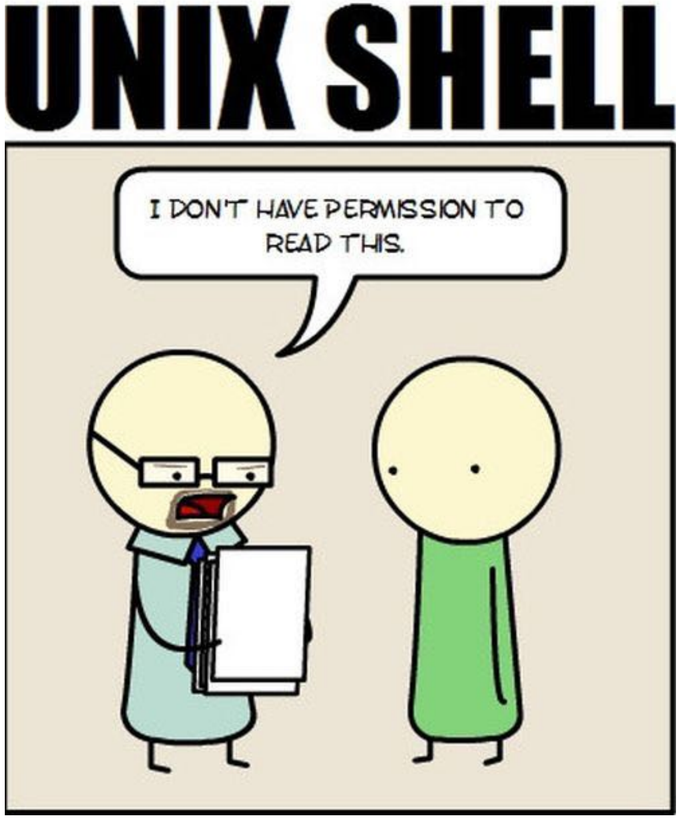
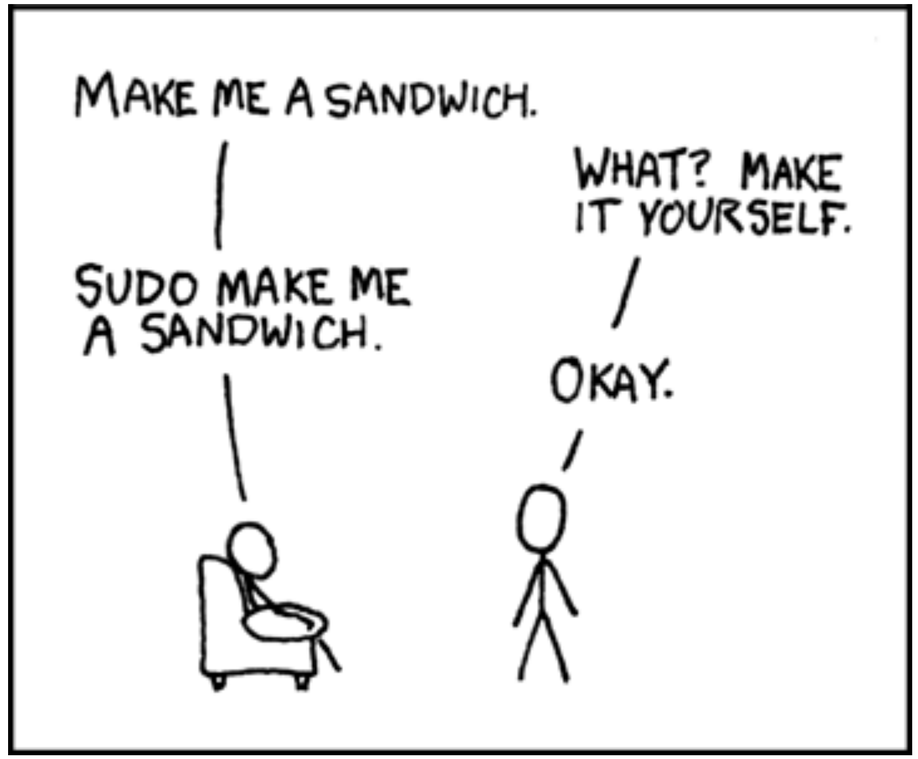
Shell Scripting

Fall 2018
PCfB Class 3
September 14, 2018





Why use shell scripts?

- Automate a series of commands
  - particularly useful when each command takes a long time to run
- Creates a record of commands that have been run
- Easy format for rerunning commands

Shell scripts

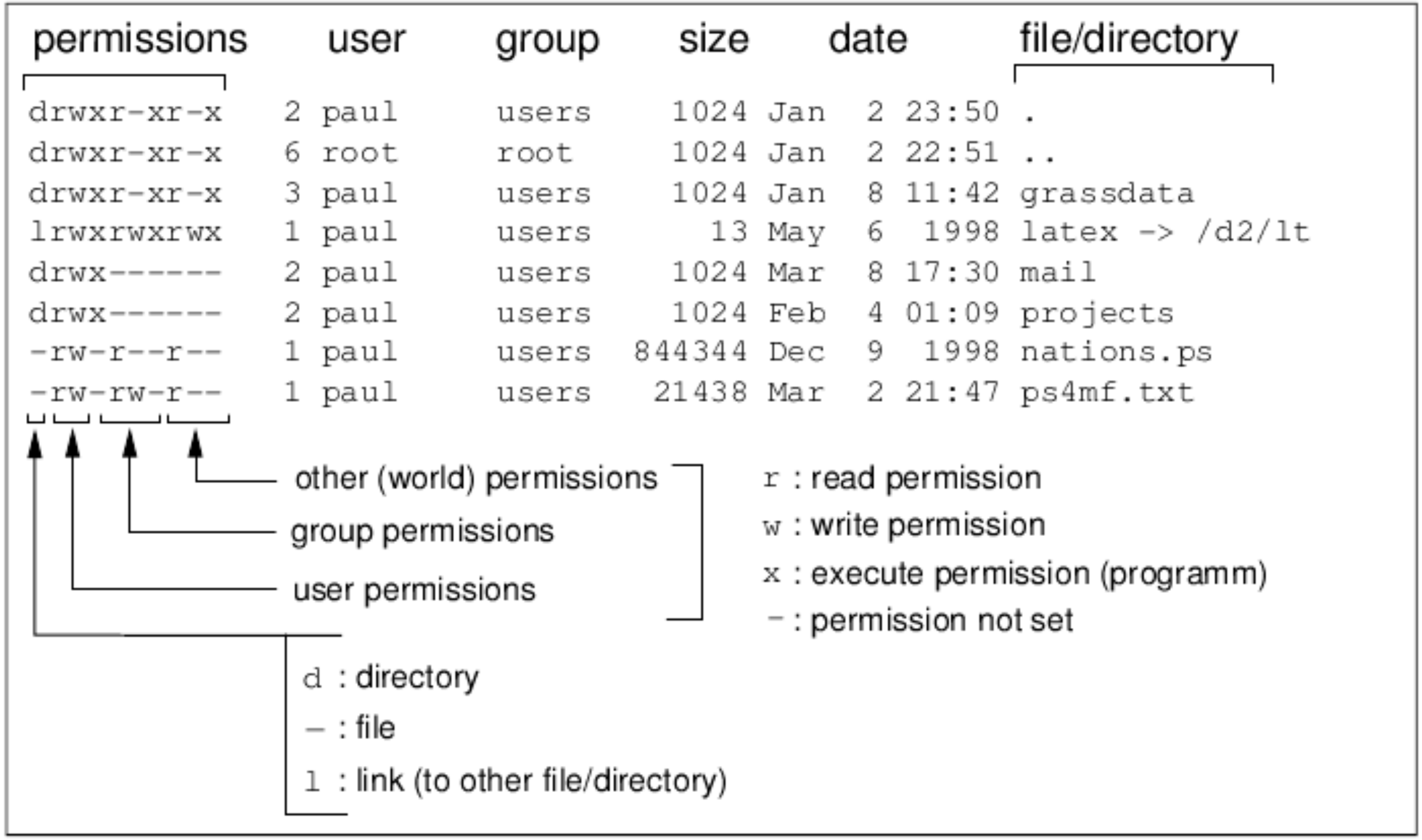
- Text file containing a series of commands that will be executed one after the other

- First line = shebang
  - Turns a text files into software
  - Tells computer which program to use to interpret commands

File extensions

- Recommended (but not required) to save script with specific file extension
- Allows recognition from file name
- Syntax-specific coloring in text editors
- For shell script: .sh

Permissions (`ls -l`)



Permissions (‘ls -l’)

permissions		user	group	size	date	file/directory
drwxr-xr-x	2	paul	users	1024	Jan 2 23:50	.
drwxr-xr-x	6	root	root	1024	Jan 2 22:51	..
drwxr-xr-x	3	paul	users	1024	Jan 8 11:42	grassdata
lrwxrwxrwx	1	paul	users	13	May 6 1998	latex -> /d2/lt
drwx------	2	paul	users	1024	Mar 8 17:30	mail
drwx------	2	paul	users	1024	Feb 4 01:09	projects
-rw-r--r--	1	paul	users	844344	Dec 9 1998	nations.ps
-rw-rw-r--	1	paul	users	21438	Mar 2 21:47	ps4mf.txt

other (world) permissions

group permissions

user permissions

`r` : read permission

`w` : write permission

`x` : execute permission (programm)

`-` : permission not set

`d` : directory

`-` : file

`l` : link (to other file/directory)

To run on command line, scripts must be executable for YOU

Changing Permissions

`chmod` (change mode)

Add execute for User: `chmod u+x file.txt`

Add read and write for Group: `chmod g+rw file.txt`

Remove write and execute for Other: `chmod o-wx file.txt`

All three in one command: `chmod u+x,g+rw,o-wx file.txt`

$PATH

- A list of directories
  - Locations your computer looks for command-line software
- Searched in the order listed
- To view: `echo $PATH`
- To add a directory: `PATH="$PATH:path/to/new/dir"`

Shell + Regexp method

1. Use the shell to generate a list of files/directories
2. Use regular expressions within your text editor to turn those file/directory names into a list of commands

`for` loop

- Simple, but powerful way to repeatedly execute the same commands for different files, parameter values, etc.
- Can be included in scripts or run directly on command line

For loop - basic syntax

```
for file in *.sh; do chmod u+x $file; done
```

- Initiates loop
- Defines variable

- Repeated command

- Terminates loop

For loop examples

```
for file in *.sh; do chmod u+x $file; done
```

```
for file in *.fasta; do
print_fasta_seq_lengths.py $file; done
```

```
for file in *txt; do cp $file copy_$file; mkdir
dir_$file; mv $file dir_$file; done
```

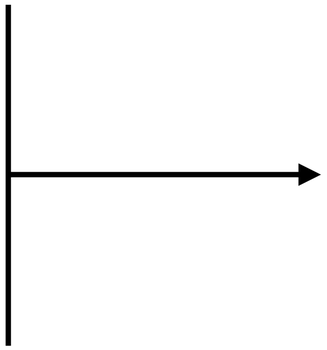
Additional repeated commands

Functions

- Mini scripts that are assigned to names within the shell environment
- Useful for long commands or a series of commands that you will use commonly

Function example

```
cpmv(){
cp $1 copy_$1
mkdir dir_$1
mv $1 dir_$1
}
```



Same commands as last for loop example

```
for file in *txt; do cpmv $file; done
```

Aliases

- Used to create shortcuts for commonly used commands
- Examples:
  - `alias ms="ssh jtl276@monsoon.hpc.nau.edu"`
  - `alias cux="chmod u+x”`
- **Caution:** best not to become too dependent on commands that are not standard
  - they won’t be available on other computers you may need to work on

Exercises

- Set up ‘scripts’ directory
- Regexp to generate shell scripts
- For loops, functions
- Custom script

Jupyter Notebook

(prep for next class)

In [114]:

```
#Read in Bayes Factors calculated from the full data set
bffile = "/Users/jtladner/JTL_GDrive/MyPapers/WAITING_for_others/EBOV_NICD/BEAST/full/v1.8.4/combo/NAU3_RIID2_BIrem
fin = open(bffile, "r")
bf_dict = {}
for line in fin:
    cols = line.strip().split("\t")
    bf_dict[(cols[0],cols[1])] = float(cols[3])
```

In [116]:

```
#Read in info about sublineage ancestral probs
ancest_probs = '/Users/jtladner/JTL_GDrive/MyPapers/WAITING_for_others/EBOV_NICD/BEAST/full/v1.8.4/combo/m2_using/s
fin = open(ancest_probs, "r")
linecount=0
aprob_dict={}
ancest_key={}
for line in fin:
    linecount+=1
    cols = line.strip().split('\t')
    district = cols[0]
    for i,v in enumerate(cols[1:]):
        if linecount==1:              #For header row
            ancest_key[i]=v
            aprob_dict[v]={}
        else:
            if float(v)>=20:
                aprob_dict[ancest_key[i]][district] = float(v)
print aprob_dict

```

```
{'lin3.1.2_add4_combo_m2_trans.txt': {'WesternUrban': 27.68550337, 'WesternRural': 69.88990709}, 'lin3.2.5_rmvJ0162_c
ombo_m2_trans.txt': {'Kenema': 98.92398032}, 'lin3.2.4_rmv2_m2_combo_trans.txt': {'Kenema': 87.90508475}, 'lin3.1.1_r
mv1_combo_m2_trans.txt': {'Kenema': 72.53769243, 'WesternRural': 20.70782415}, 'lin3.2.1_combo_m2_trans.txt': {'Bomba
li': 84.50227011}, 'lin3.2.3_combo_m2_trans.txt': {'Kenema': 98.60022219}, 'lin3.2.2_rmv4_combo_m2_trans.txt': {'Kene
ma': 97.21271059}}
```