Bash Lightning Talk

Updated: 2015-11-16

JP Vossen, CISSP
bashcookbook.com

http://www.jpsdomain.org/public/bash_lightning.pdf
http://www.jpsdomain.org/public/bash_lightning.odp

debugging

- DO NOT CALL YOUR TEST SCRIPT 'test'!

- PS4='+xtrace $LINENO: '
  - First character is duplicated to show nesting level, that's why I have '+' there
  - $LINENO should be in the default PS4 prompt!

- bash -n path/to/script   # gross syntax check
- bash -x path/to/script   # run-time debugging

- set -x & set +x   # debug on / off
- set -v & set +v   # verbose on / off

function log

```
function Log {
    printf '%(%F %T %z)T: %s\n' '-1' "$*"
}
```

- printf *%(datefmt)T* added in **bash 4.2**, but default was Epoch (1969-12-31 19:00:00 EDT)
  - So this won't work in the stone-age default `bash` on a Mac!
- Default changed to "now" in bash 4.3
- Use '-1' for consistent behavior
  - Can also use '-2' for time shell was started or 'Epoch integer' to display some other time

Snippets

```
############################ JP settings ###################################
# Only if interactive with a terminal!
[ -t 1 ] && {
    PS1='\n[\u@\h:T\l:L$SHLVL:C\!:J\j:\D{%Y-%m-%d_%H:%M:%S_%Z}]\n$PWD\$ '
    alias ll='ls --color=auto -F -h -l'
    alias md='mkdir'
    alias vzip='unzip -lvM'
    export LESS='--LONG-PROMPT --LINE-NUMBERS --ignore-case --QUIET --no-init'
    bind '"\e[A": history-search-backward'
    bind '"\e[B": history-search-forward'
    bind '"\C-i": menu-complete'
    bind 'set completion-ignore-case on'
    export HISTCONTROL='erasedups:ignoredups:ignorespace'
    export HISTIGNORE='&:[ ]*'    # bash >= 3, omit dups & lines starting with space
    export MANWIDTH=80            # man page width, use < 80 if COLUMNS=80 & less -N
    export VISUAL='vi'
    function mcd {
        mkdir -p "$1"
        cd "$1"
    }
}
```

String Stuff

- Pattern Matching:
  - ${variable#pattern} ${variable##pattern}
  - ${variable%pattern} ${variable%%pattern}
  - ${variable/pattern/string} ${variable//pattern/string}
  - So, `basename $0` is ${0##*/}
- String Operations:
  - ${variable:-word} # **Return** a default value
  - ${variable:=word}   # **Set** a default value
  - ${variable:?word}   # Catch undefined vars (error)
  - ${variable:+word}   # Test existence (or return null)
  - ${variable:offset:length}   # Substrings

URLs, Wrap-up and Q&A

- URLs:
  - TONS of resources: http://www.bashcookbook.com/bashinfo/

  - These slides: http://www.jpsdomain.org/public/bash_lightning.pdf
    or http://www.jpsdomain.org/public/bash_lightning.odp

  - Bash vs. Dash: http://www.jpsdomain.org/public/2008-JP_bash_vs_dash.pdf and
    aptitude install devscripts then use *checkbashisms*

  - The sample script: http://www.jpsdomain.org/public/cdburn

  - STDIN, STDOUT, STDERR: http://en.wikipedia.org/wiki/Standard_streams

  - Revision Control: http://www.jpsdomain.org/public/Revision_Control_for_the_Rest_of_Us.pdf
    and (older) http://www.jpsdomain.org/public/PANTUG_2007-06-13_appd=Revision_Control=JP.pdf

  - Windows Shell Scripting (cmd.exe): http://www.jpsdomain.org/windows/winshell.html

  - BASH Prompt HOWTO: http://www.tldp.org/HOWTO/Bash-Prompt-HOWTO/index.html

  - Cygwin: http://www.cygwin.com/

  - UnxUtils: http://unxutils.sourceforge.net/

  - GNU Win32 ports: http://sourceforge.net/projects/gnuwin32/

  - Win32 Perl http://www.activestate.com/solutions/perl/

  - Mac; this is awesome http://examples.oreilly.com/0636920032731/Terminal_Crash_Course.pdf

- Questions?

- I'm on the PLUG list... jp@jpsdomain.org