15
More Shell Scripting

Some more useful shell commands

- $ man - displays the manual or “man page” for a given program
- $ which - returns the path the given file or program
- $ env - displays information about the current environment
- $ cat - outputs the contents of a given file to the standard output
- $ whoami - prints the current user account name
- $ echo - displays the value of a variable - example: $ echo $PATH
- $ sleep - suspends execution of the current process for a specified interval
- $ clear - clears the standard output device - the screen
- $ read - reads input from the standard input device - keyboard
- $ uptime - displays how long the system has been running
- $ grep - search for text in files

Grep

- Tool written by Ken Thompson as a successor to the ed program which had a command g/re/p to do essentially the same thing as grep does.
- Globally search for a Regular Expression and Print
- Allows for powerful searching for text inside of files
- Also allows for using regular expression syntax for defining search patterns
- We will have a lab assignment about using grep

Grep Example Usages

- Basic usage is grep <search_pattern> <files_to_search>
- $ grep def *.py - will search for the word ‘def’ in all the files that end in .py
- Using the -E switch will let us search using a regular expression pattern
- $ grep -E “^(a|e|i|o|u){6}$” words.txt
- $ grep -E "^d(a|r)t" words.txt

Pipes

- The pipe character | (i.e. vertical line you print using Shift+backslash key)
- Using pipes, we can direct the output of one program into the input of another program
- $ grep print *.py | less
- $ git branch | grep -i derick | more
- Pipes allow us to daisy chain the I/O of programs to each other

I/O Redirection

- We can also redirect the output of a program to a file using >
- $ ls -lha /usr/bin > binary_files.txt
- $ grep -E "^d(a|r)t" words.txt | sort | uniq > filtered.txt
- We can also redirect the contents of a file using <
- $ 0 < lab4.pl

Wildcard characters

- * - everything
  $ ls s*.py

- ? - exact match at the specific position
  $ ls -lh lab*.p? - list all files starting with lab and ending wth .p followed by one more character than can match anything

- Can also use [ ] to define groups of matching characters like in a regex
  $ ls -la [bjlg]erry - list all the files named berry, jerry, lerry, or gerry

Additional Resources

- Bash guide for beginners - http://www.tldp.org/LDP/Bash-Beginners-Guide/html/index.html
- List of useful Bash resources - https://github.com/awesome-lists/awesome-bash
- Grep man page - https://www.gnu.org/software/grep/manual/grep.html
- More redirection examples - http://www.tldp.org/LDP/abs/html/io-redirection.html