

Linux 教程

- Linux 教程
- Linux 简介
- Linux 安装
- Linux 云服务器
- Linux 系统启动过程
- Linux 系统目录结构
- Linux 忘记密码解决方法
- Linux 远程登录
- Linux 文件基本属性
- Linux 文件与目录管理
- Linux 用户和用户组管理
- Linux 磁盘管理
- Linux vi/vim
- linux yum 命令
- Linux apt 命令

Shell 教程

- Shell 教程
- Shell 变量
- Shell 传递参数
- Shell 数组
- Shell 运算符
- Shell echo命令
- Shell printf命令
- Shell test 命令
- Shell 流程控制
- Shell 函数
- Shell 输入/输出重定向
- Shell 文件包含

Linux 参考手册

Linux 命令大全

Nginx 安装配置

MySQL 安装配置

Shell 文件包含

Nginx 安装配置

Linux sed 命令

Linux 命令大全

Linux sed 命令是利用脚本本来处理文本文件。
sed 可依照脚本的指令来处理、编辑文本文件。
Sed 主要用来自动编辑一个或多个文件、简化对文件的反复操作、编写转换程序等。

语法

sed [-hnV] [-e<script>] [-f<script文件>] [文本文件]

参数说明:

- e<script>或--expression=<script> 以选项中指定的script来处理输入的文本文件。
- f<script文件>或--file=<script文件> 以选项中指定的script文件来处理输入的文本文件。
- h或--help 显示帮助。
- n或--quiet或--silent 仅显示script处理后的结果。
- V或--version 显示版本信息。

动作说明:

- a：新增，a 的后面可以接字符串，而这些字符串会在新的一行出现(目前的下一行)~
- c：取代，c 的后面可以接字符串，这些字符串可以取代 n1,n2 之间的行！
- d：删除，因为是删除啊，所以 d 后面通常不接任何东东；
- i：插入，i 的后面可以接字符串，而这些字符串会在新的一行出现(目前的上一行)；
- p：打印，亦即将某个选择的数据印出。通常 p 会与参数 sed -n 一起运行~
- s：取代，可以直接进行取代的工作哩！通常这个 s 的动作可以搭配正则表达式！例如 1.20s/old/new/g 就是啦！

实例

我们先创建一个 testfile 文件，内容如下：

```
$ cat testfile #查看testfile 中的内容
HELLO LINUX!
Linux is a free unix-type operating system.
This is a linux testfile!
Linux test
Google
Taobao
Runoob
Tetestfile
Wiki
```

在 testfile 文件的第四行后添加一行，并将结果输出到标准输出，在命令行提示符下输入如下命令：

```
sed -e 4a\newline testfile
```

使用 sed 命令后，输出结果如下：

```
$ sed -e 4a\newline testfile
HELLO LINUX!
Linux is a free unix-type operating system.
This is a linux testfile!
Linux test
newline
Google
Taobao
Runoob
Tetestfile
Wiki
```

以行为单位的新增/删除

将 testfile 的内容列出并且列印行号，同时，请将第 2~5 行删除！

```
$ nl testfile | sed '2,5d'
1  HELLO LINUX!
6  Taobao
7  Runoob
8  Tetestfile
9  Wiki
```

sed 的动作为 2,5d，那个 d 是删除的意思，因为删除了 2-5 行，所以显示的数据就没有 2-5 行了，另外，原本应该是要下达 sed -e 才对，但没有 -e 也是可以的，同时也要注意的，sed 后面接的动作，请务必以 '...' 两个单引号括住哩！
只要删除第 2 行：

```
$ nl testfile | sed '2d'
1  HELLO LINUX!
3  This is a linux testfile!
4  Linux test
5  Google
6  Taobao
7  Runoob
8  Tetestfile
9  Wiki
```

要删除第 3 到最后一行：

```
$ nl testfile | sed '3,$d'
1  HELLO LINUX!
2  Linux is a free unix-type operating system.
```

在第二行后(即加在第三行) 加上drink tea? 字样：

```
$ nl testfile | sed '2a drink tea'
1  HELLO LINUX!
2  Linux is a free unix-type operating system.
drink tea
3  This is a linux testfile!
4  Linux test
5  Google
6  Taobao
7  Runoob
8  Tetestfile
9  Wiki
```

如果是要在第二行前，命令如下：

```
$ nl testfile | sed '2i drink tea'
1  HELLO LINUX!
drink tea
2  Linux is a free unix-type operating system.
3  This is a linux testfile!
4  Linux test
5  Google
6  Taobao
7  Runoob
8  Tetestfile
9  Wiki
```

如果是要增加两行以上，在第二行后面加入两行字，例如 Drink tea or 与 drink beer?

```
$ nl testfile | sed '2a Drink tea or .....\\
drink beer ?'
1  HELLO LINUX!
2  Linux is a free unix-type operating system.
Drink tea or .....
drink beer ?
3  This is a linux testfile!
4  Linux test
5  Google
6  Taobao
7  Runoob
8  Tetestfile
9  Wiki
```

每一行之间都必须要以反斜杠 \ 来进行新行标记。上面的例子中，我们可以发现在第一行的最后面就有 \ 存在。

以行为单位的替换与显示

将第 2-5 行的内容取代成为 No 2-5 number 呢？

```
$ nl testfile | sed '2,5c No 2-5 number'
1  HELLO LINUX!
No 2-5 number
6  Taobao
7  Runoob
8  Tetestfile
9  Wiki
```

透过这个方法我们就能够将数据整行取代了。

仅列出 testfile 文件内的第 5-7 行：

```
$ nl testfile | sed -n '5,7p'
5  Google
6  Taobao
7  Runoob
```

可以透过这个 sed 的以行为单位的显示功能， 就能够将某一个文件内的某些行号选择出来显示。

数据的搜寻并显示

搜索 testfile 有 oo 关键字的行：

```
$ nl testfile | sed -n '/oo/p'
5  Google
7  Runoob
```

如果 root 找到，除了输出所有行，还会输出匹配行。

数据的搜寻并删除

删除 testfile 所有包含 oo 的行，其他行输出

```
$ nl testfile | sed '/oo/d'
1  HELLO LINUX!
2  Linux is a free unix-type operating system.
3  This is a linux testfile!
4  Linux test
6  Taobao
8  Tetestfile
9  Wiki
```

数据的搜寻并执行命令

搜索 testfile，找到 oo 对应的行，执行后面花括号中的一组命令，每个命令之间用分号分隔，这里把 oo 替换为 kk，再输出这行：

```
$ nl testfile | sed -n '/oo/{s/oo/kk;p;q}'
5  Gkkgle
```

最后的 q 是退出。

数据的查找与替换

除了整行的处理模式之外，sed 还可以用行为单位进行部分数据的查找与替换<。

sed 的查找与替换的与 vi 命令类似，语法格式如下：

```
sed 's/要被取代的字符串/新的字符串/g'
```

将 testfile 文件中每行第一次出现的 oo 用字符串 kk 替换，然后将该文件内容输出到标准输出：

```
sed -e 's/oo/kk/' testfile
```

g 标识符表示全局查找替换，使 sed 对文件中所有符合的字符串都被替换，修改后内容会到标准输出，不会修改原文件：

```
sed -e 's/oo/kk/g' testfile
```

选项 i 使 sed 修改文件：

```
sed -i 's/oo/kk/g' testfile
```

批量操作当前目录下以 test 开头的文件：

```
sed -i 's/oo/kk/g' ./test*
```

接下来我们使用 /sbin/ifconfig 查询 IP：

```
$ /sbin/ifconfig eth0
eth0  Link encap:Ethernet  HWaddr 00:90:CC:A6:34:84
inet  addr:192.168.1.100 Bcast:192.168.1.255 Mask:255.255.255.0
inet6  addr: fe80::290:ccff:fea6:3484/64 Scope:Link
UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
.....(以下省略).....
```

本机的 ip 是 192.168.1.100。

将 IP 前面的部分予以删除：

```
$ /sbin/ifconfig eth0 | grep 'inet addr' | sed 's/^\.*addr://g'
192.168.1.100 Bcast:192.168.1.255 Mask:255.255.255.0
```

接下来则是删除后续的部分，即：192.168.1.100 Bcast:192.168.1.255 Mask:255.255.255.0。

将 IP 后面的部分予以删除：

```
$ /sbin/ifconfig eth0 | grep 'inet addr' | sed 's/^\.*addr://g' | sed 's/Bcast.*$/g'
192.168.1.100
```

多点编辑

一条 sed 命令，删除 testfile 第三行到末尾的数据，并把 HELLO 替换为 RUNOOB：

```
$ nl testfile | sed -e '3,$d' -e 's/HELLO/RUNOOB/'
1  RUNOOB LINUX!
2  Linux is a free unix-type operating system.
```

-e 表示多点编辑，第一个编辑命令删除 testfile 第三行到末尾的数据，第二条命令搜索 HELLO 替换为 RUNOOB。

直接修改文件内容(危险动作)

sed 可以直接修改文件的内容，不必使用管道命令或数据流重定向！不过，由于这个动作会直接修改到原始的文件，所以请你千万不要随便拿系统配置来测试！我们还是使用文件 regular_express.txt 文件来测试看看吧！
regular_express.txt 文件内容如下：

```
$ cat regular_express.txt
runoob.
google.
taobao.
facebook.
zhihu-
weibo-
```

利用 sed 将 regular_express.txt 内每一行结尾若为 . 则换成！

```
$ sed -i 's/\.$!/g' regular_express.txt
$ cat regular_express.txt
runoob!
google!
taobao!
facebook!
zhihu-
weibo-
```

q;q
利用 sed 直接在 regular_express.txt 最后一行加入 # This is a test:

```
$ sed -i 'sa # This is a test' regular_express.txt
$ cat regular_express.txt
runoob!
google!
taobao!
facebook!
zhihu-
weibo-
# This is a test
```

由於 # 代表的是最后一行，而 a 的动作是新增，因此该文件最后新增 # This is a test!
sed 的 -i 选项可以直接修改文件内容，这功能非常有帮助！举例来说，如果你有一个 100 万行的文件，你要在第 100 行加某些文字，此时使用 vim 可能会崩溃！因为文件太大了！那怎么办？就利用 sed 啊！透过 sed 直接修改/取代的功能，你甚至不需要使用 vim 去修订！

Linux 命令大全

1 篇笔记

写笔记

追加的说明:

```
sed -e 4a\newline testfile
```

149 a 动作是在匹配的行之后追加字符串，追加的字符串中可以包含换行符（实现追加多行的情况）。
追加一行的话前后都不需要添加换行符 \n，只有追加多行时在行与行之间才需要添加换行符(最后一行最后也无需添加，添加的话会多出一个空行)。
man sed 信息：

```
Append text, which has each embedded newline preceded by a backslash.
```

例如：

4 行之后追加一行：

```
sed -e '4 a newline' testfile
```

4 行之后追加 2 行：

```
sed -e '4 a newline\nnewline2' testfile
```

4 行之后追加 3 行(2 行文字和 1 行空行)

```
sed -e '4 a newline\nnewline2\n' testfile
```

4 行之后追加 1 行空行：

```
#错误: sed -e '4 a \n' testfile
sed -e '4 a \ ' testfile 实际上
```

实际上是插入了一个含有一个空格的行，插入一个完全为空的空行没有找到方法（不过应该没有这个需求吧，都要插入行了插入空行干嘛呢？）

添加空行：

```
# 可以添加一个完全为空的空行
sed '4 a \'
```

```
# 可以添加两个完全为空的空行
sed '4 a \n\'
```

Jachin 4 年前 (2019-09-02)