Basic Linux Command Line

Hugh Brown
Sr. Systems Administrator
ITS Enterprise Infrastructure

University of Iowa

August 9, 2013

Introduction

Who am I?

- Linux User and Sysadmin for 15 years
- Worked in big business, small business and university environments
- Taught Linux courses for the university

My assumptions:

- You have limited experience with the Linux command line.
- You are using bash as your shell.

Overview

- Why learn the Linux command line?
- Riding the learning curve
- Basic Linux ideas
- User accounts
- File system layout
- File system navigation and manipulation
- Bash shell basics and commands
- Permissions
- Troubleshooting Permissions
- Next steps

Why learn the Linux command line?

- Sometimes you don't have a choice
- Productivity
- Automation

  Rename 100 files that end in .html to .html.2013-08-08

```
ls *.html | while read file; do
  mv $file ${file}.$(date +%F)
done
```

Why learn the Linux command line?

- Sometimes you don't have a choice
- Productivity
- Automation

  Rename 100 files that end in .html to .html.2013-08-08

```
ls *.html | while read file; do
  mv $file ${file}.$(date +%F)
done
```

Why learn the Linux command line?

- Sometimes you don't have a choice
- Productivity
- Automation

Rename 100 files that end in .html to .html.2013-08-08

```
ls *.html | while read file; do
  mv $file ${file}.$(date +%F)
done
```

Why learn the Linux command line?

- Sometimes you don't have a choice
- Productivity
- Automation

  Rename 100 files that end in .html to .html.2013-08-08

```
ls *.html | while read file; do
  mv $file ${file}.$(date +%F)
done
```

Riding the learning curve
Linux in 45 minutes?

- You won't learn all you need to know from this presentation
- To fully learn most things (Drupal, a new OS, BBQ), you have to spend the time to master it.
- The biggest challenge is often where to start:

  The Linux Command Line
  `http://linuxcommand.org/tlcl.php` (free pdf)

  Beginners Guide to `man` pages:
  `http://www.tfug.org/helpdesk/general/man.html`

  Bash-Scripting Guide:
  `http://www.tldp.org/LDP/abs/html/`

Riding the learning curve
Linux in 45 minutes?

- You won't learn all you need to know from this presentation
- To fully learn most things (Drupal, a new OS, BBQ), you have to spend the time to master it.
- The biggest challenge is often where to start:

  The Linux Command Line
  `http://linuxcommand.org/tlcl.php` (free pdf)

  Beginners Guide to `man` pages:
  `http://www.tfug.org/helpdesk/general/man.html`

  Bash-Scripting Guide:
  `http://www.tldp.org/LDP/abs/html/`

Riding the learning curve
Linux in 45 minutes?

- You won't learn all you need to know from this presentation
- To fully learn most things (Drupal, a new OS, BBQ), you have to spend the time to master it.
- The biggest challenge is often where to start:

  The Linux Command Line
  `http://linuxcommand.org/tlcl.php` (free pdf)

  Beginners Guide to `man` pages:
  `http://www.tfug.org/helpdesk/general/man.html`

  Bash-Scripting Guide:
  `http://www.tldp.org/LDP/abs/html/`

Riding the learning curve
Linux in 45 minutes?

- You won't learn all you need to know from this presentation
- To fully learn most things (Drupal, a new OS, BBQ), you have to spend the time to master it.
- The biggest challenge is often where to start:

  The Linux Command Line
  `http://linuxcommand.org/tlcl.php` (free pdf)

  Beginners Guide to man pages:
  `http://www.tfug.org/helpdesk/general/man.html`

  Bash-Scripting Guide:
  `http://www.tldp.org/LDP/abs/html/`

Riding the learning curve
Linux in 45 minutes?

- You won't learn all you need to know from this presentation
- To fully learn most things (Drupal, a new OS, BBQ), you have to spend the time to master it.
- The biggest challenge is often where to start:

  The Linux Command Line
  `http://linuxcommand.org/tlcl.php` (free pdf)

  Beginners Guide to `man` pages:
  `http://www.tfug.org/helpdesk/general/man.html`

  Bash-Scripting Guide:
  `http://www.tldp.org/LDP/abs/html/`

Riding the learning curve
Linux in 45 minutes?

- You won't learn all you need to know from this presentation
- To fully learn most things (Drupal, a new OS, BBQ), you have to spend the time to master it.
- The biggest challenge is often where to start:

  The Linux Command Line
  `http://linuxcommand.org/tlcl.php` (free pdf)

  Beginners Guide to `man` pages:
  `http://www.tfug.org/helpdesk/general/man.html`

  Bash-Scripting Guide:
  `http://www.tldp.org/LDP/abs/html/`

Basic Linux Ideas

- Everything is a file (or is represented by one)
- Linux is multi-user, `root` is the privileged administrative user
- A running Linux system is a collection of processes and files.
- Running the command `sudo` generally means you are doing something as the `root` user.

User accounts

- Login - sitting at the computer or remotely via ssh
- Home directory - This is your default initial working directory.

  You can use `~username` as a shortcut.

  Use '`echo ~username`' to see what it is.
- Group memberships - You can be a member of multiple groups, though it's not necessary
- Every process has an associated user (used for permissions)

File system layout

- The Linux filesystem starts at / and and continues downward in an upside down tree-like pattern with an arbitrary number of subdirectories.
- The / directory is the "root" of the filesystem.
- A path is a list of directories that have to be traversed to find a directory or file. If the target is a file, the file with its parent directories is the path

File system layout

- The Linux filesystem starts at / and and continues downward in an upside down tree-like pattern with an arbitrary number of subdirectories.
- The / directory is the "root" of the filesystem.
- A path is a list of directories that have to be traversed to find a directory or file. If the target is a file, the file with its parent directories is the path

File system layout

- The Linux filesystem starts at / and and continues downward in an upside down tree-like pattern with an arbitrary number of subdirectories.
- The / directory is the "root" of the filesystem.
- A path is a list of directories that have to be traversed to find a directory or file. If the target is a file, the file with its parent directories is the path

File system layout (cont)

- An absolute path - A location that starts from /
- A relative path - A location that starts relative to your current working directory (e.g. your home directory)
  - Use . to indicate the current directory
  - Use .. to indicate the parent directory
- Linux uses / as the path separator.
- New filesystems are spliced into the tree structure

File system layout (cont)

- An absolute path - A location that starts from /
- A relative path - A location that starts relative to your current working directory (e.g. your home directory)
  - Use . to indicate the current directory
  - Use .. to indicate the parent directory
- Linux uses / as the path separator.
- New filesystems are spliced into the tree structure

File system layout (cont)

- An absolute path - A location that starts from /
- A relative path - A location that starts relative to your current working directory (e.g. your home directory)
  - Use . to indicate the current directory
  - Use .. to indicate the parent directory
- Linux uses / as the path separator.
- New filesystems are spliced into the tree structure

File system layout (cont)

- An absolute path - A location that starts from /
- A relative path - A location that starts relative to your current working directory (e.g. your home directory)
  - Use . to indicate the current directory
  - Use .. to indicate the parent directory
- Linux uses / as the path separator.
- New filesystems are spliced into the tree structure

File system layout (cont)

- An absolute path - A location that starts from /
- A relative path - A location that starts relative to your current working directory (e.g. your home directory)
  - Use . to indicate the current directory
  - Use .. to indicate the parent directory
- Linux uses / as the path separator.
- New filesystems are spliced into the tree structure

File system layout (cont)

- An absolute path - A location that starts from /
- A relative path - A location that starts relative to your current working directory (e.g. your home directory)
  - Use . to indicate the current directory
  - Use .. to indicate the parent directory
- Linux uses / as the path separator.
- New filesystems are spliced into the tree structure

File system layout (cont)

Looking at apache configuration files and directories

```
/etc/httpd/
|-- conf
|   |-- httpd.conf
|   `-- magic
|-- conf.d
|   |-- mod_dnssd.conf
|   |-- php.conf
|   |-- README
|   `-- welcome.conf
|-- logs -> ../../var/log/httpd
|-- modules -> ../../usr/lib64/httpd/modules
`-- run -> ../../var/run/httpd
```

File system navigation and manipulation

- Every process (apache web server, your login shell, ssh, . . .) has a current working directory
- Relative paths are relative to whatever that directory is
- A few common commands

  `cd <directoryname>` : Change directory to "directoryname"
  `cd -` : takes you to your last directory,
  `cd` : with no arguments takes you to your HOME directory (abbreviated as `~/` sometimes)
  `pwd` : Print working directory.
  `mkdir` : Make a new directory.

File system navigation and manipulation (cont)

More commands

- `ls` : list directory contents, if no arguments are provided, assumes CWD
- `ls -ld <directoryname>` : to see attributes of the directory rather than the contents
- `ls -ltr <directoryname>` : to see info about files sorted by date with the newest at the bottom
- `ls -a <directoryname>` : to see "hidden" files, the ones that start with a `.`

File system navigation and manipulation (cont)

Just the files

- Create: Use a text editor: `nano`, `vi`, `emacs`
- Review: Use `less filename` or `cat filename`
- Destroy: Use `rm` or `rmdir`

File system navigation and manipulation (cont)

Just the files

- Create: Use a text editor: `nano`, `vi`, `emacs`
- Review: Use `less filename` or `cat filename`
- Destroy: Use `rm` or `rmdir`

File system navigation and manipulation (cont)

Just the files

- Create: Use a text editor: `nano`, `vi`, `emacs`
- Review: Use `less filename` or `cat filename`
- Destroy: Use `rm` or `rmdir`

Bash shell basics and commands

- Work through Linux Command Line and the Bash-scripting guide (aka too much to cover in one slide)
- If you get tired of typing the same thing over and over, you can use an `alias` to do text expansion.
  `alias goweb="cd /var/www/html; ls -l"`
- If you like a specific set of options that you don't want to type over and over, use an `alias`
  `alias ls="ls -aF"`
- If you want to keep the alias across multiple sessions, test it to make sure it does what you want. Then, add it to your shell initialization files (`~/.bash_profile` or `~/.bashrc`).

Bash shell basics and commands

- Work through Linux Command Line and the Bash-scripting guide (aka too much to cover in one slide)
- If you get tired of typing the same thing over and over, you can use an `alias` to do text expansion.
  ```
  alias goweb="cd /var/www/html; ls -l"
  ```
- If you like a specific set of options that you don't want to type over and over, use an `alias`
  ```
  alias ls="ls -aF"
  ```
- If you want to keep the alias across multiple sessions, test it to make sure it does what you want. Then, add it to your shell initialization files (`~/.bash_profile` or `~/.bashrc`).

Bash shell basics and commands

- Work through Linux Command Line and the Bash-scripting guide (aka too much to cover in one slide)
- If you get tired of typing the same thing over and over, you can use an `alias` to do text expansion.
  `alias goweb="cd /var/www/html; ls -l"`
- If you like a specific set of options that you don't want to type over and over, use an `alias`
  `alias ls="ls -aF"`
- If you want to keep the alias across multiple sessions, test it to make sure it does what you want. Then, add it to your shell initialization files (`~/.bash_profile` or `~/.bashrc`).

Bash shell basics and commands

- Work through Linux Command Line and the Bash-scripting guide (aka too much to cover in one slide)
- If you get tired of typing the same thing over and over, you can use an `alias` to do text expansion.
  `alias goweb="cd /var/www/html; ls -l"`
- If you like a specific set of options that you don't want to type over and over, use an `alias`
  `alias ls="ls -aF"`
- If you want to keep the alias across multiple sessions, test it to make sure it does what you want. Then, add it to your shell initialization files (`~/.bash_profile` or `~/.bashrc`).

Bash shell basics (cont)

.bash_profile vs .bashrc

- Both files are shell scripts that are processed at shell start up and configure the shell environment.
- `~/.bash_profile` is processed when a shell is a login shell and it is interactive (examples: an interactive ssh session, logging in at a console)
- `~/.bashrc` is processed when a shell is interactive but not a login shell (examples: opening a terminal from a Linux desktop, typing `"bash"` at the command line)
- Neither is read when a shell is non-interactive (e.g. running a shell script)
- Read the bash man page under INVOCATION for all of the details about which files are used when.

Bash shell basics (cont)

.bash_profile vs .bashrc

- Both files are shell scripts that are processed at shell start up and configure the shell environment.
- `~/.bash_profile` is processed when a shell is a login shell and it is interactive (examples: an interactive ssh session, logging in at a console)
- `~/.bashrc` is processed when a shell is interactive but not a login shell (examples: opening a terminal from a Linux desktop, typing `"bash"` at the command line)
- Neither is read when a shell is non-interactive (e.g. running a shell script)
- Read the bash man page under INVOCATION for all of the details about which files are used when.

Bash shell basics (cont)

.bash_profile vs .bashrc

- Both files are shell scripts that are processed at shell start up and configure the shell environment.
- `~/.bash_profile` is processed when a shell is a login shell and it is interactive (examples: an interactive ssh session, logging in at a console)
- `~/.bashrc` is processed when a shell is interactive but not a login shell (examples: opening a terminal from a Linux desktop, typing `"bash"` at the command line)
- Neither is read when a shell is non-interactive (e.g. running a shell script)
- Read the bash man page under INVOCATION for all of the details about which files are used when.

Bash shell basics (cont)

.bash_profile vs .bashrc

- Both files are shell scripts that are processed at shell start up and configure the shell environment.
- `~/.bash_profile` is processed when a shell is a login shell and it is interactive (examples: an interactive ssh session, logging in at a console)
- `~/.bashrc` is processed when a shell is interactive but not a login shell (examples: opening a terminal from a Linux desktop, typing `"bash"` at the command line)
- Neither is read when a shell is non-interactive (e.g. running a shell script)
- Read the bash man page under INVOCATION for all of the details about which files are used when.

Bash shell basics (cont)

.bash_profile vs .bashrc

- Both files are shell scripts that are processed at shell start up and configure the shell environment.
- `~/.bash_profile` is processed when a shell is a login shell and it is interactive (examples: an interactive ssh session, logging in at a console)
- `~/.bashrc` is processed when a shell is interactive but not a login shell (examples: opening a terminal from a Linux desktop, typing `"bash"` at the command line)
- Neither is read when a shell is non-interactive (e.g. running a shell script)
- Read the bash man page under INVOCATION for all of the details about which files are used when.

Bash shell basics (cont)

.bash_profile and .bashrc: Why should you care?

- On Redhat style systems, `~/.bash_profile` is configured to read and process `~/.bashrc` by default, so you can usually put your settings in `~/.bashrc` for your interactive shells.
- For non-interactive shells, they inherit environment variables from a parent process. Most often if you run a script which works fine when you are in interactive mode, but it fails in non-interactive mode (e.g. a `cron` job), you should check your assumptions about variable settings and working directory.
- If you find yourself saying, "It works on my account but not for the web user" ...

Bash shell basics (cont)

.bash_profile and .bashrc: Why should you care?

- On Redhat style systems, `~/.bash_profile` is configured to read and process `~/.bashrc` by default, so you can usually put your settings in `~/.bashrc` for your interactive shells.
- For non-interactive shells, they inherit environment variables from a parent process. Most often if you run a script which works fine when you are in interactive mode, but it fails in non-interactive mode (e.g. a `cron` job), you should check your assumptions about variable settings and working directory.
- If you find yourself saying, "It works on my account but not for the web user" ...

Bash shell basics (cont)

.bash_profile and .bashrc: Why should you care?

- On Redhat style systems, `~/.bash_profile` is configured to read and process `~/.bashrc` by default, so you can usually put your settings in `~/.bashrc` for your interactive shells.
- For non-interactive shells, they inherit environment variables from a parent process. Most often if you run a script which works fine when you are in interactive mode, but it fails in non-interactive mode (e.g. a `cron` job), you should check your assumptions about variable settings and working directory.
- If you find yourself saying, "It works on my account but not for the web user" ...

Permissions

- There are three categories for permissions per file/directory: User (the owner), Group, Other (all other users)
- The `root` user can change ownership with the `chown` command
- If you have appropriate permissions, you can use `chgrp` to change the group ownership
- You can use `chmod` (change mode) to change the permissions of a file
- The `root` user can ignore permissions (except in cases like SELinux)

Permissions

- There are three categories for permissions per file/directory: User (the owner), Group, Other (all other users)
- The `root` user can change ownership with the `chown` command
- If you have appropriate permissions, you can use `chgrp` to change the group ownership
- You can use `chmod` (change mode) to change the permissions of a file
- The `root` user can ignore permissions (except in cases like SELinux)

Permissions

- There are three categories for permissions per file/directory: User (the owner), Group, Other (all other users)
- The `root` user can change ownership with the `chown` command
- If you have appropriate permissions, you can use `chgrp` to change the group ownership
- You can use `chmod` (change mode) to change the permissions of a file
- The `root` user can ignore permissions (except in cases like SELinux)

Permissions

- There are three categories for permissions per file/directory: User (the owner), Group, Other (all other users)
- The `root` user can change ownership with the `chown` command
- If you have appropriate permissions, you can use `chgrp` to change the group ownership
- You can use `chmod` (change mode) to change the permissions of a file
- The `root` user can ignore permissions (except in cases like SELinux)

Permissions

- There are three categories for permissions per file/directory: User (the owner), Group, Other (all other users)
- The `root` user can change ownership with the `chown` command
- If you have appropriate permissions, you can use `chgrp` to change the group ownership
- You can use `chmod` (change mode) to change the permissions of a file
- The `root` user can ignore permissions (except in cases like SELinux)

Permissions (cont)

What are the permissions (e.g. in `ls -l` output):

Symbol	Meaning
`r`	Read
`w`	Write
`x`	Execute
`-`	Permission not granted

Permissions (cont)

Permissions mean different things based on whether they apply to a file or a directory.

Files:

- Read means the ability to look at the file
- Write means the ability to modify a file (but not delete)
- Execute means treat the file as a program that can be run

Permissions (cont)

Directory

- Read means the ability to list the contents of a directory (if execute is set too)
- Write means the ability to create/delete/rename the contents of a directory (if execute is set too)
- Execute means the ability to enter/traverse a directory (e.g. `cd directory`)

Permissions (cont)

Examples

- `cd /etc/httpd; ls -ld conf`

  ```
  drwxr-xr-x.  2 root root 4096 May 14 07:47 conf
  ```

  d rwx r-x r-x

  type user group other

- `cd /etc/httpd/conf; ls -l httpd.conf`

  ```
  -rw-r--r--. 1 root root 34418 Apr 29 03:09 httpd.conf
  ```

  - rw- r– r–

  type user group other

Permissions (cont)
Examples

- `cd /etc/httpd; ls -ld conf`

  `drwxr-xr-x.  2 root root 4096 May 14 07:47 conf`

  d rwx r-x r-x

  type user group other

- `cd /etc/httpd/conf; ls -l httpd.conf`

  `-rw-r--r--. 1 root root 34418 Apr 29 03:09 httpd.conf`

  - rw- r– r–

  type user group other

Permissions (cont)

Examples

- `cd /etc/httpd; ls -ld conf`

  ```
  drwxr-xr-x.  2 root root 4096 May 14 07:47 conf
  ```

  d rwx r-x r-x

  type user group other

- `cd /etc/httpd/conf; ls -l httpd.conf`

  ```
  -rw-r--r--. 1 root root 34418 Apr 29 03:09 httpd.conf
  ```

  - rw- r– r–

  type user group other

Permissions (cont)

Examples

- `cd /etc/httpd; ls -ld conf`

  ```
  drwxr-xr-x.  2 root root 4096 May 14 07:47 conf
  ```

  d rwx r-x r-x

  type user group other

- `cd /etc/httpd/conf; ls -l httpd.conf`

  ```
  -rw-r--r--. 1 root root 34418 Apr 29 03:09 httpd.conf
  ```

  - rw- r– r–

  type user group other

Permissions (cont)
Examples

- `cd /etc/httpd; ls -ld conf`

  ```
  drwxr-xr-x.  2 root root 4096 May 14 07:47 conf
  ```

  d rwx r-x r-x

  type user group other

- `cd /etc/httpd/conf; ls -l httpd.conf`

  ```
  -rw-r--r--. 1 root root 34418 Apr 29 03:09 httpd.conf
  ```

  - rw- r-- r--

  type user group other

Permissions (cont)

Examples

- `cd /etc/httpd; ls -ld conf`

  `drwxr-xr-x.  2 root root 4096 May 14 07:47 conf`

  d rwx r-x r-x

  type user group other

- `cd /etc/httpd/conf; ls -l httpd.conf`

  `-rw-r--r--. 1 root root 34418 Apr 29 03:09 httpd.conf`

  - rw- r– r–

  type user group other

Permissions (cont)

Examples

- `cd /etc/httpd; ls -ld conf`

  `drwxr-xr-x.  2 root root 4096 May 14 07:47 conf`

  d rwx r-x r-x

  type user group other

- `cd /etc/httpd/conf; ls -l httpd.conf`

  `-rw-r--r--. 1 root root 34418 Apr 29 03:09 httpd.conf`

  - rw- r– r–

  type user group other

Permissions (cont)

Examples

- `cd /etc/httpd; ls -ld conf`

  `drwxr-xr-x.  2 root root 4096 May 14 07:47 conf`

  d rwx r-x r-x

  type user group other

- `cd /etc/httpd/conf; ls -l httpd.conf`

  `-rw-r--r--. 1 root root 34418 Apr 29 03:09 httpd.conf`

  - rw- r– r–

  type user group other

Permissions (cont)

Examples

- `cd /etc/httpd; ls -ld conf`

  `drwxr-xr-x.  2 root root 4096 May 14 07:47 conf`

  d rwx r-x r-x

  type user group other

- `cd /etc/httpd/conf; ls -l httpd.conf`

  `-rw-r--r--. 1 root root 34418 Apr 29 03:09 httpd.conf`

  - rw- r– r–

  type user group other

Permissions (cont)

Examples

- `cd /etc/httpd; ls -ld conf`

  `drwxr-xr-x.  2 root root 4096 May 14 07:47 conf`

  d rwx r-x r-x

  type user group other

- `cd /etc/httpd/conf; ls -l httpd.conf`

  `-rw-r--r--. 1 root root 34418 Apr 29 03:09 httpd.conf`

  - rw- r– r–

  type user group other

Permissions (cont)
Changing permissions

- You use the `chmod` command to change the permissions
- You specify the mode you want symbolically, via `u,g,o` with `r,w,x` and `+ - =`
- To add the ability for the user to read and write to `filename`, do: `chmod u+rw filename`
- To set the group permissions to exactly read and execute, do: `chmod g=rx filename`
- To remove the ability for others to do anything, do: `chmod o-rwx filename`
- If you want them all together, do: `chmod u+rw,g=rx,o-rwx filename`
- See the man page for `chmod` for full details

Permissions (cont)
Changing permissions

- You use the `chmod` command to change the permissions
- You specify the mode you want symbolically, via `u,g,o` with `r,w,x` and `+ - =`
- To add the ability for the user to read and write to `filename`, do: `chmod u+rw filename`
- To set the group permissions to exactly read and execute, do: `chmod g=rx filename`
- To remove the ability for others to do anything, do: `chmod o-rwx filename`
- If you want them all together, do: `chmod u+rw,g=rx,o-rwx filename`
- See the man page for `chmod` for full details

Permissions (cont)
Changing permissions

- You use the `chmod` command to change the permissions
- You specify the mode you want symbolically, via `u,g,o` with `r,w,x` and `+ - =`
- To add the ability for the user to read and write to `filename`, do: `chmod u+rw filename`
- To set the group permissions to exactly read and execute, do: `chmod g=rx filename`
- To remove the ability for others to do anything, do: `chmod o-rwx filename`
- If you want them all together, do: `chmod u+rw,g=rx,o-rwx filename`
- See the man page for `chmod` for full details

Permissions (cont)

Changing permissions

- You use the `chmod` command to change the permissions
- You specify the mode you want symbolically, via `u,g,o` with `r,w,x` and `+ - =`
- To add the ability for the user to read and write to `filename`, do: `chmod u+rw filename`
- To set the group permissions to exactly read and execute, do: `chmod g=rx filename`
- To remove the ability for others to do anything, do: `chmod o-rwx filename`
- If you want them all together, do: `chmod u+rw,g=rx,o-rwx filename`
- See the man page for `chmod` for full details

Permissions (cont)

Changing permissions

- You use the `chmod` command to change the permissions
- You specify the mode you want symbolically, via `u,g,o` with `r,w,x` and `+ - =`
- To add the ability for the user to read and write to `filename`, do: `chmod u+rw filename`
- To set the group permissions to exactly read and execute, do: `chmod g=rx filename`
- To remove the ability for others to do anything, do: `chmod o-rwx filename`
- If you want them all together, do: `chmod u+rw,g=rx,o-rwx filename`
- See the man page for `chmod` for full details

Permissions (cont)

Changing permissions

- You use the `chmod` command to change the permissions
- You specify the mode you want symbolically, via `u,g,o` with `r,w,x` and `+ - =`
- To add the ability for the user to read and write to `filename`, do: `chmod u+rw filename`
- To set the group permissions to exactly read and execute, do: `chmod g=rx filename`
- To remove the ability for others to do anything, do: `chmod o-rwx filename`
- If you want them all together, do: `chmod u+rw,g=rx,o-rwx filename`
- See the man page for `chmod` for full details

Permissions (cont)
Changing permissions

- You use the `chmod` command to change the permissions
- You specify the mode you want symbolically, via `u,g,o` with `r,w,x` and `+ - =`
- To add the ability for the user to read and write to `filename`, do: `chmod u+rw filename`
- To set the group permissions to exactly read and execute, do: `chmod g=rx filename`
- To remove the ability for others to do anything, do: `chmod o-rwx filename`
- If you want them all together, do: `chmod u+rw,g=rx,o-rwx filename`
- See the man page for `chmod` for full details

Permissions operate on a first match basis. Keeping that in mind, here's a process for checking them:

- Which user are we dealing with (hbrown)?

  `id -un`, reading config files, reading man pages, research, ...

- Which groups is this user a member of (hbrown, apache)?

  `groups <username>`

- What are the permissions on the file/directory starting from / and check all of its parent directories?

  `ls -ld /full/path/to/file`

Permissions operate on a first match basis. Keeping that in mind, here's a process for checking them:

- Which user are we dealing with (hbrown)?

  `id -un`, reading config files, reading man pages, research, ...
- Which groups is this user a member of (hbrown, apache)?

  `groups <username>`
- What are the permissions on the file/directory starting from / and check all of its parent directories?

  `ls -ld /full/path/to/file`

Permissions operate on a first match basis. Keeping that in mind, here's a process for checking them:

- Which user are we dealing with (hbrown)?

  `id -un`, reading config files, reading man pages, research, ...

- Which groups is this user a member of (hbrown, apache)?

  `groups <username>`

- What are the permissions on the file/directory starting from `/` and check all of its parent directories?

  `ls -ld /full/path/to/file`

Troubleshooting Permissions

- In my experience, permission problems are the number one stumbling block for users doing webapps (if you've ever done `chmod 777 dir`, you've experienced this)..
- Performing a simple drupal install from a permissions perspective.
- If needed (and you have root), make use of 'su - apache -s /usr/bin/bash'

Troubleshooting Permissions

- In my experience, permission problems are the number one stumbling block for users doing webapps (if you've ever done `chmod 777 dir`, you've experienced this)..
- Performing a simple drupal install from a permissions perspective.
- If needed (and you have root), make use of 'su - apache -s /usr/bin/bash'

Troubleshooting Permissions

- In my experience, permission problems are the number one stumbling block for users doing webapps (if you've ever done `chmod 777 dir`, you've experienced this)..
- Performing a simple drupal install from a permissions perspective.
- If needed (and you have root), make use of '`su - apache -s /usr/bin/bash`'

Next steps

- Learn about shell globbing (wildcards)
- Learn about regular expressions
- Learn how to use vi or emacs
- Write shell scripts to automate your tasks