

19 RTFM 阅读那该死的手册

更新时间：2019-07-09 09:52:42



“

天才免不了有障碍，因为障碍会创造天才。

——罗曼·罗兰

”

内容简介

1. 前言
2. man 命令，显示使用手册
3. SYNOPSIS 区域
4. apropos 命令：查找命令
5. 其它阅读手册的方式
6. 总结

1. 前言

上一课软件安装，如虎添翼 中，我们领略了 Linux 下安装软件的便捷，今天这一课也比较轻松。

在 Linux 下，阅读命令手册应该成为一个反射动作，一个自然而然的反应。虽说一开始要我们去看手册会有点让人生畏，但是我们学习各种 Linux 命令使用方法的最好去处就是手册了。

我们这个系列教程也绝不能和手册相比，因为我们的教程里对于每一个知识点，不可能做到巨细靡遗，全盘托出。

比如，对于某个 Linux 的命令，我们不可能详尽地列出每个参数及其用法，只能举几个最常用的参数。所以，如果要深入了解一个命令，阅读命令的手册就是最好方式了。

这一课就是为了揭开 Linux 手册的神秘面纱，带大家学习如何阅读手册。

这一课是系列课程中极重要的，因为如果你学会了如何阅读手册，那么大部分问题都可以自己解决。

今天标题里的 RTFM 稍微有点小“暴力”，因为它是 Read The F**king Manual 的缩写，意思是“阅读那该死的手册”。

这个缩写经常被 Linux 高手用来回复一些初学者，并无侮辱之意，只是提醒及表示无奈，因为这些初学者的问题大多在手册中可以轻松找到解答。

就好比我们会说：“唉，这么简单的问题，百度或 Google 一下就搜索出来了，何必要问呢？”

2. man 命令，显示使用手册

man 是 manual 的缩写，就是英语“使用手册”的意思。

man 命令用法很简单，后接你想要显示使用手册的命令、函数等等。

此命令用于查看系统中自带的各种参考手册，但是手册分为好几个类别，如下所示：

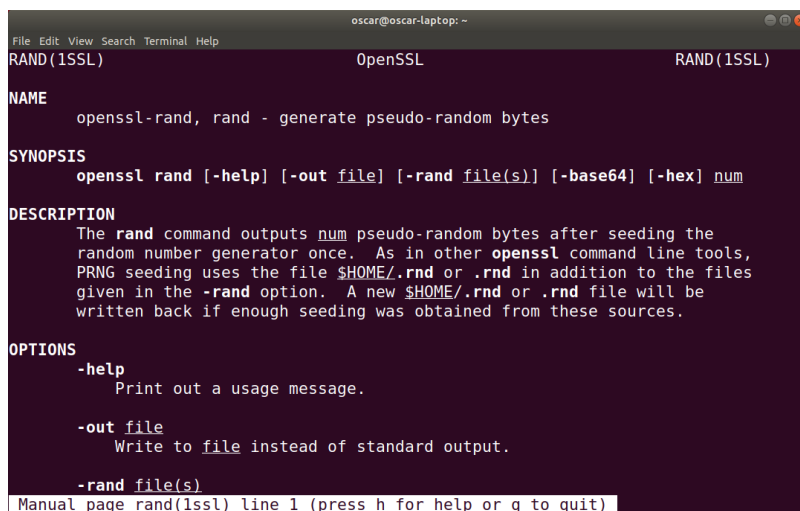
1. 可执行程序或 Shell 命令；
2. 系统调用（Linux 内核提供的函数）；
3. 库调用（程序库中的函数）；
4. 文件（例如 /etc/passwd）；
5. 特殊文件（通常在 /dev 下）；
6. 游戏；
7. 杂项（比如 man(7)，groff(7)）；
8. 系统管理命令（通常只能被 root 用户使用）；
9. 内核子程序。

Ubuntu 中默认的 man 手册是不全的，我们可以运行下面的命令来安装缺失的一些手册：

```
sudo apt-get install manpages-de manpages-de-dev manpages-dev glibc-doc manpages-posix-dev manpages-posix
```

在终端中输入 man + 数字 + 命令/函数，即可以查到相关的命令和函数。若不加数字，那 man 命令默认从数字较小的手册中寻找相关命令和函数。

例如，我们想寻找 C 语言标准库中的 rand() 函数，但直接使用 man rand 得到的是一个 rand 命令的手册，此命令是用来产生一个随机数，这个内容在手册第 1 部分中：



```
oscar@oscar-laptop: ~
File Edit View Search Terminal Help
RAND(1SSL)                                OpenSSL                                RAND(1SSL)

NAME
  openssl-rand, rand - generate pseudo-random bytes

SYNOPSIS
  openssl rand [-help] [-out file] [-rand file(s)] [-base64] [-hex] num

DESCRIPTION
  The rand command outputs num pseudo-random bytes after seeding the
  random number generator once. As in other openssl command line tools,
  PRNG seeding uses the file $HOME/.rnd or .rnd in addition to the files
  given in the -rand option. A new $HOME/.rnd or .rnd file will be
  written back if enough seeding was obtained from these sources.

OPTIONS
  -help
    Print out a usage message.

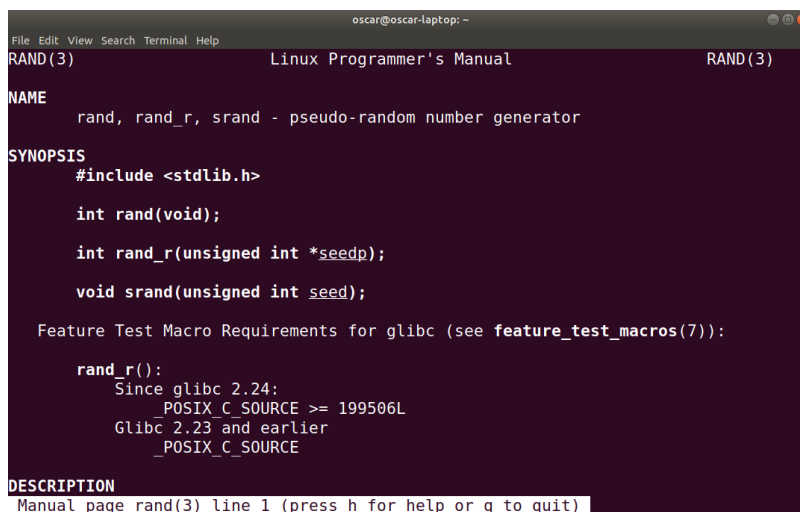
  -out file
    Write to file instead of standard output.

  -rand file(s)

Manual page rand(1ssl) line 1 (press h for help or q to quit)
```

如果我们使用 `man 3 rand` 就可以得到想要的 C 语言标准库中 `rand()` 的用法了：

```
man 3 rand
```



```
oscar@oscar-laptop: ~
File Edit View Search Terminal Help
RAND(3) Linux Programmer's Manual RAND(3)

NAME
    rand, rand_r, srand - pseudo-random number generator

SYNOPSIS
    #include <stdlib.h>

    int rand(void);

    int rand_r(unsigned int *seedp);

    void srand(unsigned int seed);

    Feature Test Macro Requirements for glibc (see feature_test_macros(7)):

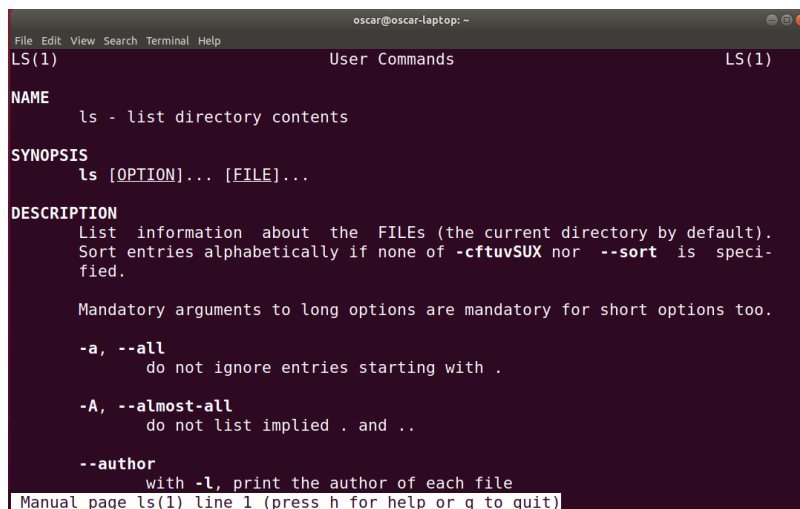
    rand_r():
        Since glibc 2.24:
            _POSIX_C_SOURCE >= 199506L
        Glibc 2.23 and earlier
            _POSIX_C_SOURCE

DESCRIPTION
Manual page rand(3) line 1 (press h for help or q to quit)
```

`man` 这个词本身在英语中是“男人”的意思。所以 Linux 和 Unix 世界流行着一句有趣的话：【不懂就问一问“男人”】。

比如，我们举个例子，用 `man` 命令来显示 `ls` 命令的使用手册：

```
man ls
```



```
oscar@oscar-laptop: ~
File Edit View Search Terminal Help
LS(1) User Commands LS(1)

NAME
    ls - list directory contents

SYNOPSIS
    ls [OPTION]... [FILE]...

DESCRIPTION
    List information about the FILES (the current directory by default).
    Sort entries alphabetically if none of -cftuvSUX nor --sort is speci-
    fied.

    Mandatory arguments to long options are mandatory for short options too.

    -a, --all
        do not ignore entries starting with .

    -A, --almost-all
        do not list implied . and ..

    --author
        with -l, print the author of each file

Manual page ls(1) line 1 (press h for help or q to quit)
```

如上图所见，运行 `man ls` 后，就显示了 `ls` 命令的使用手册，这里有你想知道的关于 `ls` 命令的一切。

在手册中移动

我们来学习如何在手册页面中移动：

- 键盘上的方向键：向上键和向下键可以使我们实现上一行和下一行的跳转；
- 键盘上的 `PgUp` 和 `PgDn`（或者空格键）键：实现上一页和下一页的跳转；
- 键盘上的 `Home` 和 `End` 键：实现开始和结尾的跳转；
- 键盘上的 `/` 键（斜杠）：实现搜索，和之前在 `less` 命令中功能类似；
- 键盘上的 `Q` 键：退出手册页。

如果我们使用图形界面的终端来打开手册页，那么可以使用鼠标的滚轮来实现上下滚动。

手册页的不同区域

正如我们在上图中所看到的，手册页分为不同的区域。这些区域的名字是用大写和粗体表示，且靠左对齐：

- **NAME**：英语“名字”的意思。手册页对应的命令或函数名字，后接简单描述；
- **SYNOPSIS**：英语“概要，梗概，大意，摘要”的意思。使用此命令的所有方法。下面我们会详述这个区域，因为此区域的内容极为关键；
- **DESCRIPTION**：英语“描述”的意思。命令的更深入的描述，这个区域也会包括所有参数及其用法。一般来说这个区域是文字最多的；
- **AUTHOR**：英语“作者”的意思。命令的作者，有时候不止一个作者；
- **REPORTING BUGS**：英语“报告故障”的意思。如果使用时遇到 bug（程序的缺陷），有汇报问题的联系方式，通常是邮件；
- **COPYRIGHT**：英语“版权”的意思。版权，也就是用了什么许可证（Licence），大多数命令都是在 GPL 许可证之下的开源代码；
- **SEE ALSO**：英语“另见”的意思。与此命令有关的其它命令，也就是扩展阅读。

一般来说，前三个区域（NAME、SYNOPSIS、DESCRIPTION）是最重要的。

手册页的语言

如你所见，手册页的语言全是英语。并不是所有人都精通英语，那假如你英语不好怎么办呢？

我只能给出一个最好的建议，也许你不会喜欢：花点时间学学英语。

不需要精通英语，但至少要学会阅读，而且大胆使用金山词霸、有道词典、谷歌翻译、灵格斯翻译家等等。

如果你做编程工作，在信息技术领域做研究，那么英语可以说是必不可少的语言。虽然你可以完全不会英语，但这会变成你晋升为编程专家的阻碍。

接下来我们学习一下 SYNOPSIS 这个区域。

3. SYNOPSIS 区域

SYNOPSIS 这个区域是手册页中很重要的一个区域，但也是最难理解的区域之一。

SYNOPSIS 这个区域的作用就是列出使用命令的所有可能方法。简单地说，SYNOPSIS 这个区域显示了所有可能的参数组合。

有些命令的 SYNOPSIS 区域比较简单，但有些就很复杂了，会是一长串，看得人头晕。

我们下面就用一些实际的例子来学习。

man mkdir

我们从 mkdir 命令的使用手册开始，运行：

```
man mkdir
```

```
oscar@oscar-laptop: ~  
File Edit View Search Terminal Help  
MKDIR(1) User Commands MKDIR(1)  
  
NAME  
    mkdir - make directories  
  
SYNOPSIS  
    mkdir [OPTION]... DIRECTORY...  
  
DESCRIPTION  
    Create the DIRECTORY(ies), if they do not already exist.  
  
    Mandatory arguments to long options are mandatory for short options too.  
  
    -m, --mode=MODE  
        set file mode (as in chmod), not a=rwx - umask  
  
    -p, --parents  
        no error if existing, make parent directories as needed  
  
    -v, --verbose  
        print a message for each created directory  
  
    -Z      set SELinux security context of each created directory to the  
Manual page mkdir(1) line 1 (press h for help or q to quit)
```

如上图所见，mkdir 命令的使用手册的 NAME 区域是：

```
mkdir - make directories
```

make 是英语“创建”的意思，directories 是 directory（英语“目录”）的复数。所以我们知道 mkdir 命令用于创建目录。

mkdir 命令的 SYNOPSIS 区域如下：

```
mkdir [OPTION]... DIRECTORY...
```

虽然这个 SYNOPSIS 区域比较短，但已经包含不少信息了，它告诉我们如何使用 mkdir 命令。

我们一点点解析此 SYNOPSIS 区域：

- **mkdir**：使用 mkdir 命令，我们当然必须先输入mkdir，这是很好理解的；
- **[OPTION]**：option 是英语“选项”的意思，表示在 mkdir 之后，我们可以加选项参数（就是 -p, -v 之类）。在 SYNOPSIS 区域中，中括号中的内容表示可选，所以我们并不一定要添加选项参数；
- **DIRECTORY**：directory 是英语“目录”的意思。就是需要用 mkdir 命令创建的目录。这个参数是必须的，因为没有用中括号括起来。这也不难理解，因为 mkdir 命令就是用于创建目录的，那最少也要输入一个目录的名字吧；
- **...**：省略号表示可以有多个此类内容。在 [OPTION] 和 DIRECTORY 后都有省略号，分别表示可以有多个选项和多个目录名，也即是说可以同时指定多个选项参数，也可以用 mkdir 同时创建多个目录。

那么为什么 mkdir 使用粗体，而 OPTION 和 DIRECTORY 都是用下划线呢？

SYNOPSIS 区域中，粗体的文字表示要原封不动地输入，下划线的文字表示要用实际的内容替换。

这也不难理解：mkdir 命令肯定要原封不动地输入，而 DIRECTORY 就要替换成实际要创建的目录名称。

用例

根据 SYNOPSIS 区域的语法，我们使用 mkdir 命令时至少要输入一个创建的目录名。

例如：

```
mkdir photo
```

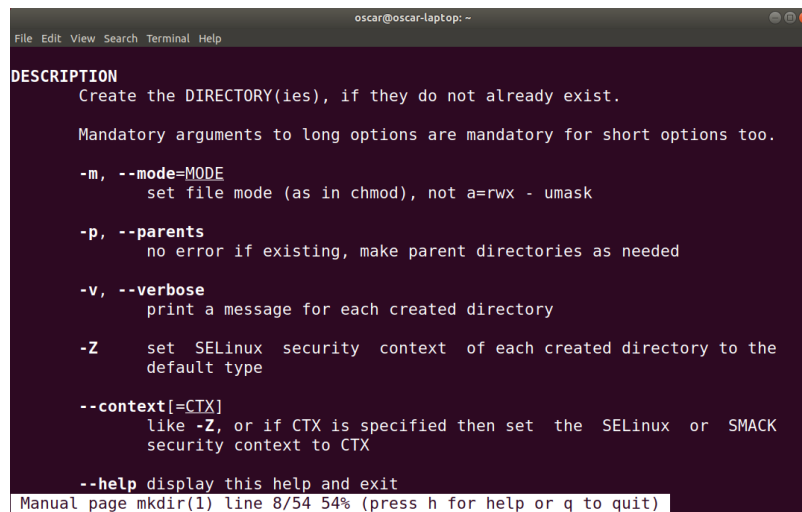
用于创建一个名为 photo 的目录。

而且，DIRECTORY 之后的省略号表明我们可以同时输入多个目录名，用于同时创建多个目录，例如：

```
mkdir photo video music
```

用于创建 photo、video、music 三个目录。

我们也可以加选项参数，选项参数在 DESCRIPTION 区域中：



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

DESCRIPTION
  Create the DIRECTORY(ies), if they do not already exist.

  Mandatory arguments to long options are mandatory for short options too.

  -m, --mode=MODE
        set file mode (as in chmod), not a=rwx - umask

  -p, --parents
        no error if existing, make parent directories as needed

  -v, --verbose
        print a message for each created directory

  -Z      set SELinux security context of each created directory to the
        default type

  --context[=CTX]
        like -Z, or if CTX is specified then set the SELinux or SMACK
        security context to CTX

  --help display this help and exit

Manual page mkdir(1) line 8/54 54% (press h for help or q to quit)
```

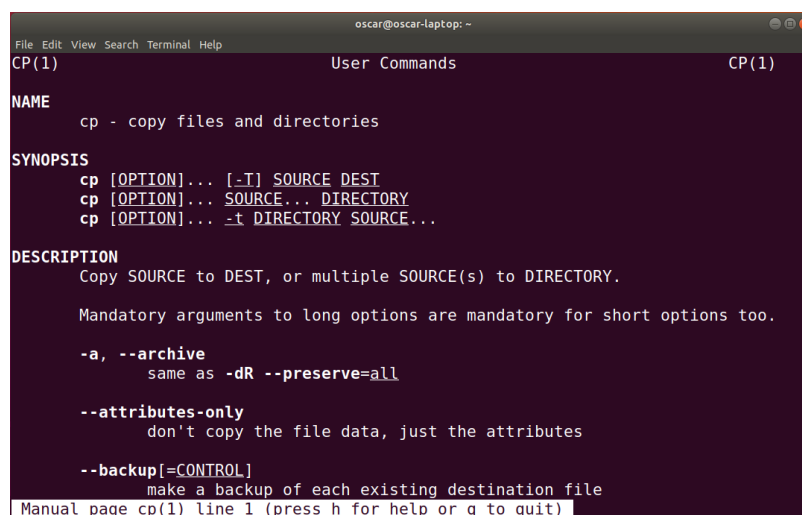
以上所有的选项参数都可以替换 [OPTION]，而且可以同时写多个选项参数，例如 -pv。

man cp

我们再来看一个比较复杂的 SYNOPSIS 区域的例子：cp 命令的 SYNOPSIS 区域。

我们之前学过 cp 命令，它用于拷贝文件或目录。运行：

```
man cp
```



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

CP(1)                                User Commands                                CP(1)

NAME
  cp - copy files and directories

SYNOPSIS
  cp [OPTION]... [-T] SOURCE DEST
  cp [OPTION]... SOURCE... DIRECTORY
  cp [OPTION]... -t DIRECTORY SOURCE...

DESCRIPTION
  Copy SOURCE to DEST, or multiple SOURCE(s) to DIRECTORY.

  Mandatory arguments to long options are mandatory for short options too.

  -a, --archive
        same as -dR --preserve=all

  --attributes-only
        don't copy the file data, just the attributes

  --backup[=CONTROL]
        make a backup of each existing destination file

Manual page cp(1) line 1 (press h for help or q to quit)
```

上图是 cp 命令的手册，可以看到 SYNOPSIS 区域如下：

```
cp [OPTION]... [-T] SOURCE DEST
cp [OPTION]... SOURCE... DIRECTORY
cp [OPTION]... -t DIRECTORY SOURCE...
```

是否有些惊讶？一下子来了三行。

为什么有三行而不是一行呢？原因很简单，因为没法只用一行表达清楚 cp 命令的使用方法。

别担心，我们逐行来解释：

- cp [OPTION]... [-T] SOURCE DEST 拷贝源文件到目标文件。除了 cp 是原封不动输入外，还必须输入 SOURCE 和 DEST 这两部分内容，分别用实际的源文件名和目标文件名代替，也可以加上一个或多个选项参数（替换 OPTION），还可以加上 -T 参数；
- cp [OPTION]... SOURCE... DIRECTORY 拷贝一个或多个源文件到目录，除了 cp 是原封不动输入外，还必须输入 SOURCE 和 DIRECTORY 这两部分内容，分别用实际的源文件名和目录名代替，也可以加上一个或多个选项参数（替换 OPTION）；
- cp [OPTION]... -t DIRECTORY SOURCE... 拷贝一个或多个源文件到目录。除了 cp 是原封不动输入外，还必须输入 DIRECTORY 和 SOURCE 这两部分内容，分别用实际的目录名和源文件名代替。与第二种方法其实效果是一样的，只不过这里把目录名放在了前面，后接需要拷贝到目录的文件名（一个或多个），但是这里的 -t 参数是必须加的，也可以加上一个或多个选项参数（替换 OPTION）。

也不难，对吧？比如我们可以这样用：

```
cp file.jpg file_copy.jpg
```

是将 file.jpg 拷贝为 file_copy.jpg：

```
cp file.jpg file_copy.jpg photo/
```

是将 file.jpg 和 file_copy.jpg 这两个文件拷贝到 photo 这个文件夹中。

man apt-get

我们最后来看一个更复杂一些的实例：apt-get 的使用手册。运行：

```
man apt-get
```



```
oscar@oscar-laptop: ~  
File Edit View Search Terminal Help  
APT-GET(8) APT APT-GET(8)  
  
NAME  
    apt-get - APT package handling utility -- command-line interface  
  
SYNOPSIS  
    apt-get [-asqdyfmubV] [-o=config_string] [-c=config_file]  
            [-t=target_release] [-a=architecture] {update | upgrade |  
            dselect-upgrade | dist-upgrade |  
            install pkg [{=pkg_version_number | /target_release}]... |  
            remove pkg... | purge pkg... |  
            source pkg [{=pkg_version_number | /target_release}]... |  
            build-dep pkg [{=pkg_version_number | /target_release}]... |  
            download pkg [{=pkg_version_number | /target_release}]... |  
            check | clean | autoclean | autoremove | {-v | --version} |  
            {-h | --help}}  
  
DESCRIPTION  
    apt-get is the command-line tool for handling packages, and may be  
    considered the user's "back-end" to other tools using the APT library.  
    Several "front-end" interfaces exist, such as aptitude(8), synaptic(8)  
    and wajig(1).  
  
Manual page apt-get(8) line 1 (press h for help or q to quit)
```

如上图所见，我们的 apt-get 命令的手册的 SYNOPSIS 区域如下：

```
apt-get [-asqdyfmubV] [-o=config_string] [-c=config_file]  
        [-t=target_release] [-a=architecture] {update | upgrade |  
        dselect-upgrade | dist-upgrade |  
        install pkg [{=pkg_version_number | /target_release}]... |  
        remove pkg... | purge pkg... |  
        source pkg [{=pkg_version_number | /target_release}]... |  
        build-dep pkg [{=pkg_version_number | /target_release}]... |  
        download pkg [{=pkg_version_number | /target_release}]... |  
        check | clean | autoclean | autoremove | {-v | --version} |  
        {-h | --help}}
```

虽然这次的 SYNOPSIS 区域多得有点吓人。

但是比起 cp 命令，至少这个 SYNOPSIS 区域只有一行，虽然这一行很长。

我们逐步分析：

- apt-get：这个粗体部分是必须原封不动输入的，这不难理解；
- [-asqdyfmubV]：我们可以使用 -a、-s、-q、-d、-y、-f、-m、-u、-b、-V 这些选项参数中的一个或多个。例如 -s、-sqfm、-bV 等等；
- -o、-c、-t、-a：这四个选项参数比较特别，也是可选的参数，但是如果使用它们，必须要接上 =xxx，xxx 是参数的值；
- 用大括号括起来的（从 update，到最后的 -h | --help）：大括号括起来的是一个整体，而每个部分又由 | 号隔开，说明必须要使用大括号里的某一项，但是每次只能使用一项，不能多项同时使用。

例如：

- update：更新软件包的缓存；
- upgrade：更新所有已安装的软件；
- install pkg：安装 pkg 字段对应的软件包，实际使用时，用实际的软件包名替换 pkg；
- 等等。

当然，可以仔细阅读 DESCRIPTION 区域，以便了解每个参数的不同含义。

用例

根据 apt-get 手册的 SYNOPSIS 区域语法，我们使用 apt-get 时，须要选择大括号里用一竖隔开的任意一个选项。因此，我们可以这样使用：


```
apt-get install xxx
```

或者：

```
apt-get update
```

或者：

```
apt-get autoclean
```

等等。

但是不能同时使用大括号里用 | 号隔开的两个或以上的选项。例如，下面的用法是**错误**的：

```
apt-get update install xxx # 不正确的用法
```

SYNOPSIS 区域的语法总结

- **粗体**：原封不动地输入；
- 下划线的部分：用实际的内容替换；
- [-hvc]：表示 -h，-v 和 -c 选项都是可选的、非强制性的；
- a | b：你可以输入 a 或者 b 选项，但是不能够同时输入 a 和 b；
- option...：省略号表示前面的内容可以输入多个。

4. apropos 命令：查找命令

在使用 `man` 命令时，我们需要先知道我们要显示使用手册的那个命令的名字。那如果不知道那个命令叫什么，该怎么办呢？

这时候我们就需要请 `apropos` 命令上场了，`apropos` 是英语“关于...”的意思。其实这个单词是从法语引进的（英语中大约有 30% 的词汇来源于法语），在法语中原来是 *à propos*，也是“关于...”的意思。

`apropos` 命令的用法很简单，只要后接一个关键字，`apropos` 命令就会为你在所有手册页中查找相关的命令。

因此 `apropos` 命令与 `man` 命令有点对立的关系：`man` 命令用于显示命令的使用手册，而 `apropos` 命令用于根据手册中的关键字来找到命令。

我们用实际例子来学习：你想知道如何用终端的命令来控制音量。

你可以这样查找：

```
apropos sound
```

为什么用 `sound` 这个关键字呢？因为 `sound` 是英语“声音，音量”的意思。

```
File Edit View Search Terminal Help
oscar@oscar-laptop:~$ apropos sound
alsabat (1) - command-line sound tester for ALSA sound card driver
alsactl (1) - advanced controls for ALSA soundcard driver
alsamixer (1) - soundcard mixer for ALSA soundcard driver, with ncurses ...
amixer (1) - command-line mixer for ALSA soundcard driver
aplay (1) - command-line sound recorder and player for ALSA soundcar...
arecord (1) - command-line sound recorder and player for ALSA soundcar...
canberra-gtk-play (1) - play a sound event
default.pa (5) - PulseAudio Sound Server Startup Script
pacat (1) - Play back or record raw or encoded audio streams on a Pu...
pacmd (1) - Reconfigure a PulseAudio sound server during runtime
pactl (1) - Control a running PulseAudio sound server
pamon (1) - Play back or record raw or encoded audio streams on a Pu...
paplay (1) - Play back or record raw or encoded audio streams on a Pu...
parec (1) - Play back or record raw or encoded audio streams on a Pu...
parecord (1) - Play back or record raw or encoded audio streams on a Pu...
pulseaudio (1) - The PulseAudio Sound System
start-pulseaudio-x11 (1) - PulseAudio Sound Server X11 Startup Script
oscar@oscar-laptop:~$
```

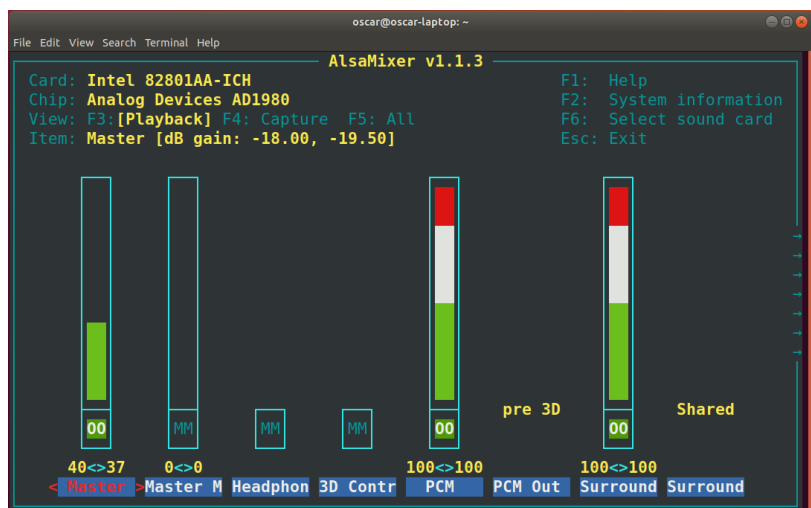
如上图所见，运行 `apropos sound` 之后，就列出了所有使用手册中有 `sound` 这个关键字的命令。

可以看到，左侧是命令的名字，后边是命令手册中出现关键字的句子。

我们就知道了要调节音量可以用的命令是：

```
alsamixer
```

我们运行此命令：



可以用加号和减号来调节，按 `Esc` 键可以退出。

5. 其它阅读手册的方式

虽说 `man` 命令和 `apropos` 命令是最常用的两个有关手册的命令，但是还有其它查阅手册的方式。

那么哪些是 `man` 的替代品呢？

-h 或 --help 参数

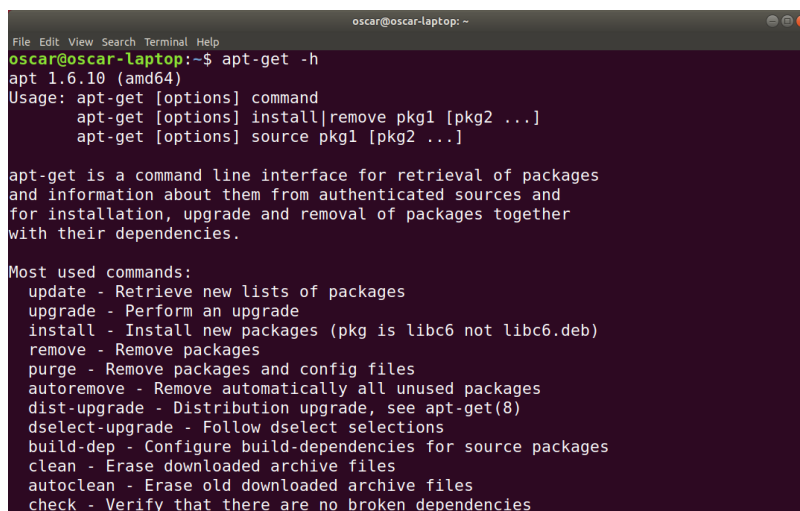
大多数命令一般都接受 `-h` 这个参数（一般来说，`--help` 是它的等价参数）。

`-h` 参数中的 `h` 是 `help` 的首字母，`help` 是英语“帮助”的意思，所以 `-h` 或 `--help` 参数的作用是显示帮助文档。

一般来说，帮助文档没有 `man` 命令显示的使用手册那么详细，但是也很有用，且往往比 `man` 显示的手册易于阅读。

比如我们要查看 apt-get 命令的帮助文档，可以这样：

```
apt-get -h
```

A terminal window titled 'oscar@oscar-laptop: ~' showing the output of the command 'apt-get -h'. The output includes the version 'apt 1.6.10 (amd64)', usage instructions, a description of apt-get as a command line interface, and a list of most used commands with their functions.

```
oscar@oscar-laptop:~$ apt-get -h
apt 1.6.10 (amd64)
Usage: apt-get [options] command
       apt-get [options] install|remove pkg1 [pkg2 ...]
       apt-get [options] source pkg1 [pkg2 ...]

apt-get is a command line interface for retrieval of packages
and information about them from authenticated sources and
for installation, upgrade and removal of packages together
with their dependencies.

Most used commands:
  update - Retrieve new lists of packages
  upgrade - Perform an upgrade
  install - Install new packages (pkg is libc6 not libc6.deb)
  remove - Remove packages
  purge - Remove packages and config files
  autoremove - Remove automatically all unused packages
  dist-upgrade - Distribution upgrade, see apt-get(8)
  dselect-upgrade - Follow dselect selections
  build-dep - Configure build-dependencies for source packages
  clean - Erase downloaded archive files
  autoclean - Erase old downloaded archive files
  check - Verify that there are no broken dependencies
```

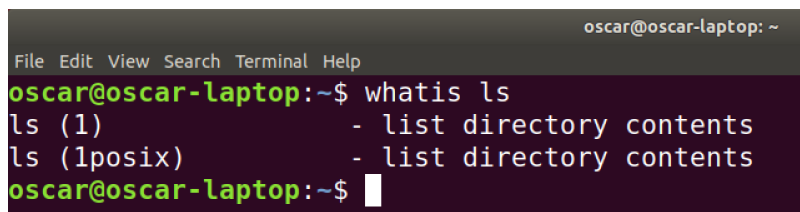
whatis 命令

what 是英语“什么”的意思，is 是英语“是”的第三人称单数形式。因此，what is 就是“是什么”的意思。

whatis 命令可以说是 man 命令的精简版，它只会列出 man 命令显示的手册开头部分，就是概述命令的作用。

例如，用 whatis 命令来查看 ls 命令的作用：

```
whatis ls
```

A terminal window titled 'oscar@oscar-laptop: ~' showing the output of the command 'whatis ls'. The output lists two entries for 'ls', both describing it as a command to list directory contents.

```
oscar@oscar-laptop:~$ whatis ls
ls (1) - list directory contents
ls (1posix) - list directory contents
oscar@oscar-laptop:~$
```

只显示 list directory contents，就是说 ls 命令用于列出目录的内容。

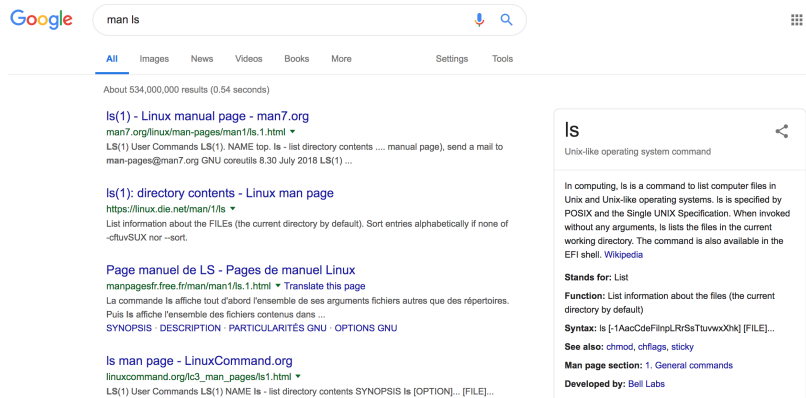
有两行是因为第二行是 POSIX 标准的手册（因为我们之前安装了 POSIX 标准的手册，前面用 `sudo apt-get install manpages-posix-dev manpages-posix` 安装的）的解释。关于 POSIX 标准，我们就不细说了，感兴趣的话可以到网上搜索。

在网络上查看手册

我们不仅仅能在 Linux 系统里查找到手册，在网络上也可以找到。

如果你暂时不在 Linux 系统里，又想查一个命令或函数的使用手册，可以用搜索引擎如 Google 或者百度来搜索。

例如，我要查看 ls 命令的手册，可以在 Google 里搜索：man ls。一般搜索结果的第一个就是你要找的：



小结

1. 在 Linux 中，命令和函数都有使用手册。每次遇到命令和函数的用法不明确的情况，都可以通过查考手册来解决；
2. man 命令可以显示命令或函数的使用手册，man 后接要显示使用手册的名字，比如：man touch；
3. 在手册页时，我们可以用键盘上的上下箭头、PgUp、PgDn、空格来移动到手册的其它部分，用 /（斜杠）键来查找，用 q 键来退出手册；
4. 手册的 SYNOPSIS 区域很重要，列出了我们使用此命令的所有方法，中括号里的内容是可选的；
5. apropos 命令可以用来查找有相应关键字的命令。比如我们要找和音量、声音有关的命令，可以用 apropos sound。

今天的课就到这里，一起加油吧！

← 18 软件安装，如虎添翼

20 查找文件，无所遁形 →

精选留言 1

欢迎在这里发表留言，作者筛选后可公开显示

我是小萌

apt-get 那里既有可选的参数-a又有-a=xxx这个是怎么理解的呢？是如果写了-a，那一定要写成-a=xxx的形式这样的嘛？

👍 0 回复

5天前