CS 208
Fri, 5 May 2023
JMP

rip

instruction pointer
jmp changes the value of rip
jmpq 0x420000(,%rax,8)

\[
0x400000 + \%rax*8
\]
lea vs. mov

lea: computes some number and puts it in its destination

mov: copies data from here to there
Addressing modes: indirect with offset

```
lea 0x8(%rsp), %rbx
rbx = 0x8 + rsp
mov 0x8(%rsp), %rbx
```

- go to mem. addr 8 + rsp
- get the 8-byte block there
- put it in r$bx
Function calling

caller (e.g., phase4)
function (e.g., fun4)

caller:
- set up parameter(s)
  - call 9

function:
- make space for local vars on stack
- save old values of registers that caller might be using

Func4(le, user's 3rd #)

How does function know where to return to?
Function returning
- Tear down stack
- Restore saved registers
- Pop return address
  \[ \text{add some value to rsp} \]
  \[ \text{pop} \]
  \[ \text{return address} \]
  \[ \text{jmp there} \]
  \[ \text{retg} \]
  \[ \text{modifies rsp & rip} \]