# tmux
## A Great Terminal Multiplexer

lgfang

plexus platform

May 18, 2012

# Agenda

# Agenda

## It is annoying

- Loosing everything when the connection to server broken?

# It is annoying

- Switching among dozens of terminal windows.

# It is annoying

- Drag&Drop *again and again* to tile terminal windows

# It is annoying

- Can not share *current* terminal with others.

# It is annoying

- Can not open one more window over one connection.

# Agenda

## Detach and re-attach

```
tmux [new -s session_name]
tmux attach [ -t session_name]
tmux list-sessions
```

# Windows grouped into sessions

# Split at will

# Or, preset layouts
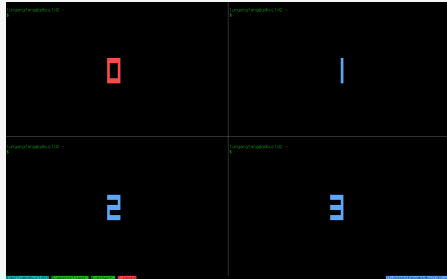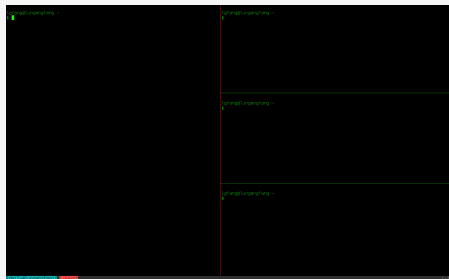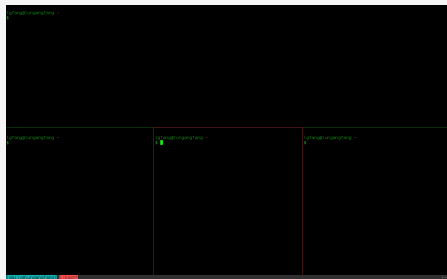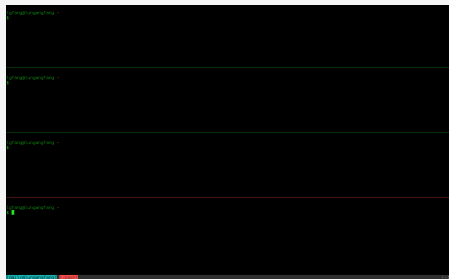
# When shared, auto-resize according to smallest window

# The Simpler The Better

# Easy to configure

Configure file `$HOME/.tmux.conf`

```
# default prefix 'C-b'
set-option -g prefix F12
bind-key | split-window -h
bind-key _ split-window -v
```

# Even support mouse

```
set-option -g mouse-select-pane on
set-option -g mouse-resize-pane on
set-option -g mouse-select-window on
set-window-option -g mode-mouse on
```

# Easy to use

This is enough to get you started[1].

| | | |
|---:|:---:|:---:|
| Detach | `F12` | `d` |
| New window | `F12` | `c` |
| Split vertically | `F12` | `_` |
| Split horizontally | `F12` | `|` |
| Change layout | `F12` | `space` |
| Kill pane [2] | `F12` | `x` |

---

[1] I bind prefix to `F12`

[2] "kill pane" when it does not response.

# Eye candy

```
[lgfang@huang ~]$
[lgfang@huang ~]$
[lgfang@huang ~]$ sed -n '58,71p' .tmux.conf
#** colour-theme-blocks
set-option -g status-left "#[bg=cyan,fg=black][tmux:#S@#h]"
set-option -g status-right "[#T]"
set-option -g status-attr none
set-option -g status-bg colour236
set-option -g status-fg white
set-window-option -g window-status-attr reverse
set-window-option -g window-status-fg green
set-window-option -g window-status-current-attr reverse
set-window-option -g window-status-current-fg red
set-window-option -g window-status-activity-attr reverse
set-window-option -g window-status-activity-fg yellow
set-option -g pane-border-fg green
set-option -g pane-active-border-fg red
[lgfang@huang ~]$
```

```
[lgfang@huang ~]$

[lgfang@huang ~]$ for i in {0..255} ; do printf "
\x1b[48;5;${i}m "; done;printf "\x1b[0m\n"

[lgfang@huang ~]$ cd tmp
[lgfang@huang tmp]$
```

[daily@huang]  1:bash  2:bash  3:bash*  4:bash#  9:qdbbuild2-              [lgfang@huang:~/tmp]

# Screen log

Just save when you want. Save all screen output of current pane: F12
Ctrl + h , or

   Start logging F12  h

   Stop logging F12  H  [3]

```
bind-key C-h command-prompt -p "save log to:" "copy-mode; send-keys
    'M->' C-e C-space 'M-<' C-a C-w; save-buffer %%"
bind-key h pipe-pane "cat >>$HOME/#S-#I-#P.log" \; display-message
    "Start logging($HOME/#S-#I-#P.log)"
bind-key H pipe-pane \; display-message "Stop logging(#S-#I-#P.log)"
```

---

[3] shift + h

# Easy to install

If you can not persuade your system administrator, install one under your HOME.

```
# install libevent if not already
tar xvfz libevent-1.4.14b-stable.tar.gz && cd
    libevent-1.4.14b-stable
./configure --prefix=$HOME/.local && make && make install
# set up LD_LIBRARY_PATH, for example
echo 'LD_LIBRARY_PATH=$HOME/.local/lib' >> ~/.bash_profile

tar xvfz tmux-1.6.tar.gz && cd tmux-1.6/
./configure --prefix=$HOME/.local
    CFLAGS="-I$HOME/local/include"
    LDFLAGS="-L$HOME/local/lib"
make && make install
# Add $HOME/.local/bin to PATH, for example
echo 'PATH=$HOME/.local/bin:$PATH' >> ~/.bash_profile
```

# Easy scripting

It is simple to control tmux in a shell script. Refer to
for an example.

# Free as in "freedom"

Last but not least: it is BSD-licensed.
Which means, it is free to use, modify, re-distribute, or even *make profit from (sell)* it.

What are you waiting for? Attach to tmux once you connected to a server.

# Agenda

3 Notes on terminal emulator

# Configuring terminal emulator

You connect to servers with a terminal emulator[4]. Therefore, you need to make sure:

- your terminal emulator cope with UTF8 chars correctly.
- your terminal emulator does not intercept hot-key sent to tmux.
- hold down `shift` when you copy & paste using mouse as usual.

Here is an example:

---

### Example (Seting up putty)

`Window` $\rightarrow$ `Translation` $\rightarrow$ `data ...  character set` : *UTF-8* [a]

`Terminal` $\rightarrow$ `Keyboard` $\rightarrow$ `The Function keys and keypad` : *XtermR6*

---

[a]your putty can receive and display Chinese characters(UTF-8) now

---

[4]such as putty, secureCRT etc

## auto-login and hotkeys

Some can not live without

- auto-login: the emulators remember hostip/username/password.
- auto-hotkey: a windows software enable you to define your own hotkeys.

Do not worry, I wrote "to" in expect, which is (IMHO) more convenient.

# Agenda

Feel free to leave. And, get back when you find yourself need more skills.

# Panes

Select pane `F12` `Tab` (or one of `←` `↑` `→` `↓` )

Swap two adjacent panes `F12` `{`

Break current pane to a new window `F12` `!`

Join a pane to current window `F12` `j` *x.y*

Swap with pane y of window x `F12` `m` *x.y*

```
bind-key Tab select-pane -t:.+
bind-key BTab select-pane -t:.-
set-option -g display-time 3000
bind-key j command-prompt "join-pane -s '%1'"
bind-key m command-prompt -p "move (swap) current pane with: "
    "swap-pane -s %1"
```

# Windows

Select window `F12` `0` - `9`
Last window `F12` `1`
Rename window `F12` `,` *new name*

# Sessions within tmux client

New session `F12` *:new-session -s name*

Choose(switch) session `F12` `s`

move window among sessions `F12` `:move-window -t sessionName`

# Copy mode

Entering "copy mode" to view,search and copy screen output.

- Entering copy mode `F12` `[` or `F12` `PageUp`
- View/search: using emacs (or vi) key bindings according to configure.
- Paste: `F12` `]`
- Show "clip board": `F12` `=`

```
set-window-option -g mode-keys emacs
set-option -g history-limit 600000
```

## Nested tmux

There are sometimes nested tmux sessions (refer to ⟨▸ my start up script⟩).

1. Start a tmux session on local linux box.
2. From that tmux session, ssh to a server and run another tmux session.

To send prefix to

nested tmux session    Press `F11` .

outer tmux session    `F12` .

tmux session nested twice    `F11` `C-b`

one more layer[5] of nest    `F11` `C-b` `C-b` .

```
bind-key -n F11 send-prefix
bind-key C-b send-prefix
```

---

[5]do you really want a nested nested nested ... session?

# Command Line Commands

Please keep in mind that tmux CLI commands almost always help you out when there is no hotkey available.

# My start-up script

Here is an example for controlling tmux in shell script. It automatically start a local session, and a remote session if possible.

```bash
#!/bin/bash
# Created: Fang lungang 2011; Modified: Fang lungang 04/01/2012 09:59>

SESSION_NAME=daily;SVR="135.252.41.248"

tmux has-session -t $SESSION_NAME 2>/dev/null && exec tmux attach -t $SESSION_NAME

tmux new-session -d -s $SESSION_NAME
if hostname | grep -q "\(lgfang\|lungang\)"; then # On my local PC
    if ping -c 1 -w 1 $SVR; then
        tmux new-window -d -n qdbbuild2 -t $SESSION_NAME:9 \
            "ssh -t lungangfang@$SVR 'source .bash_profile; .local/bin/tmux-go.sh'"
        tmux select-window -t $SESSION_NAME:9
        # make it feels like I'm working on SVR locally
        tmux set-option -g -t $SESSION_NAME status off
        tmux set-option -g -t $SESSION_NAME prefix F11
    fi
else # on servers, a window for emacs (start it if needed).
    tmux new-window -d -n emacsclient -t $SESSION_NAME:9 \
        "emacsclient -t || (emacs --daemon && emacsclient -t)"
fi
exec tmux attach -t $SESSION_NAME
```

# Agenda

5 Q&A

# Thanks

# Questions

## Thanks

# Questions
Read The ... Manual for more.

# Reference card according to my configure

|  |  |  |
|---:|:---|:---|
| Detach | `F12` `d` | |
| New window | `F12` `c` | |
| Split vertically | `F12` `-` | |
| Split horizontally | `F12` `|` | |
| Change layout | `F12` `space` | |
| Kill pane | `F12` `x` | |
| Select window | `F12` `0` - `9` | |
| Last window | `F12` `l` | |
| Rename window | `F12` `,` *new name* | |
| Save log | `F12` `Ctrl` + `h` | |
| Select pane | `F12` `Tab` (or one of `←` `↑` `→` `↓` ) | |
| Break current pane | `F12` `!` | |
| Join a pane | `F12` `j` *x.y* | |
| Swap two panes | `F12` `{` | |