

## linux基础命令之：vi模式下查找和替换 - nylic的博客 - CSDN博客

### 一、查找

查找命令 `/pattern<Enter>` :                      向下查找 `pattern` 匹配字符串 `?pattern<Enter>`:                      向上查找 `pattern` 匹配字符串 使用了查找命令之后，使用如下两个键快速查找:                      `n`: \_\_\_\_\_>  
按照同一方向继续查找                      `N`: \_\_\_\_\_> 按照反方向查找 `pattern` 是需要匹配的字符串，例如：  
`/hello<Enter>`                      #查找 `hello`                      `/hello<Enter>`                      #查找 `hello` 单词  
(注意前后的空格) 除此之外，`pattern` 还可以使用一些特殊字符，包括 (`/`、`^`、`$`、`*`、`.`)，其中前三个这两个是 `vi` 与 `vim` 通用的，`\"/>`

- 1
- 2
- 3
- 4
- 5
- 6
- 7
- 8
- 9
- 10
- 11
- 12
- 13
- 14
- 15
- 16
- 17
- 1
- 2
- 3
- 4
- 5
- 6
- 7
- 8
- 9
- 10
- 11
- 12
- 13
- 14
- 15
- 16
- 17

## 二、替换

### a) 常用替换:

`: [range] s/pattern/string/[c,e,g,i] range` 指的是范围, 1,7 指从第一行至第七行,  
`1,$` 指从第一行至最后一行, 也就是整篇文章, 也可以 `%` 代表。 `%` 是目前编辑的文章,  
`#` 是前一次编辑的文章。—————`pattern` 就是要被替换掉的字符串, 可以用  
`regexp` 来表示。 `string` 将 `pattern` 由 `string` 所取代。—————`c`  
`confirm`, 每次替换前会询问。 `e` 不显示 `error`。 `g` `globe`, 不询问, 整行替换。 `i`  
`ignore` 不分大小写。 `I` `ignore` 大小写敏感。

- 1
- 2
- 3
- 4
- 5
- 6
- 7
- 8
- 9
- 10
- 11
- 12
- 13
- 14
- 15

- 1
- 2
- 3
- 4
- 5
- 6
- 7
- 8
- 9
- 10
- 11
- 12
- 13
- 14
- 15

### b) . 基本替换

`:s/lantian/sky/#` 替换当前行第一个 `lantian` 为 `sky`: `:s/lantian/sky/g` 替换当前行所有 `lantian` 为 `sky`: `:n,$s/lantian/sky/` # 替换第 `n` 行开始到最后一行中每一行的第一个 `lantian` 为 `sky`: `:n,$s/lantian/sky/g` # 替换第 `n` 行开始到最后一行中每一行所有 `lantian` 为 `sky` # (`n` 为数字, 若 `n` 为 `.`, 表示从当前行开始到最后一行): `:%s/lantian/sky/#` (等同于 `:g/lantian/s//sky/`) 替换每一行的第一个 `lantian` 为 `sky`: `:%s/lantian/sky/g#` (等同于 `:g/lantian/s//sky/g`) 替换每一行中所有 `lantian` 为 `sky` 可以使用 `#` 或 `+` 作为分隔符, 此时中间出现

的 / 不会作为分隔符:s#lantian/#sky/# 替换当前行第一个 lantian/ 为  
sky/:%s+/oradata/apras/+/user01/apras1+ (使用+ 来 替换 / ): /oradata/apras/替换  
成/user01/apras1/

- 1
- 2
- 3
- 4
- 5
- 6
- 7
- 8
- 9
- 10
- 11
- 12
- 13
- 14

- 1
- 2
- 3
- 4
- 5
- 6
- 7
- 8
- 9
- 10
- 11
- 12
- 13
- 14

### 3. 删除文本中的^M

问题描述: 对于换行, window下用回车换行 (0A0D) 来表示, linux下是回车 (0A) 来表示。这样, 将window上的文件拷到unix上用时, 总会有个^M, 请写个用在unix下的过滤windows文件的换行符 (0D) 的shell或c程序。使用命令: cat filename1 | tr -d ``^V^M`` > newfile;使用命令: sed -e ``s/^V^M//``

filename > outputfilename需要注意的是在1、2两种方法中, ^V和^M指的是Ctrl+V和Ctrl+M。你必须手工进行输入, 而不是粘贴。在vi中处理: 首先使用vi打开文件, 然后按ESC键, 接着输入命

令: :%s/^V^M//:%s/^M\$/g如果上述方法无用, 则正确的解决办法是: tr -d ``/r`` < src >desttr -d ``/015`` deststrings A>B

- 1
- 2
- 3
- 4
- 5
- 6

- 7
- 8
- 9
- 10
- 11
- 12

- 1
- 2
- 3
- 4
- 5
- 6
- 7
- 8
- 9
- 10
- 11
- 12

#### 4. 其它用法

:s/str1/str2/#用字符串 str2 替换行中首次出现的字符串 str1:s/str1/str2/g#用字符串 str2 替换行中所有出现的字符串 str1:.,\$ s/str1/str2/g #用字符串 str2 替换正文当前行到末尾所有出现的字符串 str1:1,\$ s/str1/str2/g #用字符串 str2 替换正文中所有出现的字符串 str1:g/str1/s//str2/g#功能同上

- 1
- 2
- 3
- 4
- 5

- 1
- 2
- 3
- 4
- 5

#### 5. g的总结说明

从上述替换命令可以看到：g 放在命令末尾，表示对指定行的搜索字符串的每次出现进行替换；不加 g，表示只对指定行的搜索字符串的首次出现进行替换；g 放在命令开头，表示对正文中所有包含搜索字符串的行进行替换操作。也就是说命令的开始可以添加影响的行，如果为g表示对所有行；命令的结尾可以使用g来表示是否对每一行的所有字符串都有影响。

- 1
- 2
- 3
- 4
- 1

- 2
- 3
- 4

[来源点这](#)