COMP61511 (2017) (http://studentnet.cs.manchester.ac.uk/pgt/COMP61511/): Labs

The “Unix Philosophy” is to build a set of small, “sharp” tools with distinct purposes that can be freely combined to accomplish more complex tasks. We are going to build a clone of one of the classic Unix tools: `wc` as a requirement analysis, code creation, and testing exercise.

🎓 Prerequisites

You should be competent with:

1. A command line shell (`bash` , preferably)
2. Python programming (nothing fancy)
3. The basic Python infrastructure (e.g., running Python programs)
4. The Python Documentation (https://docs.python.org/3/) esp. the Library Reference (https://docs.python.org/3/library/index.html).

For this work, we don’t *require* you to use version control, though, of course, it is always good practice.

Topics

1. Extracting a Spec (wc-requirements.html) (depends! but you should work through a first cut before leaving)
2. A first implementation of `wc` (wc-first-implementation.html) (depends! take your time!)

Submission details

1. You will upload to Blackboard a zipped archive called `yourusername_CW1.zip` which contains a directory called `yourusername_CW1` . That directory will contain a single file called `miniwc.py` .
2. No prep script or stub file for this one! It’s very simple and you have a model from this morning.
3. NOTE: This is due WED NIGHT by 19:00!!! This is to allow us to prepare for the next day lab! TAKE NOTE!!!

Other Resources

- The Wikipedia article (https://en.wikipedia.org/wiki/Unix_philosophy) on the Unix Philosophy is a good starting place with a good overview of the various flavours of “the” Unix Philosophy.
- The Art Of Unix Programming (http://www.catb.org/esr/writings/taoup/) is a fairly canonical, if contentious, text on the subject. The key chapter (http://www.catb.org/esr/writings/taoup/html/ch01s06.html) overlaps a lot with the Wikipedia page.
- This description (http://www.linfo.org/unix_philosophy.html) is helpful, though one must take care in interpreting the historical context and contrast with proprietary systems section.

The format was derived from the Software Carpentry (http://software-carpentry.org/license/) template. The lessons are sort of patterned on and inspired by the Software Carpentry style.