CS 2230
Grading Guidelines
Daniel Boyer
daniel.l.boyer@wmich.edu

The following are general guidelines to consider when writing your code. Because you are now dealing with a low-level language, things like commenting, indentation, etc. become extremely important. I would recommend that you write all of your assignments in an appropriate text editor. For Windows/Mac/Linux I recommend Notepad++/Tincta/Geany (respectively) which are all free.

Comments/Indentation

This is the most important aspect of any assembly language program you write. If you turn in an assignment with zero comments, I will flip over my desk and scream, “NO!” and politely ask you to comment your program so I can effectively grade it, i.e., you will get one warning before I take off points.

What NOT to do:

```
loop2:
and %i0, 0xF, %l2
srl %i0, 4, %i0
ldub [%i5+%l2], %l2
stb %l2, [%i4 + %l0 ]
cmp %l0, %g0
bne loop2
dec %l0
```

Look at that mess! This is incredibly difficult to debug and I simply do not have enough time.

What is expected:

```
loop2:
          and  %i0,  0xF,  %l2    ! get low nibble of data
          srl  %i0,    4,  %i0    ! shift right 4 bits for
                                  ! next mask
          ldub [%i5 + %l2],  %l2  ! asciify the nibble
          stb  %l2, [%i4 + %l0 ]  ! fill byte in “empty”
          cmp  %l0,  %g0          ! have we reached 0?
          bne  loop2              ! if not, loop again
          dec  %l0                ! decrement loop counter
```

Much better. Do you need to comment every line? While this is the ideal, especially toward the end of the semester as your programs grow larger, I will not take points off if every line does not have a comment. Just remember that when in doubt, comment! If I can understand what your program is doing, chances are that your grade will reflect that as well. Also take note that everything under a particular label is indented appropriately. This helps with the clarity of your code and makes it easier for grading. Keep these things in mind when writing your C code as well. While your C code will be primarily used to call the assembly functions that you write, comments are always helpful.

Typescript

Since I will not be compiling and running your programs, you will be required to turn in a typescript of your work. You MUST turn this in. Failure to do so will result in a zero. When creating your typescript, please do the following:

1. Start your typescript (obviously).
2. *cat* your .c and .s file(s).
3. Compile your program via a *make* or *gcc* command.
4. Run your program.
5. Exit the typescript.

Dr. Trenary will go over the details on how to do all of this; this is just the order that I would like to see when you turn in your assignment.

When creating your typescript, I would recommend opening the file in a text editor to make sure everything is formatted correctly before printing it out. Does this mean that you can alter the file? Yes. Will I notice if you change your output to hide a faulty program? Of course. Obviously if you do this you will receive a zero for the assignment, but more importantly, you'll receive a very disappointed look from me and be sent to your room to think about what you've done.

Grades

In terms of individual assignments and point totals for each one, that is still to be determined. More than likely, each assignment will be worth ~25 points. Because of the way your final grade is calculated, I would recommend not worrying too much about points but rather focus on not falling behind and preparing well for your tests. Remember that doing well and understanding all of your assignments is essential for doing well on your tests and succeeding in the class.

Questions?

Feel free to email me any questions you might have in regards to grading or the class in general. I recently took this course so a lot of the material is still fresh in my mind. If you would rather meet in person we can schedule a time for that as well. Mondays, Wednesdays, and Fridays work best for me. Good luck!

How to fix any computer

The Oatmeal http://theoatmeal.com

Windows

Step 1. Reboot

Did that fix it?

No? Proceed to step 2

Step 2.

Format hard drive.
Reinstall Windows.

Lose all your files. Quietly weep.

Apple

Step 1. Take it to an Apple store.

Did that fix it?

No? Proceed to step 2

Step 2. Buy a new Mac.

Overdraw your account. Quietly weep.

Linux

Step 1.

Learn to code in C++. Recompile the kernel. Build your own microprocessor out of spare silicon you had lying around. Recompile the kernel again. Switch distros. Recompile the kernel again but this time using a CPU powered by refracted light from Saturn. Grow a giant beard. Blame Sun Microsystems. Turn your bedroom into a server closet and spend ten years falling asleep to the sound of whirring fans. Switch distros again. Abandon all hygiene. Write a regular expression that would make other programmers cry blood. Learn to code in Java. Recompile the kernel again (but this time while wearing your lucky socks).

Did that fix it?

No? Proceed to step 2

Step 2.

Revert back to using Windows or a Mac.

Quietly weep.