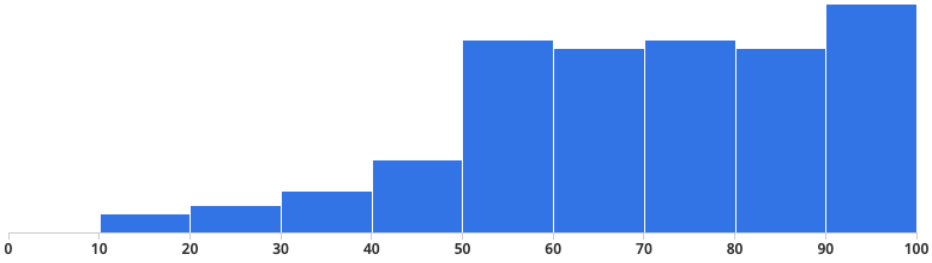
Taking over the World!

CS 2130: Computer Systems and Organization 1
February 25, 2026

Announcements

- Homework 4 **due Monday after break** on Gradescope
  - You have written most of this code already
  - Hint: Lab 7 may provide a fast way to get started

Exam 1



Median	Maximum	Mean	Std Dev
72.0	102.0	70.07	19.93

ToyISA Instructions

icode	b	meaning
0		`rA = rB`
1		`rA &= rB`
2		`rA += rB`
3	0	`rA = ~rA`
	1	`rA = !rA`
	2	`rA = -rA`
	3	`rA = pc`
4		`rA` = read from memory at address `rB`
5		write `rA` to memory at address `rB`
6	0	`rA` = read from memory at `pc + 1`
	1	`rA` &= read from memory at `pc + 1`
	2	`rA` += read from memory at `pc + 1`
	3	`rA` = read from memory at the address stored at `pc + 1`
		For icode 6, increase `pc` by 2 at end of instruction
7		Compare `rA` as 8-bit 2's-complement to 0 if `rA <= 0` set `pc = rB` else increment `pc` as normal

Storing Variables in Memory

So far... we/compiler chose location for variable
Consider the following example:

```
f(x):
    a = x
    if (x <= 0) return 0
    else return f(x-1) + a
```

Recursion

- The formal study of a function that calls itself

Storing Variables in Memory

```
f(x):
    a = x
    if (x <= 0) return 0
    else return f(x-1) + a
```

Where do we store `a`?

The Stack

Stack - a last-in-first-out (LIFO) data structure

- *The* solution for solving this problem

`rsp` - Special register - the *stack pointer*

- Points to a special location in memory
- Two operations most ISAs support:
  - `push` - put a new value on the stack
  - `pop` - return the top value off the stack

The Stack: Push and Pop

`push r0`

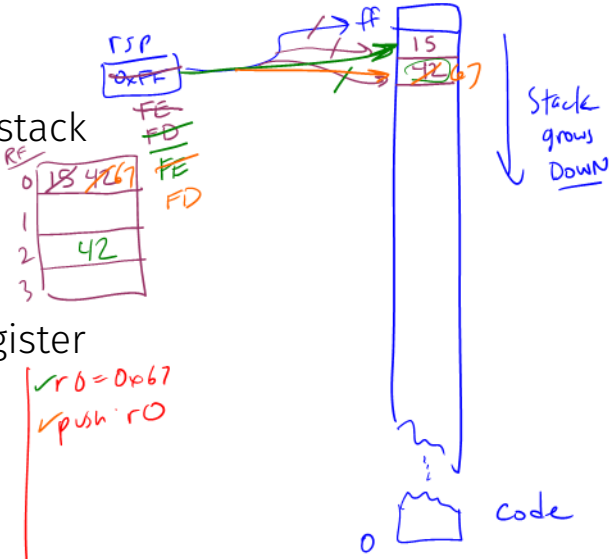
- Put a value onto the "top" of the stack

```
rsp -= 1
M[rsp] = r0
```

`pop r2`

- Read value from "top", save to register

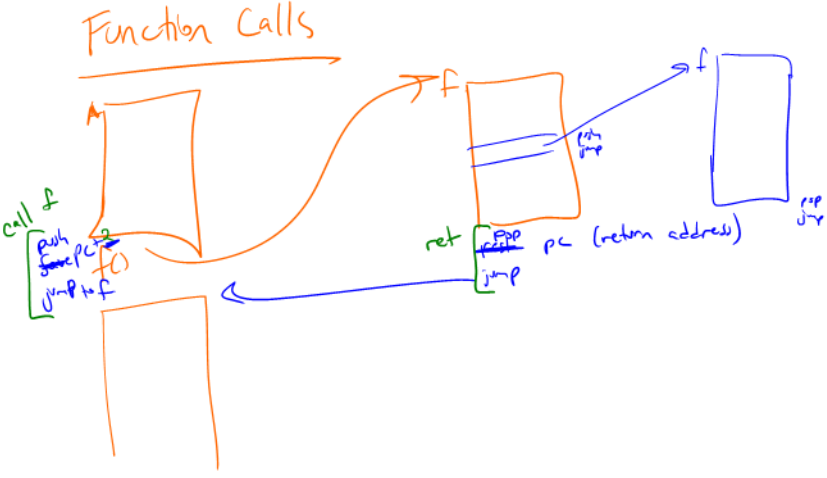
```
r2 = M[rsp]
rsp += 1
```



r0 = 0x15
push r0
r0 = 0x42
push r0
pop r2

r0 = 0x67
push r0

The Stack: Push and Pop



- parameters: r_2, r_3
- return value: r_0
- "not touch": r_1

ToyISA Instructions

icode	b	meaning
0		`rA = rB`
1		`rA &= rB`
2		`rA += rB`
3	0	`rA = ~rA`
	1	`rA = !rA`
	2	`rA = -rA`
	3	`rA = pc`
4		`rA` = read from memory at address `rB`
5		write `rA` to memory at address `rB`
6	0	`rA` = read from memory at `pc + 1`
	1	`rA` &= read from memory at `pc + 1`
	2	`rA` += read from memory at `pc + 1`
	3	`rA` = read from memory at the address stored at `pc + 1`
		For icode 6, increase `pc` by 2 at end of instruction
7		Compare `rA` as 8-bit 2's-complement to 0 if `rA <= 0` set `pc = rB` else increment `pc` as normal

A short aside...

Time to take over the world!

Backdoors

Backdoor: secret way in to do new *unexpected* things

- Get around the normal barriers of behavior
- Ex: a way in to allow me to take complete control of your computer

Vulnerability: flaws or weaknesses in design or implementation

Backdoors

Exploit - a way to use a vulnerability or backdoor that has been created

- Our exploit today: a **malicious payload**
  - A passcode and program
  - If it ever gets in memory, run my program regardless of what you want to do

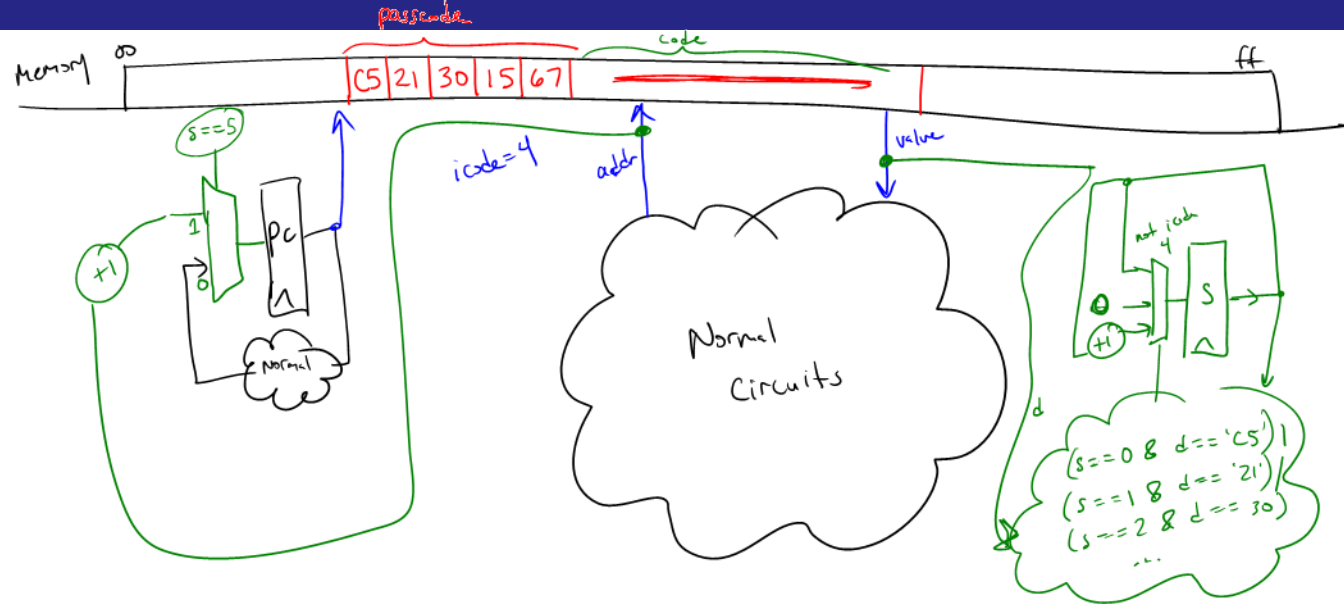
Our Hardware Backdoor

Our exploit will have 2 components

- Passcode: need to recognize when we see the passcode
- Program: do something bad when I see the passcode

Backdoor will recognize and run

Our Hardware Backdoor



Our Hardware Backdoor

Will you notice this on your chip?

Our Hardware Backdoor

Will you notice this on your chip?

- Modern chips have **billions** of transistors
- We're talking adding a few hundred transistors

Our Hardware Backdoor

Will you notice this on your chip?

- Modern chips have **billions** of transistors
- We're talking adding a few hundred transistors
- *Maybe with a microscope? But you'd need to know where to look!*

Our Hardware Backdoor

Have you heard about something like this before?

Our Hardware Backdoor

Have you heard about something like this before?

- Sounds like something from the movies

Our Hardware Backdoor

Have you heard about something like this before?

- Sounds like something from the movies
- People claim this might be happening

Our Hardware Backdoor

Have you heard about something like this before?

- Sounds like something from the movies
- People claim this might be happening
- To the best of my knowledge, no one has ever *admitted* to falling in this trap