A Nifty Log-book Script to Seamlessly log what Commands were run and Where for Easy Organisation of Your Various Projects

Imperial College
London

1st March 2016
Ben Krikler
My Super Hacky Logbook
Script Proof-of-principle Tool that Might be Useful to Others Already but is Seriously Super Hacky

Imperial College
London

1st March 2016
Ben Krikler
Introduction

- I found myself writing bigger and bigger command lines
- Commands are often only relevant if issued in the right location
- My BASH_HISTORY set up was already a big help:
  - `shopt -s histappend`
  - `export HISTIGNORE="ls:rm -r*:\[bf\]g:exit:history *[0-9]*"`
  - `export HISTCONTROL="ignoredups:erasedups"`
  - `export HISTTIMEFORMAT="%A %d/%m/%y %T "`
  - `export HISTSIZE=100000`
  - `export HISTFILESIZE=100000`
History controls in bashrc

- **Enable history appending rather than over-writing**
  - `shopt -s histappend`

- **Ignore certain commands**
  - `export HISTIGNORE="ls:rm -r*:^[bf]g:exit:history *[0-9]*"`

- **Only log duplicated commands once**
  - `export HISTCONTROL="ignoredups:erasedups"

- **Store the time each command was executed**
  - `export HISTTIMEFORMAT="%A %d/%m/%y %T "`

- **Keep a lot of history**
  - `export HISTSIZE=100000`
  - `export HISTFILESIZE=100000`

- **But no nice way to store the working directory in bash history**
  - As far as I could see
Log-book script design goals

- Stores selected commands to a log book file
  - Include the time, date and directory of execution
- Needs to be told which commands to store; not logging everything
- Specific to a given project, but works from any sub-directory of that project
  - Similar to using git
- Can print and review later on
Obtaining and Installing

- **Clone from github**
  - `cd ~/bin`

- **Make sure directory is contained in PATH**
  - `PATH=$HOME/bin:"$PATH"

- **Need to have bash write the history to the file after each command. In bashrc do one of:**
  - `PROMPT_COMMAND='history -w'
  - `prompt_cmd(){
      history -w
  }
PROMPT_COMMAND='prompt_cmd'

- **Make sure to source your bashrc before using**
Usage

$ lb -h
lb: [options] [command]

Valid commands:
init            Initialise a log in the current directory
append          Append commands to the log
print           Print the contents of the log

Valid global options:
-h, --help      Show this help and quit
--debug         Enable debug output
-l, --log-file <filename>        Provide the name of a log file to use
--log-file-format <filename>      The log file format to search for
-n <number>      The number of previous commands to append to the log
--hist-file <filename>     The name of the history file to extract commands from
Usage

- **Go to a project directory**
  - $ cd Inane_example/

- **Initialise the log book:**
  - $ lb init
    
      Initialised new log at: /vols/comet00/users/bek07/Inane_example/.log

- **Do some magical work on the command line**
  - $ ...
    $ for nfiles in 1 4 10; do
      IcedustControl_SubmitToBatch -Q '-q heplong.q'
      -I <(shuf -n $nfiles InputFiles) -s $(TD)-files_$nfiles RunAnalysis.cfg ;done

- **Save the last command that was executed to the log book**
  - $ lb append
  - (or just lb)
    
      Appending the following to log
      (/vols/comet00/users/bek07/Inane_example/.log):
      9860 Tuesday 01/03/16 09:32:02 for nfiles in 1 4 10; do
      IcedustControl_SubmitToBatch -Q '-q heplong.q' ; -I <(shuf -n $nfiles InputFiles) -s $(TD)-files_$nfiles RunAnalysis.cfg ;done
Usage

- Change down into a sub-directory and run some commands
  - `cd well_named_sub_directory`
  - `max_shift=50; for shift in {030..070..10}; do for height in {600..700..20}; do echo DIO Height: $height, Shift: $shift; sed -e 's!%MAX_SHIFT !$max_shift!'g' -e 's!%SHIFT%!'$shift!'g' -e 's!%DIOHEIGHT%!'$height!'g' ElectronAcceptance.cfg.in > batch3/ElectronAcceptance-shiftBack_${max_shift}-shift_${shift}-DIOHeight_${height}.cfg; done; done`

- Append to log book:
  - `lb`
  - Appending the following to log (/vols/comet00/users/bek07/Inane_example/.log):
    99860 Tuesday 01/03/16 09:33:59 max_shift=50; for shift in {030..070..10}; do ...............
Usage

- **Review the log:**
  - `$ lb print`

```bash
= /vols/comet00/users/bek07/Inane_example =
99860  Tuesday 01/03/16  09:32:02 for nfiles in 1 4 10; do
IcedustControl_SubmitToBatch -Q '-q heplong.q' ; -I <(shuf -n
$nfiles InputFiles) -s $(TD)-files_$nfiles RunAnalysis.cfg
;done

= /.../Inane_example/well_named_sub_directory =
99860  Tuesday 01/03/16  09:33:59 max_shift=50;for shift in
{030..070..10}; do for height in {600..700..20}; do echo DIO
Height: $height, Shift: $shift;  sed -e 's!%MAX_SHIFT
%!'$max_shift'!g' -e 's!%SHIFT%!'$shift'!g' -e 's!%DIOHEIGHT
%!'$height'!g' ElectronAcceptance.cfg.in >
batch3/ElectronAcceptance-shiftBack_${max_shift}-shift_{$shift}-DIOHeight_${height}.cfg ;done ;done

- **Time stamp formatting is coming from raw BASH history controls, use**
  HISTTIMEFORMAT **to control**
Future Additions
(Making it Less Hacky)

- Nicer printing
- Finer controls over command appending, Eg:
  - Using Regex to find last command
  - Last n-th command (sort of works already)
- Switch away from shell scripting (→ python)
- Append files
  - Simple scripts and text files
  - Plots
- Version control hooks
  - What tag of a piece of software was in use?
  - Tag of experiment software or analysis code
Summary

- Shown a log book script that's in its early stage of development
- Feel free to use!
- Let me know if there are issues
- Welcome to contribute
  - Fix issues
  - Add features
  - Just open a pull request when ready
- Intend to develop further in the future (post-PhD)