

Shell 的 18 条常用命令整理

1、ls: 类似于 dos 下的 dir 命令

ls 最常用的参数有三个: `-a -l -F`。

```
ls -a
```

Linux 上的文件以 . 开头的文件被系统视为隐藏文件, 仅用 ls 命令是看不到他们的, 而用 ls -a 除了显示一般文件名外, 连隐藏文件也会显示出来。

```
ls -l
```

该参数显示更详细的文件信息。

```
ls -F
```

使用这个参数表示在文件的后面多添加表示文件类型的符号, 例如*表示可执行, /表示目录, @表示连结文件, 这都是因为使用了-F 这个参数。但是现在基本上所有的 Linux 发行版本的 ls 都已经内建了-F 参数, 也就是说, 不用输入这个参数, 我们也能看到各种分辨符号。

2、cd: 用于切换用户当前工作目录

```
cd aaa 进入 aaa 目录
```

```
cd 命令后不指定目录, 会切换到当前用户的 home 目录
```

```
cd ~ 作用同 cd 后不指定目录, 切换到当前用户的 home 目录
```

```
cd - 命令后跟一个减号, 则会退回到切换前的目录
```

```
cd .. 返回到当前目录下的上一级目录
```

3、pwd: 用于显示用户当前工作目录

4、mkdir 和 rmdir

mkdir:创建目录 rmdir:删除目录

两个命令都支持-p 参数，对于 mkdir 命令若指定路径的父目录不存在则一并创建，对于 rmdir 命令则删除指定路径的所有层次目录，如果文件夹里有内容，则不能用 rmdir 命令

如下：

```
mkdir -p 1/2/3
```

```
rmdir -p 1/2/3
```

5、cp: 复制命令

复制一个文件到另一目录: cp 1.txt ../test2

复制一个文件到本目录并改名: cp 1.txt 2.txt

复制一个文件夹 a 并改名为 b: cp -r a b

6、mv: 移动命令

将一个文件移动到另一个目录: mv 1.txt ../test1

将一个文件在本目录改名: mv 1.txt 2.txt

将一个文件一定到另一个目录并改名: mv 1.txt ../test1/2.txt

7、rm 命令

rm 命令用于删除文件，与 dos 下的 del/erase 命令相似，rm 命令常用的参数有三个：-i，-r，-f。

-i : 系统在删除文件之前会先询问确认，用户回车之后，文件才会真的被删除。需要注意，linux 下删除的文件是不能恢复的，删除之前一定要谨慎确认。

-r: 该参数支持目录删除，功能和 rmdir 命令相似。

-f: 和-i 参数相反，-f 表示强制删除

8、du、df 命令

du 命令可以显示目前的目录所占用的磁盘空间，df 命令可以显示目前磁盘剩余空间。

如果 du 命令不加任何参数，那么返回的是整个磁盘的使用情况，如果后面加了目录的话，就是这个目录在磁盘上的使用情况。

```
du -hs 指定目录    查看指定目录的总大小
```

```
du -hs ./ *    查看当前目录下的所有文件夹和文件的大小
```

这两个命令都支持 -k, -m 和 -h 参数，-k 和 -m 类似，都表示显示单位，一个是 k 字节一个是兆字节，-h 则表示 human-readable，即友好可读的显示方式。

9、cat 命令

cat 命令的功能是显示或连结一般的 ascii 文本文件。cat 是 concatenate 的简写，类似于 dos 下面的 type 命令。用法如下：

```
cat file1    显示 file1 文件内容
```

```
cat file1 file2    依次显示 file1, file2 的内容
```

```
cat file1 file2 > file3    把 file1, file2 的内容结合起来，再“重定向 (>)”到 file3 文件中。
```

“>”是右重定向符，表示将左边命令结果当成右边命令的输入，注意：如果右侧文件是一个已存在文件，其原有内容将会被清空，而变成左侧命令输出内容。如果希望以追加方式写入，请改用“>>”重定向符。

如果“>”左边没有指定文件，如：`cat >file1`，将会等用户输入，输入完毕后再按 [Ctrl]+[c] 或 [Ctrl]+[d]，就会将用户的输入内容写入 file1。

10、echo 命令

echo 命令的使用频率不少于 ls 和 cat，尤其是在 shell 脚本编写中。

语法：`echo [-ne][字符串]`

功能：echo 会将输入的字符串送往标准输出，输出的字符串间以空白字符隔开，并在最后加上换行符。

参数：

-n 显示字符串时在最后自动换行
-e 支持以下格式的转义字符， -E 不支持以下格式的转义字符
/a 发出警告声；
/b 删除前一个字符；
/c 最后不加上换行符号；
/f 换行但光标仍旧停留在原来的位置；
/n 换行且光标移至行首；
/r 光标移至行首，但不换行；
/t 插入 tab；
/v 与/f 相同；
// 插入/字符；
/nnn 插入 nnn（八进制）所代表的 ASCII 字符；

示例：

```
Oracle@vxbus:~/vxworks> echo "123" "456"
123 456
oracle@vxbus:~/vxworks> echo "123/n456"
123/n456
oracle@vxbus:~/vxworks> echo -e "123/n456"
123
456
oracle@vxbus:~/vxworks> echo -E "123/n456"
123/n456
oracle@vxbus:~/vxworks> echo -E "123///456"
123//456
oracle@vxbus:~/vxworks> echo -e "123///456"
123/456
oracle@vxbus:~/vxworks> echo -e "123/100456"
123@456
```

注意事项：

在 Linux 使用的 bash 下，单引号 ' ' 和双引号是有区别的，单引号忽略所有的转义，双引号不会忽略以下特殊字符：

Dollar signs (\$), Back quotes (`), Backslashes (/), Exclamatory mark(!)

示例如下：

```
oracle@vxbus:~> echo ``TEST``
-bash: TEST: command not found
oracle@vxbus:~> echo '`TEST`'
`TEST`
oracle@vxbus:~> echo "$TEST"

oracle@vxbus:~> echo '$TEST'
$TEST
oracle@vxbus:~> echo "//TEST"
/TEST
oracle@vxbus:~> echo '//'TEST'
//TEST
oracle@vxbus:~> echo "Hello!"
echo "Hello"
Hello
oracle@vxbus:~> echo 'Hello!'
Hello!
```

11、more, less, clear

more, less 命令

这两个命令用于查看文件，如果一个文件太长，显示内容超出一个屏幕，用 cat 命令只能看到最后的内容，用 more 和 less 两个命令可以分页查看。more 指令可以使超过一页的文件内容分页暂停显示，用户按键后才继续显示下一页。而 less 除了有 more 的功能以外，还可以用方向键往上或往下的滚动文件，更方便浏览阅读。

less 的常用动作命令：

回车键 向下移动一行；

y 向上移动一行；

空格键 向下滚动一屏；

b 向上滚动一屏；

d 向下滚动半屏；

h less 的帮助；

u 向上滚动半屏；

w 可以指定显示哪行开始显示，是从指定数字的下一行显示；比如指定的是 6，那就从第 7 行显示；

g 跳到第一行；

G 跳到最后一行；

p n% 跳到 n%，比如 10%，也就是说比整个文件内容的 10%处开始显示；

/pattern 搜索 pattern，比如 /MAIL 表示在文件中搜索 MAIL 单词；

v 调用 vi 编辑器；

q 退出 less

!command 调用 SHELL，可以运行命令；比如 !ls 显示当前列当前目录下的所有文件；

clear 命令

clear 命令是用来清除当前屏幕显示的，不需任何参数，和 dos 下的 cls 命令功能相同。

12、head, tail

head 和 tail 命令

都用于查看文本文件，区别在于：head 显示文件的头 n 行，tail 显示文件的尾 n 行，缺省情况 n 都为 10 行。可以通过 -n 方式指定行数，如：

head -100 file 和 tail -100 file 分别表示显示文件头 100 行和尾 100 行内容。

tail -f 命令

可以实时查看文件新增内容。

13、wc 命令

该命令用于统计指定文件中的字节数、字数、行数。该命令各选项含义如下：

-l 统计行数

-w 统计字数
-c 统计字节数

这些选项可以组合使用。输出列的顺序和数目不受选项的顺序和数目的影响。总是按下述顺序显示并且每项最多一列。

行数、字数、字节数、文件名

如果命令行中没有文件名，则输出中不出现文件名。

例如：

```
oracle@vxbus:~> wc 1.txt 2.txt
```

```
460    1679 16353 1.txt
```

```
300    1095 10665 2.txt
```

```
760    2774 27018 总用量
```

```
oracle@vxbus:~> wc -l 1.txt
```

```
460 1.txt
```

缺省参数为-lcw，即 `wc file1 file2` 命令的执行结果与上面一样。

14、grep 命令

grep 是 (global search regular expression(RE) and print out the line) 的缩写，用于从文件面搜索包含指定模式的行并打印出来，它是一种强大的文本搜索工具，支持使用正则表达式搜索文本。grep 的工作方式是这样的，它在一个或多个文件中搜索字符串模板。如果模板包括空格，则必须被” ” 引用，模板后的所有字符串被看作文件名。搜索结果送到屏幕，不影响原文件内容。

grep 可用于 shell 脚本，因为 grep 通过返回一个状态值来说明搜索的状态，如果模板搜索成功，则返回 0，如果搜索不成功，则返回 1，如果搜索的文件不存在，则返回 2。我们利用这些返回值就可进行一些自动化的文本处理工作。

示例：

```
$ ls -l | grep '^a'
```

通过管道过滤 `ls -l` 输出的内容，只显示以 a 开头的行。

```
$ grep 'test' d*
```

显示所有以 d 开头的文件中包含 test 的行。

```
$ grep 'test' aa bb cc
```

显示在 aa, bb, cc 文件中匹配 test 的行。

```
$ grep '[a-z]{5/}' aa
```

显示所有包含每个字符串至少有 5 个连续小写字母的字符串的行。

```
$ grep 'w(es)t.*1' aa
```

如果 west 被匹配，则 es 就被存储到内存中，并标记为 1，然后搜索任意个字符（.*），这些字符后面紧跟着另外一个 es (/1)，找到就显示该行。如果用 egrep 或 grep -E，就不用 "/" 号进行转义，直接写成 'w(es)t.*1' 就可以了。

15、man, logout 命令

man 命令

man 是 manual 的缩写，相当于 Unix/Linux 的联机 Help，每个系统命令和调用都有非常详细的说明，绝大多数都是英文。如：man ls 即是查看 ls 命令的使用说明，一般还有另一种方法用来查看帮助，如：ls -help，这种方式绝大多数命令都支持。

logout 命令

该命令用于退出系统，与 login 命令对应。

16、管道和 xargs

管道：

利用 Linux 所提供的管道符“|”将两个命令隔开，管道符左边命令的输出就会作为管道符右边命令的输入。连续使用管道意味着第一个命令的输出会作为第二个命令的输入，第二个命令的输出又会作为第三个命令的输入，依此类推。

注意：管道左边命令的输入作为管道右边命令的输入（命令的输入是一定的），不是参数，并不是所有命令都支持管道

例子：ls | grep a 查看当前目录下名称包含 a 的文件或文件夹

xargs:

大多数 Linux 命令都会产生输出：文件列表、字符串列表等。但如果要使用其他某个命令并将前一个命令的输出作为参数该怎么办？例如，file 命令显示文件类型（可执行文件、ascii 文本等）；你能处理输出，使其仅显示文件名，目前你希望将这些名称传递给 ls -l 命令以查看时间戳记。xargs 命令就是用来完成此项工作的。

注意：find 命令把匹配到的文件传递给 xargs 命令，而 xargs 命令每次只获取一部分文件而不是全部，不像-exec 选项那样。这样它可以先处理最先获取的一部分文件，然后是下一批，并如此继续下去

例子：

1、在整个系统中查找内存信息转储文件(core dump) ，然后把结果保存到

/tmp/core.log 文件中：

```
$ find / -name "core" -print | xargs echo "" >/tmp/core.log
```

2、当一个目录下文件太多时，直接用 rm * 命令会包参数过长,用如下方法可以全部删除

```
$ls | xargs rm
```

17、basename 和 dirname

basename 用于查看文件不含路径的名字，dirname 则用于查看文件路径，使用效果我们测试一下便知：

```
> basename /home/hj/1.txt
```

```
1.txt
```

```
> dirname /home/hj/1.txt
```

```
/home/hj
```

```
> basename 1.txt
```

```
1.txt
```

```
> dirname 1.txt
```

```
.
```

18、不带任何参数的 set 命令将显示 shell 的全部变量