GNU Screen for the Impatient

Paul Nijjar

KWartzlab

October 28, 2009

Why Screen?

- AUGH! Too many terminal windows!
- AUGH! Too many terminal tabs!
- AUGH! My remote terminal session is flaky!
- AUGH! I have a big job running but I need to log out!

Getting Started

All commands begin with a control sequence, which by default is `<Ctrl>+a` .

There are tonnes of commands (`<ctrl>-a ?`) but you only need a few to get started:

`<ctrl>-a c`	Add a screen
`<ctrl>-a "`	List/select screens
`<ctrl>-a a`	Toggle to last screen
`<ctrl>-a <space>`	Cycle through screens
`<ctrl>-a A`	Give a screen a useful title

More!

There is lots I won't tell you about. Go learn it yourself.

`<ctrl>-a [` Copy/scrollback (`<ctrl>-a ]` to paste)

`<ctrl>-a d` Detach your session (`screen -r` to reattach)

Also: things having to do with setting "hardstatus lines" at the bottom of your terminal,

Also: remapping keys, making copy/paste easier, ...

Lipstick on a pitbull: byobu

Byobu is an Ubuntu project which adds some configuration scripts to `screen` . It comes installed with Karmic.

- TUI configuration editor (okay)
- Easy to configure hardstatus section (good)
- Remappings of the easy keys to function keys (meh)
- Angry fruit salad colours (yay!)

Fortunately standard commands work in Byobu. However, the config file locations are separate.

Setting titles automatically

The default screen titles are dumb (and make `<ctrl>-a "` useless).

Setting screen titles manually is stupid.

What can we do?

runwithtitle.sh

Put this in your `$HOME/bin` directory:

```
#!/bin/sh

#set the screen title
progname=‘basename $0‘
  case $TERM in
      screen*)
          screen -X title "$progname $@"
          ;;
  esac
/usr/bin/$progname $@
```

Using `runwithtitle.sh`

Now (also in `$HOME/bin`) alias commonly-used interactive commands:

```
ln -s runwithtitle.sh ssh
ln -s runwithtitle.sh aptitude
ln -s runwithtitle.sh w3m
ln -s runwithtitle.sh vi
```

Of course, if your binaries do not live in `/usr/local/bin` then you need to make adjustments.

Hacking the bash prompt

Adding the following snippet to `.bashrc` will set screen titles to your cwd (if you’re lucky):

```
# This sends the sequence ESC k (\w) ESC \
# The \w is a prompt string.
case "$TERM" in
screen*)
    SCREENTITLE=’\[\ek\w\e\\\]’
    PS1="${SCREENTITLE}${PS1}"
    ;;
esac
```