Tip 79: Try a multiplexer

Ben Klemens

8 March 2012

level: command-liner
purpose: so very much

I always have two terminals open when coding: one with the code in an editor, and one for compiling and running the program (probably in a debugger). Working on an R package, I'll have a terminal with C side code, a terminal with the R side code, a compilation/run terminal, and because R is so undocumented a window with R's source code.

Deftly jumping among terminals has suddenly become incredibly important.

There are two terminal multiplexers to choose from, on either side of the great GNU|BSD rivarly: GNU Screen[1] and tmux[2].

Not to take sides on the license thing, but tmux is more recently written and has learned from some of GNU Screen's mistakes. I like Screen's copy mode better. Maybe next year all that will be reversed.

Your package manager will probably install either or both of them.

Both work via a single command key. GNU Screen defaults to <ctrl>-A. Tmux defaults to <ctrl>-B, but the consensus seems to be that everybody remaps that to use <ctrl>-A instead, by adding

```
unbind C-b
set -g prefix C-a
bind a send-prefix
```

to `.tmux_conf` in their home directory. There are lots of other things that you can add to your configuration files. I'm not going to read you the manual here, just point out to you how really fabulous these things are so you're compelled to try them. [When searching for tips and documentation, notice that *GNU Screen* is the name to type into your search engine, because *Screen* by itself will get you nowhere.]

Once you've done that, <ctrl>-A <ctrl>-A jumps between two windows, and you can RTFM for the <ctrl>-A (otherkey) combinations that will let you step forward or backward in the window list, or display the full list of windows so you can just pick from the list.

So both of these guys solve the multi-window problem. But they do so very much more:

[1]`http://www.gnu.org/software/screen/`
[2]`http://tmux.sourceforge.net/`

- OK, you have a session with several windows up. <ctrl>-A D will detach the session, meaning that your terminal no longer displays the various virtual terminals under the multiplexer's control. But they're still running in the background.
  - Got a spotty connection to your server in Belize? No problem, because after you're unexpectedly disconnected, you can reattach to pick up where you left off. [If the detach didn't happen cleanly, use `screen -r -d` or `tmux detach; tmux attach`.]
  - Log in to your server in the morning, work all day with GNU Screen/Tmux, detach at the end of the day (or let your session time out, which is an auto-detach). Tomorrow morning, use `screen -r` or `tmux attach` to reattach to the full set of screens you left last night. No need to kill ten minutes restoring where you were last night.
  - While you were sleeping, anything you had running is still running. So if you need something to run all night and don't remember the commands and tricks to get a program to keep running after you close the terminal (hint: `nohup`), then don't worry about it: the multiplexer keeps its virtual terminals open even after you've detached.
- There's a cut/paste feature.
  - OK, now we're really mouseless: once in copy mode, you can page through what's passed through the terminal lately, highlight a section, and copy it to the multiplexer's internal clipboard via <ctrl>-A [(or one or two other keys). Then, back in regular mode, <ctrl>-A] pastes.
  - While you're browsing for things to cut, you can scroll through the history comfortably. Both systems provide a search while in copy mode, which means that you finally have a search function for your terminal.

OK, how's that for a sell? These multiplexers really take that last step from making the terminal a place to work to being a fun place to work.