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

- Stack
- Heap
- Static Data
- Strings
- Instructions
Program Memory Layout

- **STACK**: Writeable; not executable
- **HEAP**: Writeable; not executable
- **STATIC DATA**: Writeable; not executable
- **STRINGS**: Read-only; not executable
- **INSTRUCTIONS**: Read-only; executable
IA32/Linux Register Usage

- Caller Save:
  - %eax
  - %edx
  - %ecx
- Callee Save:
  - %ebx
  - %edi
  - %esi
- Special:
  - %ebp
  - %esp
### x86-64 Register Usage

<table>
<thead>
<tr>
<th>Register</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>%rax</td>
<td>Return Value</td>
</tr>
<tr>
<td>%rbx</td>
<td>Callee Saved</td>
</tr>
<tr>
<td>%rcx</td>
<td>Argument #4</td>
</tr>
<tr>
<td>%rdx</td>
<td>Argument #3</td>
</tr>
<tr>
<td>%rsi</td>
<td>Argument #2</td>
</tr>
<tr>
<td>%rdi</td>
<td>Argument #1</td>
</tr>
<tr>
<td>%rsp</td>
<td>Stack Pointer</td>
</tr>
<tr>
<td>%rbp</td>
<td>Callee Saved</td>
</tr>
<tr>
<td>%r8</td>
<td>Argument #5</td>
</tr>
<tr>
<td>%r9</td>
<td>Argument #6</td>
</tr>
<tr>
<td>%r10</td>
<td>Caller Saved</td>
</tr>
<tr>
<td>%r11</td>
<td>Caller Saved</td>
</tr>
<tr>
<td>%r12</td>
<td>Callee Saved</td>
</tr>
<tr>
<td>%r13</td>
<td>Callee Saved</td>
</tr>
<tr>
<td>%r14</td>
<td>Callee Saved</td>
</tr>
<tr>
<td>%r15</td>
<td>Callee Saved</td>
</tr>
</tbody>
</table>
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:

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

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

```sh
gdb multi-param64
```

GDB commands:

```sh
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