

# Tmux - A Great Terminal Multiplexer

## Table of Contents

- 1. About this slides
- 2. The Itches
- 3. Why Tmux
- 4. Why NOT xxx
  - 4.1. Why NOT VNC
  - 4.2. Why NOT GNU "screen"
  - 4.3. Why NOT secureCRT
- 5. Feature Highlights
  - 5.1. Detach and re-attach
  - 5.2. Multiple "tabbed" windows on one connection
  - 5.3. Remote share and auto-resize
  - 5.4. Split windows at will
  - 5.5. Tile windows with preset layouts
  - 5.6. Group windows into sessions
  - 5.7. Easy to use
  - 5.8. Supports scripting
  - 5.9. Free (as in "freedom")
- 6. Demo
- 7. Installation
  - 7.1. The most simple way
  - 7.2. The preferred way
  - 7.3. Install from source
- 8. Setup
- 9. Basic operations
- 10. Enable mouse support
- 11. Intermediate Usages
  - 11.1. Important concepts
  - 11.2. Customization
  - 11.3. Eye candy
  - 11.4. Windows
  - 11.5. Panes
  - 11.6. Copy mode
  - 11.7. Sessions within tmux client
  - 11.8. Run complicated commands
  - 11.9. About configuration
- 12. Advanced
  - 12.1. Log screen outputs
  - 12.2. Another way
  - 12.3. Nesting tmux
  - 12.4. Types of hot-keys
  - 12.5. Maximize a pane temporary
  - 12.6. Command line auto-complete
- 13. Q&A
- 14. Backup
  - 14.1. Reference card according to my conf
  - 14.2. Auto-login and hot-keys

# 1 About this slides

This slides explains why we want to use tmux as well as the basic features and configurations of tmux.

This page is a html slides. To view it in slides mode, press **[x]** then click mouse to start browsing. For more help, press **[?]**.

## 2 The Itches

- Loosing **everything** when the connection to a server break
- Cannot share **current** terminal with others
- Drag & drop **again and again** to tile terminal windows
- Switching among **dozens** of terminal windows
- Unable (or tedious) to open multiple connections

## 3 Why Tmux

A terminal multiplexer capable of

1. tabbed windowing
2. detach/re-attach
3. remote-share, pair-working (esp. remote pair-working)
4. window split and tiling
5. grouping windows
6. re-sizing windows
7. ...

Meanwhile, it is

1. Easy to install & use
2. Well defined command line interface
  - For interactive use and for scripting.
3. Well documented
4. BSD licensed

## 4 Why NOT xxx

### 4.1 Why NOT VNC

The cons of VNC:

1. Depends on X (GUI environment).
2. Consume more CPU and network traffic (I think).
3. Can not resize the desktop (Latest realvnc can).
4. Does not tile (To be accurate, you can start a tiling WM in VNC. Average engineers just don't bother or able to do that).
5. Non-native windows that response slowly

The pros of VNC:

1. Uses dedicated password instead of login/password for sharing
2. Supports X applications

## 4.2 Why NOT GNU "screen"

Tmux is kind of the next-gen of screen. People say tmux is superior since it

1. Is easier to configure.
2. Is easier to split window and change layout.
3. Is able to resize window.
4. Is easier to scripting.
5. Cleaner architecture, code, document etc.

## 4.3 Why NOT secureCRT

I prefer "putty + tmux + to" since, as far as I know, secureCRT

1. can not detach/attach
2. can not group windows per projects(or bug/escalation etc.)
3. can not remote share
4. can not tile (?)
5. seems it does not support UTF-8
6. not convenient if the host list is really long
7. can not jump to destination host via an intermediate server
8. your hot-keys is not available other PCs, or after re-install
9. non-free

# 5 Feature Highlights

## 5.1 Detach and re-attach

- Network unstable?
- Has to bring laptop home or meeting room?
- Laptop crashed?
- Mistakenly closed the terminal?

Does NOT matter! just re-attach.

## 5.2 Multiple "tabbed" windows on one connection

```

lgfang@lungangfang ~/tmp/tmux-1.6
$ tmux -V
tmux 1.6

lgfang@lungangfang ~/tmp/tmux-1.6
$ █

[daily@lungangfang] 1:~ 2:~ 3:~ 4:~ 5:~ 6:~ 7:~ 8:~ 9:qdbbuid2# [~/tmp/tmux-1.6]

```

## 5.3 Remote share and auto-resize

```

requests from Unconfigured source - logged in sipa_protocol_error
;;
telica 2011-12-05 10:43:52
M 0 COMPLD
  "IP=169.254.125.3-10000-DEFAULT-NOTPRSNT-NOTPRSNT:TGN=1"
/* RTRV-SIP-TGMAPADDR */
;
Telica> rtrv-sip-tgmaFADDR;
telica 2011-12-05 10:44:08
M 0 COMPLD
  "IP=169.254.125.3-10000-DEFAULT-NOTPRSNT-NOTPRSNT:TGN=1"
/* RTRV-SIP-TGMAPADDR */
;
Telica>
telica 2011-12-05 10:48:51
A 2 REPT EVT COM
  "Com:GP,TC,2011-12-05,10-48-51,,,,,\SHELF-0-ACH-3-INST-1-
requests from Unconfigured source - logged in sipa_protocol_error
;
telica 2011-12-05 10:53:51
A 3 REPT EVT COM
  "Com:GP,TC,2011-12-05,10-53-51,,,,,\SHELF-0-ACH-3-INST-1-
requests from Unconfigured source - logged in sipa_protocol_error
;
telica 2011-12-05 10:58:51
A 4 REPT EVT COM
  "Com:GP,TC,2011-12-05,10-58-51,,,,,\SHELF-0-ACH-3-INST-1-
requests from Unconfigured source - logged in sipa_protocol_error
;
]

.ddl.sql does not exist. IGNORE
Mon Dec 5 09:56:23 CST 2011: Note: run /Telica/swCPU/CurrRel/sy
.isup.ddl.sql finished.
Mon Dec 5 09:56:23 CST 2011: CLEANING SORM Data
Mon Dec 5 09:56:23 CST 2011: CLEANING SORM Data Done
Mon Dec 5 09:56:23 CST 2011: Internal Data Loading Done
Mon Dec 5 09:56:23 CST 2011:
Mon Dec 5 09:56:23 CST 2011: =====
Mon Dec 5 09:56:23 CST 2011: === END MYSQL INSTALLATION ===
Mon Dec 5 09:56:23 CST 2011: =====
Mon Dec 5 09:56:23 CST 2011:
Telica_IP already supports IPv6 data
mgt0 data is provisioned, creating mgt0 aliases with IPADDR=135
255.255.255.240 PORT=68
sigad data is provisioned, creating sigad aliases with IPADDR=1
55.255.255.248 PORT=66 IPV6=none/0
mgt1 data is provisioned, creating mgt1 aliases with IPADDR=135
255.255.255.240 PORT=68
sigal data is provisioned, creating sigal aliases with IPADDR=1
55.255.255.248 PORT=66 IPV6=none/0
WARN: No AMC SS7 card found
<qdi02-s00c04h0:root>/Telica/swCPU/CurrRel/system/scripts:
> reboot
Broadcast message from root (pts/0) (Mon Dec 5 10:28:41 2011):
The system is going down for reboot NOW!
<qdi02-s00c04h0:root>/Telica/swCPU/CurrRel/system/scripts:
>
telnet> q
Connection closed.
lungangfang@qdbuild2 -
$

1:expect* 2:aluac* 3:expect 4:~ 5:~ 6:~ 7:~ 8:~ 9:emac*~
[daily@qdbuild2]

```

This is a window auto-resized since my peer's terminal window is smaller than mine.

## 5.4 Split windows at will

### 5.4.1 Real-world example: working with TL1

```

ibc4-s00c03h0:root>/Telica/swCPU/CurrRel/system/scripts:
> vi ALGP_Create_MGC.sql
<ibc4-s00c03h0:root>/Telica/swCPU/CurrRel/system/scripts:
> grep -i siproute *
ALGP_Create_MGC.sql:CREATE TABLE siproute (
ALGP_Create_MGC.sql:sipRouteName VARCHAR(11) NOT NULL,
ALGP_Create_MGC.sql:PRIMARY KEY(sipRouteName, db_index)
ALGP_Create_MGC.sql:sipRouteName VARCHAR(11),
defaults.sql:INSERT INTO DbpTableInit VALUES (4581, 9999, 'siproute');
Binary file schemaupgrade matches
<ibc4-s00c03h0:root>/Telica/swCPU/CurrRel/system/scripts:
>
shell will timeout in 60 seconds due to inactivity
<ibc4-s00c03h0:root>/Telica/swCPU/CurrRel/system/scripts:
> -ksh: timed out waiting for input

Connection to 10.84.13.207 closed.
lungangfang@qdbuild2 ~ /tmp
$

db_thrAlertLvl | int(11) | NO | NULL |
db_thrAbateLvl | int(11) | NO | NULL |
db_genAlarm | tinyint(3) unsigned | NO | NULL |
8 rows in set (0.00 sec)

mysql> select * from sessioncacdef;
Empty set (0.00 sec)

mysql>

ibc4_1 2011-09-21 22:50:37
M 0 COMPLD
/* ENT-CAC-SESSION */

ibc4_1 2011-09-21 22:50:37
A 21 REPT DBCHG
"DATE=2011-09-21,TIME=22-50-37,USERID=plexview1,DBCHGSEQ=80:ENT-CAC-SESSION:1,ItCacSession:MAXTOTALCONCURRCALLS=32000,MAXINGRESSCONCURRCALLS=32000,MAXEGRESSCONCURRCALLS=32000,ALARM=INGRESS&EGRESS&TOTAL"
V
Telica>

ent-prfl-sip::1::bearERNA=y,codeCREORDER=agnostic,sendPRACK=y,hdrPassSetName=hdrPassSetAll,NONINVITEPROXY=Y,TRANSITINDIALOGREFER=Y,HRTBTOPTIONS=LOCALHOST,PASSTHRUF
CONVY,PASSTHRUPALD=y,PASSTHRUCONTACT=y;
ent-prfl-sdp1::1::code1=PCMW,code2=PCMA,code3=G726,code4=G729;
ENT-CAC-SESSION::defaultCacSession::MAXTOTALCONCURRCALLS=32000,MAXINGRESSCONCURRCALLS=32000,MAXEGRESSCONCURRCALLS=32000,ALARM=INGRESS&EGRESS&TOTAL;

-UU-:8---F2 tmp.txt 40k (29,0) (Text yas AC)-----
[daily@qdbuild2] 2:expect* 3:expect- 4:bash qdbuild2:-

```

One window for each of the following:

1. Emacs to compose TL1 commands
2. TL1 session to launch TL1 commands
3. MYSQL session to inspect DB
4. Another session to query DB to verify TL1 commands
5. Misc. tasks

## 5.4.2 Real-world example: upgrade ATCA blades simultaneously

```

Creating /home file system
Creating /logs file system
Creating /storage file system
Done.

Mounting filesystem for LV set jumpstart....
LV /dev/VG1/LV0001.root for LV set jumpstart mounted on /mnt/jumpstart
LV /dev/VG1/LV0001.appl for LV set jumpstart mounted on /mnt/jumpstart/appl
boot partition mounted on /mnt/jumpstart/boot
LV /dev/VG1/LV0001.var for LV set jumpstart mounted on /mnt/jumpstart/var
LV home for LV set common mounted on /mnt/jumpstart/home
LV logs for LV set common mounted on /mnt/jumpstart/logs
LV storage for LV set common mounted on /mnt/jumpstart/storage
LV cores for LV set common mounted on /mnt/jumpstart/cores
Done.

Creating fstab file for LV set jumpstart....
Done.

Updating /tmp/fstab file for LV set jumpstart....
Done.
lvm_admin "init" action succeeded.
Done

Installing operating system -- please be patient

ID 37017186
Preparing packages for installation...
smartmontools-5.38-2.el5
OpenIPMI-libs-2.0.16-7.el5
OpenIPMI-2.0.16-7.el5
molene-1.4-1.2
NOTE: RPMpkg package installation completed

Installing grub for /dev/sda
echo "Creating static route to DHCP server: 135.251.237.21"
echo "Creating static route to LSP server: 135.252.41.246"

Installing SSH keys
Jumpstart of qd19-s00c03h0 succeeded at Wed Aug 15 02:03:02 UTC 2012

INIT: Stopping sshd: [ OK ]
Stopping xinetd: [ OK ]
Shutting down kernel logger: [ OK ]
Shutting down system logger: [ OK ]
Shutting down interface eth0:

or network, you are consenting to such monitoring and information retrieval
for law enforcement and other purposes. Users should have no expectation
of privacy as to any communication on or information stored within the
computer system or network, including information stored locally or remotely
on a hard drive or other media in use with this computer system or network.

Command line editor? (default = vi): ne

TERM: vt100-nav + EDITOR: + DISPLAY:

<qd19-s00c05h0:JUMPSTART ENVIRONMENT>/root:
> jumpstart

Starting jumpstart of qd19-s00c05h0 at Wed Aug 15 02:03:03 UTC 2012
Version=5.0

Hardware Clock updated to Wed Aug 15 02:03:08 UTC 2012.

Transferring /data0/jumpstart/R23.17.00/RPMpkg.zip from 135.251.237.21

Transferring RPM files:
% Total % Received % Xferd Average Speed Time Time Time Current
Dload Upload Total Spent Left Speed
9 168M 9 15.5M 0 0 11.1M 0 0:00:15 0:00:01 0:00:14 11.1M

(qdbuild2.daily) 1:qd19-h7* 2:qd11-mil- 3:bash 9:sh#

Bye
<qd19-s00c06h0:root>/root:
>
<qd19-s00c06h0:root>/root:
>
<qd19-s00c06h0:root>/root:
> reboot

Broadcast message from root (pts/0) (Wed Aug 15 10:02:53 2012):

The system is going down for reboot NOW!
<qd19-s00c06h0:root>/root:
>

#
# exit
Connection closed by foreign host.
root@qd19-s00c07h0:~#
root@qd19-s00c07h0:~#
root@qd19-s00c07h0:~# console -s0 -c 6
Enter the SQL password:
Error in open session response message : insufficient resources for session

Error: Unable to establish IPMI v2 / RMCP+ session
Error: No response activating SQL payload
root@qd19-s00c07h0:~#
lungangfang@qdbuild2:~/tmp

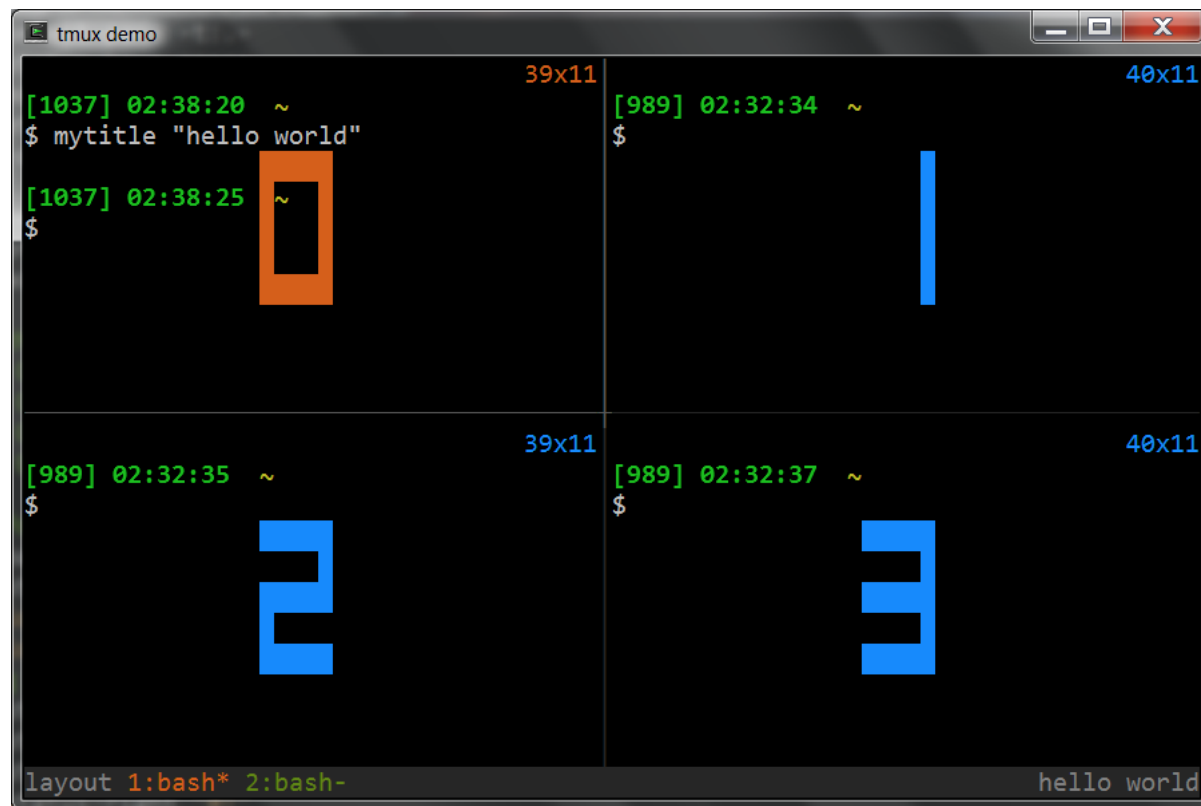
```

Each window monitors one blade.

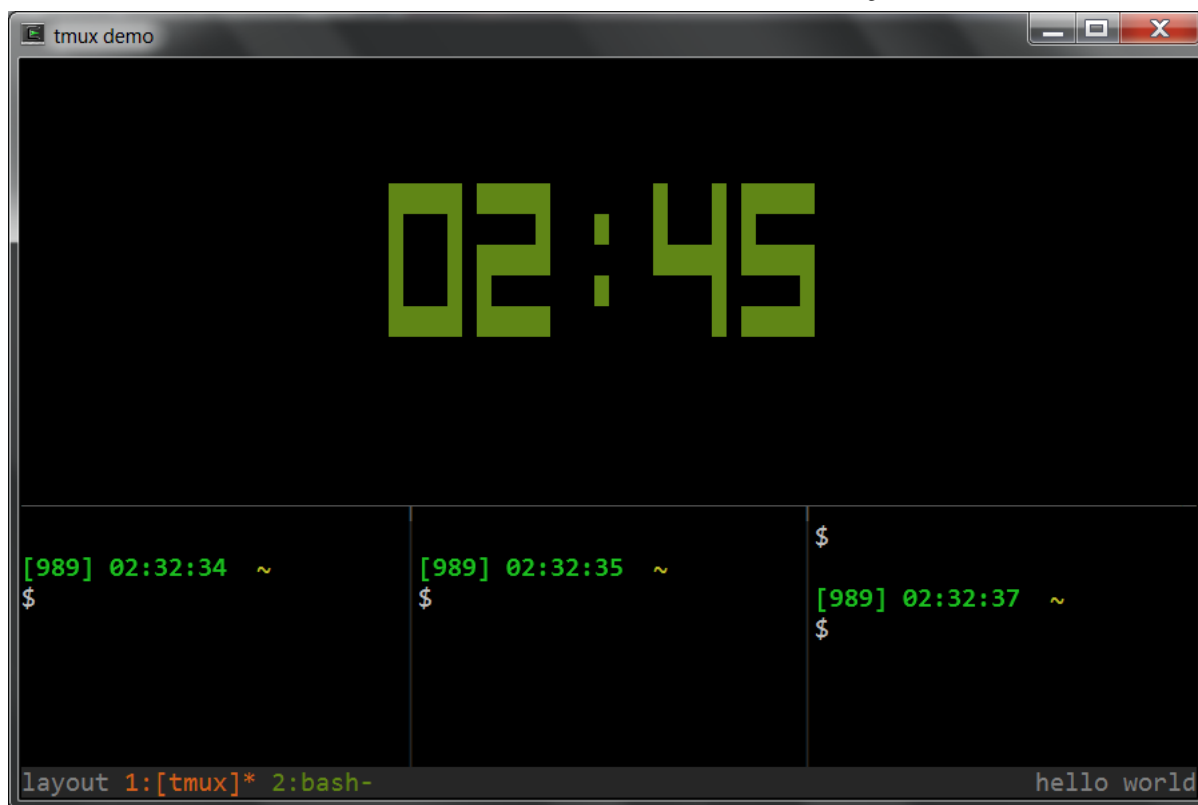
## 5.5 Tile windows with preset layouts

[Up](#) / [HOME](#) / [HELP](#) / [toggle view](#)

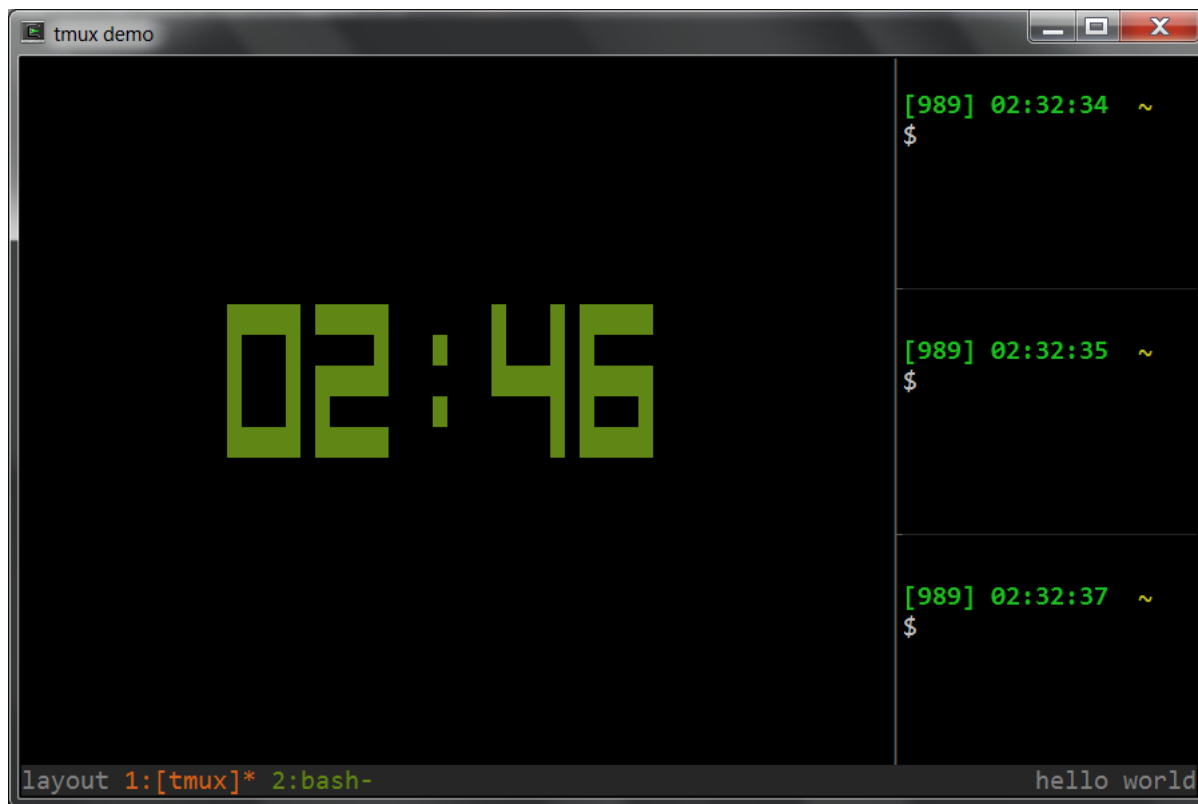
### 5.5.1 tiled



### 5.5.2 main horizontal



### 5.5.3 main vertical



Other layouts not shown in this slides:

- even horizontal
- even vertical

## 5.6 Group windows into sessions

```
(0) 88690: 4 windows [88x28] (attached)
(1) daily: 4 windows [88x28]
(2) support: 5 windows [167x54]
```

The screenshot shows a tmux terminal window with a black background and green text. At the top, it lists three sessions: (0) 88690: 4 windows [88x28] (attached), (1) daily: 4 windows [88x28], and (2) support: 5 windows [167x54]. The bottom status bar shows the current session as [88690@qdbuild2], followed by window tabs for 1: [tmux]\*, 2: bash#, 3: bash#, and 4: bash#-. The user's name and host are shown as [lungangfang@qdbuild2:~].

Focus on what you are working on right now.

Say, one session per project/work-item. Easily switch among the handful windows within current session. Windows in other sessions does not distract you.

## 5.7 Easy to use

- A simplistic setup procedure ([the section of setup](#))
- A minimal set of hotkeys ([the basic operations](#))
- A nice support of mouse even though it is a CLI ([how to enable mouse support](#))

## 5.8 Supports scripting

- Automatically create or attach to a session

```
function tg {
    local session_name="$1"

    if [ -n "$session_name" ]; then
        tmux -2 attach -t "$session_name" || tmux -2 new -s "$session_name"
    else
        tmux -2 attach || tmux -2 new -s 'misc'
    fi
}
```

- Retrieve content of the clipboard of tmux in Emacs



```
(defun lgfang-get-tmux-copied ()  
  "Get current tmux buffer"  
  (interactive)  
  (call-process "tmux" nil t nil "show-buffer"))
```

- A more complicated example can be found [here](#)

## 5.9 Free (as in "freedom")

Last but not least: it is BSD-licensed.

That means it is free to

- Use
- Modify
- Re-distribute
- Or even make profit from it.

## 6 Demo

If scheduled, demonstrate window/pane/attach/session/nesting

1. Connect to server HOST-A (say, a server hosts UDR virtual machines) which I already has at least one tmux session.
2. Create a demo session with "tg demo"
3. Create windows/panes and ssh to HOST-B, which is only accessible from HOST-A (say the pilot card with 10.\* IPs)
4. Changing layout, rotate panes within the window
5. Join/swap panes
6. Share
7. Disconnect and re-attach
8. Create a nested tmux session on HOST-B and create windows within it.
9. Switch between the sessions on host-A
10. Explain the naming rule "session\_name:window\_index.pane\_index"
11. Moving windows/panes between sessions
12. Automatically attach to tmux upon login
  - the putty way
  - the command line way

## 7 Installation

### 7.1 The most simple way

Copy a statically-linked "tmux" binary to the server.

### 7.2 The preferred way

Most modern linux/unix variants have package manage system like yum/ports which download and install tmux for you in a single command.

For instance, for fedora/rhel

```
yum install tmux
```

## 7.3 Install from source

Here is how you install tmux from source code in case needed. (NOTE: compile a **statically linked, 32-bit** binary for portability)

```
# install libevent if not already
tar xvfz libevent-1.4.14b-stable.tar.gz
cd libevent-1.4.14b-stable
./configure --prefix=$HOME/.local
make && make install
tar xvfz tmux-1.6.tar.gz && cd tmux-1.6/
./configure --enable-static --prefix=$HOME/.local \
    CFLAGS="-I$HOME/.local/include" LDFLAGS="-L$HOME/.local/lib"
make && make install

# echo "LD_LIBRARY_PATH=$HOME/.local/bin:$LD_LIBRARY_PATH" >> ~/.bash_profile
echo "PATH=$HOME/.local/bin:$PATH" >> ~/.bash_profile
```

For cygwin, refer to [here](#).

## 8 Setup

### 1. Server side

Make sure your locale is a utf-8 one, such as `en_US.utf8` or `zh_CN.utf8`.

```
# in your profile
export LC_CTYPE=zh_CN.utf8 # or "LC_ALL" if you prefer
```

### 2. Make sure your terminal emulator

1. deals with UTF8 chars correctly.
2. does not intercept hot-keys sent to tmux.
3. has a way to NOT send keys to application.

Let's take PuTTY for example:

- `Window -> Translation -> Received data...character set`: UTF-8

Your putty can display UTF-8 characters (the TMUX pane border and the Chinese chars).

- `Terminal -> Keyboard -> The Function keys and keypad`: XtermR6
- `Window -> Selection -> Shift overrides applications use of mouse`.

This enables you to copy/paste in your PuTTY as usual except you need to hold down `<shift>` key in the meantime.

**NOTE:** refer to *copy mode* for more sophisticate copy/paste.

## 9 Basic operations

All that you need for your daily work (Note that I bind the prefix key to `F12`):

1. New session :: `tmux [new -s session-name]`
2. Attach :: `tmux attach [-t session-name]`
3. Detach :: `F12 d`
4. New window :: `F12 c`
5. Split vertically :: `F12 _`
6. Split horizontally :: `F12 |`
7. Change layout :: `F12 <space>`
8. Kill pane :: `F12 x` (Use this only when the pane does not response)

```
set-option -g prefix F12
```

## 10 Enable mouse support

```
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
```

## 11 Intermediate Usages

You are ready to get started already. However, you may find following slides useful.

### 11.1 Important concepts

- Server instance
- Sessions
- Windows
- Panes
- Handover of key-strokes

### 11.2 Customization

Refer to my `~/tmux.conf` at [/tmux.html](http://tmux.html).

Some examples:

```
# Emacs can not use default prefix 'C-b' :)
set-option -g prefix F12
```

```
bind-key | split-window -h
bind-key _ split-window -v
```

## 11.3 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 ~]$ 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: qdbuild2 [lgfang@huang:~/tmp]

## 11.4 Windows

1. Select previous working window :: **F12 1**
2. Select the N<sup>th</sup> window :: **F12 N**
3. Select next window (index+1) :: **F12 n**
4. Select previous window (index-1) :: **F12 p**
5. To rename a window :: =F12 ,= then input new name

## 11.5 Panes

1. Select pane :: **F12 <Tab>/<left>/<right>/<up>/<down>**
2. Swap two adjacent panes :: **F12 {**
3. Break current pane to a new window :: **F12 !**
4. Join pane y in window x to current window :: **F12 j** then type in **x.y**
5. Swap with pane y of window x :: **F12 m** then type in **x.y**
6. Toggle Synchronize mode, i.e. duplicate input (key-strokes) to all panes in the same window :: **F12 ctrl-s**. Sometimes, this is can be handy for testers

```
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-par
bind-key C-s setw synchronize-panes
```

## 11.6 Copy mode

1. `F12` `[` or `F12` `<PageUp>` to enter ``copy mode'', and then view, search and copy screen output as described below.
  - Scroll up/down: `<PageUp>` / `<PageDown>`, or Emacs(or vi if you configured that way) key bindings, or mouse wheel.
  - Search, just use Emacs key bindings for search (or that of vi based on your configure):
    1. `Ctrl-r`
    2. Type in string to search
    3. `Enter`
    4. Press `n/N` for next/previous match
  - Copy: Emacs/vi key bindings
2. Paste: `F12` `]`
3. Show "clip board": `F12` `=`
4. Customize if you want

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

## 11.7 Sessions within tmux client

1. New session :: `F12` `:` `new-session -s session_name`
2. Choose(switch) session :: `F12` `s`
3. Move window among sessions :: `F12` `:` `move-window -t session_name`

## 11.8 Run complicated commands

1. Command mode: `F12` `:` `source-file .tmux.conf`
2. Command line command: `tmux source-file .tmux.conf`

## 11.9 About configuration

1. `F12` `:` `source-file .tmux.conf`
2. "restart" tmux does not make new configure take effect?

You just "restarted" a session, not the tmux server.

3. What does `-g` in `set-option -g` mean?

Without `-g`, the option applies to current session only.

# 12 Advanced

## 12.1 Log screen outputs

When you want screen log of current pane, save it with `F12` `Ctrl-h`

```
bind-key C-h command-prompt -p "save log to:" "copy-mode; send-keys 'M->
# # if "set-window-option -g mode-keys vi", use this one:
# bind-key C-h command-prompt -p "save log to:" "copy-mode; send-keys g
```

## 12.2 Another way

Though I do not recommend this one, here it is:

1. Start logging :: `F12 h`
2. Stop logging :: `F12 H`

```
# for mode-keys emacs
bind-key h pipe-pane "cat >> $HOME/#S-#I-#P.log" \; display-message "Star
bind-key H pipe-pane \; display-message "Stop logging(#S-#I-#P.log)"
```

## 12.3 Nesting tmux

There are sometimes when need to nest tmux sessions. For example: Start a tmux session on local Linux box. Then, from within that tmux session, ssh to a server and run another tmux session.

To send prefix to

1. Outer tmux session :: `F12`
2. Nested tmux session :: `F11`
3. tmux session nested twice :: `F11 Ctrl-b`
4. One more layer :: `F11 Ctrl-b Ctrl-b` (do you really want a nested nested nested ... session?)

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

## 12.4 Types of hot-keys

1. Normal (follows the prefix) :: `bind-key Tab select-pane -t:.+`
2. Repeatable :: `bind-key -r Space next-layout`
3. Without prefix :: `bind-key -n F11 send-prefix`

## 12.5 Maximize a pane temporary

Use cases:

1. Show more in the pane for a little while
2. Maximize, copy (drag mouse) and restore

For latest tmux (version 1.8), already has built-in support. `F12 z` to toggle.

For older versions, *here* is a trick :

```

unbind z
bind-key z run "if [[ $(tmux list-window) =~ MAX ]]; then \
    tmux last-window;\
    tmux swap-pane -s MAX.0; \
    tmux kill-window -t MAX; \
    else tmux new-window -d -n MAX; \
    tmux swap-pane -s MAX.0; \
    tmux select-window -t MAX;fi"

```

## 12.6 Command line auto-complete

```

if [ -n "$BASH_VERSION" -a -f $HOME/.local/bin/bash_completion_tmux.sh ]
    source $HOME/.local/bin/bash_completion_tmux.sh
fi
# -n $BASH_VERSION ==> we are in bash

```

## 13 Q&A

### References

- The official site: <http://tmux.sourceforge.net/>
- More tips from me: [/tmux.html](http://tmux.html)

## 14 Backup

### 14.1 Reference card according to my conf

---

List Keys	F12 ?
Detach	F12 d
New window	F12 c
Split vertically	F12 _
Split horizontally	F12
Change layout	F12 <space>
Kill pane	F12 x
Select window	F12 idx
Last window	F12 l
Rename window	F12 , new-name
Save log	F12 Ctrl-h
Select pane	F12 <Tab>/<left>/<right>/<up>/<down>

Break current pane F12 !

Join a pane F12 j x.y

Swap two panes F12 {

---

## 14.2 Auto-login and hot-keys

Some can not live without

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

I wrote "to" in expect, which is IMHO more convenient.

Created: 2016-09-18 Sun 18:45 by [Emacs](#) 24.5.1 ([Org](#) mode 8.2.10)

0 Comments **lgfang**

 Login ▾

 Recommend  Share




Sort by Best ▾



Start the discussion...

Be the first to comment.

---

 Subscribe  Add Disqus to your siteAdd DisqusAdd  Privacy