Unix Tutorial 2: Unix Pipes

Today, we are going to practise chaining together simple commands using the Unix pipe facility, which is written as a vertical bar i.e. |. Most of our pipelines will have the following shape—there is a 'source' command that generates some textual data as an *output stream*, which is then filtered by another command that processes this textual data as its *input stream*. In this tutorial, we will work through two examples.

Counting Word Occurrences

Download a textfile containing *The Adventures of Sherlock Holmes* from the web, using this command[1].

```
$ curl -O \
  https://www.gutenberg.org/cache/epub/1661/pg1661.txt
$ ls pg1661.*
```

Let's rename this file to a better filename.

```
$ mv pg1661.txt holmes.txt
$ ls *.txt
```

Notice that a file `holmes.txt` has been downloaded to your filespace, into your working directory. You may inspect the contents of file using the less command:

```
$ less holmes.txt
```

Press space bar to scroll down one screen and b to scroll back one screen. Press q to quit the less program.

Sometimes, you will want to look at just the first (or last) few lines of a long textfile, e.g. a log file. Use the head (or tail) command for this:

```
$ head -n 10 holmes.txt
$ tail -n 10 holmes.txt
```

To print the whole text file out to the console output, use the cat command as follows:

```
$ cat holmes.txt
```

We are going to use cat as the source command in our pipeline. Suppose we want to find all the instances where Holmes says, 'My dear Watson.' Let's build a pipeline that searches for this phrase in our text file:

[1]Sorry, you might have to type in this command rather than cutting+pasting. The PDF version of this file uses different character encodings for – and other characters, that the console will not recognize. Also, this command should be on one line but the URL makes it too long!

```
$ cat holmes.txt | grep -i "my dear watson"
```

When you run this command, it should print out three lines which are the three occasions when Sherlock utters his memorable phrase to his friend. Notice the | is the pipe operator, sending the output from the cat command to the grep command directly. Notice the `-i` flag for grep, which makes the search case-insensitive. There are plenty of other useful flags for grep. You might want to print out a line or two of context before each quotation, or the position in the source file where each expression occurs. Try this command:

```
$ cat holmes.txt | grep -inC3 "my dear watson"
```

which will print line numbers for the matching lines, and include 3 lines of context each side of the matching line. Notice that we can run together the flags after a single dash—this saves typing `-i -n -C3` as separate flags.

Here are some more search terms for you—try looking for **elementary** and **cocaine**. How many times does each word occur in the text file?

Counting Word Occurrences

For a second exercise, we are going to find out the current temperature in Glasgow from the BBC website. We fetch the Glasgow weather page with a curl command:

```
$ curl https://www.bbc.co.uk/weather/2648579
```

This command prints out the whole HTML webpage to the console standard output. We want to filter this command, so it just prints out the relevant information. Let's use the grep command to search for the latest observation of weather[2].

```
$ curl -s https://www.bbc.co.uk/weather/2648579 |
    grep wr-day__weather-type-description
```

Note the `-s` flag to curl makes it operate *silently* so it does not print out its progress information as it downloads the data. The `grep` command prints out matching lines for the wr-day . . . div class.

Now we need to use some search-and-replace commands (`sed`) to strip out the HTML tags, with their angle-bracket start and end characters:

```
$ curl -s https://www.bbc.co.uk/weather/2648579 |
grep wr-day__weather-type-description |
sed -e 's/<[^>]*>/ /g'
```

This returns lots of JSON key-value pairs, which we can split into separate lines with a `tr` command, a single character search-and-replace.

[2]This is a long command, so it may spread over multiple lines, but you can carry on typing directly on a single line that gets wrapped by the console.

```
curl -s https://www.bbc.co.uk/weather/2648579 |
grep wr-day__weather-type-description |
sed -e 's/<[^>]*>/ /g' |
tr ',' '\n'
```

We only want the first three lines, since these contain the most recent weather report, for the current time. We use the `head` command to get the first few lines.

```
curl -s https://www.bbc.co.uk/weather/2648579 |
grep wr-day__weather-type-description |
sed -e 's/<[^>]*>/ /g' |
tr ',' '\n' |
head -n 3
```

Finally, let's save the weather report to a textfile called glasgowtemp.txt. We can use the > operator to redirect the output of the pipeline to a file.

```
curl -s https://www.bbc.co.uk/weather/2648579 |
grep wr-day__weather-type-description |
sed -e 's/<[^>]*>/ /g' |
tr ',' '\n' |
head -n 3 > glasgowtemp.txt
```

Some more challenges:

1. (easy) Adapt this pipeline to print out the temperature in Singapore.
2. (difficult) Write a bash script that takes a single argument which is a UK placename. Your script will use the Open Weather service to find the temperature in that place, then it will print out "The current temperature in XXX is Y degrees". The relevant URL for Open Weather will have the form: `http://api.openweathermap.org/data/2.5/weather?q=CITYNAME`, e.g. `http://api.openweathermap.org/data/2.5/weather?q=glasgow`. This web address will return a JSON string, which you will have to parse to extract the main.temp field, which measures the temperature in Kelvin units. Relevant Unix utilities are: curl, cut, echo, bc. Check out the man pages for these utilities to find out how to use them. **Update**: To use OpenWeatherMap, you need to sign up for an API key — how annoying! Check out the details at `http://openweathermap.org/appid`. **Note!** If you use a curl command and you have `$` or `&` characters in the URL, then you need to enclose the *entire* URL in double quotes, to avoid the shell misinterpreting these special characters.

Further Reading

For further investigation today, use the man command to find out the flags for all the commands you have already executed.

There are lots of grep tutorials online, e.g. see `http://www.thegeekstuff.com/2009/03/15-practical-unix-grep-command-examples/`. Also, here is an exhaustive tutorial on sed: `https://www.grymoire.com/Unix/Sed.html`.

If you would like to play around with more web APIs using curl, then check out the online test site at `https://restful-api.dev/`.