> author：编程界的小学生
>
> date：2021/02/16
>
> PS：高频能提升工作效率的命令总结，并非命令大全哈，命令大全的话自己看官网咯。都是自己亲身总结。

一、vim

1、复制

- yy：复制一整行
- 3yy：复制三整行
- 5yy：复制五整行
- ...以此类推
- y+shift+4（也就是y$）：复制光标位置到这行末尾

2、剪切

- dd：剪切一整行
- 3dd：剪切三整行
- d$（d+shift+4）：剪切光标位置到行尾

3、粘贴

- p：粘贴

4、撤销、前进

- u：撤销，也就是windows上的ctrl+z
- ctrl+r：前进，相当于windows的ctrl+y

5、显示行号

- :set nu：显示行号

6、移动光标

- g：光标移动到文本第一行的最前面
- shift+g：光标移动到文本最后一行的最前面
- 3+shift+g：光标移动到第3行

7、替换

- :s/old/new：将光标定位的这行的**第一个**old值换成new
- :s/old/new/g：将光标定位的这行的所有old值换成new
- :%s/old/new：将文本**每一行的第一个old**值换成new
- :%s/old/new/g：将全文本中的old值换成new
- :3,5s/old/new/g：将第三行到第五行之间（包含第三行和第五行）的old值换成new

二、正则表达式符号

- .

> 匹配除换行符外的任意单个字符

- *

> 匹配任意一个跟在它前面的字符

- [ ]

> 匹配方括号中的字符类中的任意一个

- ^

> 匹配开头

- $

> 匹配结尾

- \

> 转义后面的特殊字符

- +

> 匹配前面的正则表达式出现一次

- ?

> 匹配前面的正则表达式出现零次或一次

- |

> 匹配他前面或者后面的正则表达式

三、find

1、按名字查找

```
find /home/chentongwei -name 'minio*'
```

> find -name语法
>
> 在/home/chentongwei目录下查找以minio开头的文件

2、按正则表达式查找

```
find /home/chentongwei -regex .*log$
```

> find -regex语法
>
> 在/home/chentongwei目录下查找以log结尾的文件

3、按类型查找

```
find /home/chentongwei -type f -regex '.*\.log$'
```

> find -type语法
>
> 在/home/chentongwei目录下查找是文件类型的（将非文件类型排除，比如目录类型、块类型等，只保留文件类型）且以.log结尾的文件。

4、按照时间查找

> 语法：find -a(m|c)time days、find -a(m|c)min minutes
>
> 使用stat 文件名 可以查看amc三个时间。

- 查找20分钟前的文件

```
find /home/chentongwei -type f -mmin +20
```

- 查找20天前的文件

```
find /home/chentongwei -type f -mtime +20
```

- 补充说明

> n都支持正（+）负（-）数
>
> -amin n 查找系统中最后n分钟访问的文件。比如cat文件也会更新这个时间。 -atime n 查找系统中最后n天访问的文件。比如cat文件也会更新这个时间。 -cmin n 查找系统中最后n分钟被改变文件状态的文件。比如chown/vim啥也没改就保存文件也会更新这个时间。 -ctime n 查找系统中最后n天改变文件状态的文件。比如chown/vim啥也没改就保存文件也会更新这个时间。 -mmin n

> 查找系统中最后n分钟被改变文件数据的文件。只有修改数据内容才会更新这个时间。 -mtime n
> 查找系统中最后n天被改变文件数据的文件。只有修改数据内容才会更新这个时间。

5、查找到文件后进行操作

> 语法：find -exec 命令 {} \;

- 找到以xxx.log结尾的文件且删除

```
find /home/chentongwei -regex ".*xxx.log$" -exec rm {} \;
```

- 显示20分钟前的文件

```
find /home/chentongwei -type f -mmin +20 -exec ls -l {} \;
```

- 删除20天前的文件

```
find /home/chentongwei -type f -mtime +20 -exec rm {} \;
```

四、sed

> 替换语法：`sed 's/old/new/标志位' filename1 filename2 filename3 ...`
>
> **s前面支持正则表达式。**
>
> **old和new都支持正则。**
>
> **标志位可有可无。**

1、替换单个匹配项（不修改原文件）

将test.txt文件内容的第一个a变成aa且输出，此操作不会修改test.txt文件的内容，只是替换后输出到控制台。

```
sed 's/a/aa/' test.txt
```

2、替换多个匹配项（不修改原文件）

将test.txt文件内容的第一个a变成aa，将文本内容的第一个b变成bb且输出，此操作不会修改test.txt文件的内容，只是替换后输出到控制台

- 方式一：-e参数

```
sed -e 's/a/aa/' -e 's/b/bb/' test.txt
```

- 方式二：分号隔开

```
sed 's/a/aa/;s/b/bb/' test.txt
```

3、替换多个匹配项且写入原文件

将test.txt文件内容的第一个a变成aa，将文本内容的第一个b变成bb且将修改写入原文件test.txt

```
# -i参数
sed -i 's/a/aa/;s/b/bb/' test.txt
```

4、替换多个匹配项且写入原文件

将test.txt文件内容的全部a变成aa，将文本内容的全部b变成bb且将修改写入原文件test.txt

```
# g参数
sed -i 's/a/aa/;s/b/bb/g' test.txt
```

5、替换第几个出现的字母

找到文本中第二个a，且将第二个出现的a替换成b并写入原文件test.txt

```
# 标志位后面加数字即可
sed -i 's/a/aa/2' test.txt
```

6、按照行号进行替换

替换文本第6行中的a为x并写入原文件test.txt

```
# s前面加数字即可
sed -i '6s/a/x/' test.txt
```

替换文本第2-5行（包含2和5）中的a为x并写入原文件test.txt

```
# s前面加数字,数字即可
sed -i '2,5s/a/x/' test.txt
```

7、删除某一行

找到文本中包含bc字符串的那行，且删除（只会删除第一个匹配到bc的行），且不会修改原文件

```
# /d参数
sed '/bc/d' test.txt
```

8、向上插入一行

找到文本中包含bc字符串的那些行，且在每一个匹配行上面插入hello world，且不会修改原文件

```
# i参数
sed '/bc/i hello world' test.txt
```

9、向下插入一行

找到文本中包含bc字符串的那些行，且在每一个匹配行下面插入hello world，且不会修改原文件

```
# a参数
sed '/bc/a hello world' test.txt
```

五、awk

`print`是打印命令，`$0`代表当前整行，依次用`$1`、`$2`、`$3`代表第一个字段、第二个字段、第三个字段等等。$n配合-F可以切割出想要的字段。

1、输出以x开头的行

```
awk '/^x/{print $0}' test.txt
```

2、输出以x开头的行且带序号，也就是每行前面带1,2,3...

```
awk '/^x/{print x++, $0}' test.txt
```

3、输出以r开头且按照:分割开的第1个字段

如下命令就会打印出全部r开头的用户名，因为passwd第一个字段是用户名。

```
awk -F ':' '/^r/{print $1}' /etc/passwd
```