CSE 351 Section 4

Program Stack & Procedure Calls

Bomb Lab!

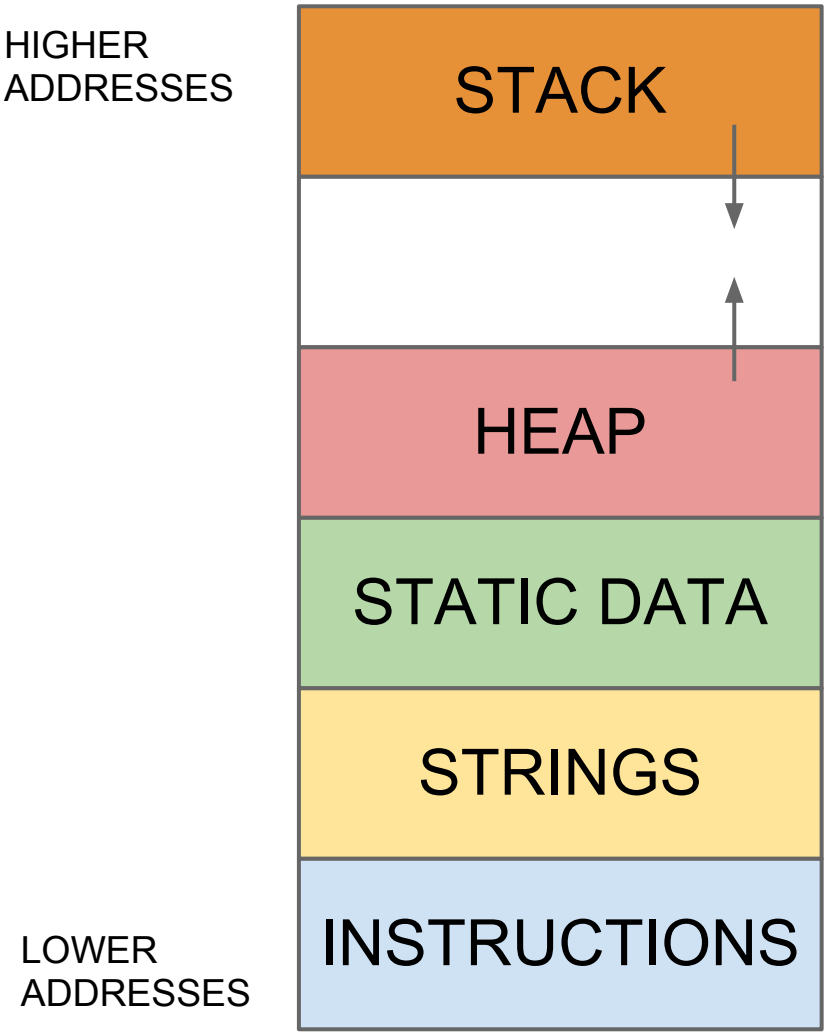
- How is it going?

Bomb-defusing tips

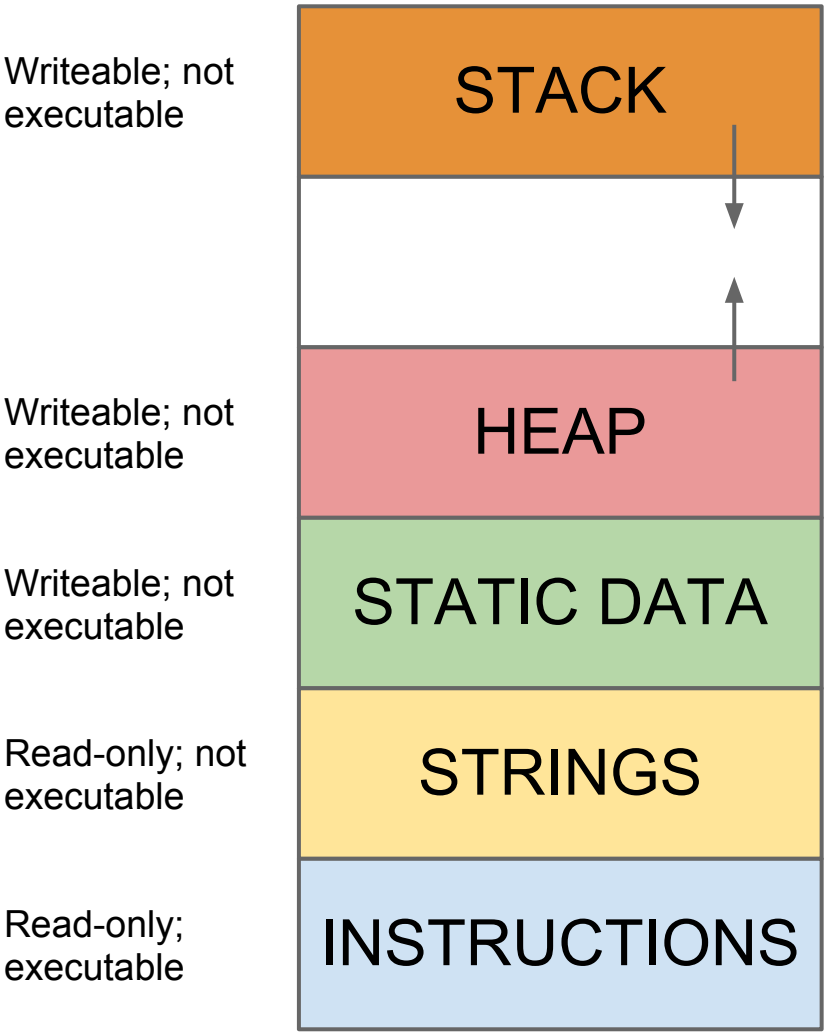
- If you have trouble figuring out what a phase is doing, try working backwards
- Write out everything that happens, and you might be able to look back and see a pattern
- Try to understand what the phase is doing based on assembly code, then predict variable values and check your understanding in gdb

Memory Layout

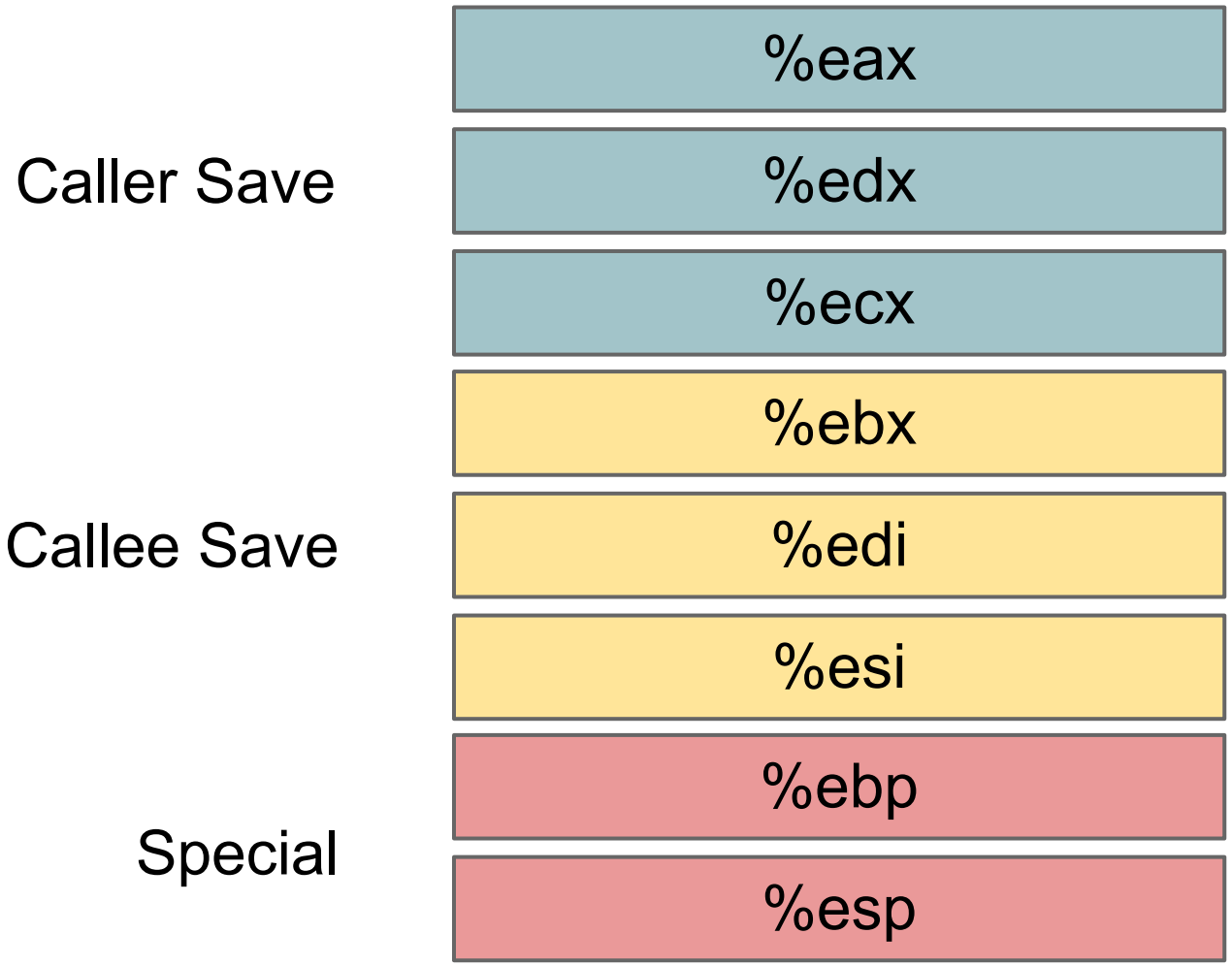
Program Memory Layout



Program Memory Layout



IA32/Linux Register Usage



x86-64 Register Usage

%rax	Return Value	%r8	Argument #5
%rbx	Callee Saved	%r9	Argument #6
%rcx	Argument #4	%r10	Caller Saved
%rdx	Argument #3	%r11	Caller Saved
%rsi	Argument #2	%r12	Callee Saved
%rdi	Argument #1	%r13	Callee Saved
%rsp	Stack Pointer	%r14	Callee Saved
%rbp	Callee Saved	%r15	Callee Saved

Demos

Download these demos

Multiple parameters demo

http://www.cs.washington.edu/education/courses/cse351/12au/section-slides/multi-param.c

wget http://www.cs.washington.edu/education/courses/cse351/12au/section-slides/multi-param.c

Recursive stack frame demo

http://www.cs.washington.edu/education/courses/cse351/12au/section-slides/fact_check.c

wget http://www.cs.washington.edu/education/courses/cse351/12au/section-slides/fact_check.c

Multiple Parameters Demo

Demo commands:

```
gcc -g -m64 multi-param.c -o multi-param64
gcc -g -m32 multi-param.c -o multi-param32

objdump -d multi-param64 | less
objdump -d multi-param32 | less

gdb multi-param64
```

GDB commands:

```
break addEight, run, disas, info registers
```

Recursive Stack Frame Demo

Demo commands:

```
gcc -g fact_check.c -o fact_check

objdump -d fact_check.c | less

gdb fact_check
```

GDB commands:

```
break factorial, run, disas, info frame,
x /20x $esp
```