

# PSUMAC205: Intro to Scripting

(builds on PSUMAC 202)

PENNSTATE



MACADMINIS  
CONFERENCE  
2011

# My Intro...

- ✦ Jeremy Hill
  - ✦ Programmer/Analyst, Clearinghouse for Military Family Readiness
  - ✦ formerly...Systems Design Specialist, Systems Administrator, Computer Support Assistant -- ITS
  - ✦ best known as a Systems Integrator...

# Why do we want to script?

As seen on a t-shirt: Go away or I will  
replace you with a very small shell script

We're lazy...not  
that it's a bad  
thing!

If you script them, repetitive or  
menial tasks take care of  
themselves...only to be heard  
from if you want them to be  
heard from. Just set it and  
forget it!

It's FUN!

# When do we want to script?

If the logic is  
simple enough

If the task is repetitive

If the task feels like a  
waste of your time...

If you can write down the  
instructions for someone  
else to follow....

If users need to  
do something  
that shouldn't  
need sys admin  
time

**SCRIPT IT!**

# How to script?

- ✦ Aye, there's the rub....
- ✦ So many different ways to write scripts!
  - ✦ Shell (aka Bash)
  - ✦ P's -- Perl, PHP or Python
  - ✦ C's -- C, C++, Java
  - ✦ Ruby
  - ✦ Applescript/Automator

# How to script?

Yes, we will be talking concepts of programming in relation to system administration in this session...

# Shell Scripting

- ✦ What is shell?
  - ✦ A working environment (command line)
  - ✦ An interpreter
  - ✦ Takes input from stdin or file and figures out what to do with it
  - ✦ sh, csh, tcsh, bash, ksh

# Shell Scripting

- ✦ What is a shell script?
  - ✦ series of command line instructions with optional user input and logic wrapping
  - ✦ bash

# Shell Scripting

- ✦ Main parts
  - ✦ `#!/bin/bash`
  - ✦ `echo`
  - ✦ `read`
  - ✦ `if then elif else fi`
  - ✦ `loops`
  - ✦ `exit`

# Shell Scripting Start

- ✦ open a file, make extension `.sh`
- ✦ at the top `#!/bin/bash`
- ✦ `echo "Hello World!"`
- ✦ `exit`

Run it! `sh ./file.sh`

# Shell Scripting Read User Input

- ✦ echo "Please type your name:"
- ✦ read NAME
- ✦ change "World" to "\$NAME"

Run it! `sh ./file.sh`

# Shell Scripting Assigning Variables

- ✦ `a=blah`
  - ✦ no spaces
  - ✦ `=` is assignment operator, can also be test operator (synonym for `==` in the right context) so BE CAREFUL!!
  - ✦ variables are untyped, BE AWARE!

# Shell Scripting Referencing Variables

- `echo $a`
  - `$` must precede variable for the interpreter to grab the value instead of the name

# Shell Scripting Logic

```
if then elif else fi
```

```
if [ test condition ]
```

```
then
```

```
do something
```

```
elif [ other test ]
```

```
then
```

```
do something
```

```
else
```

```
do something else
```

```
fi
```

# Shell Scripting Logic

## File Test Operators

- ✦ -e (file exists)
- ✦ -f (file is a regular file not a directory or device)
- ✦ -h (file is a symlink)
- ✦ -s (file is not empty)
- ✦ -r (file can be read)
- ✦ -w (file can be written)
- ✦ -x (file can be executed)

# Shell Scripting Logic

## File Test Operators

- ✦ -O (user is owner of file)
- ✦ -G (group-id of file is same as user)
- ✦ -N (file modified since last read)
- ✦ -nt (newer than)
- ✦ -ot (older than)
- ✦ ! (negation)

# Shell Scripting Logic

## File Test Operators

```
if [ -e file ]  
then  
do (or don't do) something  
elif [ file1 -nt file2 ]  
do something different  
fi
```

# Shell Scripting Looping

- ✦ for loops
  - ✦ VERY different from any other programming looping structure
  - ✦ Takes an arg and assigns it to each element in a list
- ✦ while loops
- ✦ until loops

# Shell Scripting Exit

- ✦ `exit --` will exit with the status of last command run
- ✦ `exit 0 --` will exit as “successful”
- ✦ `exit 1-255 --` indicative of some “unsuccessful” exit

# Shell Scripting Example

Tempting the demo gods....

# What if I don't wanna shell?

- ✦ Ok, use another “scripting” language (Perl, PHP, Python, Ruby...even C, etc.)
- ✦ invoke via separate shell script (or not) same as you would run by command line
  - ✦ ex: `/usr/bin/perl perlscript.pl`
  - ✦ provides excellent looping workaround
  - ✦ flexibility of using the right tool for the job

# What if I want to do it “The Apple Way”

- ✦ Applescript
  - ✦ /Applications/Utilities/Applescript Editor
- ✦ Automator
  - ✦ /Applications/Automator
- ✦ Either of these can run other scripts too
- ✦ Can provide a nice interface to otherwise clunky scripts

# Applescript/Automator

Tempting the demo gods....

# What if I want to repeat it?

- ✦ Cron
  - ✦ time based execution
  - ✦ easier, less features
- ✦ launchd
  - ✦ time or event based execution
  - ✦ harder, more features

# Cron

- ✦ crontab -e
- ✦ runs as user who owns the crontab

- ✦ basic setup:



```
30 0 * * * sh /usr/bin/sync.sh >/dev/null 2>&1
```

Every day, every month at 00:30 run sync.sh and just erase output

# Cron

- Don't ignore output, instead e-mail me!

```
MAILTO: yoursysadmin@psu.edu  
30 0 * * * sh /usr/bin/sync.sh
```

OR

```
30 0 * * * sh /usr/bin/sync.sh | mail -s "Subject  
Cron" yurothersysadmin@psu.edu
```

# Cron

- ✦ By default, cron is very “chatty”
- ✦ Try using quiet modes of commands
- ✦ Send stdout to /dev/null redirect stderr to be sent via e-mail

# launchd

- ✦ PID 1 -- yep, it starts the OS at boot
- ✦ more event based
- ✦ takes the place of cron, initd, rc, watchdog, (x)inetd
- ✦ more difficult to configure, more powerful too
- ✦ interface via launchctl command
- ✦ items are configured via .plist files
- ✦ no time to dig in...

# Other types of “scripting”

- ✦ Many management apps have scripting concepts
- ✦ Puppet “scripts”
- ✦ JavaScript
  - ✦ kidding.....
- ✦ Any others?

# Useful links

- <http://tldp.org/LDP/abs/html/> -- Advanced Bash-Scripting Guide
- <http://steve-parker.org/sh/sh.shtml> -- Steve's Bourne / Bash shell scripting tutorial
- <http://adminschoice.com/crontab-quick-reference>
- <http://www.afp548.com/article.php?story=20050620071558293> (or just go to [afp548.com](http://www.afp548.com) and search for launchd)
- <http://www.devdaily.com/mac-os-x/launchd-examples-launchd-plist-file-examples-mac>
- <http://www.macenterprise.org/>
- <http://macadmins.psu.edu/>
- Google terms

Scripting Brainstorm!!!

# End of PSUMAC205

Questions?

[j.hill@psu.edu](mailto:j.hill@psu.edu)