



Week2: The POSIX shell

☰ Topics Covered

Regular Expressions

grep

Pipes:

```
$ ls -1 | head
```

```
# ls -1 will create a vertical output to the screen
```

```
# The head programm shows you the top level directories of the c
```

```
#The output of ls -1 is then passed to head
```

```
$ ls -1 | head -n 4
```

```
# will show you only four directories
```

▼ grep

- grep allows you to search for your files within your system

Example: search for a name in a text file:

```
$ grep "Hello" HelloWorld.txt
#output : Hello if word is found in the text file
# But if you had Helloworld, then the grep command would return nothing
# To get an exact match: use the -w flag. Where -w is Whole word

$ grep -w "Hello" HelloWorld.txt
## Will return Hello
```

- Other useful flag to use in grep:

```
-i : Will ignore case sensitivity
-n : will show you the exact line where the match was found
-B <number> : Will show you the lines before your match as well as the match
-A <number> : similar to -B but will show you lines after the match
-C <number> : shows you <number> of lines before and <number> of lines after the match
./* : Will show you the match in all the files in the directory
-r : Recursive Search will search for within sub-directories
-l : will only show the files that contain the match
-s : suppresses error messages if you want to search all the directories
-iP : flag
^<match> : This will search for matches that start with the match
-P : will search for matches that begin with P (combine it with -i and you can search for matches that begin with P)
```

- You can use the grep command with pipes,

- A very useful example of this is if you want to search for your git commits, if you type `history` on the command line you will get a list of all the commands you have typed.
- But if you instead did this:

```
$ history | grep "git commit"
```

- The output from history is passed to grep which then shows you only the commands containing the text "git commit"
- Now lets say you wanted to search for commits made to the main branch, well if you can use pipe again:

```
$ history | grep "git commit" | grep "main"
```

- This will now show you only the commits made to the main branch.

Redirecting :

- To save a output to a file you can use the `>` e.g.

```
cat infile | sort > reducedinfile.txt
```

- `>` will overwrite to a file
- `>>` will append an existing file

- Output is in two forms:

1. Standard output
2. Standard error

The standard error is the error message you get whenever a programme does not produce the expected results.

- You can re-direct the error to a file, via the command:

`command > file 2>&1` This will output the standard output (1) and the standard error (2) to the file.

e.g. `cat nonexistentfile > temp.txt 2>&1`

- If we cat a nonexistentfile it will produce an error that will say the file does not exist
- Instead of the message showing to the screen, it will be shown to the file temp.txt
- If there is no error and the file does exist then this will also be contained in the temp.txt file as we used the `&1`

Advanced

- sed:

Sed is a stream editor it can change texts using regular expression as it passes from the input to the output

e.g:

```
$ echo "Hello World"
# Will print Hello world to the standard output
```

```
# Say we wanted to say Hello Universe or Sup World:
```

```
$ echo "Hello World" | sed -e 's/World/Universe/'
```

```
output: Hello Universe
```

```
$ echo "Hello World" | sed -e 's/Hello/Sup/'
```

```
output: Sup World
```

Task:

Navigate to the folder: `/usr/share/dict/words` which contains a list of English words in alphabetical order used for spell checking purposes.

Using the tools described above, complete the following tasks:

- Count the number of lines in the file:

```
$ wc -l words
```

```
#output = 479828
```

- The 6171st word what is it?

```
$ sed -n '6171p' words
```

```
#output aethers
```

- Count the number of words containing the capitalised Q :

```
$ grep Q words | wc -l
```

```
#output 363
```

- All words ending in j

```
$ grep 'j$'
```

```
#output 363
```

- All words containing the sequence "kp", but not "ckp"

```
$ grep -w kp
```

```
#output 363
```