

# Tmux 使用教程 - 阮一峰的网络日志

Tmux 是一个终端复用器 (terminal multiplexer)，非常有用，属于常用的开发工具。  
本文介绍如何使用 Tmux。



## 一、Tmux 是什么？

### 1.1 会话与进程

命令行的典型使用方式是，打开一个终端窗口 (terminal window，以下简称"窗口")，在里面输入命令。用户与计算机的这种临时的交互，称为一次"会话" (session)。

会话的一个重要特点是，窗口与其中启动的进程是连在一起的。打开窗口，会话开始；关闭窗口，会话结束，会话内部的进程也会随之终止，不管有没有运行完。

一个典型的例子就是，SSH 登录远程计算机，打开一个远程窗口执行命令。这时，网络突然断线，再次登录的时候，是找不回上一次执行的命令的。因为上一次 SSH 会话已经终止了，里面的进程也随之消失了。

为了解决这个问题，会话与窗口可以"解绑"：窗口关闭时，会话并不终止，而是继续运行，等到以后需要的时候，再让会话"绑定"其他窗口。

### 1.2 Tmux 的作用

Tmux 就是会话与窗口的"解绑"工具，将它们彻底分离。

- (1) 它允许在单个窗口中，同时访问多个会话。这对于同时运行多个命令行程程序很有用。
- (2) 它可以让新窗口"接入"已经存在的会话。
- (3) 它允许每个会话有多个连接窗口，因此可以多人实时共享会话。
- (4) 它还支持窗口任意的垂直和水平拆分。

类似的终端复用器还有 GNU Screen。Tmux 与它功能相似，但是更易用，也更强大。

## 二、基本用法

### 2.1 安装

Tmux 一般需要自己安装。

```
# Ubuntu 或 Debian
$ sudo apt-get install tmux

# CentOS 或 Fedora
$ sudo yum install tmux

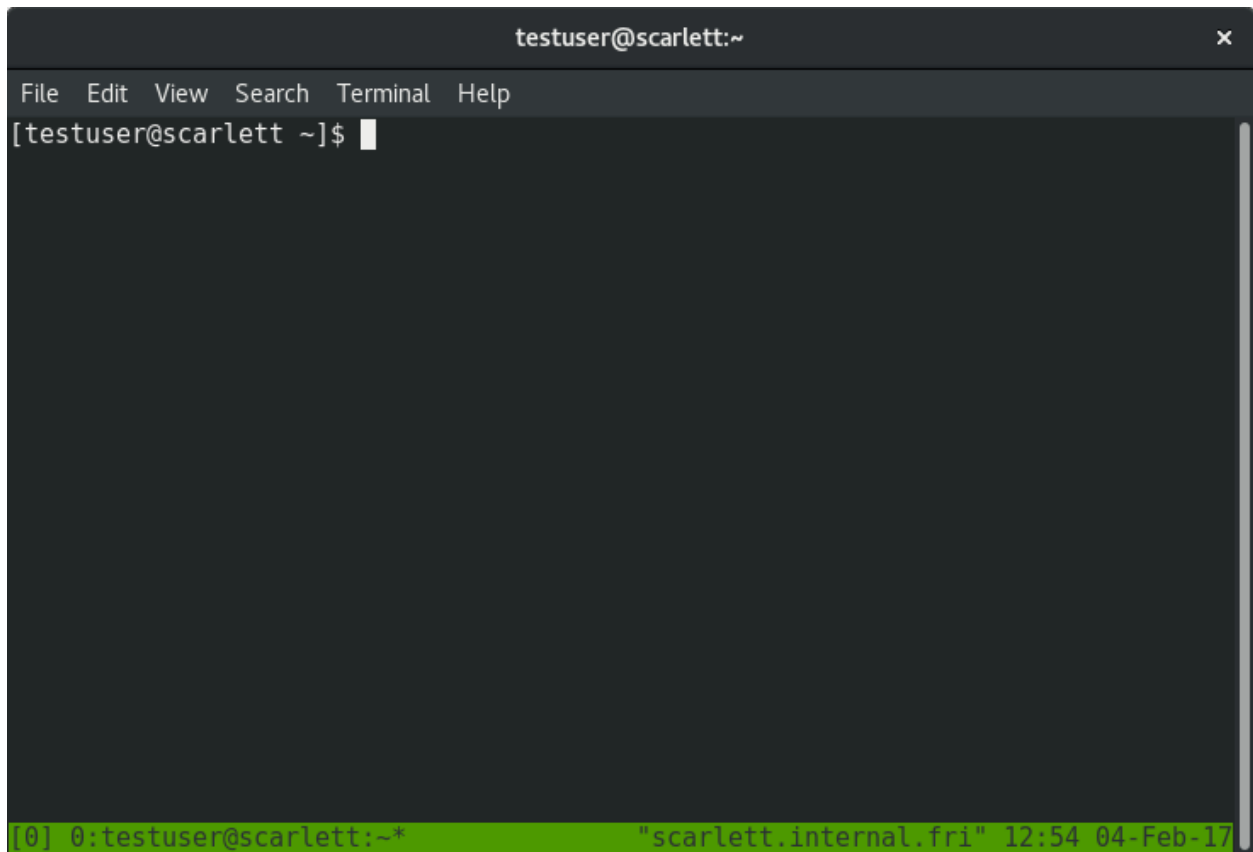
# Mac
$ brew install tmux
```

### 2.2 启动与退出

安装完成后，键入 `tmux` 命令，就进入了 Tmux 窗口。

```
$ tmux
```

上面命令会启动 Tmux 窗口，底部有一个状态栏。状态栏的左侧是窗口信息（编号和名称），右侧是系统信息。



按下 `Ctrl+d` 或者显式输入 `exit` 命令，就可以退出 Tmux 窗口。

```
| $ exit
```

## 2.3 前缀键

Tmux 窗口有大量的快捷键。所有快捷键都要通过前缀键唤起。默认的前缀键是 `Ctrl+b`，即先按下 `Ctrl+b`，快捷键才会生效。

举例来说，帮助命令的快捷键是 `Ctrl+b ?`。它的用法是，在 Tmux 窗口中，先按下 `Ctrl+b`，再按下 `?`，就会显示帮助信息。

然后，按下 ESC 键或 `q` 键，就可以退出帮助。

# 三、会话管理

## 3.1 新建会话

第一个启动的 Tmux 窗口，编号是 `0`，第二个窗口的编号是 `1`，以此类推。这些窗口对应的会话，就是 0 号会话、1 号会话。

使用编号区分会话，不太直观，更好的方法是为会话起名。

```
| $ tmux new -s <session-name>
```

上面命令新建一个指定名称的会话。

## 3.2 分离会话

在 Tmux 窗口中，按下 `Ctrl+b d` 或者输入 `tmux detach` 命令，就会将当前会话与窗口分离。

```
| $ tmux detach
```

上面命令执行后，就会退出当前 Tmux 窗口，但是会话和里面的进程仍然在后台运行。

`tmux ls` 命令可以查看当前所有的 Tmux 会话。

```
| $ tmux ls  
# or  
$ tmux list-session
```

## 3.3 接入会话

`tmux attach` 命令用于重新接入某个已存在的会话。

```
| # 使用会话编号  
$ tmux attach -t 0  
  
# 使用会话名称  
$ tmux attach -t <session-name>
```

## 3.4 杀死会话

`tmux kill-session` 命令用于杀死某个会话。

```
| # 使用会话编号  
$ tmux kill-session -t 0
```

# 使用会话名称

\$ tmux kill-session -t <session-name>

### 3.5 切换会话

`tmux switch` 命令用于切换会话。

# 使用会话编号

\$ tmux switch -t 0

# 使用会话名称

\$ tmux switch -t <session-name>

### 3.6 重命名会话

`tmux rename-session` 命令用于重命名会话。

\$ tmux rename-session -t 0 <new-name>

上面命令将0号会话重命名。

### 3.7 会话快捷键

下面是一些会话相关的快捷键。

- `Ctrl+b d` : 分离当前会话。
- `Ctrl+b s` : 列出所有会话。
- `Ctrl+b $` : 重命名当前会话。

## 四、最简操作流程

综上所述，以下是 Tmux 的最简操作流程。

1. 新建会话 `tmux new -s my_session`。

2. 在 Tmux 窗口运行所需的程序。
3. 按下快捷键 `ctrl+b d` 将会话分离。
4. 下次使用时，重新连接到会话 `tmux attach-session -t my_session`。

## 五、窗格操作

Tmux 可以将窗口分成多个窗格（pane），每个窗格运行不同的命令。以下命令都是在 Tmux 窗口中执行。

### 5.1 划分窗格

`tmux split-window` 命令用来划分窗格。

# 划分上下两个窗格

\$ `tmux split-window`

# 划分左右两个窗格

\$ `tmux split-window -h`

The screenshot shows a terminal window with a menu bar (File, Edit, View, Terminal, Go, Help) and a title bar (Terminal). The main content is divided into two panes. The left pane displays system status information, including CPU usage (9.3%, 21.6%, 44.8%, 44.8%, 14.2%), memory usage (1641/4013MB), tasks (260; 5 running), load average (0.79 0.28), and uptime (6 days, 11:15:24). Below this is a table of running processes with columns for PID, USER, CPU%, MEM%, and Command. The right pane shows a Grails build configuration file (BuildConfig.groovy) with sections for repositories, plugins, and dependencies. The bottom status bar shows the current directory (~/dev/nineteen/src/server), the command being run (grails), and the time (04:29 01-Nov-13).

```
1 [||||| 9.3%] 5 [||||| 21.6%]
2 [||||| 65.2%] 6 [||||| 44.8%]
3 [||||| 14.0%] 7 [||||| 44.8%]
4 [||||| 15.9%] 8 [||||| 14.2%]
Mem [||||| 1641/4013MB] Tasks: 260; 5 running
Swp [||||| 2/8191MB] Load average: 0.79 0.28
Uptime: 6 days, 11:15:24

PID USER CPU% MEM% Command
14574 root 0.0 13.7 /usr/lib/jvm/java-7-openj
14521 root 0.0 12.4 /usr/lib/jvm/default-java
907 root 213. 3.5 /usr/lib/jvm/default-java
12495 ntp 0.0 1.0 /usr/sbin/mysqld --based
11911 ntp 0.0 0.7 /usr/sbin/named -u bind
17123 root 0.0 0.1 vim
15381 root 0.0 0.1 vim /etc/varnish/default.
F1:help F2:Setup F3:Search F4:Filter F5:Free F6:Sort F7:

nineteen@manager:~/dev/nineteen/src/server# grails
| Configuring classpath

35 repositories {
36     inherits true // Whether to inherit repository definition
37     s from plugins
38     grailsPlugins()
39     grailsHome()
40     mavenLocal()
41     grailsCentral()
42     mavenCentral()
43     // uncomment these (or add new ones) to enable remote dep
44     endency resolution from public Maven repositories
45     //mavenRepo "http://repository.codehaus.org"
46     //mavenRepo "http://download.java.net/maven/2/"
47     //mavenRepo "http://repository.jboss.com/maven2/"
48 }
49 dependencies {
50     // specify dependencies here under either 'build', 'compi
51     le', 'runtime', 'test' or 'provided' scopes e.g.
52 }
53 plugins {
54     // plugins for the build system only
55     build ":tomcat:7.0.42"
56
57     // plugins for the compile step
58     //compile ":hibernate:3.6.10.2"
59     //compile ":quartz:1.0-RC13"
60
61     // plugins needed at runtime but not for compilation
62     runtime ":resources:1.2.1"
63 }
64 }
teen/src/server/grails-app/conf/BuildConfig.groovy 55,30 Bot

[0] 0:nineteen 1:helium* 2:r4b- 3:ocf 4:stk 5:trenders "chris-mba-debian" 04:29 01-Nov-13
```

## 5.2 移动光标

`tmux select-pane` 命令用来移动光标位置。

# 光标切换到上方窗格

\$ `tmux select-pane -U`

# 光标切换到下方窗格

\$ `tmux select-pane -D`

# 光标切换到左边窗格

\$ `tmux select-pane -L`

# 光标切换到右边窗格

\$ `tmux select-pane -R`

## 5.3 交换窗格位置

`tmux swap-pane` 命令用来交换窗格位置。

# 当前窗格上移

\$ `tmux swap-pane -U`

# 当前窗格下移

\$ `tmux swap-pane -D`

## 5.4 窗格快捷键

下面是一些窗格操作的快捷键。

- `Ctrl+b %` : 划分左右两个窗格。
- `Ctrl+b "` : 划分上下两个窗格。
- `Ctrl+b <arrow key>` : 光标切换到其他窗格。 `<arrow key>` 是指向要切换到的窗格的方向键, 比如切换到下方窗格, 就按方向键 `↓`。
- `Ctrl+b ;` : 光标切换到上一个窗格。
- `Ctrl+b o` : 光标切换到下一个窗格。
- `Ctrl+b {` : 当前窗格与上一个窗格交换位置。
- `Ctrl+b }` : 当前窗格与下一个窗格交换位置。
- `Ctrl+b Ctrl+o` : 所有窗格向前移动一个位置, 第一个窗格变成最后一个窗格。
- `Ctrl+b Alt+o` : 所有窗格向后移动一个位置, 最后一个窗格变成第一个窗格。
- `Ctrl+b x` : 关闭当前窗格。
- `Ctrl+b !` : 将当前窗格拆分为一个独立窗口。



- `Ctrl+b z` : 当前窗格全屏显示, 再使用一次会变回原来大小。
- `Ctrl+b Ctrl+<arrow key>` : 按箭头方向调整窗格大小。
- `Ctrl+b q` : 显示窗格编号。

## 六、窗口管理

除了将一个窗口划分成多个窗格, Tmux 也允许新建多个窗口。

### 6.1 新建窗口

`tmux new-window` 命令用来创建新窗口。

```
$ tmux new-window
```

```
# 新建一个指定名称的窗口
```

```
$ tmux new-window -n <window-name>
```

### 6.2 切换窗口

`tmux select-window` 命令用来切换窗口。

```
# 切换到指定编号的窗口
```

```
$ tmux select-window -t <window-number>
```

```
# 切换到指定名称的窗口
```

```
$ tmux select-window -t <window-name>
```

### 6.3 重命名窗口

`tmux rename-window` 命令用于为当前窗口起名 (或重命名)。

```
$ tmux rename-window <new-name>
```

### 6.4 窗口快捷键

下面是一些窗口操作的快捷键。

- `Ctrl+b c` : 创建一个新窗口, 状态栏会显示多个窗口的信息。
- `Ctrl+b p` : 切换到上一个窗口 (按照状态栏上的顺序) 。
- `Ctrl+b n` : 切换到下一个窗口。
- `Ctrl+b <number>` : 切换到指定编号的窗口, 其中的 `<number>` 是状态栏上的窗口编号。
- `Ctrl+b w` : 从列表中选择窗口。
- `Ctrl+b ,` : 窗口重命名。

## 七、其他命令

下面是一些其他命令。

```
# 列出所有快捷键, 及其对应的 Tmux 命令
```

```
$ tmux list-keys
```

```
# 列出所有 Tmux 命令及其参数
```

```
$ tmux list-commands
```

```
# 列出当前所有 Tmux 会话的信息
```

```
$ tmux info
```

```
# 重新加载当前的 Tmux 配置
```

```
$ tmux source-file ~/.tmux.conf
```

## 八、参考链接

(完)

[API 开发利器](#)

## 相关文章