Local History Aachen-style

Carmen Hager, Sasa Hasan, and Jon Graehl

July 23, 2014

(Text written by Jonathan May.)

This comes from Carmen Heger and Sasa Hasan, by way of Aachen, where Arne Mauser was the apparent originator. It's since been adapted and cleaned up by Jon Graehl. Please feel free to share this but cite them (not Jonathan May, unless you want to cite this specific note). If you want to skip to the instructions they're at the bottom.

The idea is a .history file is created in every directory you work in with the list of all the commands you typed (except for a few that match a list of boring stuff like ls).

This is if you want to save yourself the trouble of cut-and-pasting into a README in each directory. I still do that, actually, but this functions as a readme of last resort. It's also great for seeing what steps others have done if you're trying to debug their inability to reproduce your results.

example:

```
[jonmay@Jonathan-Mays-MacBook-Pro demo]$ echo "README: this is a
demonstration"
[jonmay@Jonathan-Mays-MacBook-Pro demo]$ time
[jonmay@Jonathan-Mays-MacBook-Pro demo]$ whoami
[jonmay@Jonathan-Mays-MacBook-Pro demo]$ mkdir foo
[jonmay@Jonathan-Mays-MacBook-Pro demo]$ cd foo
[jonmay@Jonathan-Mays-MacBook-Pro foo]$ echo "README: this is where i’m
going to do my secret work"
[jonmay@Jonathan-Mays-MacBook-Pro foo]$ for i in ‘seq 1 10‘; do echo $i
$((i/2)); done | sed s’/0/h/g’ > blarg
[jonmay@Jonathan-Mays-MacBook-Pro foo]$ cd ..
[jonmay@Jonathan-Mays-MacBook-Pro demo]$ cat .history
2014-01-16.17-39-12.Jonathan-Mays-MacBook-Pro.local echo "README: this is
a demonstration"
2014-01-16.17-39-16.Jonathan-Mays-MacBook-Pro.local time
2014-01-16.17-39-20.Jonathan-Mays-MacBook-Pro.local whoami
2014-01-16.17-39-26.Jonathan-Mays-MacBook-Pro.local mkdir foo
[jonmay@Jonathan-Mays-MacBook-Pro demo]$ cat foo/.history
2014-01-16.17-39-43.Jonathan-Mays-MacBook-Pro.local echo "README: this is
where i’m going to do my secret work"
2014-01-16.17-40-41.Jonathan-Mays-MacBook-Pro.local for i in ‘seq 1 10‘;
do echo $i $((i/2)); done | sed s’/0/h/g’ > blarg
```

Instructions: Put this in your bashrc/bash_profile (I'm never clear on which one, even after reading this. (For Aliya Deri on Linux, .bashrc works fine.)

CODE:

```
# User specific aliases and functions
# output '0' if no matches (not using -q because PIPEFAIL might not work)
function useHistory() {
    \egrep -v '^top$|^pwd$|^ls$|^ll$|^l$|^lt$|^cd |^h |^bg$|^fg$'
}
function owner() {
    \ls -ld "${1:-$PWD}" | \awk '{print $3}'
}
function lastHistoryLine() {
    history 1 | HISTTIMEFORMAT= \sed 's:^ *[0-9]* *::'
}
function localHistory()
{
    if [[ `owner` = ${USER:=$(whoami)} ]] ; then
        local line=$(lastHistoryLine)
        if [[ $(echo "$line" | useHistory) ]]; then
            # date hostname cmd >> $PWD/.history
            echo $(date +'%FT%T').${HOST:=$(uname -n)} "$line" >> .history
            2>> /dev/null
        fi
    fi
}
function addPromptCommand() {
    # convenience command to enable adding of the prompt
    if [[ $PROMPT_COMMAND != *$1* ]]; then
        if [[ $PROMPT_COMMAND ]]; then
            # exists with content
            if [[ $PROMPT_COMMAND =~ \;[\ \ ]*$ ]]; then
                # already ends in semicolon and space or tab
                PROMPT_COMMAND+="$1"
            else
                # does not end in semicolon
                PROMPT_COMMAND+=" ; $1"
            fi
        else
            PROMPT_COMMAND="$1"
        fi
        export PROMPT_COMMAND
    fi
}
function h() {
    # look for something in your cwd's .history file
    if [[ -r .history ]]; then
        if ! [[ $1 ]]; then
            \cat .history
        else
            # permit looking for multiple things: h 'foo bar' baz
            local f
            for f in "$@"; do
                grep -a "$f" .history # allow aliased grep --color etc
            done
        fi
    else
        echo "Warning: .history not accessible" 1>&2
    fi
}

### to enable,

addPromptCommand localHistory
```