Today

- Function (procedure) calls

Assignments

- HW6 out (due Thurs)
- Quiz 6
Implementing Procedures in Assembly

Things we need to consider: Assume P calls Q, Q executes and returns to P

Passing Control:
- program counter set to starting address of Q ... the call
- then set to return to next instruction in P ... the return

Passing Data:
- P must provide parameter values to Q ... the call
- Q can return a value to P ... the return

Local Data:
- Q needs to store local vars ... e.g., via stack and/or registers

⇒ x86-64 has special instructions and conventions for supporting procedures
  - we’ll look at instructions
  - then move on to conventions
calls Label

- Label is address of procedure to jump to (e.g., call Q)
- pushes return address A onto stack (next instruction after call)
- program counter is set to address of Q's first instruction

P: ...
call Q  # pushes A, jumps to Q
...  

Q: ...  # body of Q

rets

- returns control to caller
- pops address A from stack and jumps to A

P: ...
call Q  # pushes A, jumps to Q
...  

Q: ...  # body of Q
ret  # pops A, jumps to P (after call)

Notes:

- s denotes size of address, which is assumed / must be q in x86-64
- Since call/ret push/pop, care needed when passing values via stack