## 1 CMD

Sed

Stream Editor文本流编辑，sed是一个“非交互式的”面向字符流的编辑器。能同时处理多个文件多行的内容，可以不对原文件改动，把整个文件输入到屏幕,可以把只匹配到模式的内容输入到屏幕上。还可以对原文件改动，但是不会再屏幕上返回结果。

sed [option] ‘sed command ’ filename

-n 只打印匹配的结果

-e 直接在命令模式上进行sed动作编辑

-f 将sed的动作写进一个文件内 –f filename

-r 支持扩展表达式

-i 直接修改文件内容

Sed –n ‘2p’ data 打印文件的第二行

Sed –n ‘/second/p’ data 打印文件中匹配second的行

Tail –n(行数、+k(从头显示)) –q(不显示文件头) –f(新内容)

[root@linux ~]# tail /etc/passwd #默认，显示最后10 行。

[root@linux ~]# tail -n 2 /etc/passwd #显示最后2行

[root@linux ~]# tail -q -n k file1 file2 file3 #显示多文件最后k行，并且不显示文件名的文件头

[root@linux ~]# tail -n +k /etc/passwd #从开头第k行处开始输出。

[root@linux ~]# tail -f /var/log/messages #参数-f使tail不停地去读最新的内容，因此有实时监视的效果，用Ctrl＋c来终止

# **tail -n+10 file.txt | head -1** #显示file.txt的第10行

Vim:

h,j,k,l 上，下，左，右  
ctrl-u 上翻半页  
ctrl-d 下翻半页  
w 跳到下一个字首，按标点或单词分割  
b 跳到上一个字  
$ 跳至行尾  
gg 跳至文首  
G 调至文尾  
5gg/5G 调至第5行

yy 复制本行

p 粘贴本行

dd删除本行

terminal：

ctrl a 首位

ctrl e 末尾

ctrl k 删除此处至末尾所有的内容

ctrl u 删除此处至开始所有的内容

ctrl l clear

ctrl r 历史命令

## 2 C++

### 2.1 thread

C++11 的新特性 thread基本API以及使用

Join()------等待线程执行完毕

Detach()--------与主线程分离

Mutex-----------互斥锁，线程同步，共享数据

**用类内部函数在类外部创建线程：**



用类内部函数在类内部创建线程



### 2.2 类静态数据和函数

C++类的数据成员可以分为静态数据成员和非讲台数据成员。静态成员使用static 修饰，可以通过类名::静态数据成员访问，静态数据成员在全局存储区，类没有实例化也存在。类的静态函数不能访问类的非静态数据和类的非静态函数；而类的非静态函数可以访问类的静态数据成员以及函数。

面向过程：

静态全局变量是在全局变量前加上关键字static。特点：

1. 该变量在全局数据区分配内存
2. 如果不显示的初始化，那么将被初始化为0
3. 本源文件可见，其他文件不可见。

静态全局变量和静态局部变量全部都存储在全局数据区。包括静态局部变量。

全局变量可以实现文件共享。

静态函数和普通函数不同，它只能在声明它的文件中可见，不能被其他文件使用。在其他文件可以定义名字相同的函数不会发生冲突。

Extern:

全局变量在外部使用声明时，extern关键字是必须的，如果变量没有extern修饰且没有显式的初始化，同样成为变量的定义，编译器在此标记存储空间在执行时加载内并初始化为0, 因此此时必须加extern。而局部变量的声明不能有extern的修饰，且局部变量在运行时才在堆栈部分分配内存。