

什么是tmux?

tmux是一个终端复用器 (terminal multiplexer) , 主要作用是分屏, 将主窗口分为多个小窗口, 方便多文件同时操作。且最重要的是可以后台运行命令, 例如, 远程终端操作某些持续执行的命令, 一旦发生意外(断网、断电、内存不足.....) 终端连接断开, 命令也会终止。而tmux可以实现断开后重新连接, 命令依然在持续执行。



```
1 # tmux安装
2 sudo apt install tmux
```

比如执行top指令, 相当于Windows系统的任务管理器:

```
top - 09:29:14 up 13 min, 1 user, load average: 0.09, 0.19, 0.17
Tasks: 266 total, 1 running, 265 sleeping, 0 stopped, 0 zombie
%Cpu(s): 0.7 us, 1.0 sy, 0.0 ni, 98.3 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st
MiB Mem : 1941.6 total, 223.3 free, 941.3 used, 777.0 buff/cache
MiB Swap: 2048.0 total, 2048.0 free, 0.0 used. 847.8 avail Mem
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
1478	hioier	20	0	303700	73624	40948	S	1.0	3.7	0:01.78	Xorg
1685	hioier	20	0	3417516	236080	106400	S	1.0	11.9	0:05.31	gnome-+
133	root	20	0	0	0	0	I	0.3	0.0	0:01.55	kworke+
2180	hioier	20	0	817528	50860	38312	S	0.3	2.6	0:00.76	gnome-+
3246	root	20	0	0	0	0	I	0.3	0.0	0:00.01	kworke+
3248	hioier	20	0	15068	4172	3396	R	0.3	0.2	0:00.01	top
1	root	20	0	167536	11120	8024	S	0.0	0.6	0:01.34	systemd
2	root	20	0	0	0	0	S	0.0	0.0	0:00.01	kthrea+
3	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	rcu_gp
4	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	rcu_pa+
5	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	netns
7	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	kworke+
9	root	0	-20	0	0	0	I	0.0	0.0	0:00.22	kworke+
10	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	mm_per+
11	root	20	0	0	0	0	S	0.0	0.0	0:00.00	rcu_ta+
12	root	20	0	0	0	0	S	0.0	0.0	0:00.00	rcu_ta+
13	root	20	0	0	0	0	S	0.0	0.0	0:00.18	ksofti+

强制关闭窗口, top命令也会中断运行。

tmux基本概念

- session, 会话
- windows, 窗口
- pane, 窗格

在普通的终端中，窗口和其中由于session而启动的进程是连在一起的，关闭窗口，session就结束了，session内部的进程也会终止，不管是否运行完。但是在具体使用中，我们希望当前的session隐藏起来，在终端中做其他事情，但是又不希望session及其进程被关闭。这样就需要用到tmux，对session进行解绑。之后再想继续出来这个session的时候，再次绑定就可以回到之前的工作状态。

对于window可以理解为一个工作区，一个窗口。

对于一个session，可以创建好几个window，对于每一个窗口，都可以将其分解为几个pane小窗格。

所以，关于session、window、pane的关系是：

$[pane \in window] \in session$

session操作

启动和退出

新建session，可以在terminal上输入 `tmux` 命令，会自动生成一个id为0的session

```
1 | tmux
```

也可以在建立时显式地说明session的名字，这个名字可以用于解绑后快速的重新进入该session：

```
1 | tmux new -s your-session-name
```

列出当前所有tmux的窗口

```
1 | tmux ls
2
3 # 快捷键
4 Ctrl+a s
```

```
hioier@pc:~$ tmux ls
0: 1 windows (created Wed Nov 30 09:41:07 2022) (attached)
tmux_flask: 1 windows (created Wed Nov 30 09:41:10 2022) (attached)
```

ctrl+d或exit退出当前窗口。注：仅仅退出tmux窗口中一个pane，全部pane退出之后，才会真正结束当前窗口。

前缀键和配置文件

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。

然而，一般情况下，`ctrl+b` 快捷键按下距离较远，可以考虑将前缀快捷键变为 `ctrl+a`，且tmux window中如果划分出多个pane，默认不支持鼠标选择pane进行切换，可以更改配置文件，支持鼠标操作。

在home目录下，新建隐藏文件 `.tmux.conf`：

```
1 set -g mouse on
2 set -g prefix C-a
3 unbind C-b
```

重新加载当前的 Tmux 配置：`tmux source-file ~/.tmux.conf`

分离会话

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

```
1 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。

接入会话

`tmux attach` 命令用于重新接入某个已存在的会话。

```
1 # 使用会话编号
2 tmux attach -t 0
3 tmux a -t 0
4
5 # 使用会话名称
6 tmux attach -t <session-name>
```

杀死会话

`tmux kill-session` 命令用于杀死某个会话。

```
1 # 使用会话编号
2 tmux kill-session -t 0
3
4 # 使用会话名称
5 tmux kill-session -t <session-name>
```

切换会话

`tmux switch` 命令用于切换会话。

```
1 # 使用会话编号
2 tmux switch -t 0
3
4 # 使用会话名称
5 tmux switch -t <session-name>
```

重命名会话

`tmux rename-session` 命令用于重命名会话。

```
1 tmux rename-session -t 0 <new-name>
2
3 # 快捷键
4 Ctrl+a $
```

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。

```
1 Ctrl+a %: 划分左右两个窗格。
2 Ctrl+a ": 划分上下两个窗格。
3 Ctrl+a <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↓。
4 Ctrl+a x: 关闭当前窗格。
5 Ctrl+a !: 将当前窗格拆分为一个独立窗口。
6 Ctrl+a z: 当前窗格全屏显示，再使用一次会变回原来大小。
7 Ctrl+a Ctrl+<arrow key>: 按箭头方向调整窗格大小。
8 Ctrl+a q: 显示窗格编号。
```