

The Five Most Interesting Things I Learned in UNIX Tools

Connor Hausmann

December 6, 2018

1 The Grep Command

The grep command was great for using regex to search strings in a file. Using grep allowed me to further my understanding of regular expressions and it gave me a very user-friendly way to specify how we managed that patter matching with command line operations. Some of those command line operations can be found in the table below: [3]

Table 1: Sample Commands

Command	Description
-c	Prints the number of lines
-h	Displays the matched lines, not filenames
-l	Displays list of filenames
-e	Specifies an expression
-w	positive

2 The Man Pages

Though they were not explicitly taught in this class, I had not used them properly until taking this class. The MAN pages as well as the MAN command have opened my eyes to a powerful resource that has been available to me since beginning programming. It is often faster to search the man pages for knowledge on a command rather than having to look for it online. As we all know, unix is a powerful programming language to learn because commands are easier and faster to type than using a mouse at times. Below is a screenshot of me accessing the ‘man’ pages for the command ‘-ls’ and below that is a screenshot of the man pages redirection.

```

haus @Razer:~/cop4342$ man ls
haus @Razer:~/cop4342$

LS(1)                                User Commands                                LS(1)

NAME
    ls - list directory contents

SYNOPSIS
    ls [OPTION]... [FILE]...

DESCRIPTION
    List information about the FILES (the current directory by default). Sort entries
    alphabetically if none of -cftuvSUX nor --sort is specified.

    Mandatory arguments to long options are mandatory for short options too.

    -a, --all
        do not ignore entries starting with .

    -A, --almost-all
        do not list implied . and ..

    --author
        with -l, print the author of each file

    -b, --escape
        print C-style escapes for nongraphic characters

    --block-size=SIZE
        scale sizes by SIZE before printing them; e.g., '--block-size=M' prints
        sizes in units of 1,048,576 bytes; see SIZE format below

    -B, --ignore-backups
        do not list implied entries ending with ~

    -c      with -lt: sort by, and show, ctime (time of last modification of file status
            information); with -l: show ctime and sort by name; otherwise: sort by
            ctime, newest first

```

Figure 1: LS Commands

3 Latex Implementation

Possessing the ability to create and export a PDF file based on an input file is something else we learned and it is referred to as using Latex. This is helpful because this document preparation and markup system allows us to produce PDF files that contain tables, pictures and many other features. This assignment is a very good representation for how latex files work. As this PDF was created using bibtex, latex, import and xfig.

4 Perl in General

Perl programming was completely new to me prior to taking this course and there are many aspects of the language that I like, including the use of hashes and keys. Having a scalable data type that has no particular order made manipulating the hash values via the use of keys a low easier. This was my first real use of a simple database. Below is a list of other Perl commands/functions that I had never used before but found helpful.[2]

- Scalar Variables using ‘my’
- The ‘print’ expression as opposed to ‘printf’
- Reading in a file to a line using ;STDIN;
- The Chomp operator to remove the newlines
- Shift/Unshift
- Foreach

5 The Sed Command

Heading back to the beginning of the semester, I found myself using Sed to complete many of my assignments. Sed is great for making changes to more than one file according to editing commands. This stream editor then writes the results to standard output for the user to easily access. [1]

```
# Example: delete the 4th line in a file
$ sed '4d' input.txt > output.txt

# Example: replace every occurrence of 'hello' with 'world' on lines 10-20
$ sed '10,20s/hello/world/' input.txt > output.txt
```

Figure 2: Sed Example

6 Conclusion

To conclude this semester of UNIX tools at Florida State University, I can say that I learned two new languages in one class as well as tools and applications on how to use them. This class will help me further my abilities in Computer Science and will ultimately help me become a better code writer. Because, after all, sometimes it is faster to do a few key strokes rather than using a mouse.

References

- [1] GNU.org. A gnu head, using sed, 2015.
- [2] R. Schwartz. *Learning Perl*. O’Riley Media, 2011.
- [3] R. Stanson. Grep command in unix/linux, 2017.