Code Jam 4

Names:

Complete the following exercises in 40 minutes. This activity is open book, open computer. All work should be your own group.

Question 1

Short answer

1) How is machine code generated from source code? What is the relationship between machine code and assembly?

2) What is a register?

3) What two steps occur when the CPU executes the instruction `push %rbp` ?

4) What two steps occur when the CPU executes the instruction `pop %rbp` ?

5) What does `callq` do?

Question 2

Consider the following code segment.

```
Dump of assembler code for function main:
=> 0x0000555555555149 <+0>:     endbr64
   0x000055555555514d <+4>:     push   %rbp
   0x000055555555514e <+5>:     mov    %rsp,%rbp
   0x0000555555555151 <+8>:     sub    $0x10,%rsp
   0x0000555555555155 <+12>:    movl   $0xfffffff7,-0x8(%rbp)
   0x000055555555515c <+19>:    mov    -0x8(%rbp),%eax
   0x000055555555515f <+22>:    shl    $0x2,%eax
   0x0000555555555162 <+25>:    mov    %eax,-0x4(%rbp)
   0x0000555555555165 <+28>:    mov    -0x4(%rbp),%edx
   0x0000555555555168 <+31>:    mov    -0x8(%rbp),%eax
   0x000055555555516b <+34>:    mov    %eax,%esi
   0x000055555555516d <+36>:    lea    0xe90(%rip),%rdi        # 0x555555556004
   0x0000555555555174 <+43>:    mov    $0x0,%eax
   0x0000555555555179 <+48>:    callq  0x555555555050 <printf@plt>
   0x000055555555517e <+53>:    mov    $0x0,%eax
   0x0000555555555183 <+58>:    leaveq
   0x0000555555555184 <+59>:    retq
```

Suppose memory has the following values:

Address	Value
M[0x555555556004]	"%d %d"

And suppose our registers have the following values to start:

Register	Value
%eax	0xd0d
%edx	0xe148
%rbp	0x0
%rsp	0x048
%esi	0xe138
%edi	0x1
%rip	0x0000555555555149

1) What is the value 0x10 as a base 10 integer?

2) What is the value 0xfffffff7 as a base 10 *signed* integer (two's complement)?

3) Draw the contents of the registers and stack after executing the instruction `sub $0x10,%rsp` (0x0000555555555151)

Register	Value
%eax	
%edx	
%rbp	
%rsp	
%esi	
%edi	
%rip	

"Stack top"

Address	Stack value
0x048	

4) What is the translation of the memory form `-0x8(%rbp)` ?

5) What is the `shl` instruction?

6) What are the contents of %eax after executing instruction 0x000055555555515c?

7) What are the contents of %eax after executing instruction 0x000055555555515f?

8) What are the contents of %rsi after executing instruction 0x000055555555516b?

9) What are the contents of %rdi after executing instruction 0x000055555555516d?

Question 3

In this question use GDB and objdump to reverse engineer to binary executable `secret`

```
$ ./secret
Guess the mysterious number: 39
You are wrong!
```

1) What functions does `main` call (please list in execution order)?

2) What are the values of %esi and %edi before the first function call in main?

3) What is the return value from the first function?

4) What is the value of %edi before the second function call in main?

5) What is the secret number?

Last updated 2022-03-30 14:17:29 -0400