

Homework 1: Numerical Representations and Object Files

Assigned	Friday, April 12, 2013
Due Date	<i>Friday, April 19, 2013 at 17:00</i>

Introduction

The purpose of written homework assignments is to get you thinking about the topics being covered in lecture and in readings in the textbook which are not represented in the hands-on, programming lab assignments. These written assignments also better prepare you for course examinations. It is worth noting that the book contains many practice problems similar to the problems we ask on these written assignments. The solutions for those practice problems are located at the end of each chapter and should give you a feel for the kind of answers we expect you to turn in for these kind of assignments.

Logistics

Since these written homeworks are not programming lab assignments, you will turn in your assignment online as electronic documents in PDF format. We will not accept any paper submissions.

Grading

Due to the large number of students in the class, we reserve the right to grade only a subset of the problems assigned in any written homework assignment for correctness. However, we expect you to answer all of the questions in an assignment and therefore will take that into account when calculating your grade for an assignment. For example, if we choose only to grade the 1st and 4th

questions in this assignment, you would receive full credit for the assignment by answering these problems correctly. However, if you were not to answer all of the questions in an assignment, we would scale your score by a factor of $\text{num_questions_answered} / \text{num_questions_assigned}$. So in this example, if you only submitted answers to the 1st, 3rd and 4th questions, you would receive only 3/4 of your score. In conclusion, try your best and be sure to answer all questions in each written homework assignment.

We will provide solutions to all of the problems in the written homework assignments in a timely fashion after the assignment is due. Since the late policy affords you a maximum of three late days, we cannot release solutions until the Wednesday following the assignment's due date, at the earliest.

Questions

Answer the following problems from the textbook. If a practice problem is listed, try to solve the problem on your own first, then check your answer at the end of the chapter. Make sure you understand the solution provided, then complete the additional questions below. Your write-up only needs to contain the information necessary to understand your answer to the additional questions - you do NOT need to turn in the answer to the practice problem in its entirety.

1. Homework Problem 2.76, parts A, B, C, and D.
2. Homework Problem 2.88, parts A, B, C, D, and E.
3. What, if anything, is wrong with the code below?

```
int test(float f, double d) { return ((double)f) == d; }
```

4. Recall that a floating point number is of the form: $(-1)^S \times M \times 2^E$. M is the mantissa and 2^E is the exponent. Also recall that M ranges from 1.0 to (almost) 2.0. Tell us the values of M and E for the number 15.0. Note we do not expect you to encode this in IEEE 754 representation, we want you to just tell us in decimal the values of M and E , and the binary value of the frac (recall $\text{frac} = M - 1.0$).
5. For this problem you will use the jack-of-all-trades tool **objdump** to examine executable programs on Linux. Be sure to work on this problem on *attu* or the CSE home VM. You will need a copy of the code that you turned in for Lab 1 (your code does not have to be entirely correct or complete, but at least some of your puzzle functions should be filled in).

- A. Run **make** in your Lab 1 directory to build your code. Run the command **objdump -t btest | grep text** and examine the output. What do the strings in the rightmost column of the output represent?
- B. Use **objdump** to disassemble your btest code (review the lecture slides or check the man page to find the right flag). Find the labels in the assembly code corresponding to your bit puzzle functions. What x86 assembly instructions appear to perform right-shift and left-shift operations?
- C. Most Linux distributions include several other useful programs for examining binary files containing machine code. Try out commands such as **ldd**, **file**, **nm**, **strings** and **readelf**, and try other flags with **objdump**. Report something neat, unusual, or unexpected that you find. Don't forget to use the man pages to learn more about all of these programs.
- D. Extra Credit (a little): run the **objdump -t** command on a different program on the system - choose one in the `/usr/bin/` directory. You will likely see the output `"SYMBOL TABLE: no symbols"`. Why does **objdump -t** appear to work on your program, but not on others?

Notes

Make **sure** you are using the 2nd edition of Computer Systems: A Programmer's Perspective. If you're not using the right book, you might be doing the wrong problems!

Submitting Your Work

Please turn in a PDF file containing your answers to the **Catalyst Drop Box for this assignment**. **We will not accept submissions that are not in PDF format.**