

Linux 常用命令指南

2022 年 10 月 24 日

编写说明:

针对许多同学对使用命令行编程不习惯、不熟悉的情况，此参考资料介绍了部分 linux 常用命令。正文内容由计试 2101 王珩编写，计试 2101 杨思成对排版做了些许改动。

由于时间仓促和人手问题，本稿中难免存在错误，欢迎大家批评指正！

钱院学辅-计学组网站：

<https://books.shinonomelab.net/>

目录

1 前言	2
2 常用命令	3
2.1 man	3
2.2 pwd	3
2.3 ls	3
2.4 cd	4
2.5 mkdir	4
2.6 mv	4
2.7 cp	5
2.8 touch	5
2.9 wc	5
2.10 tar	5
2.11 find	6
2.12 grep	7
2.13 history	8
3 小结	8

1 前言

程序设计基础这门课要求大家在 Linux 环境下编程。Linux 是一种自由和开放的类 Unix 的操作系统，相比于其他常见的操作系统，如 Windows 和 macOS，Linux 最大的特点就是自由，这一点大家在今后的学习生活中必然会逐步有更深入的理解。下面主要讲一讲 Linux 中最为常用且有用的命令，帮助大家快速熟悉 Linux。前面的命令相对简单一些，大家熟悉的话可以迅速带过。

2 常用命令

2.1 man

man 是第一个要跟大家介绍的命令，因为 man 可以帮助我们理解其他命令的用法。当我们不知道一个命令（比如 ls 命令）的用法时，我们可以输入以下命令来查看 ls 的 manual（手册）。

```
$ man ls
```

然而 man 显示出来的 manual 内容过于丰富（比如 ls 的 manual 就有 14 页），也许大多数时候我们不需要这么多内容。这时有一个叫做 tldr 的工具就登场了，该命令可能需要通过 `sudo apt install tldr` 下载。使用 `tldr <command>`，可以简明地列出 `<command>` 的一些主要用法（配有例子），例如：

```
$ tldr ls
```

非常推荐大家下载 tldr 命令，并在今后的学习中多多使用。

2.2 pwd

pwd 命令简单而常用，全称为 print work directory，即显示当前工作目录。

```
$ pwd
```

2.3 ls

ls 全称为 list directory contents，用于列出当前目录下的文件和子目录。ls 命令有许多选项可选，常见的有 -a 选项，即“all”，可显示所有文件（包括隐藏文件）；-l 选项可以将文件权限、拥有者、大小等信息列出；-h 选项可以将 -l 选项显示的文件大小从不太好读的字节数转换成容易理解的以 K，M，G 等为单位的大小。例如我们经常会用以下命令查看文件夹内各个文件的大小：

```
$ ls -lh
```

小练习（选自 the missing semester）：阅读 man ls 并写一个 ls 命令满足以下四个要求：

- 包含所有文件（包括隐藏文件）
- 文件大小是便于理解的格式，如显示 454M 而不是 454279954
- 文件按日期排序
- 输出带有颜色

2.4 cd

cd 即 change directory，用于切换工作目录，目录名可以是绝对路径或者相对路径，例如，切换到当前目录的上一级目录：

```
$ cd ..
```

2.5 mkdir

mkdir 即 make directory，用于创建目录。可以同时创建多个目录，例如：

```
$ mkdir folder_1 folder_2
```

也可以通过-p 选项创建嵌套的目录，例如：

```
$ mkdir father_folder/son_folder
```

与之相对应的有 rmdir，用于删除目录，如删去刚刚创建的 folder_1 和 folder_2：

```
$ rmdir folder_1 folder_2
```

2.6 mv

mv 全称 move file，用于移动文件。注意如果 mv 后面最后一个参数是目录时，前面所有的东西都会被移到这个目录中，例如：

```
$ mv A.cpp B.cpp homework_1
```

则 A.cpp 和 B.cpp 都会被移动到 homework_1 中。mv 还可用于为文件重命名，如：

```
$ mv studet.cpp student.cpp
```

将 studet.cpp 移动到 student.cpp，实际上就将名字从 studet.cpp 重命名为 student.cpp。

2.7 cp

cp 全称 copy file，用于复制文件。如：

```
$ cp source destination
```

复制目录时，需要使用-r 选项（recursively）将目录下的所有东西复制，如

```
$ cp folder_1 folder_2
```

将 folder_1 中的所有东西复制到 folder_2。

2.8 touch

touch 可用于创建空的文件，若该文件已存在，则会修改文件的时间属性。如：

```
$ touch A.cpp
```

若 A.cpp 不存在时，则会创建一个空的 A.cpp 文件。

2.9 wc

wc 命令可以用于查看文件的 byte 数，行数和字数。

```
$ wc A.cpp
```

```
# output:
```

```
$ 6 10 48 A.cpp
```

上面的例子中，输出代表 A.cpp 中有 6 行，10 个单词，字节数为 48. 有的时候，如果只想查看部分信息，比如只想获取 A.cpp 到底有多少行，可以使用-l 选项，即：

```
$ wc -l A.cpp
```

```
# output:
```

```
$ 6 A.cpp
```

2.10 tar

tar 全称为 tape archive，用于将多个文件归到一起，创建一个 archive（不是很确定应该怎么翻译）。例如：

```
$ tar -cf archive.tar A.cpp B.cpp
```

可以把 A.cpp 和 B.cpp 打包成 archive.tar，其中-c 选项表示 create，-f 选项用于指定 archive。而解开这个 archive.tar 文件，我们只需要：

```
$ tar -xf archive.tar -C new_code_directory
```

其中-x 选项表示 extract，如果加上-C 选项，则可以把 archive 的内容解开到指定目录中（上面的例子中就把 A.cpp 和 B.cpp 解开了到名叫 new_code_directory 的目录）。然而 tar 最常用的可能是将若干个文件打包后进行压缩，即生成.tar 文件后再将其压缩，变成.tar.gz 的压缩文件。具体用法如下：

```
$ tar -czvf archive.tar.gz A.cpp B.cpp C.cpp
```

其中-z 选择即代表压缩，而-v 选项可以可视化压缩过程，个人在使用时比较喜欢加上。如果要解压缩 archive.tar.gz，则可以使用：

```
$ tar -xvf archive.tar.gz
```

至于创建 archive 和压缩到底区别在哪，为什么要先创建 archive 再压缩，感兴趣的同学可以上 StackOverflow 进一步查找相关资料了解。

2.11 find

查找文件是我们平时经常需要使用的操作。find 命令可以用于查找文件。下面主要给出两个例子加以说明。

- 假设我前两天写了一些 C++ 代码，现在有点想不起来把它们放哪了，那我们可以通过以下命令查找：

```
$ find . -type f -name "*.cpp" -mtime -2
```

其中，. 表示在当前路径下查找（当然也可以换成别的路径）；-type 用于指定查找目标的类型，这里想查找 cpp 文件，故类型为 f(file)；-name 用于指定查找文件的名称，这里可以使用正则表达式，如 “*.cpp” 即查找所有后缀名为 cpp 的文件；-mtime 表示修改时间（m 即 modify），-mtime 后跟着-2 的话表示修改时间在 2 天内（类似的，如果是 +2 则可表示修改时间为两天以上的）。

- find 在找到结果后，后面还能紧接着对查找结果进行操作，比如我突然想删掉当前路径下的最近一天内写的所有 cpp 文件（笑）：

```
$ find . -name "*.cpp" -mtime -1 -exec rm {} \;
```

其中 {} 在执行的时候将会被填充进查找到的文件名，注意结尾的 \; 符号。

2.12 grep

上面的 find 命令主要用于查找文件和目录，而我们有时想按文件内容查找，这时 grep 命令就派上用场了。grep 的全称是 global regular expression print。grep 命令非常非常的有用。比较常规的用法有：

```
$ grep -nC 2 "sort()" A.cpp
```

该命令会在 A.cpp 中查找含有 "sort()" 的部分并输出所在行的内容，-n 选项可以显示出查找到的行号；-C 选项可以显示上下文，后面的 2 表示同时显示查找到的上下两行。常用的还包括 -i 选项，可以使查找对大小写不敏感。

grep 还常常跟 pipe（管道）结合，用于过滤别的命令结果的输出。首先介绍一下什么是 pipe。pipe 运算符为 |，可以将程序连接起来，在 | 前面程序的输出将做为后面程序的输入。用上面讲过的 wc 命令举个例子：

```
$ ls -al | wc -l
```

ls -al 输出的内容将会被当作 wc -l 的输入，则 ls -al 应该显示多少行，最后输出就应该为多少。

在实际中，有时候我们运行了一个名叫 train.py 的 python 文件，现在我想结束它，但已经无法直接结束了，这时一般可以选择直接 kill 掉相应的进程（Windows 大家可能会打开任务管理器去结束相应进程）。

```
$ ps -aux | grep train
```

```
# 假设输出的进程号是 12345
```

```
$ kill -9 12345
```

其中，ps 全称为 process status，即显示当前进程的状态，-aux 选项用于显示所有包含其他使用者的进程。进程的状态输出后，将会成为 grep train 的输入，grep train 去匹配其中名字含有 train 的进程，最后会输出相应的 pid(process identifiers)，假设输出的进程号是 12345，那我们就可以通过 kill 对应进程号，结束进程。

2.13 history

有时候我们还需要去查找过去输过的命令。history 命令可以让我们看到过去使用过的命令。例如，查找过去命令中含有 g++ 的命令：

```
$ history | grep g++
```

以上三个命令主要关于各个方面的查找，下面给出一个小练习巩固一下。

小练习：递归地找到所有 cpp 文件，并将他们打包压缩。

3 小结

由于时间仓促和个人水平有限，给出的 Linux 常用命令指南难免有疏漏和不足。希望大家能在使用中熟练掌握 Linux 的各种工具，熟悉并爱上 Linux，而不要在大一上就对 Linux 产生恐惧心理。进一步学习请参考<https://missing.csail.mit.edu/>。