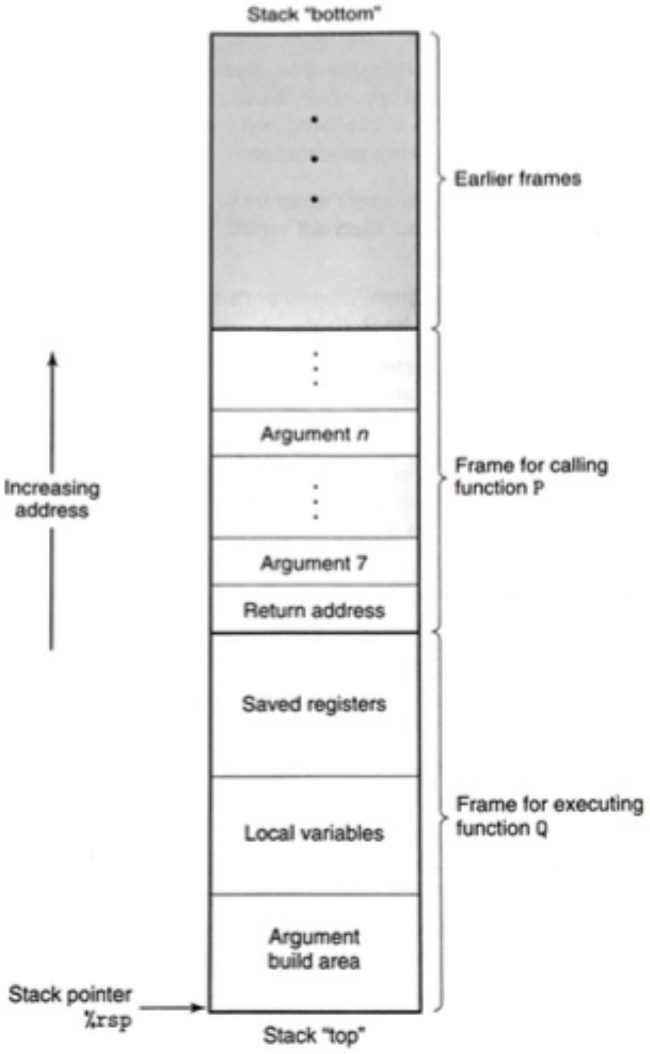
Chapter 3.7: Procedures

- procedures provide a way to package code that implements some functionality with a designated set of arguments and an optional return value
- there are many attributes that must be handled for machine-level support for procedures
- Suppose P calls procedure Q, and Q executes and returns back to P
    - *Passing control.* The program counter must be set to the starting address of the code for Q upon entry and then set to the instruction in P following the call to Q upon return.
    - *Passing data*. P must be able to provide one or more parameters to Q, and Q must be able to return a value back to P.
    - *Allocating and deallocating memory*. Q may need to allocate space for local variables when it begins and then free that storage before it returns.

3.7.1: The Run-Time Stack



- when P calls Q, control and data info are added to the end of the stack
    - this info gets deallocated when P returns
- stack grows toward lower addresses
- the stack pointer `%rsp` points to the top element of the stack
- when an x86-64 procedure requires torage beyond register capacity, it allocates space on the stack
    - this allocated space is referred to as the procedure’s **stack frame**
- when procedure P calls procedure Q, it will push the *return address* onto the stack, indicating where in P the program should resume execution once Q returns
    - this info is considered to be in P’s stack frame, since it is relevant to procedure P
- stack frames for most procedures are of fixed size, allocated at the beginning of the procedure
- some procedures don’t require a stack frame because they can hold all the variables and arguments in the registers
    - these are called **leaf procedures**

3.7.2 Control Transfer

- when called in P, the instruction `callq` pushes an address **A** onto the stack and sets the PC to the beginning of Q
    - **A** is referred to as the *return address* and is computed as the address of the instruction immediately following the `callq` instruction
- the counterpart instruction, `ret`, pops an address **A** off the stack and sets the PC to **A**