CS 208
Wed
2 March 2022
Call a function
- Put parameters where the fn expects them
  (common: rdi, rsi, ...) (too big? stack)
- Call g
  pushes return addr on stack
  rsp = rsp - 8
  jump to function
Function f just got called

- Subtract from rsp to make stack frame for local vars
- Push regs f plans to use, so we can save their values
Return From Function

- Pop the used registers so orig values are back
- Add to rsp to "tear down" the stack frame
- Put return value where the caller expects it (commonly: rax)
- Retg — Pop return addr \((\text{rsp} + 8)\) + Jump there