* Gedit命令 输入以后，可以键入一连串命令，再gedit窗口被关闭以后才执行所有的命令
* 统计

Wc -w/c/l filelist ，按w/c/l对文件进行统计

W：对word计数，

C:一个字符

L：一行

Ex: ls -l | wc -l(统计该目录下的entry个数) ls -l可以显示entry ls命令都显示的是entry 只是ls -a能显示隐藏文件

* Df 查看磁盘的存储容量

Du 查看磁盘下某目录所占据的空间

* 进程

Ps 直接使用可以查看所有进程的pid数值，pid不同，进程就不是同一个以及时间等信息，ps本身也是一个进程（动态的，同一个名字的程序，如不同时间调用ps，都可以显示一个名为ps的进程但是其实不是同一个，pid不同）

Ps只是查看当前终端下的进程 Ps aux显示系统内所有的进程信息

Kill 杀死pid对应的进程，传入pid参数值,用ps aux查看，用grep筛选需要查看的

&

Gedit &进入后台执行gedit 再终端中可以继续执行命令，而不是等gedit界面执行完关闭再进行后续命令

多任务系统就需要分前后台，

单个终端中 contr z挂到后台， fg再回来，如vim可以挂到后台

Nohup（不要挂断，no hung up）+命令，可以不挂断执行，除了关机其他都不会中断

Tmux最好用

1. Tmux ls查看有几个 htop相当于任务管理器
2. Tmux

* 文件系统装在卸载

Mount unmount

Ex: mount （-t iso9660） 设备文件 挂载点(-tk以不要，设备文件和挂在点缺一不可,挂在点一般是空目录。挂载的文件系统会覆盖掉挂载点所有的文件，（不是删除损坏，只是不显示原来的文件内容），当卸载所挂的文件系统之后，原来的文件又会显示出来)

Unmount后面可以加设备文件或者挂载点（都可以进行卸载）

* Alias给复杂命令行取别名

Alias 别名=’命令’

Ex: alias mntcd= ‘ount -t iso9660 设备名 挂载点’

* Vi编辑器

1. insert i，再当前光标前插入模式
2. Esc退出插入模式 ，用 hjkl来进行移动 左移h 右移l(与键盘上位置同) 往下j 网上k 0回到行首 $到达行尾，数字+hjkl表示执行几次
3. 大写I再行首插入，小写a在光标所在后插入，大写A行尾插入，插入新行小写o，在该行前插入空行大写O
4. 跳到某一行，非insert模式下 用 ：行号
5. 一般认为三种模式 ： insert模式 esc模式 esc ：下的模式
6. 往后跳：w 一个单词一个单词跳 b返回

单词e 跳到每个单词尾部

删除x字符 dw删除单词 dd删除一行且下面的都上移一行 ，有剪切功能删掉后保存在剪切板 后面用p ddp就是交换两行位置

复制yy一行

粘贴p粘贴在后面一行

P粘贴在前面一行

1. Vim file 有进入 没创建
2. Vim直接进入 保存的时候用 w filename可以创建
3. Wq保存 q!不保存
4. 大于号缩进 小于号反方向缩进
5. ：命令下 ce ri le 可以让该行命令位置改变

：set number（set nu）会有行号显示

：set nonumber（去掉）

：set cursorline（给一条线显示当前在哪一行）

：set colorscheme 加参数（改颜色） 该成evening

9测验讲解 装vim插件 完成实验楼作业