1. 查看某文件权限为rwxr-xr-- ，则其所属组权限为只读。
2. 对一个目录有777权限，表示可以修改目录下文件的内容。
3. .tar.gz和.tar.bz2格式的压缩包都可以使用tar -xvf解压缩
4. mv剪切目录时需要加-r选项。
5. /sbin目录下的命令只有root可以使用。
6. find根据文件大小查找使用选项为-size ，默认单位为KB
7. grep查找时反向查找可以使用-i
8. 改变文件的所有者可以使用命令chgrp

vi编辑器

Linux图形化界面下的文本编辑器

gedit 、 libre office 、 evince PDF阅读器

①gedit是一个GNOME桌面环境下兼容UTF-8的文本编辑器。它使用GTK+编写而成，因此它十分的简单易用，有良好的语法高亮，对中文支持很好，支持包括gb2312、gbk在内的多种字符编码。gedit是一个自由软件。

②LibreOffice是OpenOffice.org 办公套件衍生版， 同样免费开源，以Mozilla Public License V2.0许可证分发源代码，但相比OpenOffice增加了很多特色功能。LibreOffice拥有强大的数据导入和导出功能，能直接导入PDF文档、微软Works、LotusWord，支持主要的OpenXML格式。软件本身并不局限于Debian和Ubuntu平台，OpenXML格式Windows、Mac、PRM packageLinux等多个系统平台。

③Evince是一套文件阅读工具,可以用来阅读许多种格式的资料,像是Postscript、PDF 或是其他格式的文件

1） 文本编辑器的作用：

创建或者修改文本文件

维护Linux系统中的各种配置文件

2） 常用版本：

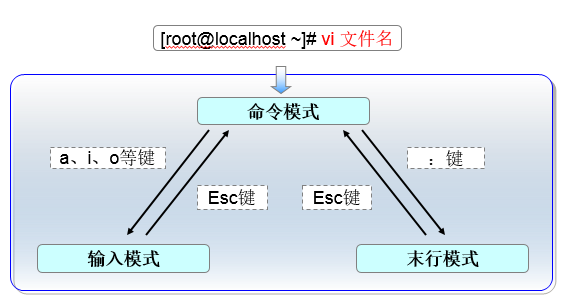
vi ：类Unix系统中默认的文本编辑器

vim ：vi编辑器的增强版本，习惯上也称为vi

3） 工作模式：

命令模式、输入模式、末行模式

三种模式之间的互相切换方式



输入模式下命令：

|  |  |
| --- | --- |
| 命令 | 作用 |
| a | 在光标所在字符后插入 |
| A | 在光标所在行尾插入 |
| i | 在光标所在字符前插入 |
| I | 在光标所在行行首插入 |
| o | 在光标下插入新行 |
| O | 在光标上插入新行 |

命令模式下的操作：

|  |  |  |
| --- | --- | --- |
| 操作类型 | 操作键 | 功能 |
| 光标方向移动 | 🡩、🡫、🡨、🡪 | 上、下、左、右 |
| 字母移动 | H j k l | 左、下、上、右 |
| 翻页 | Page Down或Ctrl+F(大) | 向下翻动一整页内容 |
| Page Up或Ctrl+B（大） | 向上翻动一整页内容 |
| 行内快速跳转 | Home键或“^”、数字“00” | 跳转至行首 |
| End键或“$”键 | 跳转到行尾 |
| 行间快速跳转 | 1G或者gg | 跳转到文件的首行 |
| G | 跳转到文件的末尾行 |
| #G | 跳转到文件中的第#行 |

|  |  |  |
| --- | --- | --- |
| 操作类型 | 操作键 | 功能 |
| 删除 | x或Del | 删除光标处的单个字符 |
| dd | 删除当前光标所在行 |
| #dd | 删除从光标处开始的#行内容 |
| d^ (d shift+6） | 删除当前光标之前到行首的所有字符（不包含光标） |
| d$ | 删除当前光标处到行尾的所有字符（包含光标） |
| 复制 | yy | 复制当前行整行的内容到剪贴板 |
| #yy | 复制从光标处开始的#行内@容 |
| 粘贴 | p | 将缓冲区中的内容粘贴到光标位置处之后（下一行） |
| P（大） | 粘贴到光标位置处之前（上一行） |
| 替换 | r | 替换光标所在字符 |
| R | 从光标出开始替换，按ESC键结束 |

|  |  |
| --- | --- |
| 操作键 | 功能 |
| /word | 从上而下在文件中查找字符串“word” |
| ?word | 从下而上在文件中查找字符串“word” |
| n | 定位下一个匹配的被查找字符串 |
| N | 定位上一个匹配的被查找字符串 |

那么有人会问，两个都是全文查找，怎么区分是从下往上，还是从上往下。用n定位来确定

|  |  |
| --- | --- |
| 操作键 | 功能 |
| u | 按一次取消最近的一次操作  多次重复按u键，恢复已进行的多步操作 |
| U | 用于取消对当前行所做的所有编辑 |
| Shift+zz（小）或者ZZ | 保存当前的文件内容并退出vi编辑器 |

末行模式下操作：

|  |  |  |
| --- | --- | --- |
| 功能 | 命令 | 备注 |
| 保存文件 | :w | 保存所做操作 |
| :w /root/newfile | 另存为其它文件 |
| 退出vi | :q | 未修改退出 |
| :q! | 放弃对文件内容的修改，并退出vi |
| :wq | 保存文件退出 |
| 删除多行文件 | :n,nd |  |

|  |  |  |  |
| --- | --- | --- | --- |
| 命令 | | 功能 | |
| :e ~/install.log | | 打开新的文件进行编辑 | |
| :r /etc/filesystems | | 在当前文件中读入其他文件内容 | |
| :!命令 | | 直接执行命令打印到屏幕上 | |
| :r !命令 | | 导入命令结果到光标所在处 | |
| 行号显示 | :set nu | | 在编辑器中显示行号 |
| 取消行号 | :set nonu | | 取消编辑器中的行号显示 |
| 颜色开关 | :syntax off | | 关闭颜色显示 |
| :syntax on | | 开启颜色显示 |

|  |  |
| --- | --- |
| 命令 | 功能 |
| :s /old/new/ | 将当前行中查找到的第一个字符“old” 串替换为“new” |
| :s /old/new/g | 将当前行中查找到的所有字符串“old” 替换为“new” |
| :#,#s /old/new/g | 在行号“#,#”范围内替换所有的字符串“old”为“new” |
| :% s /old/new/g | 在整个文件范围内替换所有的字符串“old”为“new” |
| :s /old/new/cg | 在替换命令末尾加入c命令，将对每个替换动作提示用户进行确认 |

vim 使用技巧--设置快捷键

:map 快捷键 快捷键执行的命令

例1：

:map ^P I#<esc>

当输入ctrl+p快捷键时，执行在首行添加#注释

^P 不能手动输入，需要执行ctrl+V+P 来定义。或者ctrl+V，然后ctrl+P

例2：

:map ^B ^x

当输入ctrl+B快捷键时，光标会移动到行首，然后删除一个字符

例3：

:ab mymail liuchuan@itxdl.cn

注：如果想让这些快捷键永久生效，可以写入vim的配置文件。