Build Your Own wc Tool

This challenge is to build your own version of the Unix command line tool wc!

The Unix command line tools are a great metaphor for good software engineering and they follow the Unix Philosophies of:

- Writing simple parts connected by clean interfaces – each tool does just one thing and provides a simple CLI that handles text input from either files or file streams.
- Design programs to be connected to other programs – each tool can be easily connected to other tools to create incredibly powerful compositions.

Following these philosophies has made the simple unix command line tools some of the most widely used software engineering tools – allowing us to create very complex text data processing pipelines from simple command line tools. There's even a Coursera course on Linux and Bash for Data Engineering.

You can read more about the Unix Philosophy in the excellent book The Art of Unix Programming.

The Challenge – Building wc

The functional requirements for wc are concisely described by it's man page – give it a go in your local terminal now:

```
man wc
```

The TL/DR version is: **wc** – word, line, character, and byte count. You can see the result in action in the video below:

Step Zero

Like all good software engineering we're zero indexed! In this step you're going to set your environment up ready to begin developing and testing your solution.

I'll leave you to setup your IDE / editor of choice and programming language of choice. After that here's what I'd like you to do to be ready to test your solution.

Download the this text and save it as `test.txt`.

Step One

In this step your goal is to write a simple version of wc, let's call it ccwc (cc for Coding Challenges) that takes the command line option -c and outputs the number of bytes in a file.

If you've done it right your output should match this:

```
>ccwc -c test.txt
  342190 test.txt
```

If it doesn't, check your code, fix any bugs and try again. If it does, congratulations! On to…

Step Two

In this step your goal is to support the command line option -l that outputs the number of lines in a file.

If you’ve done it right your output should match this:

```
>ccwc -l test.txt
    7145 test.txt
```

If it doesn’t, check your code, fix any bugs and try again. If it does, congratulations! On to…

Step Three

In this step your goal is to support the command line option -w that outputs the number of words in a file. If you’ve done it right your output should match this:

```
>ccwc -w test.txt
   58164 test.txt
```

If it doesn’t, check your code, fix any bugs and try again. If it does, congratulations! On to…

Step Four

In this step your goal is to support the command line option -m that outputs the number of characters in a file. If the current locale does not support multibyte characters this will match the -c option.

You can learn more about programming for locales here

For this one your answer will depend on your locale, so if can, use wc itself and compare the output to your solution:

```
>wc -m test.txt
  339292 test.txt
>ccwc -m test.txt
  339292 test.txt
```

If it doesn’t, check your code, fix any bugs and try again. If it does, congratulations! On to…

Step Five

In this step your goal is to support the default option - i.e. no options are provided, which is the equivalent to the -c, -l and -w options. If you’ve done it right your output should match this:

```
  >ccwc test.txt
      7145   58164  342190 test.txt
```

If it doesn’t, check your code, fix any bugs and try again. If it does, congratulations! On to…

The Final Step

In this step your goal is to support being able to read from standard input if no filename is specified. If you’ve done it right your output should match this:

```
  >cat test.txt | ccwc -l
      7145
```

If it doesn’t, check your code, fix any bugs and try again. If it does, congratulations! You’ve done it, pat yourself on the back, job well done!

Help Others by Sharing Your Solutions!

If you think your solution is an example other developers can learn from please share it, put it on GitHub, GitLab or elsewhere. Then let me know - ping me a message via Twitter or LinkedIn or just post about it there and tag me.

Get The Challenges By Email

If you would like to recieve the coding challenges by email, you can subscribe to the weekly newsletter on SubStack here:

Tags: unix beginner