

Re: bash history customizing

Source: <http://unix.derkeiler.com/Newsgroups/comp.unix.shell/2004-05/1063.html>

From: Stephane CHAZELAS (*this.address_at_is.invalid*)

Date: 05/28/04

Date: Fri, 28 May 2004 13:55:36 +0000

2004-05-28, 14:12(+02), Gabkin:

[...]

> *As for zsh, I'm a bash man! (but does zsh have an evolutionary history*

> *facility like this?)*

Yes, and much more. Some customizations possible for history:

APPEND_HISTORY <D>

If this is set, zsh sessions will append their history list to the history file, rather than overwrite it. Thus, multiple parallel zsh sessions will all have their history lists added to the history file, in the order they are killed.

EXTENDED_HISTORY <C>

Save each command's beginning timestamp (in seconds since the epoch) and the duration (in seconds) to the history file. The format of this prefixed data is:

```
`:<BEGINNING TIME>:<ELAPSED SECONDS>:<COMMAND>'
```

HIST_ALLOW_CLOBBER

Add `|' to output redirections in the history. This allows history references to clobber files even when CLOBBER is unset.

HIST_BEEP <D>

Beep when an attempt is made to access a history entry which isn't there.

HIST_EXPIRE_DUPS_FIRST

If the internal history needs to be trimmed to add the current command line, setting this option will cause the oldest history event that has a duplicate to be lost before losing a unique event from the list. You should be sure to set the value of HISTSIZE to a larger number than SAVEHIST in order to give you some room for the duplicated events, otherwise this option will behave just like HIST_IGNORE_ALL_DUPS once the history fills up with unique events.

HIST_FIND_NO_DUPS

When searching for history entries in the line editor, do not display duplicates of a line previously found, even if the duplicates are not contiguous.

HIST_IGNORE_ALL_DUPS

If a new command line being added to the history list duplicates an older one, the older command is removed from the list (even if it is not the previous event).

HIST_IGNORE_DUPS (-h)

Do not enter command lines into the history list if they are duplicates of the previous event.

HIST_IGNORE_SPACE (-g)

Remove command lines from the history list when the first character on the line is a space, or when one of the expanded aliases contains a leading space. Note that the command lingers in the internal history until the next command is entered before it vanishes, allowing you to briefly reuse or edit the line. If you want to make it vanish right away without entering another command, type a space and press return.

HIST_NO_FUNCTIONS

Remove function definitions from the history list. Note that the function lingers in the internal history until the next command is entered before it vanishes, allowing you to briefly reuse or edit the definition.

HIST_NO_STORE

Remove the history (fc -l) command from the history list when invoked. Note that the command lingers in the internal history until the next command is entered before it vanishes, allowing you to briefly reuse or edit the line.

HIST_REDUCE_BLANKS

Remove superfluous blanks from each command line being added to the history list.

HIST_SAVE_NO_DUPS

When writing out the history file, older commands that duplicate newer ones are omitted.

HIST_VERIFY

Whenever the user enters a line with history expansion, don't execute the line directly; instead, perform history expansion and reload the line into the editing buffer.

INC_APPEND_HISTORY

This options works like APPEND_HISTORY except that new history lines are added to the \$HISTFILE incrementally (as soon as they are

entered), rather than waiting until the shell is killed. The file is periodically trimmed to the number of lines specified by `$SAVEHIST`, but can exceed this value between trimmings.

`SHARE_HISTORY <K>`

This option both imports new commands from the history file, and also causes your typed commands to be appended to the history file (the latter is like specifying `INC_APPEND_HISTORY`). The history lines are also output with timestamps ala `EXTENDED_HISTORY` (which makes it easier to find the spot where we left off reading the file after it gets re-written).

By default, history movement commands visit the imported lines as well as the local lines, but you can toggle this on and off with the `set-local-history` zle binding. It is also possible to create a zle widget that will make some commands ignore imported commands, and some include them.

If you find that you want more control over when commands get imported, you may wish to turn `SHARE_HISTORY` off, `INC_APPEND_HISTORY` on, and then manually import commands whenever you need them using ``fc -RI'`.

I have:

```
HISTFILE=~/.zsh-history.$ZSH_VERSION
HISTSIZ=500
SAVEHIST=50000
setopt HIST_IGNORE_ALL_DUPS HIST_REDUCE_BLANKS \
  HIST_SAVE_NO_DUPS HIST_VERIFY INC_APPEND_HISTORY NO_BANG_HIST
```

--

Stephane