How Do I Shoot Myself in the Foot
Use the Command Line

William Killian
KILLIAN@UDEL.EDU
UNIVERSITY OF DELAWARE
OCTOBER 27, 2015
BACKGROUND – WHO AM I

• **5th year PhD Student, research in HPC, Compilers, Machine Learning**
• **Using Linux since 2005, regularly since 2008, full-time since 2011**

• **Regular expressions are my actual favorite**
• **Arch Linux, Gnome, Bash, Emacs**
HISTORY OF COMPUTING INTERACTIONS

Keyboard • Terminal

Mouse • GUI

Touch • Ubiquitous

Why do we want to revert back?
YOUR HANDS DON'T MOVE
ANATOMY OF A (MODERN-DAY) *NIX SYSTEM

- Hardware
- Operating System
- Kernel
- Drivers
- Shell
- Userspace
The shell acts as our gateway to the system

- But mastering a command-line shell can be difficult

[wkillian@liz] ~$
Display all 3655 possibilities? (y or n)
WHAT DO YOU WANT TO LEARN?

• How to...
  • Increase productivity when using a system
  • Become more successful with using computers
  • Automate your workflow
  • Be a Wizard
## Basic Commands – Moving Around

<table>
<thead>
<tr>
<th>Command</th>
<th>Arguments</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>cd</td>
<td>&lt;directory&gt;</td>
<td>Change directory</td>
</tr>
<tr>
<td>mv</td>
<td>&lt;src&gt; &lt;dest&gt;</td>
<td>Rename src to dest</td>
</tr>
<tr>
<td>mv</td>
<td>&lt;srclist&gt; &lt;dest&gt;</td>
<td>Move srclist to dest</td>
</tr>
<tr>
<td>mkdir</td>
<td>[-p] &lt;directory&gt;</td>
<td>[recurse] Create a directory</td>
</tr>
<tr>
<td>rm</td>
<td>&lt;list&gt;</td>
<td>Remove list of files</td>
</tr>
<tr>
<td>rm</td>
<td>-r &lt;directory&gt;</td>
<td>Recursively delete directory</td>
</tr>
</tbody>
</table>

### Important things to know:
- Current directory
- Parent directory
- Home directory
- Root directory
BASIC SYNTAX – UNDERSTANDING I/O

• /dev/stdin
  • Standard Input
  • Also known as 0

• /dev(stdout
  • Standard Output
  • Also known as 1

• /dev/stderr
  • Standard Error
  • Also known as 2

• /dev/null
  • Empty
  • /dev/random
    • Random sequence of bytes

> Output redirection
2>&1 Redirect stderr to stdout
< Input redirection
>> Output append
p1 | p2 Pipe output of p1 into input of p2
If I'd known computer science was going to be like this, I'd never have given up being a rock 'n' roll star.

-- G. Hirst
HOW DO I KNOW WHAT’S IMPORTANT?

• OR MORE IMPORTANTLY, **WHAT TO USE?** PRACTICE, TRY IT OUT
• HOW DO I KNOW WHAT **SOMETHING DOES?** MAN PAGES

• GNU CoreUtils: [HTTPS://WWW.GNU.ORG/SOFTWARE/COREUTILS/](https://www.gnu.org/software/coreutils/)
• GNU FindUtils: [HTTP://WWW.GNU.ORG/SOFTWARE/FINDUTILS/](http://www.gnu.org/software/findutils/)
• GNU DiffUtils: [HTTPS://WWW.GNU.ORG/SOFTWARE/DIFFUTILS/](https://www.gnu.org/software/diffutils/)
BATCH PROCESSING TEXT

- **Filter/modify text**, use **sed, awk, tr, or grep**

- **Grab columns of data**, use **awk or cut**

- **Merging lines**, use **paste**

- **Sorting, unique, set differences, etc…** use **sort, comm, uniq**
BATCH PROCESSING FILES

- **Find** is your best friend
- **Xargs** is its BFF

```
find ~/ -name '*.pdf' | xargs cp -t ~/PDFs/
```

**Find all PDF files in the home directory and copy them to ~/PDFs**
PROCESSING TEXT

But before I **Kill** you, **Mr. Bond**... (erm, use most text processing utilities)

I must show you **Regular Expressions**
REGULAR EXPRESSIONS

^\(banana.*\)\{2,\}orange$  two or more ‘banana’ followed by ‘orange’

^  begin of line
\( \)  group
.  any character
*  zero or more
\([m],[n]\)  minimal m, maximum n
$  end of line
[a-z]  character group (a-z); [aeiou] vowels; [^aeiou] no vowels
Regular Expressions can be awesome
REGULAR EXPRESSIONS CAN BE HARD
Sample Data: http://www.udel.edu/002771

DEMO TIME

THIS PART IS INTERACTIVE – I CAN GO OVER ANYTHING