

# The Hardware/Software Interface

## Section 4: Procedure calls

# Homework 1 Debrief

- **Double check your stuff**
- **Show your work**
- **Question 2 was very tricky**
  - int to float/double conversion
- **Question 3 had lots of issues**
  - Don't compare floating-point values with ==
- **Many people didn't attempt the EC**

# Lab 2 Questions?

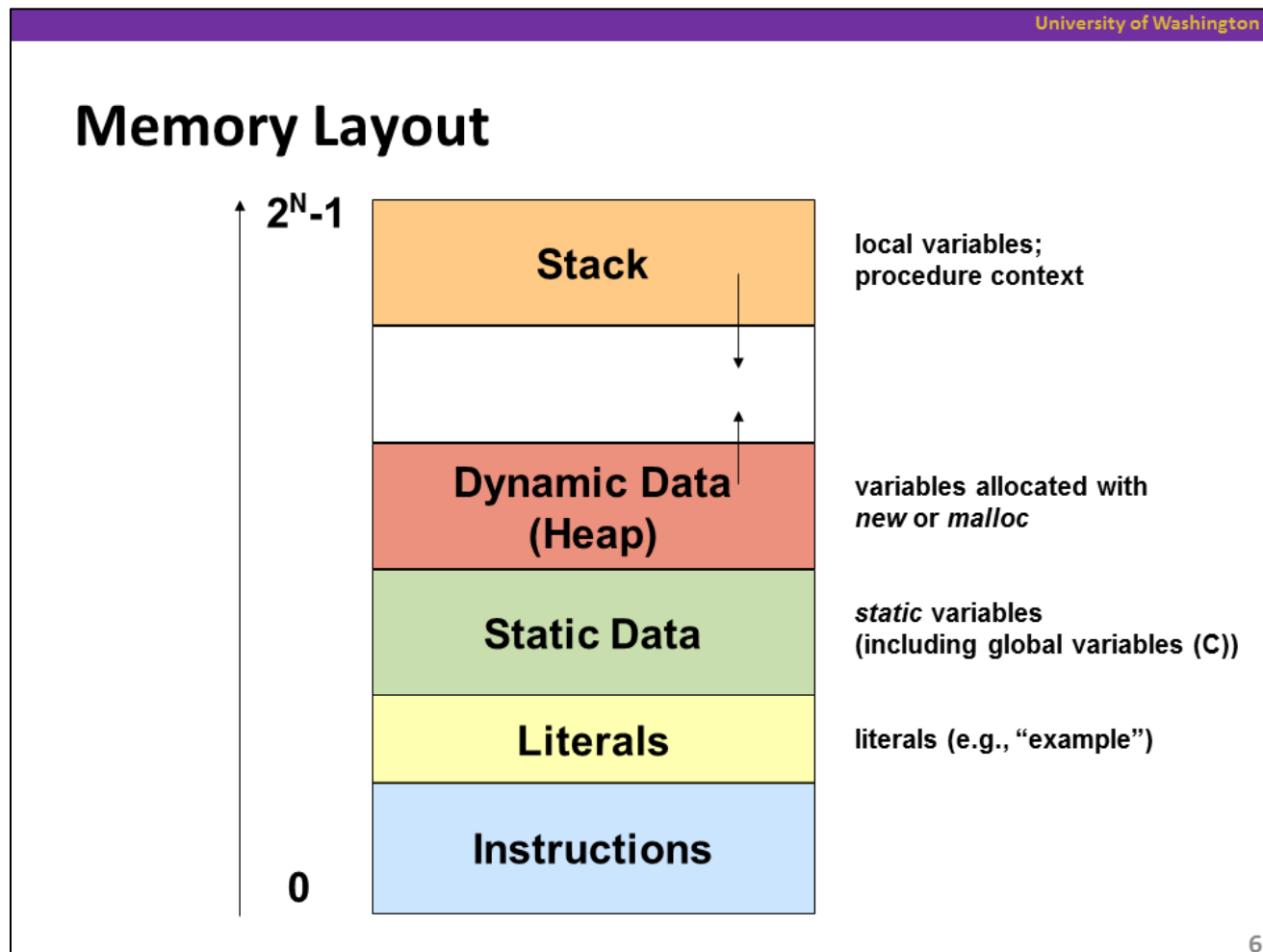
# Procedure Calls

- **X86 vs x86-64**

- But before those differences, we should review....

# The Stack

## ■ Recall:

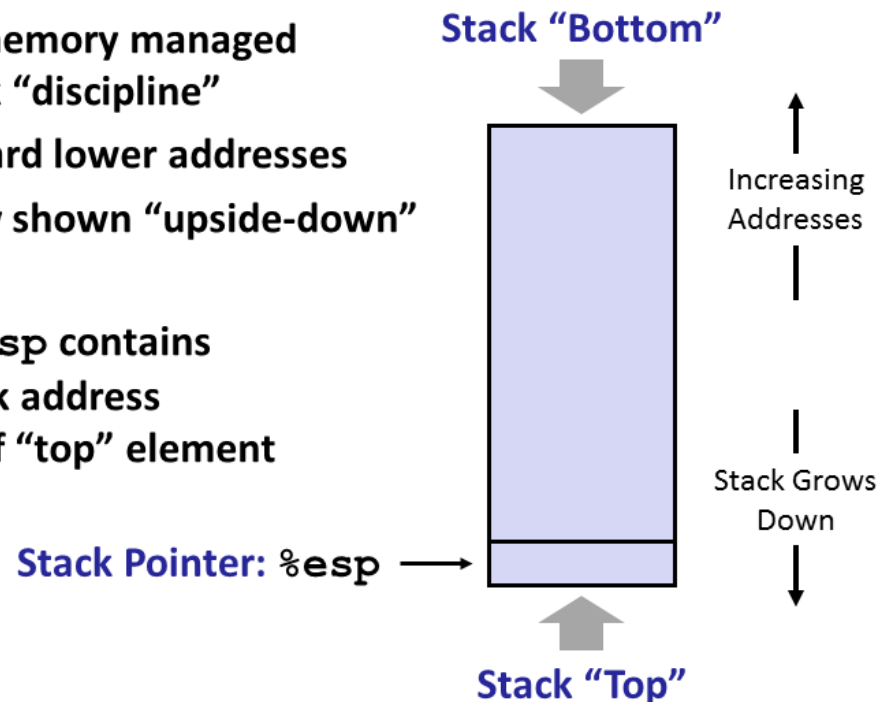


# The Stack

## ■ Recall:

### IA32 Call Stack

- Region of memory managed with a stack “discipline”
- Grows toward lower addresses
- Customarily shown “upside-down”
- Register `%esp` contains lowest stack address = address of “top” element



# The Stack

## ■ Definitely recall (Also textbook page 220):

University of Washington

### IA32/Linux Stack Frame

- **Current Stack Frame (“Top” to Bottom)**
  - “Argument build” area (parameters for function about to be called)
  - Local variables (if can’t be kept in registers)
  - Saved register context (when reusing registers)
  - Old frame pointer (for caller)
- **Caller’s Stack Frame**
  - Return address
    - Pushed by `call` instruction
  - Arguments for this call

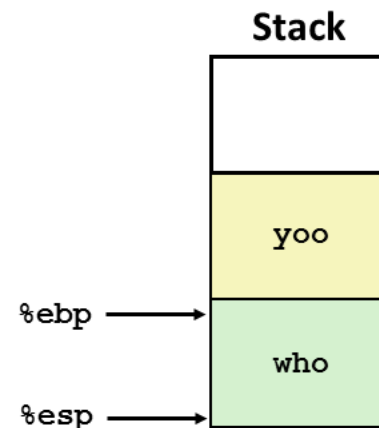
The diagram illustrates the stack frame layout. The stack grows downwards. The current frame (top) contains the argument build area, local variables, saved register context, and the old frame pointer. The caller's frame (bottom) contains the return address and arguments. The frame pointer `%ebp` points to the old frame pointer, and the stack pointer `%esp` points to the bottom of the current frame.

41

# x86 Conventions

- Push arguments onto the stack
- Caller-save registers
  - %eax, %ecx, %edx
- Callee-save registers
  - %ebx, %edi, %esi
- Special
  - %ebp, %esp
- Return value in %eax

University of Washington





# x86-64 Conventions

- **Store arguments in registers**
  - %rdi, %rsi, %rdx, %rcx, %r8, %r9
  - If you run out, then push onto stack
- **Caller-save registers**
  - %rax, %r10, %r11
- **Callee-save registers**
  - %rbx, %r12-%r15
- **Special**
  - %rbp, %rsp
- **Return value in %rax**

# Let's take a look