Class exercise: Analyzing conditional control flow

COMP 40

November 4, 2009

Group

<table>
<thead>
<tr>
<th>Keeper of the record:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Other group members:</td>
</tr>
</tbody>
</table>

Conditional control flow

Here is assembly code for procedure classphase:

```
0000000000400590 <classphase>:
400590: 48 83 ec 08 sub $0x8,%rsp
400594: 85 ff test %edi,%edi
400596: 74 13 je 4005ab <classphase+0x1b>
400598: 85 f6 test %esi,%esi
40059a: 74 0f je 4005ab <classphase+0x1b>
40059c: 8d 04 3e lea (%rsi,%rdi,1),%eax
40059f: 39 d0 cmp %edx,%eax
4005a1: 75 08 jne 4005ab <classphase+0x1b>
4005a3: 89 f8 mov %edi,%eax
4005a5: 29 f0 sub %esi,%eax
4005a7: 39 c8 cmp %ecx,%eax
4005a9: 74 05 je 4005b0 <classphase+0x20>
4005ab: e8 c8 ff ff ff callq 400578 <explode_bomb>
4005b0: 48 83 c4 08 add $0x8,%rsp
4005b4:  c3 retq
```

The `test` instruction is to and as `cmp` is to sub: it does a bitwise and of the two arguments, throws away the result, but sets the flags SF (sign bit) and ZF (result equal to zero). The instruction

```
lea (%rsi,%rdi,1),%eax
```

is load effective address; it is a popular way of doing arithmetic without tying up the integer unit and without touching flags. This instance is the same as

```
%eax = %rsi + 1 * %rdi
```

1. How many parameters does `classphase` expect, and of what types?
2. How do you know?
3. What values will keep the bomb from blowing up?

Please return your work to the course staff.