Commandline Customization, Tips & tricks

How to make your linux environment your own

What is the commandline in linux

- In linux your command line is a program called a shell.
- The shell is an independent process that will be the parent to other processes, even other shells.
- There are many shells and shell family trees. The default is often /bin/sh (though this is often implemented as a feature set of /bin/bash) and IBM really wants you to be using ksh. They have some different features and syntax. We'll focus on bash

So shells run commands?

- In a shell there are 4 types of commands, shell builtins, external programs, aliases, and functions.
- The builtin commands tend to do something or act as a control structure.
- The external programs are called with a full path “/bin/bash”, a relative path “../bin/bash”, or directly “bash”
  - Directly issued commands are found with the $PATH environmental variable which is a colon separated list of directories to look in for those programs

- The aliases are macro’s instantiated by the user.

- bash supports user defined functions

You mentioned variables and macros those seem like a pain to setup each time?

- Your shell process does some initialization, first from the system and then from your home directory. So in your personal initialization you can put the things you want every time.
- In this initialization process your shells will be flagged login shells or non-login shells. This is a legacy thing that mostly just makes customizing shell initialization grosser than it needs to be.
  - For bash when your shell is firing up there are a number of different filenames it will look for in your home directory and will only use the first one it finds. The list is different depending if it's a login shell. These files are treated like a list of commands that are done one after another as a "script"

$PATH customization

- You should only prepend and append to the $PATH, the system shell initialization will grab all the goodies and put them in your $PATH, let it help you out.
- You should never put relative directories in your path, this can be used to get you to run malware.
- Things may appear multiple times in the directories in $PATH. The first one that appears is the one that's going to win
- Ex: `export PATH=/awesome/bin:$PATH:/meh/bin`

aliases!

- You probably already have some of these. You can check with the which command. Ex: `which ls`
- These are macro work at the beginning of a command.
- Great for long commands with lots of arguments, especially if those commands are in directories you don't want in your path.
- Arguments go after the alias'ed command
- Ex: `alias lshell=" /opt/umsappy/bin/shell -L ridge"`

I want an alias, but BETTER

- If you want an alias where the argument isn't just at the end or is used multiple times, you can declare a function
- You probably already have some of these. They don't show up in which. You can check with the declare command. Ex: `declare -F`
- These arguments go to numbered variables $1, $2, etc
- This can make debugging hard
- Ex: `mkcd () { mkdir -p -- $1 && cd -P -- $1 ;}`
- Can also do the same thing by making a script

The command history can be tweaked a lot.

- $HISTFILE contains where your history is kept
- $HISTSIZE and $HISTFILESIZE control the size of history
- $HISTCONTROL can be used to ignore duplicates, commands that begin with a space, or both with the values ignoredups ignorespace ignoreboth
- $HISTIGNORE is a colon separated list of command patterns to ignore. Good for a concise history bad for debugging
- $HISTFORMAT can be used to add things like the time executed to the history

Other bash behaviors that can be customized with variables.

- $TMOUT sets the shell idle timeout in seconds.
- $PAGER sets the command for anything that has builtin paging
- $EDITOR sets the default editor for commands like git commit
- $TMPDIR sets the directory used for storing temp files, good for when /tmp is full

Customizing the Prompt! (This could be a class)

- There are actually 5 prompts. PS0, PS1, PS2, PS3, PS4
- PS1 is the main one.
- Before displaying the prompt the commands in $PROMPT_COMMAND will be run, it's a colon separated list
- PS1 is set as a string with a bunch of macro replacements
  - \u the user
  - \h the hostname
  - \W the current directory
  - \a the bell character
  - \$ $ unless you are root than #
  - \# the number of this command in this shell session
  - … Seriously just look these up
- $PROMPTDIRTRIM controls how deep in the prompt directory tree to go
- EX: [\u@\h \W]\$ results in prompts like [ridge@moon etc]$

Motd or the login banner.

-Motd stands for message of the day.

- On unix /etc/motd typically hosts a static login banner. Put important messages or ASCII art in it.

- If you want dynamic just put something in the login shell initialization