

Commands used in this workshop:

Getting around

pwd - prints the current directory path to the screen. Stands for "print working directory."

cd - changes into a different directory.

ls - lists the contents of the directory. Note that the ls command has many useful options

File/folder operations

mkdir - creates a new directory. Example: mkdir my_new_directory

rmdir - removes a directory

cp - copy a file

rm - remove a file

mv - move a file

Working with files

cat - concatenates files, but is often used just to print the contents of a single file

more - displays output of a file for perusal

less - a lot like more

head - print the first part of a file

tail - print the last part of a file

wc - counts words, lines, characters, and bytes in a file (stands for word count)

gunzip - unzip a zipped file or directory

nano - edit a file using the nano text editor. Other text editors available include vi (vim) and pico

grep - search the contents of a file for some text or a pattern

uniq - removes repeated lines

sort - sorts lines

shuf - shuffles lines

More commands

history - prints out the history of commands you ran

man - prints the manual for a command

echo - prints text to the console

wget - downloads files from the web

| - pipe the output of one command to the next one

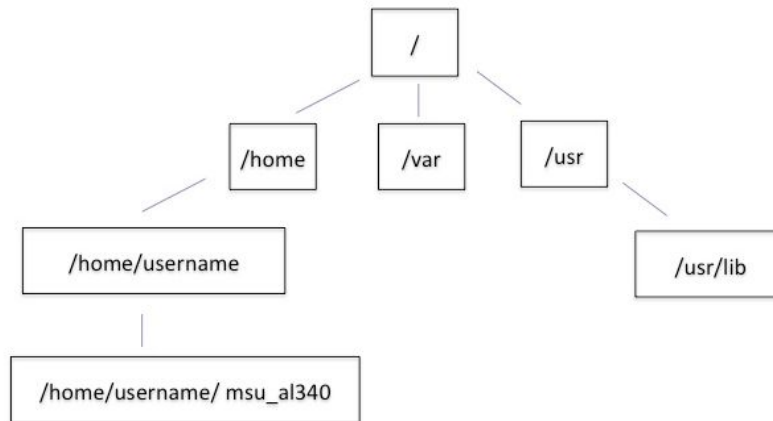
> - redirect the output to a file (and will overwrite)

>> - redirects and appends the output to a file

;- separates between commands in a list

The Filesystem

We refer to the base of the linux filesystem as the **root directory**, denoted by a forward-slash: /



Source: <http://www.datacarpentry.org/2015-11-04-ACUNS/shell-intro/>

Path shortcuts:

- .. the parent directory, i.e. "one level up"
- ~ your home directory (example: /home/yourusername)
- * wildcard (example: data*.csv - may match several filenames)

Absolute vs. relative paths:

Example of a full/absolute path: /home/janedoe/data

Example of a relative path: ../data (does **not** start with /)

Piping and Redirection

Commands can often be linked together, so that the output of one command can be fed into another command. You can "pipe" commands using the | character. For example:

```
echo "this is my command"
```

Prints the phrase "this is my command" to the screen. Instead we can pipe the result to wc -w

```
echo "this is my command" | wc -w
```

The output of this command would be 4 since four words were sent to the word count program.

You can also redirect the output to a file. Use > to create or overwrite a file, or >> to append to a file. For example, the following command would create a file called words.txt :

```
echo "this is my command" | wc -w > words.txt
```