CS170 – Spring 2017 — Homework 1

Ninh DO, SID 25949105, `cs161-aqz`

1. Policy

Please read and check that you understand the course policies on that page. If you have any questions, please ask for clarification on Piazza. To receive credit for having read the policies, submit the following statement as your answer for Q1: “I understand the course policies.”

Solution I understand the course policies.

2. Policy

You’re working on a course project. Your code isn’t working, and you can’t figure out why not. Is it OK to show another student (who is not your project partner) your draft code and ask them if they have any idea why your code is broken or any suggestions for how to debug it?

Solution NO

3. Hidden Backdoor

We have hidden a secret password (a “backdoor”) on the web page you visited in Question 1. If you entered the URL correctly as above (substituting your last name and userid), you’ll be able to access the password - if you know the trick.

Solution l33tskillzTQS

4. Feedback

Optionally, feel free to include feedback. What’s the single thing we could do to make the class better? Or, what did you find most difficult or confusing from lectures or the rest of class, and what would you like to see explained better? If you have feedback, submit your comments as your answer to Q4.

Solution Please do not make GSI’s office hours overlap. There are many time slots available. The overlapping office hours waste your time while students do not benefit from it.

5. Memory Layout

Consider the following C code:

```
 1 int dog(int i, int j, int k) {
 2     int t = i + k;
 3     char *newbuf = malloc(4);
 4     t = t + 4;
 5     t = t - j;
 6     return t;
 7 }
 8
 9 void cat(char *buffer) {
10     int i[2];
11     i[1] = 5;
12     i[0] = dog(1, 2, i[1]);
13 }
14
15 int main() {
16     char buf[8];
17     cat(buf);
18     return 0;
19 }
```

The code is compiled and run on a 32-bit x86 architecture (i.e., IA-32). Assume the program is run until line 6, meaning everything before line 6 is executed (i.e., a breakpoint was set at line 6). We want you to sketch what the layout of the program's stack looks like at this point. In particular, print the template provided on the next page and fill it in. Fill in each empty box with the value in memory at that location. Put down specific values in memory, like 1 or 0x00000000, instead of symbolic names, like buf. Also, on the bottom, fill in the values of %ebp and %esp when we hit line 6.

Assumptions you should make:

- memory is initially all zeros
- execution starts at the very first instruction during the usual invocation of main(), and
- the call to malloc returns the value 0x12341234
- the address of the code corresponding to line 4 is 0x01111180
- the address of the code corresponding to line 13 is 0x01111134
- the address of the code corresponding to line 18 is 0x01111100
- a char is 1 byte, an int is 4 bytes
- no function uses general-purpose registers that need to be saved (other than %ebp)

Solution

0xa0000060:	0xa0000064
0xa000005c:	0x00000000
0xa0000058:	0x00000000
0xa0000054:	0xa000005c
0xa0000050:	0x01111100
0xa000004c:	0xa0000060
0xa0000048:	0x00000000
0xa0000044:	5
0xa0000040:	5
0xa000003c:	2
0xa0000038:	1
0xa0000034:	0x01111134
0xa0000030:	0xa000004c
0xa000002c:	8
0xa0000028:	0x12341234
0xa0000024:	
0xa0000020:	
0xa000001c:	
0xa0000018:	
0xa0000014:	

⋮

%ebp $= 0xa0000030$

%esp $= 0xa0000028$