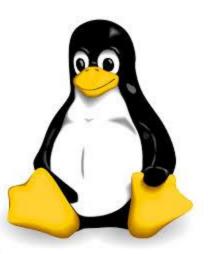# Introduction to Scientific Computing
Meeting 4
Unix Commands

Jeremiah Lant, Hydrologist
USGS Kentucky Water Science Center
jlant@usgs.gov

Last meeting

- Learned some more Unix commands
  - Creating directories and files; **mkdir, touch**
  - Looking at file contents; **cat, head, tail, less**
  - Moving and copying files and directories; **mv, cp, cp -r**
  - Deleting files and directories; **rm, rmdir, rm –r**
  - Finding text in files; **grep**
  - Count number of lines, words, and characters in files; **wc**

FYI

- For now, sharing slides and recordings through my Google Drive.
  - Recordings are large (~100MB) so I will share via WebEx, and will archive a recording locally.
- Software Carpentry has updated some of there material. Continuing to use Version 4 for now.
  - http://software-carpentry.org/lessons.html

Today's Objective

1. **Learn** some more Unix commands
   - Wildcard command; *****
   - Re-direct output from a command to a file; **>**
   - Append the output from a command to a file; **>>**
   - Take output from one command and "pipe" it as input to another command; **|**

- **Why?**
  - Good basis for learning how to program.
  - Becoming more comfortable on command line
  - Becoming more efficient
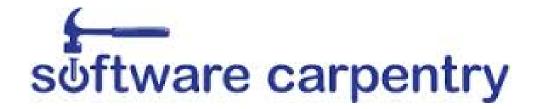
Quick Demo of Some Commands

```
$ mkdir temp
$ cd temp
$ echo hello world > hello-world.txt          # redirect output of echo to file
$ cat hello-world.txt
$ echo hello world again >> hello-world.txt   # append output of echo to file
$ cat hello-world.txt


$ cd $NWISPY
$ cd nwispy
$ ls | wc -l                      # count number of files
$ ls *.txt | wc -l                # count the number of text files
$ wc -l *.py                      # count the number of lines in python files
$ wc -l *.py | sort               # sort
$ wc -l *.py | sort | tail -1     # find total number of lines of code
$ wc -l *.py | sort  -r| head -1  # find total number of lines of code
```

Video – The Unix Shell

- Software Carpentry, Greg Wilson
  - The Unix Shell: Pipes and Filters
    - http://software-carpentry.org/v4/shell/pipefilter.html

Try Out Commands – page 1

\# make a directory called temp and do the following

$ echo hello world > hello-world.txt   # redirect output of echo to file

$ cat hello-world.txt

$ echo hello world again > hello-world.txt   # overwrites hello-world.txt

$ cat hello-world.txt

$ echo hello world again again >> hello-world.txt   # append output to file

$ echo hello world again again again >> hello-world.txt

$ cat hello-world.txt

\# remove the directory called temp

Try Out Commands – page 2

```
# move into directory called "data/nwis-files/"
$ wc *.txt                           # count lines, words, characters
$ wc -l *.txt                        # count lines


# redirecting output
$ wc -l *.txt > numlines
$ cat numlines
$ sort numlines                      # sort lines in "phone book" order
$ sort -r numlines                   # sort in reverse order


# piping output
$ sort numlines | head -1
$ wc -l *.txt | sort | head -1       # find file with least number of lines
$ wc -l *.txt | sort -r | head -1    # find total number of lines
OR
$ wc -l *.txt | sort | tail -1       # find total number of lines


# find total number of lines and file with most number of lines
$ wc -l *.txt | sort | tail -2
```

Try Out Commands – page 3

\# move into the directory called "data/sample-weekly-discharge/"
\# find the largest discharge value in a set of files
$ cat discharge-week1.txt
$ cat discharge-week1.txt | sort

\# remove duplicates (only removes adjacent duplicates)
$ cat discharge-week1.txt | uniq
$ cat discharge-week1.txt | uniq -c
$ cat discharge-week1.txt | sort | uniq -c
$ cat discharge-week1.txt | sort | uniq
$ cat discharge-week1.txt | sort | uniq | tail -1
$ cat *.txt | sort | uniq | tail -1
OR
$ cat *.txt | sort -r | uniq | head -1

\# save your command history
$ history
$ history > meeting-4-history

Next meeting

- More Unix commands
  - Creating aliases and variables; **.bashrc**
  - Finding things; **grep**
  - Start a mini project