

vi是文本编辑器，只编辑纯ASCII码的文档，没有任何多余的修饰符。

首先需要确保系统已经安装了 VI 工具，那么怎样去检查是否已经安装了呢？

使用 rpm 命令查看是否已安装 vi：

```
1 [root@exercise1 ~]# rpm -qf `which vi`  
2 vim-minimal-7.4.160-2.el7.x86_64  
3  
4 ###以上输入的查询结果表示已安装好
```

vim是vi的**增加版**，最明显的区别就是vim可以**语法加亮**，它完全兼容vi

```
1 安装  
2 yum -y install vim
```

## 模式介绍：

```
1 (1) 首次进入文件 处于“一般模式”。  
2 (2) 出现 “Insert” 处于“编辑模式”。  
3 (3) 输入 ： 处于“命令行模式”。  
4  
5 即有三种模式：一般模式、编辑模式、命令行模式
```

## 模式切换

```
1 1. 一般模式 ==> 编辑模式：  
2 (1) i：当前字符之前插入 （光标前）  
3 (2) I：行首插入 （行首）  
4 (3) a：当前字符之后插入 （光标后）  
5 (4) A：行尾插入（行尾）  
6 (5) o：下一行插入 （另起一行）  
7 (6) O：上一行插入（上一行插入）  
8  
9 2. 在一般模式下做的操作：  
10 (1) h：光标向左移动  
11 (2) j：光标向下移动
```

12 (3) **k**: 光标向上移动  
 13 (4) **l** : 光标向右移动  
 14 (5) **O** 和 **home** 键: 切换到行首, **\$**和 **end** 键表示切换到行尾  
 15 (6) **gg**: 快速定位到文档的首行, **G** 定位到末行  
 16 (7) **3gg**: 或者 **3G** 快速定位到第 3 行  
 17 (8) **u**: 撤销一步, 每按一次就撤销一次  
 18 (9) **r** : 替换  
 19 (10) **/string(字符串)**: 找到或定位你要找的单词或内容, 如果相符内容比较多,  
 20 我们可以通过 **N**、**n** 来进行向上向下查找, 并且 **vi** 会对查找到的内容进行高亮显示,  
 21 取消用 **:noh**  
 22 (11) **/^d**: **^**符号表示以什么开头, 查找以字母 **d** 开头的内容  
 23 (12) **/t\$**: **\$**符号表示以什么结尾, 查找以字母 **t** 结尾的内容  
 24  
 25 3. 一般模式---》对文本进行编辑:

对于文本的内容编辑主要有: 删除、复制、粘贴、撤销、等等常见操作。

(1) **y**: 复制 (以字符为单位) 表示对单个字符进行复制, 如果要复制整行, 用 **yy** (以行为单位)

(2) **Nyy**: 复制 **N** 行, 比如: **2yy**, 表示在光标所在的行往下复制 2 行

(3) **dd**: 删除, 以行为单位, 删除当前光标所在行

(4) **Ndd**: 删除 **N** 行, 比如: **2dd**, 表示光标所在行往下删除 2 行

(5) **p**: 在光标所在位置往下插入粘贴的内容

(7) **x**: 向后删除一个字符, 等同于 **Delete**

(8) **X**: 向前删除一个字符

(9) **D**: 从光标处删除到行尾

(10) **u**: 撤销操作

(11) **ctrl+r**: 还原撤销过的操作, 将做过的撤销操作再还原回去, 也就是说撤销前是什么样, 再还原成什么样

总结: **vi** 如何进入其它模式。

(1) **a**、**A**、**o**、**O**、**i**、**I** 都是可以进行插入, 编辑模式

(2) **:** 输入冒号进入命令行模式

(3) **r** 擦除、改写, 进入替换模式

(4) 进入以上模式后，想要退出，按 esc

#### 4. 命令行模式操作

(1) :w 保存 save

(2) :w! 强制保存

(3) :q 没有进行任何修改，退出 quit

(4) :q! 修改了，不保存，强制退出

(5) :wq 保存并退出

(6) :wq! 强制保存并退出

(7) :x 保存退出

(8) :w filename 保存到新文件

(9) :n1,n2 w filename 将n1到n2的内容保存到新文件

## 文本替换:

**格式：范围（其中%代表所有内容） s分隔符 旧的内容 分隔符 新的内容**  
(分隔符可以自定义)

**默认是每一行的第一个符合要求的词 (/g全部)**

1	:1,3 s/Jan/Feb	替换第1到3行中出现的第一个Jan进行替换为Feb
2	:1,3 s/Jan/Feb/g	替换第1到3行中查找到所有的Jan进行替换为Feb
3	:3 s/Jan/January	只把第3行中内容替换了
4	:% s/Jan/January/g	将文本中所有的Jan替换成January
5	:% s/Jan/January/gi	将文本中所有的Jan替换成January，并且忽略Jan的大小写
6	:% s@a@b@g	将文本中所有的a替换成b
7	:1,\$ s/Jan/January/g	替换第一行到最后所有的Jan替换成January

## 块选择

```
1 在一般模式下:
2  v          字符选择, 会将光标经过的地方选中
3  V          行选择, 会将光标经过的地方选中
4  ctrl+v    可以用长方形的方式选择数据
5  shift+v   进入行模式
6  y          将选中的地方复制
7  d          将选中的地方删除
```

```
1 vim异常退出:
2 方法1:
3 vim -r filename 恢复
4 方法2:
5 rm -f .filename.swp 删除
6
7 vim快速跳转至多少行:
8
9 vim proxy.conf +12
10
11
```

## 自定义 vi 使用环境

```
1 临时设置
2     (1)  :set nu  设置行号
3     (2)  :set nonu 取消设置行号
4     (3)  :N  移动到第N行
5
6 永久设置
7     [root@exercise1 ~]# vi /etc/vimrc #设置后会影响到系统
    所有的用户
8
```

```

9      [root@exercise1 ~]# vi ~/.vimrc #在用户的家目录下，创建
      一个
10                                     #以点开头的文件.vimrc 此
      修改的应用范围仅生效于
11                                     #该用户，~符号表示匹配当前登
      录用户匹配家目录，
12                                     #以.点开头的表示隐藏文件
13
14      root用户下所有文件打开默认带行号：
15      [root@exercise1 ~]# vi /root/.vimrc #使用vi命令创建
      或修改家目录下的.vimrc并添加set nu
16      set nu
17      [root@exercise1 ~]# cat /root/.vimrc
18      set nu
19      [root@exercise1 ~]# vi /etc/passwd #以上设置完成
      后，使用vi打开文件，自动会显示行号

```

## vi/vim经典报错(这些报错有选项提示是用vim操作的)

- 1 情况一：
- 2 如果发现交换文件的占用进程状态是正在运行，说明有人正在修改该文件，一般我们会选择终止，或者退出。否则，
- 3 谁最后退出，就以谁的修改版本为准。这样会造成先提交修改的用户修改的内容丢失。

```

Jan 9 09:43:19 exercise1 journal: Runtime journal is using 6.1M (max allowed
48.8M, trying to leave 73.2M free of 482.0M available - current limit 48.8M)
.
Jan 9 09:43:19 exercise1 kernel: Initializing cgroup subsys cpuset
Jan 9 09:43:19 exercise1 kernel: Initializing cgroup subsys cpu
Jan 9 09:43:19 exercise1 kernel: Initializing cgroup subsys cpuacct
Jan 9 09:43:19 exercise1 kernel: Linux version 3.10.0-693.el7.x86_64 (builde
r@kbuilder.dev.centos.org) (gcc version 4.8.5 20150623 (Red Hat 4.8.5-16) (GC
C) ) #1 SMP Tue Aug 22 21:09:27 UTC 2017
Jan 9 09:43:19 exercise1 kernel: Command line: BOOT_IMAGE=/vmlinuz-3.10.0-69
3.el7.x86_64 root=UUID=c14ecb64-e445-457e-a838-04314faa45bf ro crashkernel=au
to rhgb quiet LANG=zh_CN.UTF-8
Jan 9 09:43:19 exercise1 kernel: Disabled fast string operations
Jan 9 09:43:19 exercise1 kernel: e820: BIOS-provided physical RAM map:
Jan 9 09:43:19 exercise1 kernel: BIOS-e820: [mem 0x0000000000000000-0x000000
0000009ebfff] usable
Jan 9 09:43:19 exercise1 kernel: BIOS-e820: [mem 0x0000000000009ec00-0x000000
0000009ffff] reserved
Jan 9 09:43:19 exercise1 kernel: BIOS-e820: [mem 0x000000000000dc000-0x000000
000000ffff] reserved
Jan 9 09:43:19 exercise1 kernel: BIOS-e820: [mem 0x0000000000100000-0x000000
003fedffff] usable
Jan 9 09:43:19 exercise1 kernel: BIOS-e820: [mem 0x00000000003fee0000-0x000000
003fefffff] ACPI data
Jan 9 09:43:19 exercise1 kernel: BIOS-e820: [mem 0x00000000003feff000-0x000000
003fefffff] ACPI NVS
Jan 9 09:43:19 exercise1 kernel: BIOS-e820: [mem 0x00000000003ff00000-0x000000
003fffffff] usable
@
1,1 顶端

```

```

E325: 注意
发现交换文件 "/opt/.messages.swp"
所有者: root 日期: Mon Jan 10 08:54:31 2022
文件名: /opt/messages
修改过: 否
用户名: root 主机名: exercise1
进程 ID: 1009 (仍在运行)
正在打开文件 "/opt/messages"
日期: Sun Jan 9 14:17:52 2022

(1) Another program may be editing the same file. If this is the case,
be careful not to end up with two different instances of the same
file when making changes. Quit, or continue with caution.
(2) An edit session for this file crashed.
如果是这样，请用 ":recover" 或 "vim -r /opt/messages"
恢复修改的内容 (请见 ":help recovery")。
如果你已经进行了恢复，请删除交换文件 "/opt/.messages.swp"
以避免再看到此消息。

交换文件 "/opt/.messages.swp" 已存在！
以只读方式打开([O])，直接编辑([E])，恢复([R])，退出([Q])，中止([A])：
交换文件 "/opt/.messages.swp" 已存在！
以只读方式打开([O])，直接编辑([E])，恢复([R])，退出([Q])，中止([A])：

```

- 1 情况二：
- 2 非正常状态退出vi，导致不正常的保存退出，导致Swap文件残留。
- 3 出现以下提示。此时占据的进程号没有运行，这个时候，我们一般会利用swp文件进行之前工作的恢复操作R，
- 4 恢复后，即可删除swp文件。也可以直接在报错界面输入:q退出，直接删掉.swp文件。

```
[root@exercise1 opt]# ls -a #此时没有.swp文件
```

```
. .. 1 1d 1e 1f 2 3 4 5 i.bak id j log log.bak messages
```

```
[root@exercise1 opt]# vi /opt/messages #vi进去以后使用ctrl+z停止
```

```
[1]+ 已停止          vi /opt/messages
```

```
[root@exercise1 opt]# vi /opt/messages #此时进去会报错
```

```
[root@exercise1 opt]# ls -a #查看发现多了一个.swp文件
```

```
. .. 1 1d 1e 1f 2 3 4 5 i.bak id j log log.bak messages
```

```
.messages.swp
```

```
[root@exercise1 opt]# rm -rf .messages.swp #删掉.swp文件，删完再进去就正常了
```

```
[root@exercise1 opt]#
```

```

E325: 注意
发现交换文件 "/opt/.messages.swp"
    所有者: root    日期: Mon Jan 10 08:55:44 2022
    文件名: /opt/messages
    修改过: 否
    用户名: root    主机名: exercisel
    进程 ID: 1028 (仍在运行)
正在打开文件 "/opt/messages"
    日期: Sun Jan  9 14:17:52 2022

(1) Another program may be editing the same file.  If this is the case,
    be careful not to end up with two different instances of the same
    file when making changes.  Quit, or continue with caution.
(2) An edit session for this file crashed.
    如果是这样, 请用 ":recover" 或 "vim -r /opt/messages"
    恢复修改的内容 (请见 ":help recovery")。
    如果你已经进行了恢复, 请删除交换文件 "/opt/.messages.swp"
    以避免再看到此消息。

交换文件 "/opt/.messages.swp" 已存在!
以只读方式打开([O]), 直接编辑([E]), 恢复([R]), 退出([Q]), 中止([A]):

[root@exercisel ~]# ll -a /opt/
总用量 976
drwxr-xr-x.  6 root root    176 1月 10 09:12 .
dr-xr-xr-x. 17 root root    224 1月  9 09:31 ..
-rw-r--r--.  1 root root 653626 1月  9 14:41 1
-rw-r--r--.  1 root root     0 1月  9 13:41 1d
-rw-r--r--.  1 root root     0 1月  9 13:41 1e
-rw-r--r--.  1 root root     0 1月  9 13:41 1f
-rw-r--r--.  1 root root     0 1月  9 13:38 2
-rw-r--r--.  1 root root     0 1月  9 13:38 3
-rw-r--r--.  1 root root     0 1月  9 13:38 4
-rw-r--r--.  1 root root     0 1月  9 13:38 5
drwxr-xr-x.  3 root root     29 1月  9 14:23 i.bak
-rw-r--r--.  1 root root     45 1月  9 14:58 id
drwxr-xr-x.  2 root root     6 1月  9 13:50 j
drwxr-xr-x.  7 root root   4096 1月  9 14:19 log
drwxr-xr-x.  7 root root   4096 1月  9 14:20 log.bak
-rw-----.  1 root root 324785 1月  9 14:17 messages
-rw-----.  1 root root 4096 1月 10 09:12 .messages.swp

```

要删掉这个  
才能解决报错,  
否则下一  
次进去还  
是会报同  
样的错误

## vi练习示例1

- 1.将/etc/passwd复制到/opt目录下, 并重命名为test.txt
- 2.用vim打test.txt并显示行号
- 3.分别向下、向右、向左、向右移5个字符, 分别向下、向上翻两页
- 4.把光标移动到第10行, 让光标移动到行末, 再移动到行首, 移动到test.txt文件的最后一行, 移动到文件的首行
- 5.搜索文件中出现的root并数一下一共出现多少个, 不区分大小写搜索

- 6.把从第一行到第三行出现的root替换成-od-, 然后还原上一步操作
- 8.把整个文件中所有的root替换成-od-
- 9.把光标移动到20行, 删除本行, 还原上一步操作
- 10.删除第19行, 还原上一步操作
- 11.删除从5行到10行的所有内容, 还原上一步操作
- 12.复制2行并粘贴到11行下面, 还原上一步操作
- 13.复制从11行到15行的内容并粘贴到8行上面, 还原上一步操作
- 14.把13行到18行的内容移动文件的尾部, 还原上一步操作
- 15.光标移动到首行, 把sbin/nologin改成bin/bash
- 16.在第一行下面插入新的一行, 并输入 " # Helo! "
- 17.保存文档并退出

## vi练习示例2

```
1 [root@exercise1 opt]# cat proxy.conf
2 server {
3     listen      8080;
4     server_name  LOCALhost;
5     location / {
6         proxy_pass http://127.0.0.1:8080;
7         proxy_set_header Host $http_host;
8         proxy_set_header X-Forward-for;
9         proxy_intercept_errors on;
10        proxy_next_upstream error timeout;
11        proxy_next_upstream_timeout 3s;
12        proxy_next_upstream_tries 2;
13        error_page 500 502 403 404
14        /proxy_error.html;
15    }
16    location =/proxy_error.html {
17        root /code/proxy;
```



```
17         }  
18     }
```

- 1.使用vim打开proxy.conf文件
- 2.修改listen为LISTEN大写，并将8080修改为80
- 3.修server\_name为ServerName。
- 4.修LOCALhost为localhost
- 5.在server name行下插入一行root/code;
- 6.复制5-14行的内容，然后将其粘贴到14行下面
- 7.删除与proxy\_set\_header相关的两行全部删除
- 8.如上操作完成后，在13-20行前面加上#号
- 9.删除21-23的行，然后保存当前文件