

(Due: Jan. 29)

This warm-up homework helps you understand what is really inside of a running program and what the operating system needs to deal with. Login a Linux workstation in FH133E. Type

```
tar xvfz ~cis345s/pub/hw1.tar.gz
```

to uncompress and extract the files (i.e. `lbcount.c`, `map.c`) to your working directory. Next, use the following commands to compile and build the needed executable files:

```
gcc lbcount.c -g -o lbcount
```

```
gcc map.c -g -o map
```

Load up your `lbcount` executable in `gdb`, set a breakpoint at the second `if` statement, and start running your program with the single input file `lbcount.c`. When the execution stops at the breakpoint, type `continue` three times. Take a screenshot of the terminal window and print it. Think about the following questions and put your answers in the file `hw1_report.txt`.

1. What is the value of `argv`? (hint: `print argv`)
2. What is pointed to by `argv`? (hint: `print argv[0]`)
3. What is the value of `byteCount`? `lineCount`?
4. What is the address of the function `main`?
5. Try `info stack`. Explain what you see.
6. Try `info frame`. Explain what you see.

Next, type `objdump -x -d lbcount` to look the executable file `lbcount`. You will see that your program has several segments, names of functions and variables in your program correspond to labels with addresses or values. And the guts of everything is chunks of stuff within segments. In the `objdump` output these segments are under the section heading. There is actually a slight nuance between these two terms which you can read more about online. While you are looking through the `objdump`, try and think about the following questions and put the answers in the file `hw1_report.txt`.

7. What file format is used for this binary? And what architecture is it compiled for?
8. What segment/section contains `main` (the function) and what is the address of `main`? (The last few hex digits should be the same as what you saw in `gdb`)
9. Do you see the stack segment anywhere? What about the heap? Explain.

Now, you are ready to run the executable `map`. Think about the following questions and put the answers in `hw1_report.txt`.

10. Use `objdump` with the `-D` flag on the `map` executable. Which of the addresses from the output of running `./map` are defined in the executable, and which segment/section is each defined in?
11. Where is the heap? What direction is it growing in?

12. Are the two malloc()ed memory areas contiguous? (e.g. is there any extra space between their addresses?)
13. What direction is the stack growing in?
14. How large is the stack frame for each recursive call?

### **Turnin**

Each student has to submit this homework electronically using the following turnin command (on grail):

```
turnin -c cis345s -p hw1 hw1_report.txt
```

Each student also needs to hand in a hard-copy document which includes the screenshot and the file `hw1_report.txt`. The cover page should contain your picture (taken in FH128 using iMac), name, and your login id. Start on time and good luck. If you have any questions, send e-mail to [sang@cis.csuohio.edu](mailto:sang@cis.csuohio.edu).