Assembly Languages

addl %eax, %ecx → 00000010100010101100100000010000

Assembly Language specification

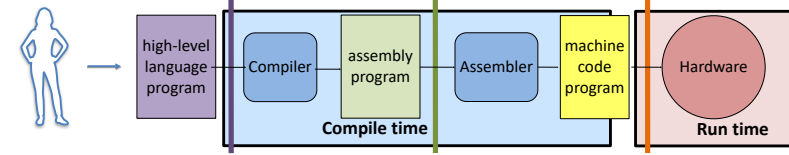


11

Higher-Level Programming Languages

x = x + y;
↓
addl %eax, %ecx → 00000010100010101100100000010000

Programming Language specification



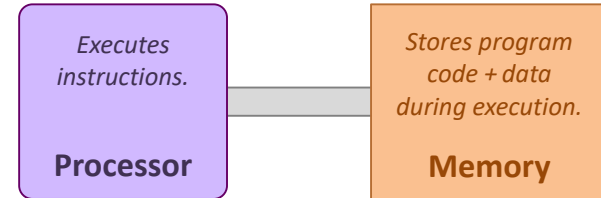
12

More and more layers...

- Operating systems
- Virtual machines
- Hypervisors
- Web browsers
- ...

13

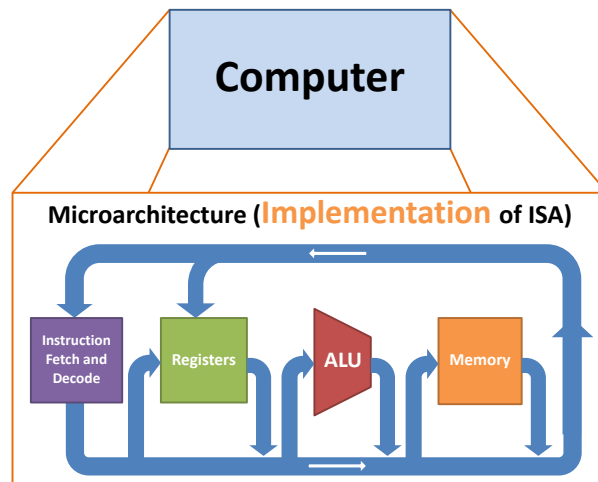
Modern Computer Organization



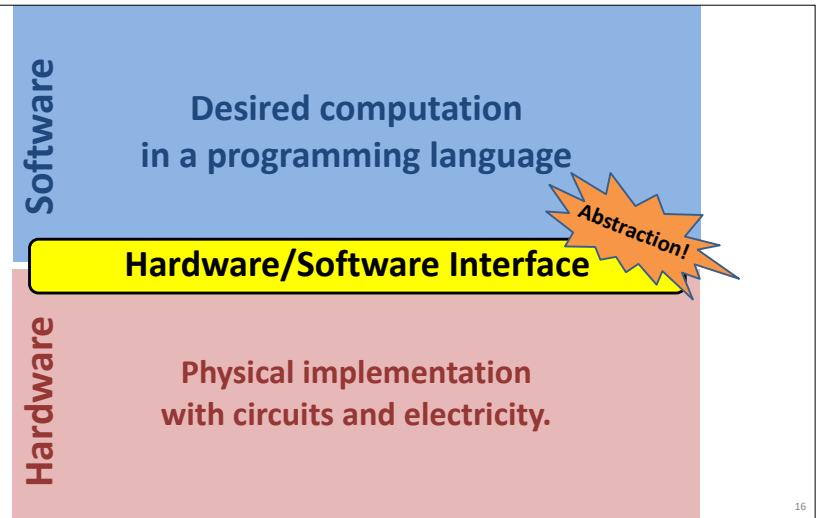
Processor repeats:

1. fetch instruction
2. fetch data used by instruction
3. execute instruction on data
4. store result or choose next instruction

14

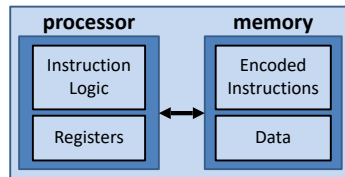


15



16

Instruction Set Architecture (HW/SW Interface)



Computer

17

Why take CS 240?

2

I just like to program.

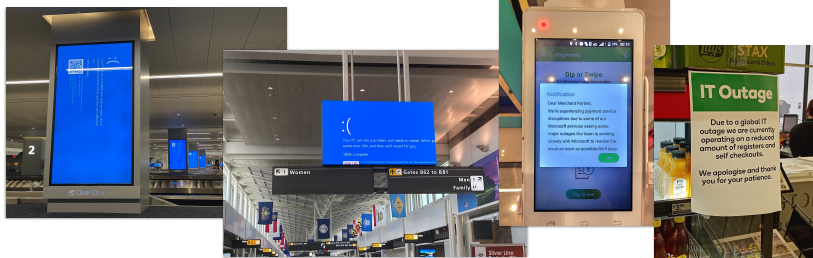
Why study the implementation?

18



Does anyone remember what was noteworthy about July 19, 2024?

...was anyone trying to travel by plane around then?



What happened?

invalid memory access in C
unchecked array length

code running in OS kernel
limitations of processor multithreading

insufficient testing & validation

... all CS240 topics!

19

2

I just like to program.

Why study the implementation?

Abstraction!

Most system abstractions "leak."

Implementation details affect your programs:

Their performance



Their correctness



Their security



20

Performance



x / 973

x / 1024

```
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];
}

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];
}
```

several times faster
due to hardware caches

21

Correctness



int ≠ integer
float ≠ real

Exploded due to **cast** of
64-bit floating-point number
to 16-bit signed number.
Overflow.



Boeing 787, 2015



"... a **Model 787 airplane** ... can lose all
alternating current (AC) electrical power ...
caused by a **software counter** internal to the
GCUs that will **overflow** after **248 days** of
continuous power. We are issuing this AD to
prevent loss of all AC electrical power, which
could result in **loss of control of the airplane.**"
--FAA, April 2015

22

Security



The **GHOST vulnerability** is a buffer overflow condition that can be easily exploited locally and remotely, which makes it extremely dangerous. This vulnerability is named after the **GethGOSTbyname** function involved in the exploit.

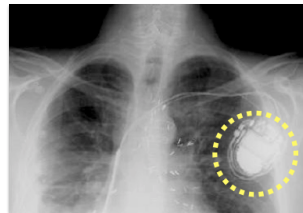


Cyber-Safe
All computers are flawed -- and the fix will
take years

By **SHAWN L. LITTON** - **SHAWN LITTON**

January 26, 2016 10:17 AM EST

Meltdown and
Spectre



23

Why take CS 240?

Learn **how** computers execute programs.
Deepen your appreciation of **abstraction**.
Improve your **critical thinking** skills.

Become a **better programmer**:

Think rigorously about execution models.
Identify limits and impacts of abstractions and representations.
Learn to use software development tools.

Foundations for:

Compilers, security, computer architecture, operating systems, ...

Have fun and feel accomplished!

24



3

<https://cs.wellesley.edu/~cs240/>

All details about the course.
Please read syllabus (About link) and
schedule (Calendar link) before Friday's
lecture and ask questions then.



4

Let's start learning about Digital Logic!
(separate slide deck)