

## 28 多个终端和Terminator软件

更新时间：2019-07-22 15:40:08



“立志是事业的大门，工作是登堂入室的旅程。

——巴斯德”

### 内容简介

1. 前言
2. screen 命令：合并多个终端
3. Terminator 软件：地表最强“终结者”
4. 总结

### 1. 前言

上一课前后台进程，灵活切换 中，我们学习了进程的几种状态，以及好多个与进程前后台切换有关的命令。

上一课最后，我们用一张原理图总结了进程的前后台切换，相信你应该已经掌握得不错了。

这一课我们来学习 screen 命令，看看如何操作和合并多个终端。如果你觉得 screen 的操作比较繁琐，可以用这节课后面介绍的 Terminator 软件，会更方便你管理终端。

### 2. screen 命令：合并多个终端

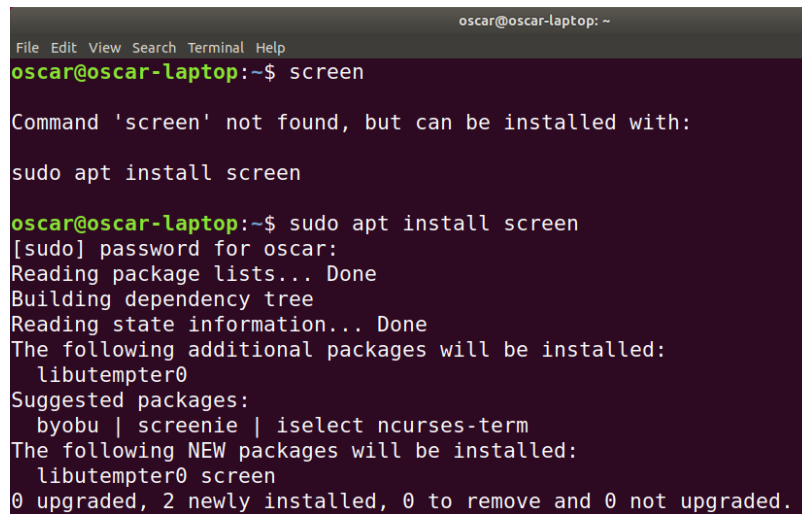
现在，我们来学习一个特殊的命令：screen。

screen 是英语“屏幕”的意思。

screen 这个程序（所有命令其实都是程序）也是 GNU 项目的，它的官网是 <https://www.gnu.org/software/screen>。

通常 screen 命令没有在 Linux 发行版里预装。如果你的 Ubuntu 系统里没有 screen 这个程序，那么可以如此安装：

```
sudo apt install screen
```



```
File Edit View Search Terminal Help
oscar@oscar-laptop: ~
oscar@oscar-laptop:~$ screen

Command 'screen' not found, but can be installed with:

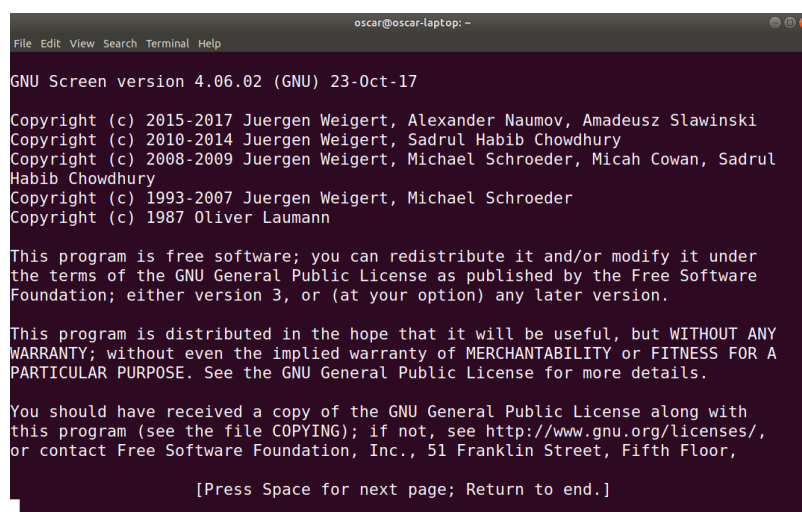
sudo apt install screen

oscar@oscar-laptop:~$ sudo apt install screen
[sudo] password for oscar:
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  libutempter0
Suggested packages:
  byobu | screenie | iselect ncurses-term
The following NEW packages will be installed:
  libutempter0 screen
0 upgraded, 2 newly installed, 0 to remove and 0 not upgraded.
```

安装完之后，你可以运行 screen 命令：

```
screen
```

会显示以下内容：



```
File Edit View Search Terminal Help
oscar@oscar-laptop: ~
GNU Screen version 4.06.02 (GNU) 23-Oct-17

Copyright (c) 2015-2017 Juergen Weigert, Alexander Naumov, Amadeusz Slawinski
Copyright (c) 2010-2014 Juergen Weigert, Sadrul Habib Chowdhury
Copyright (c) 2008-2009 Juergen Weigert, Michael Schroeder, Micah Cowan, Sadrul
Habib Chowdhury
Copyright (c) 1993-2007 Juergen Weigert, Michael Schroeder
Copyright (c) 1987 Oliver Laumann

This program is free software; you can redistribute it and/or modify it under
the terms of the GNU General Public License as published by the Free Software
Foundation; either version 3, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY
WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A
PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with
this program (see the file COPYING); if not, see http://www.gnu.org/licenses/,
or contact Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor,

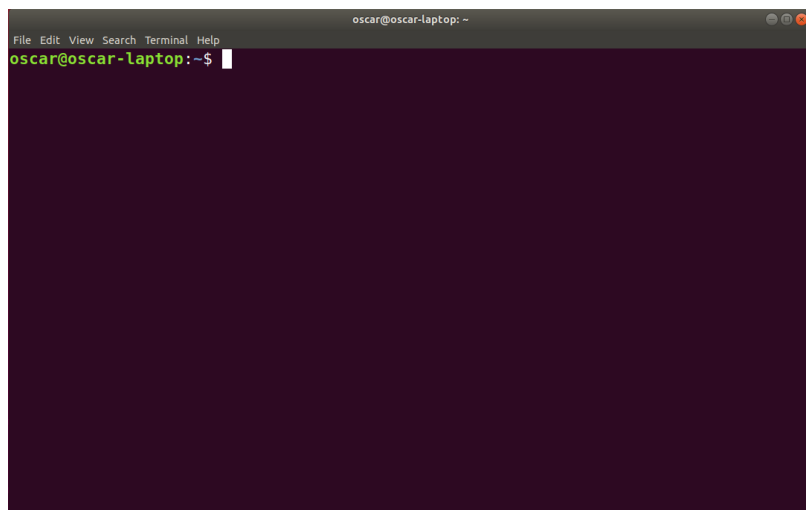
[Press Space for next page; Return to end.]
```

按回车键会直接跳过这个介绍页。按空格键，会进入介绍页的下一页，再按空格键就会跳过介绍页。

screen 命令用于在一个终端中打开多个终端，就好像在一个页面中开多个标签栏一样（使用过浏览器的朋友肯定有这种使用经验），很酷吧。

但是 screen 打开的多个终端是重叠在一起的，如果你不知道，还以为只是打开了一个终端，但是我们会学习如何在各个打开的终端间切换。

刚才，我们运行了 screen 命令，并用回车键或空格键跳过 screen 的介绍页之后，我们看到终端里好像没发生什么变化，就跟之前我们看到的终端一样。如下图：

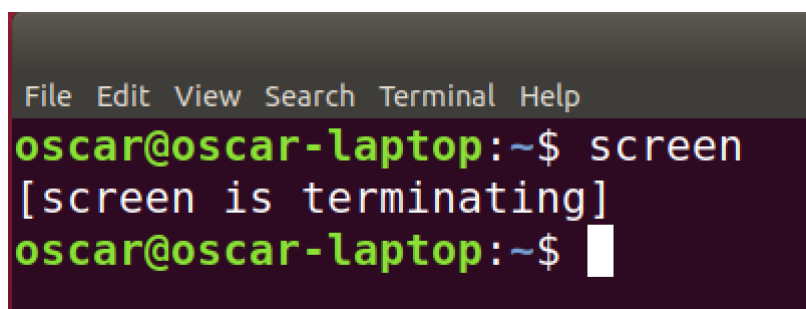


那 screen 到底做了什么呢？

其实，screen 为我们开了一个虚拟终端（Virtual Terminal），就是在当前实际的终端里又开了一个终端。

如果你再运行 screen，那么它又会新开一个虚拟终端。那么怎么退出每个新开的虚拟终端呢？可以按 Ctrl + D 或者用 exit 命令。

每次你按 Ctrl + D 或运行 exit 命令，都会关闭当前所在的虚拟终端，直到最后一个虚拟终端被关闭，screen 程序退出，回到我们的实际终端里，如下图：



上图中显示了 [screen is terminating]，表示所有 screen 开的虚拟终端都已关闭，screen 退出（terminate 是英语“终结，停止”的意思）。

现在我们已经知道如何退出 screen 了。

我们想重新回去，再输入 screen 就好了。

在 screen 程序中，几乎所有的操作都是以 Ctrl + a 开始的。

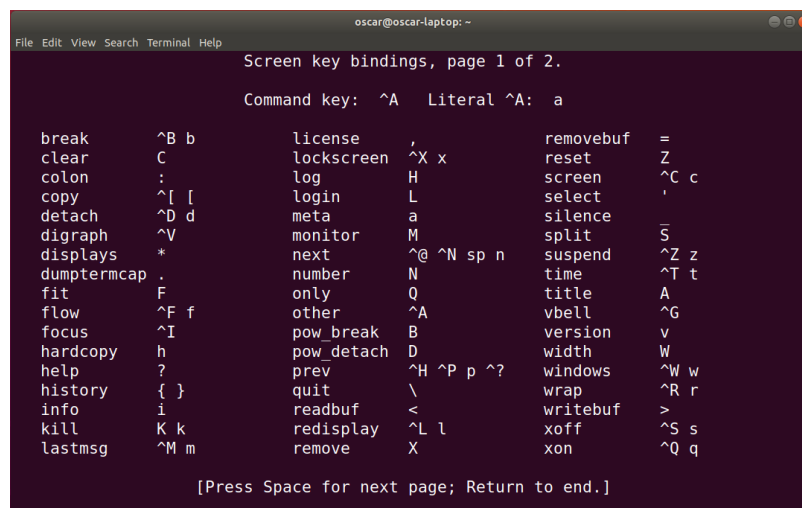
以下所有的讲解中，英文字母区分大小写。也就是说：b 和 B 是不同的，前者就是按下键盘上的 b 键，B 则需要用 Shift + b。

如何操作呢？首先，按下 Ctrl + a 键，然后松开 Ctrl 键和 a 键，再按其他键来完成一定的操作。

## Ctrl + a 再加 ? 号：显示帮助页面

我们先用 Ctrl + a 键（也就是同时按下 Ctrl 键和 a 键），然后松开这两个键，再按下 ? 号（需要 Shift + / 号）。

screen 的帮助页面就会显示：

A screenshot of a terminal window titled "oscar@oscar-laptop: ~". The terminal displays the "Screen key bindings, page 1 of 2." help page. At the top, it says "Command key: ^A Literal ^A: a". Below this is a table of key bindings. The first column lists actions like break, clear, colon, copy, detach, digraph, displays, dumptermcap, fit, flow, focus, hardcopy, help, history, info, kill, and lastmsg. The second column lists the corresponding key sequences, such as ^B b, C, :, ^[ [, ^D d, ^V, \*, ., F, ^F f, ^I, h, ?, { }, i, K k, and ^M m. The third column lists actions like license, lockscreen, log, login, meta, monitor, next, number, only, other, pow\_break, pow\_detach, prev, quit, readbuf, redisplay, and remove. The fourth column lists the corresponding key sequences, such as , ^X x, H, L, a, M, ^@ ^N sp n, N, Q, ^A, B, D, ^H ^P p ^?, \, <, ^L l, and X. The fifth column lists actions like removebuf, reset, screen, select, silence, split, suspend, time, title, vbell, version, width, windows, wrap, writebuf, xoff, and xon. The sixth column lists the corresponding key sequences, such as =, Z, ^C c, ', \_\$, ^Z z, ^T t, A, ^G, v, W, ^W w, ^R r, >, ^S s, and ^Q q. At the bottom, it says "[Press Space for next page; Return to end.]".

可以看到，帮助页面显示了各种操作的实现方法，都是英语。所以好好学英语，很有好处。

帮助页总共有 2 页，可以通过第一页第一行 [Screen key bindings, page 1 of 2] 知道。目前是在 2 页中的第 1 页，按空格可以翻到下一页，按回车退出帮助页。

那怎么来阅读这个帮助页面呢？

比如说，你想要知道 screen 的版本号，那就是 version，可以看到需要用到 v 键。但是光是按 v 键还不够，因为我们看到第一页第二行 Command key: ^A，就是说以下所有的操作，都需要先按下 Ctrl + a 键。^ 表示 Ctrl 键。

所以说，要知道 screen 的版本号，可以先按 Ctrl + a 键（也就是同时按下 Ctrl 键和 a 键），然后松开这两个键，再按下 v 键。就会在左下角显示 screen 的版本号了。

当然了，这个帮助页面还是不太容易懂，不过我们也不需要全部记住，只要会用常用的一些组合键就好了。

## 常用的组合按键

Ctrl + a，松开，再按 c：创建一个新的虚拟终端。

Ctrl + a，松开，再按 w：显示当前虚拟终端的列表。会显示在左下角，类似下图：

A screenshot of a terminal window showing the list of virtual terminals. The text "0\$ bash 1-\$ bash 2\*\$ bash" is displayed in the bottom left corner. The "0\$ bash" part is highlighted in red.

此处的 0\$ bash 1-\$ bash 2\*\$ bash 表示此时打开了 3 个虚拟终端，都叫作 bash，编号是 0，1，2。这是因为目前终端的 Shell 是用的 Bash，之后我们第五部分会开始学习 Shell（外壳程序）。

有 \*（星号）的那个虚拟终端就是我们目前所在的虚拟终端，也就是第 3 个，编号是 2。

Ctrl + a，松开，再按 A：重命名当前虚拟终端。修改后的名字，你用 Ctrl + a，松开，再按 w 时就会看到。

Ctrl + a，松开，再按 n：跳转到下一个虚拟终端。

Ctrl + a，松开，再按 p：跳转到上一个虚拟终端。

Ctrl + a，松开，再按 Ctrl + a：跳转到最近刚使用的那个虚拟终端。

Ctrl + a, 松开, 再按 0 ~ 9 数字键: 跳转到第 0 ~ 9 号虚拟终端。

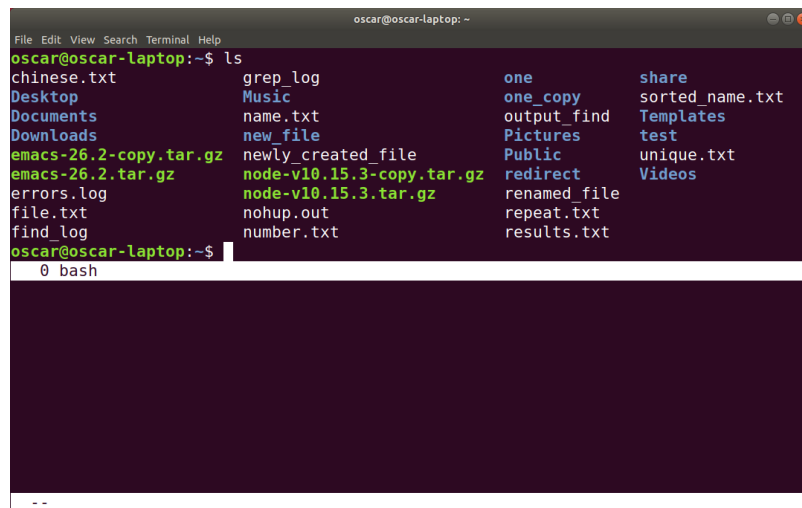
Ctrl + a, 松开, 再按 " (双引号): 会让你选择跳转到哪个虚拟终端。

Ctrl + a, 松开, 再按 k: 关闭当前终端。

以上是一些常用的 `screen` 组合键, 下面我们重点来看两个很有用的组合键, 分别用于分割虚拟终端和分离 `screen`。

## Ctrl + a, 松开, 再按 S 或 |: 分割虚拟终端为多个小虚拟终端

注意是大写的 S (是英语 `split` 的首字母, 表示“分割, 分离”)。如果这样操作一次, 则当前虚拟终端被横向分割为上下两部分。如下图所示:



```
File Edit View Search Terminal Help
oscar@oscar-laptop: ~$ ls
chinese.txt      grep_log         one              share
Desktop          Music            one_copy         sorted_name.txt
Documents        name.txt         output_find      Templates
Downloads        new_file         Pictures          test
emacs-26.2-copy.tar.gz  newly_created_file  Public           unique.txt
emacs-26.2.tar.gz  node-v10.15.3-copy.tar.gz  redirect         Videos
errors.log        node-v10.15.3.tar.gz  renamed_file
file.txt          nohup.out        repeat.txt
find_log          number.txt        results.txt
oscar@oscar-laptop: ~$
0 bash
```

如果再按这样操作, 就横向分割成 3 部分, 4 部分...

可以看到我在上面的半部分中运行了 `ls` 命令, 下面的半部分暂时还没跳转过去操作, 因此下半部分连命令行提示符也没有, 空空的。

那我们如何跳转到下半部分去操作呢?

Ctrl + a, 松开, 再按 Tab 键。

光标就会跳转到下半部分了, 但是还是没见有命令行提示符, 那是因为还没为下半部分创建虚拟终端呢。

所以我们可以新建一个: Ctrl + a, 松开, 再按 c。或者打开一个现有的虚拟终端。

```
oscar@oscar-laptop: ~  
File Edit View Search Terminal Help  
oscar@oscar-laptop:~$ ls  
chinese.txt      grep_log          one                share  
Desktop          Music             one_copy           sorted_name.txt  
Documents        name.txt          output_find        Templates  
Downloads        new_file          Pictures           test  
emacs-26.2-copy.tar.gz newly_created_file Public             unique.txt  
emacs-26.2.tar.gz node-v10.15.3-copy.tar.gz redirect          Videos  
errors.log       node-v10.15.3.tar.gz renamed_file  
file.txt         nohup.out         repeat.txt  
find_log        number.txt        results.txt  
oscar@oscar-laptop:~$  
0 bash  
oscar@oscar-laptop:~$  
1 bash
```

可以看到，我们用 `Ctrl + a`，松开，再按 `c` 之后，下半部分的左下角的 `0 bash` 变成了 `1 bash`，说明新建了一个虚拟终端，编号是 1，也就是第 2 个（虚拟终端的编号从 0 开始）。

我们可以在这个 `1 bash` 的虚拟终端里运行命令，例如 `top` 命令，如下：

```
oscar@oscar-laptop: ~  
File Edit View Search Terminal Help  
oscar@oscar-laptop:~$ ls  
chinese.txt      grep_log          one                share  
Desktop          Music             one_copy           sorted_name.txt  
Documents        name.txt          output_find        Templates  
Downloads        new_file          Pictures           test  
emacs-26.2-copy.tar.gz newly_created_file Public             unique.txt  
emacs-26.2.tar.gz node-v10.15.3-copy.tar.gz redirect          Videos  
errors.log       node-v10.15.3.tar.gz renamed_file  
file.txt         nohup.out         repeat.txt  
find_log        number.txt        results.txt  
oscar@oscar-laptop:~$  
0 bash  
top - 15:47:07 up 2:43, 3 users, load average: 0.05, 0.01, 0.00  
Tasks: 182 total, 1 running, 148 sleeping, 0 stopped, 0 zombie  
%Cpu(s): 1.0 us, 0.7 sy, 0.0 ni, 98.3 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st  
KiB Mem : 2041088 total, 518212 free, 668772 used, 854104 buff/cache  
KiB Swap: 969960 total, 969960 free, 0 used, 1153308 avail Mem  


| PID  | USER  | PR | NI | VIRT    | RES    | SHR    | S | %CPU | %MEM | TIME+   | COMMAND        |
|------|-------|----|----|---------|--------|--------|---|------|------|---------|----------------|
| 1151 | oscar | 20 | 0  | 2990580 | 271144 | 107440 | S | 0.7  | 13.3 | 0:19.52 | gnome-shell    |
| 978  | oscar | 20 | 0  | 466932  | 124008 | 93600  | S | 0.3  | 6.1  | 0:03.89 | Xorg           |
| 1110 | oscar | 20 | 0  | 125776  | 2336   | 1960   | S | 0.3  | 0.1  | 0:15.35 | VBoxClient     |
| 2230 | oscar | 20 | 0  | 798468  | 35976  | 27204  | S | 0.3  | 1.8  | 0:00.30 | gnome-termina+ |

  
1 bash
```

那么我们如何关闭新分割出来的虚拟终端呢？只要 `Ctrl + a`，松开，再按大写的 `X`。

除了横向分割，我们当然也可以纵向分割：`Ctrl + a`，松开，再按 `|`（`Shift + \`）。

我们可以先用 `Ctrl + a`，松开，再按 `Tab` 键，回到我们目前横向分割成两部分的上面那个虚拟终端，然后再将上面的虚拟终端纵向分割。纵向分割之后，我们再切到纵向分割出的地方，新建一个虚拟终端：



```
File Edit View Search Terminal Help
oscar@oscar-laptop: ~
emacs-26.2-copy.tar.gz newly_created_file
Public unique.txt
emacs-26.2.tar.gz node-v10.15.3-copy
py.tar.gz redirect Videos
errors.log node-v10.15.3.tar.gz
r.gz renamed_file
file.txt nohup.out
repeat.txt
find_log number.txt
results.txt
oscar@oscar-laptop:~$
0 bash 2 bash
top - 16:04:37 up 3:01, 4 users, load average: 0.01, 0.01, 0.00
Tasks: 183 total, 1 running, 149 sleeping, 0 stopped, 0 zombie
%Cpu(s): 0.0 us, 0.0 sy, 0.0 ni, 100.0 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st
KiB Mem : 2041088 total, 514868 free, 671568 used, 854652 buff/cache
KiB Swap: 969960 total, 969960 free, 0 used, 1150440 avail Mem

  PID USER      PR  NI   VIRT   RES   SHR  S  %CPU  %MEM     TIME+ COMMAND
 2230 oscar    20   0  798920 36292 27204 S   0.3   1.8   0:01.30 gnome-termina+
    1 root      20   0 159960  9008  6504 S   0.0   0.4   0:01.21 systemd
    2 root      20   0     0     0     0 S   0.0   0.0   0:00.00 kthreadd
    3 root      0 -20     0     0     0 I   0.0   0.0   0:00.00 rcu gp
    1 bash
    2 bash
```

我们再如法炮制，在横向的下半部分再分割出一个纵向的虚拟终端，这下我们就有四个虚拟终端了：

```
File Edit View Search Terminal Help
oscar@oscar-laptop: ~
emacs-26.2-copy.tar.gz newly_created_file
Public unique.txt
emacs-26.2.tar.gz node-v10.15.3-copy
py.tar.gz redirect Videos
errors.log node-v10.15.3.tar.gz
r.gz renamed_file
file.txt nohup.out
repeat.txt
find_log number.txt
results.txt
oscar@oscar-laptop:~$
0 bash 2 bash
top - 16:06:19 up 3:03, 5 users, load average: 0.01, 0.01, 0.00
Tasks: 183 total, 1 running, 150 sleeping, 0 stopped, 0 zombie
%Cpu(s): 0.0 us, 0.3 sy, 0.0 ni, 99.7 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st
KiB Mem : 2041088 total, 513124 free, 671568 used, 854652 buff/cache
KiB Swap: 969960 total, 969960 free, 0 used, 1150440 avail Mem

  PID USER      PR  NI   VIRT   RES   SHR  S  %CPU  %MEM     TIME+ COMMAND
 1110 oscar    20   0 125776  2336  6504 S   0.0   0.0   0:00.00 bash
 1151 oscar    20   0 2990580 271240  6504 S   0.0   0.0   0:00.00 bash
 2230 oscar    20   0  798920 36292 27204 S   0.3   1.8   0:01.30 gnome-termina+
 2272 oscar    20   0  48888  3696  6504 S   0.0   0.0   0:00.00 bash
    1 bash
    2 bash
    3 bash
    4 bash
```

上图中，我们已经创建了 4 个虚拟终端，分别是 0 bash，1 bash，2 bash 和 3 bash。

## Ctrl + a，松开，再按 d：分离 screen

如果我们在 screen 程序中，先按 Ctrl + a，松开，再按 d，就可以使 screen 程序与当前实际终端分离了，有点类似 nohup 命令的作用。

这样我们就可以重回我们自己的实际终端了，而 screen 并没有退出，还在后台运行。

```
File Edit View Search Terminal Help
oscar@oscar-laptop:~$ screen
[detached from 2249.pts-0.oscar-laptop]
oscar@oscar-laptop:~$
```

可以看到 [detached from 2249.pts-0.oscar-laptop]

表示我们的 screen 与实际终端分离（detach 是英语“分离，挣脱”的意思）了。

之后如果你要重回 screen 中，可以输入：

```
screen -r
```

就又回到刚才的 screen 的虚拟终端里了。

我们可以使好几个 screen 进入分离（detached）状态。此时如果你运行 `screen -r` 命令，那么实际终端会询问你要回到哪一个被分离出来的 screen 虚拟终端，如下图：

```
File Edit View Search Terminal Help
oscar@oscar-laptop: ~
oscar@oscar-laptop:~$ screen -r
There are several suitable screens on:
  2249.pts-0.oscar-laptop (05/18/2019 03:44:45 PM)      (Detached)
  2079.pts-0.oscar-laptop (05/18/2019 01:15:36 PM)      (Detached)
  1766.pts-0.oscar-laptop (05/18/2019 01:04:09 PM)      (Detached)
Type "screen [-d] -r [pid.]tty.host" to resume one of them.
oscar@oscar-laptop:~$
```

可以看到，目前我的情况下，有三个候选虚拟终端，编号分别是 2249，2079 和 1766。你想要回到哪一个就用：

```
screen -r 编号
```

就可以了。例如我要回到 2079 那个 screen，只要这样：

```
screen -r 2079
```

如果你在实际终端下，输入：

```
screen -ls
```

则会列出当前打开着的 screen 进程：

```
File Edit View Search Terminal Help
oscar@oscar-laptop: ~
oscar@oscar-laptop:~$ screen -ls
There are screens on:
  2249.pts-0.oscar-laptop (05/18/2019 03:44:45 PM)      (Detached)
  2079.pts-0.oscar-laptop (05/18/2019 01:15:36 PM)      (Detached)
  1766.pts-0.oscar-laptop (05/18/2019 01:04:09 PM)      (Detached)
3 Sockets in /run/screen/S-oscar.
oscar@oscar-laptop:~$
```

怎么样，screen 是不是很有趣呢？好好练习，就会比较熟练了。

用过 Emacs 或 Vim 编辑器的读者，也许觉得 screen 有点像 Emacs 或 Vim 中的分屏功能。我们以后的课程会学习 Vim 编辑器。

### 3. Terminator 软件：地表最强“终结者”

我个人感觉 screen 组合键使用起来还是比较麻烦的，上手不是很容易。

除了 screen 命令，我们还有其他的选择，例如 `tmux` 这个来自 OpenBSD 项目的命令，是和 screen 命令类似的。tmux 的官方文档在 <https://github.com/tmux/tmux/wiki>

篇幅关系，我们就不演示 tmux 了，大家有兴趣可以自己去搜索网上 tmux 的安装、配置和使用的图文教程，非常多。

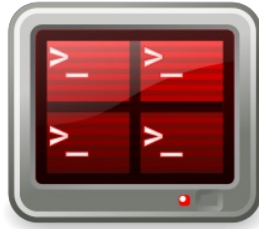
我们这里要介绍一下我个人觉得目前在 Linux 发行版（特别是 Ubuntu）中最好用的终端软件，也就是名字和《终结者》（The Terminator）电影一样的 Terminator 软件。其自带的分屏功能非常优越，比 screen 命令的分屏方便快捷很多。

当然了，Terminator 在其他 Linux 发行版和 macOS 操作系统下也可以安装。



如果你要在 Windows 系统使用 Terminator，你可以先在 Windows 的 Microsoft Store 找到 [Ubuntu 应用](#)，下载安装。这跟我们以前演示的硬盘安装 Ubuntu 或虚拟机安装 Ubuntu 是不一样的。Windows 上的 Ubuntu 这个应用允许用户在 Windows 中使用 Ubuntu 终端并运行 Ubuntu 命令行实用程序，包括 bash, ssh, git, apt 等等。然后你在这个 Ubuntu 环境里面用 `sudo apt install terminator` 命令来安装 Terminator 软件。

Terminator 的软件图标挺好看的，长这样：



Terminator 的官方网站是：<https://gnometerminator.blogspot.com/p/introduction.html>

遇到 Terminator 之前，我一直用的是 tmux 或 screen。直到有一天遇到 Terminator，一时惊为天人。我毫不犹豫地吧 Ubuntu 的默认终端给替换了，从此几乎只用 Terminator 作为默认终端，我的大部分同事也是如此。

Terminator 给我的感觉就是：

1. 颜值高
2. 上手极为容易
3. 快捷键很容易记
4. 可以自由定制
5. 功能多

如果你还没有用过 Terminator，强烈推荐下载使用一下，相信你会喜欢上它的。

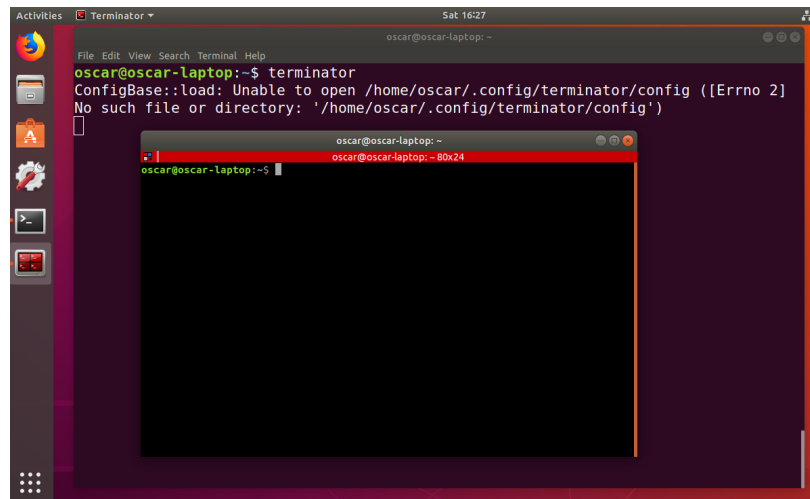
Terminator 软件是在 GNU 项目的 GPL 许可证下发布的免费软件。

我们在 Ubuntu 下安装 Terminator，运行以下命令来安装：

```
sudo apt install terminator
```

```
oscar@oscar-laptop: ~  
File Edit View Search Terminal Help  
oscar@oscar-laptop:~$ sudo apt install terminator  
[sudo] password for oscar:  
Reading package lists... Done  
Building dependency tree  
Reading state information... Done  
The following additional packages will be installed:  
  gir1.2-keybinder-3.0 libkeybinder-3.0-0 python-cairo python-dbus python-gi  
  python-gi-cairo python-psutil  
Suggested packages:  
  python-dbus-dbg python-dbus-doc python-psutil-doc  
The following NEW packages will be installed:  
  gir1.2-keybinder-3.0 libkeybinder-3.0-0 python-cairo python-dbus python-gi  
  python-gi-cairo python-psutil terminator  
0 upgraded, 8 newly installed, 0 to remove and 0 not upgraded.  
Need to get 841 kB of archives.  
After this operation, 4,595 kB of additional disk space will be used.  
Do you want to continue? [Y/n] y
```

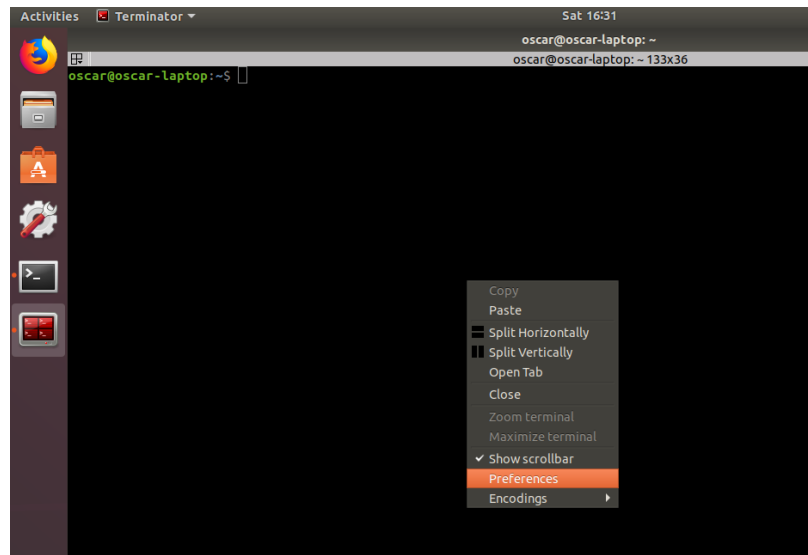
安装完毕之后，运行 `terminator` 命令，就会打开一个 Terminator 终端了：



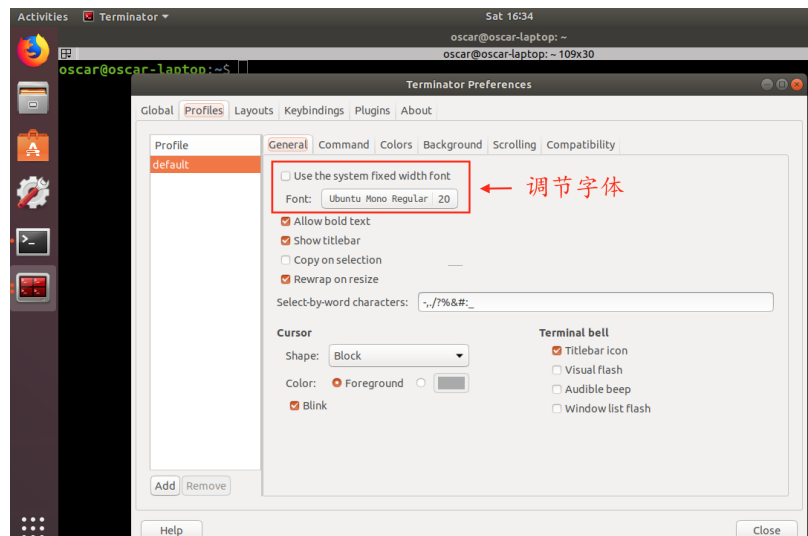
你可以将 Terminator 添加到左边的快捷启动栏里，右键 Terminator 的图标，选择“Add to Favorites”，Terminator 的快捷方式就添加好了：



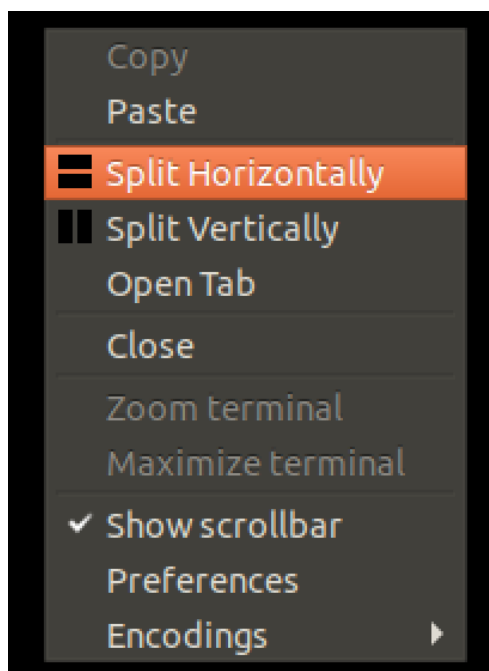
可以看到，初始的 Terminator 终端没有最大化，我们可以双击终端的窗口上栏来最大化（就和 Windows 下最大化窗口一样的操作），然后可以调节一下字体大小之类的设置。要设置 Terminator 软件，可以鼠标右键单击 Terminator 的终端区域，会调出菜单栏：



选择菜单栏中的 Preferences（“偏好”）来设置。点击菜单中的 Profiles，在 General 里面，将“Use the system fixed width font”（“使用系统固定大小的字体”）前面的勾去掉，然后你就可以自己选择字体和字体大小了。如下图：



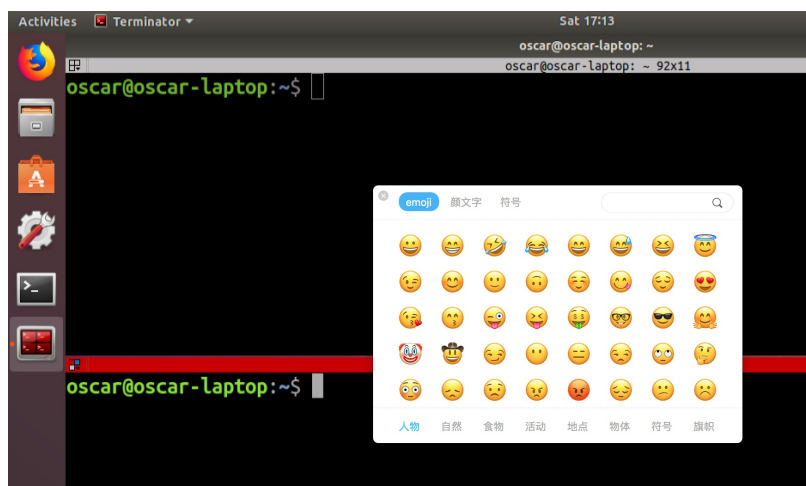
Terminator 终端的分屏操作非常简单。其实在刚才的右键调出的菜单里，也已经有分屏的操作选项了：



可以看到 Split Horizontally 和 Split Vertically 两个选项。split 我们之前说过是“分割，分离”的意思，horizontally 是“横向地”的意思，vertically 是“纵向地”的意思。

因此，这两个选项分别是用于横向分割和纵向分割。当然了，它们也有对应的组合键，分别是 Ctrl + Shift + o 和 Ctrl + Shift + e。

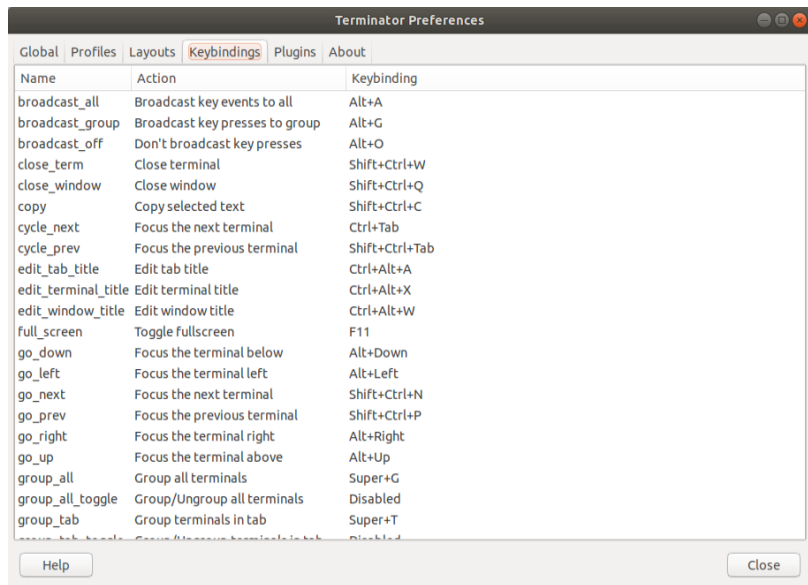
我在 Ubuntu 18.04 中，用 Ctrl + Shift + o 来横向分屏的时候，没有问题。但是当用 Ctrl + Shift + e 来纵向分屏时，竟然跳出了 emoji 表情的窗口：



原来在新版 Ubuntu 下，Ctrl + Shift + e 这个快捷键是系统默认分配给了 emoji 表情。

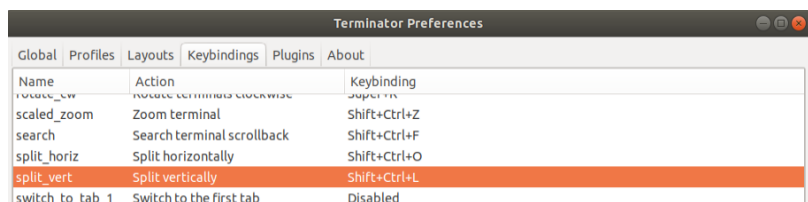
所以，你可以给垂直分割分配另外的快捷键。因为 emoji 这个默认绑定的快捷键比较顽固，我用网上给出的一些设置方法没能将其禁用。

要修改快捷键，你可以在 Preferences（“偏好”）里面的 Keybindings（“键绑定”）的菜单里面去修改：

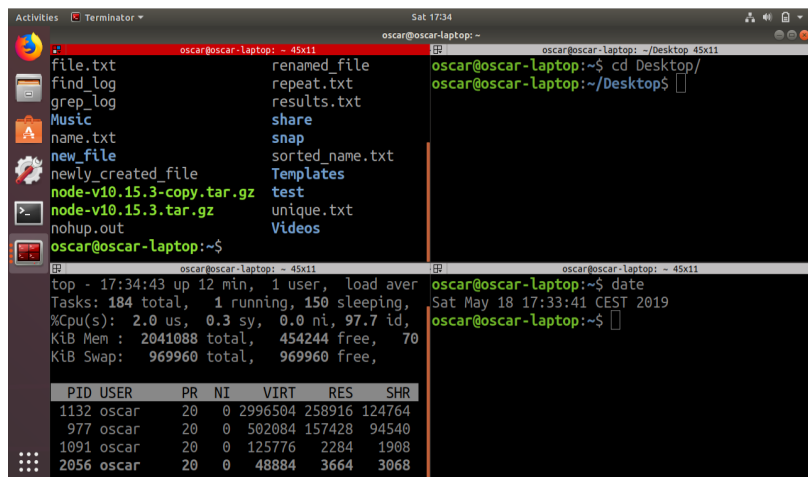


修改快捷键，只需要将鼠标点击一下 Keybinding 那一列，比如我点击一下 split\_vert（垂直分割）对应的 Ctrl + Shift + e，然后它会提示你分配新的快捷键组合，此时你按下想要设置的新按键组合，即可完成设置。

我暂时把它修改为 Ctrl + Shift + L。你当然可以指定不同的快捷键。



修改之后，就可以顺畅地进行垂直分屏了：



可以看到我们顺利分出了四个屏幕，代表四个虚拟终端。

和 screen 不同的是，Terminator 用快捷键来分割完后，在分割出的新部分里，虚拟终端是直接被创建的。不像 screen 还要切换过去，再创建新的虚拟终端。所以说 Terminator 大大简化了我们的操作。

在 Terminator 的虚拟终端里面切换，可以用 Alt + 上下左右键。

如果你要关闭所在的虚拟终端，只需要用组合键 Ctrl + Shift + w。

如果你要开一个新的 Tab（菜单栏），只需要用组合键 `Ctrl + Shift + t`。可以看到，我开了一个新的 Tab，在新的 Tab 里面又可以用快捷键来进行分屏操作。

如果你要开一个新的 Terminator 窗口，可以用快捷键 `Ctrl + Shift + i`。

Terminator 还有很多快捷键组合，我只是讲了比较常用的几个，你可以去 Keybindings 里面查找快捷键组合。

### 小结

1. screen 是一个程序，在 Debian 一族的操作系统里，你可以用 `apt-get / apt` 来安装。screen 命令使用户能够在一个终端中打开多个虚拟终端。
2. Terminator 软件是非常好用的终端软件，可以替代 Ubuntu 自带的 Gnome 的 Terminal 软件，成为我们日常使用的终端。入门非常简单，分屏操作比 screen 要方便很多。

今天的课就到这里，一起加油吧！

← 27 前后台进程，灵活切换

29 定时和延时执行，唯慢不破 →

### 精选留言 0

欢迎在这里发表留言，作者筛选后可公开显示



目前暂无任何讨论