**vim编辑器（编辑器之神）**

1. 基本介绍

1.vi编辑器是linux中最基本的文本编辑器（适用于文本编辑）

2.vim 比较高级，可以视为vi的升级版本（使用与编码）

vim的重点是：光标移动、模式切换、删除、查找替换、复制、粘贴、撤销命令的使用

1. vim三种模式（重点）

命令模式、编辑模式（输入模式）、末行模式（尾行模式）

1.命令模式：打开文件后默认进入的模式

在该模式下是不能对文件直接编辑，可以输入快捷键进行一些操作（删除行、复制行、移动光标、粘贴等）

2.编辑模式

在该模式下可以对文件内容进行编辑

3.末行模式

可以在末行输入命令对文件进行操作（搜索、替换、保存、退出、撤销、高亮）

vim打开文件的方式，总共四种

（1）# vim 文件路径 作用：打开指定的文件

(2) # vim +数字 文件路径 作用：打开指定的文件，并且将光标移动到指定行

1. #vim +/关键字 文件路径 作用：打开指定的文件，并且高亮显示关键字
2. # vim 文件路径1 文件路径2 文件路径3 作用：同时打开多个文件

！！！注意：接下来对当前目录下的paswd文件进行相关的操作

1. vim三种模式下的相关操作

1.命令模式：该模式是打开文件后进入的（第一个看到的模式）

（1）光标的移动

<1>光标移动到行首 按键: shift+6 (T字母上面的6）/^

<2>光标移动到行尾 按键:shift+4(R字母左上角）/$

<3>光标移动到首行 按键:gg

<4>光标移动到末行 按键:G

<5>翻屏

向上翻屏 按键:ctrl+b/PgUp

向下翻屏 按键:ctrl+f /PgDn

(2)复制操作

<1>复制光标所在行 按键：yy 粘贴：在想要粘贴的地方，按下p键

<2>以光标所在行为准，向下复制指定行数 按键：数字 yy 粘贴：在需要粘贴的地方，按下p键

<3>可视化复制 按键：ctrl+v -->↑/↓/←/→选择复制区域-->yy-->p

(3)剪切/删除

<1>剪切和删除光标所在行 按键:dd(删除之后，下一行上移）

<2>剪切和删除光标所在行为准（包含当前行），向下剪切/删除指定行 按键： 数字 dd (删除之后，上移）

<3>剪切和删除光标所在行,但是删除之后不上移 按键：D

1. 撤销/恢复

<1>撤销：输入 :u(属于末行模式）/ 直接按u (undo)

<2>恢复: ctrl+r 恢复（取消）之前的撤销

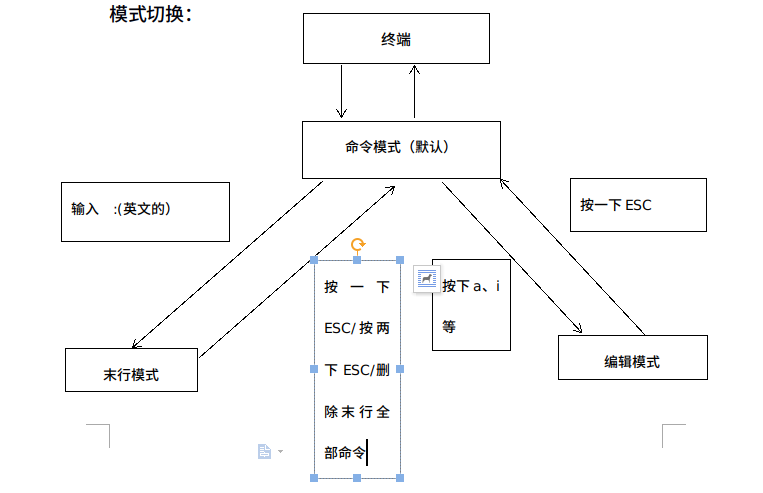
1. 扩展1：光标的快速移动

<1>快速将光标移动到指定行 按键：数组 G

<2>以当前光标为准，向上/向下移动n行 按键：数字 ↑/↓

<3>以当前光标为准,向左/向下移动n行 按键：数字 ←/→

<4>末行模式下的快速移动方式，移动到指定的行 按键：:数字



2.末行模式

进入方式：由命令模式进入，按下“：”即可进入

退出方式： (1）按一下ESC（延迟一下退出）

（2）按两下ESC（立即退出）

(3）删除末行全部输入字符

（1）保存操作 输入 “:w” (保存文件）

输入 ”:w 路径“ （另存文件）

（2）退出操作 输入 ”:q”

(3)保存并退出 输入 “:wq”

(4)强制（！） 输入 ”:q!” (刚才的修改不保存）

(5)调用外部命令 输入 “:!外部命令”

当外部命令结束之后，按下任意键回到vim

(6)搜索/查找 输入”/关键字”

在搜搜结果中切换下/上一个结果 N/n

（7）替换

用法1： “:s/搜索的关键字/新的内容” 替换光标所在 行的第一处符合条件的内容

用法2： “:s/搜索的关键字/新的内容/g” 替换光标所在行的全部符合条件的内容

用法3：“:%s/搜索的关键字/新的内容” 替换整个文档中每行第一个符合条件的内容

用法4： “:%s/搜索的关键字/新的内容/g” 替换整个文档中的符合条件的内容

%表示整个文档，g表示全局

1. 显示行号（临时表示）

输入”:set nu”

取消显示行号：”:set nonu”

(9)使用vim同时打开多个文件，在末行模式下进行切换文件

查看当前已经打开的文件名称： files(files中%a表示当前打开的文件，#表示上一个打开的文件）

切换文件方式

方式一：如果需要指定切换文件的名称 “: open 文件名“

方式二：可以通过其他命令切换上一个文件/下一个文件

输入 ”:bn”(切换到下一个文件）

输入 “:bp”(切换到上一个文件)

1. 编辑模式

进入：

i:在光标所在字符前开始插入

a:在光标所在字符后开始插入

退出：

按键:esc

1. 实用功能

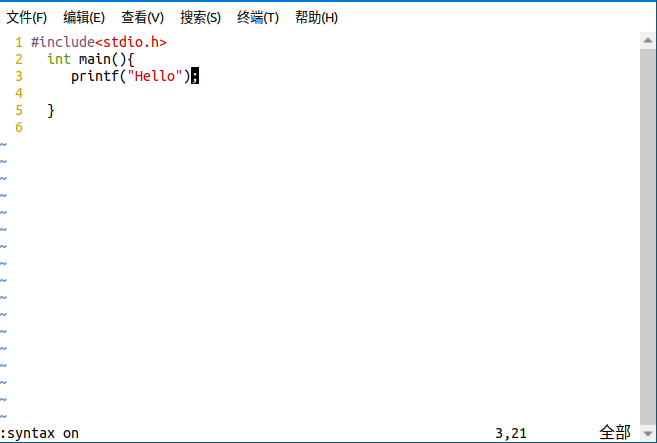
1.代码着色

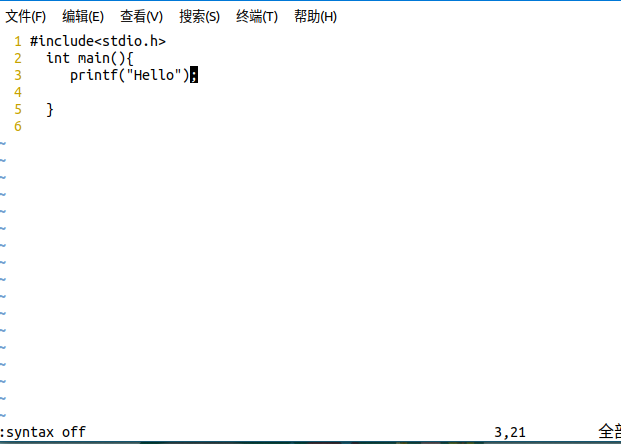
案例：首先简单的创建一个C语言程序

语法格式：

显示：”:syntax on”

关闭显示：“:syntax off”





1. vim中计算器的使用

在编辑文件的时候，突然需要实用计算器去计算一些公式，则此时需要用计算器，但是需要退出，vim自身集成了一个简易的计算器

a:进入编辑过程

b:按下按键 ctrl+R,然后输入”=“，此时光标会变到最后一行

c:输入需要计算的内容，按下回车

1. 扩展

1.vim的配置（重点）

vim是一款编辑器，编辑器也是有配置文件的

vim配置有三种情况

a:在文件打开的时候，在末行模式下输入的配置（临时的）

b:个人配置文件 .vimrc

c:全局配置文件 （vim自带的） /etc/vimrc

**问题**：如果某个设置项，在个人配置文件和全局配置文件产生冲突的时候，应该以谁为准？

验证方法：在两个配置文件里面，针对同一个配置项设置不同的值

（1）先在全局的配置文件中设置不显示行号，在个人配置里面设置显示行号，观察结果？

最后显示行号，说明以个人配置文件为准

(2）先在全局配置文件中设置显示行号，在个人配置文件中不显示行号，观察结果？

最后不显示行号，说明以个人配置文件为准

**结论**： 如果针对同一个配置项，个人配置文件中存在，则以个人配置为准。如果个人配置文件中不存在该项，则以全局配置为准。

1. 异常退出

什么是异常退出？

在编辑文件之后，并没有正常的去保存退出，而是遇到突然关闭终端或者断电的情况，则会显示异常退出

解决办法：将交换文件（在编程过程中产生的临时文件）删除即可

# rm -f .pass.swap

1. 别名机制

作用：相当于创建一些属于自己的自定义命令

例如：在windows下有cls命令，linux下可能没有这个命令，而不习惯清屏，现在可以通过别名机制来解决这个问题，可以自己创造出cls命令

别名机制依靠一个别名映射文件：～/.bashrc

alias cls=clear

**!!!注意**：如果想要命令立即有效，必须要重新登录到当前用户

1. 退出方式：

回顾：之前vim退出编辑文件，可以使用”:q”或“:wq”

除了上面的语法之外，vim 还支持另外一个保存退出方式”:x”

**说明：**

(1）“:x”在文件没有修改的情况下，表示直接退出。在文件修改的情况下，表示保存并退出

（2） 如果文件没有被修改，都是使用wq进行退出的话，则文件的修改时间会更新。但是如果文件没有被修改，使用x进行退出的话，则文件修改时间不会被更新。主要是会混淆用户对文件修改时间的认定

**因此**：以后建议使用”:x”进行保存退出

**注意**：但是不要使用X，因为大写X，是对文件进行加密操作