

Viewing files

Max Carter-Brown

Anglia Ruskin University, Wellcome Sanger Institute
max.carter-brown@aru.ac.uk

Contents

1. Outcomes	1
2. Introduction	1
3. Using <code>cat</code>	1
4. Using <code>less</code>	2
5. Using <code>head</code> and <code>tail</code>	2
6. Using a text editor	3

1. Outcomes

By the end of this document, you should be able to:

1. Open any kind of text based file using the command line, using `cat`, `less`, `head`, and `tail`
2. Know what a text editor is, and how to use one, for example using `nano` and `vim`

2. Introduction

At some point, you will want to view the files you have created. Luckily, many file types can be viewed directly in the terminal. This is useful, as it allows you to quickly check the contents of the file, and modify accordingly. Other file formats are not easily readable, and require specialised software to view and edit them - for example binary or compressed files.

3. Using `cat`

`cat` is a command that reads data from a file and outputs it to the terminal. It is a very simple command, and is often used to view the contents of a file.

Text redirection

The `>` symbol is used to take whatever is on the left and redirect the output. There's a whole tutorial on this later, so don't be worried if you haven't come across it yet. For now it is good to remember that `>` will overwrite any existing file with the same name, so be a bit careful.

Shell 1: Create a file called `hello.txt`, and view the contents of the file using `cat`

```
# Create a file called hello.txt
echo "Hello, world!" > hello.txt
# View the contents of hello.txt
cat hello.txt
```

`cat` can be used to view the contents of multiple files at the same time. For example:

Shell 2: Create two files called `hello.txt` and `world.txt`, and view the contents of both files using `cat`

```
# Create a file called hello.txt
echo "Hello, world!" > hello.txt
# Create a file called world.txt
echo "World, hello!" > world.txt
# View the contents of hello.txt and world.txt
cat hello.txt world.txt
```

If you have very large files though, `cat` will dump the entire contents of the file to the terminal. This is not always desirable, as it can be difficult to read. Instead, you can use `less`, `head`, or `tail`.

Questions

1. Use `cat` on some files in your file system. Are there any files which you can't view with `cat`? Try a Word Document, for example.
2. Can you create a copy of a text file using `cat`? How?
3. How do you add line numbers next to output using `cat`?

4. Using `less`

`less` is a command that allows you to view the contents of a file, one page at a time. This is useful for viewing large files, as you can scroll through the contents of the file, and search for specific text.

The `seq` command

`seq` is an in built BASH command which print out sequences of numbers. In its default form it will print 0 to whatever number you specify, on separate lines.

Shell 3: Create a file called `large.txt`, and view the contents of the file using `less`

```
# Create a file called large.txt
seq 1000 > large.txt
# View the contents of large.txt using less
less large.txt
```

You can scroll through the file using the arrow keys, and search for specific text by typing `/` followed by the text you want to search for. Press `n` to go to the next occurrence of the text. Press `q` to exit `less`.

5. Using `head` and `tail`

`head` and `tail` are commands that allow you to view the first or last few lines of a file, respectively. This is useful for quickly checking the contents of a file, without having to view the entire file. or example, to view the first 10 lines of a file:

Shell 4: Create a file called `numbers.txt`, and view the first 10 lines of the file using `head`

```
# Create a file called numbers.txt
seq 100 > numbers.txt
# View the first 10 lines of numbers.txt using head with the -n flag
head -n 10 numbers.txt
```

Similarly, you can view the last 10 lines of a file using `tail`:

Shell 5: View the last 10 lines of the file `numbers.txt` using `tail`

```
# View the last 10 lines of numbers.txt using tail with the -n flag
tail -n 10 numbers.txt
```

Questions

1. Extract the first line of a file using `head` and save it to a new file.
2. Extract the last line of a file using `tail` and save it to a new file.
3. How would you exclude the first line of a file using `tail`?

6. Using a text editor

It is possible to edit files using the terminal, using a text editor. There are many text editors available, but two of the most commonly used are `nano` and `vim`. `nano` is a simple text editor that is easy to use, whereas `vim` is a more complex text editor that has a steeper learning curve.

To edit a file using `nano`, simply type `nano` followed by the name of the file you want to edit:

Shell 6: Edit the file `hello.txt` using `nano`

```
# Edit the file hello.txt using nano
nano hello.txt
```

Now you can edit the contents of the file. To save the changes and exit `nano`, press `Ctrl + X`, then `Y`, and finally `Enter`.

The very basics of `vim` are as follows. To enter `vim`, type `vim` followed by the name of the file you want to edit (or the name of a file if it exists already):

Shell 7: Edit the file `hello.txt` using `vim`

```
# Edit the file hello.txt using vim
vim hello.txt
```

In `vim`, you can move the cursor using the arrow keys. To enter insert mode, press `i`. Now you can edit the contents of the file. To save the changes and exit insert mode, press `Esc`, then `:wq` (i.e. write and then quit), and finally `Enter`. If you want to exit `vim` without saving the changes, press `Esc`, then `:q!`, and finally `Enter`. This is only the start, `vim` has many so-called 'key bindings' that allow you to do things more quickly than you would be able to in `nano`. But these are beyond the scope of this document.

Questions

1. Create a file using `nano` , add some text, and save it.
2. Create a file using `vim` , add some text, and save it.
3. Go back into one of the files you just created, edit it, and save the file.