

粒子物理与核物理实验中的 数据分析

杨振伟

清华大学

第一讲：Linux环境下编程(1)

本讲摘要

- 什么是Linux
- 为什么使用Linux
- 登陆和使用Linux服务器
- Linux简介(内核、shell以及目录结构等)
- Linux终端的常用命令(!!!)
- Linux终端的常用编辑器(vi, emacs, pico, etc)
- Linux 环境变量与Shell 脚本(!!!)
- Linux下编译运行c/c++, fortran程序

什么是Linux(1)

什么是Linux ?

Linux是众多操作系统的一种



主要特点

源代码开放，自由软件/代码

众多的不同发行版

强大的shell指令以及shell编程功能：

`cd, ls, grep, find, sed...`

大量的科学计算、数据分析处理的程序包

(CERN、FermiLab、KEK以及其它众多机构提供支持)

什么是Linux(2)

粒子与核物理界普遍使用的是

Scientific Linux CERN(SLC)或Scientific Linux(SL),

(1)Redhat发行Redhat Enterprise Linux(**RHEL**)发行版

(2)CERN与FermiLab基于RHEL内核，联合发行**SL**

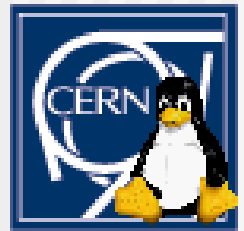
(3)CERN在SL的基础上发行**SLC**

主要定制了CERNLIB包。

本课程服务器采用的是Scientific Linux CERN3
(SLC3)版本

最新版为SLC5.6 发行版 和 SLC6 beta版

<http://linux.web.cern.ch/linux/>



为什么使用Linux(1)

为什么使用Linux？

源代码开放，自由软件/代码

Linux的强势不在桌面、游戏等方面

Linux的强势主要在于科学研究方面

尤其是需要大量计算或编程进行数据分析的科研工作

强大的shell命令和脚本，多任务长时间的运算

科研中可能会遇到的小问题(程序相关):

- 1.在某文件夹下有很多文件和子文件夹，需要将所有的.cpp文件中的yang_int改成yang_float
- 2.同样该文件夹，有很多文件中可能包含某个函数，比如Fit_Landau()，需要找到函数的定义及使用
- 3.查看某个文本文件a.txt一共有多少行
- 4.某程序，需要变更其中的参数进行多次运行。

为什么使用Linux(2)

问题1可以用下面一行命令解决：

```
find . -name "*.cpp" -exec sed -i 's/yang_int/yang_float/g' {} \;
```

即用find命令在当前目录寻找(递归)所有的cpp文件，找到后执行sed命令，其中-name和-exec是find命令的参数，分别表示按文件名寻找和执行指令。sed指令在文件中寻找(s)字符串“yang_int”，替换为“yang_float”。find后面的“.”表示在当前目录寻找，也可以改成其它想寻找的目录，比如/home/yangzw/mywork

问题2可以用grep命令解决：

```
grep -srn "Fit_Landau" /home/yangzw/mywork
```

即用grep命令递归查找/home/yangzw/mywork目录里面的所有文件，打印出所有包含“Fit_Landau”字符串的文件名称，以及该字符串出现的行号和该行的内容。其中-srn是grep的参数，s表示忽略文件不存在或无法读取等错误信息，r表示在文件夹中递归查找，n表示打印出字符串出现的行号。这些参数可以组合使用。

为什么使用Linux(3)

问题3可以用下面一行命令解决：

```
wc -l a.txt
```

即用wc命令，计算a.txt文件有多少行，其中-l参数表示计算行号。如果改成-w，则表示计算有多少word。

问题4可以用shell脚本快速解决，详见shell脚本编程。

这仅仅是几个简单的例子，Linux提供的这种指令不计其数。一般通过baidu或者google都可以查到如何实现你需要的功能。如何获取更多关于Linux的使用信息？不得不说的几句：搜索引擎是个好东西！！！！

思考题：如果在linux下获得系统当前时间并截取时间中的月份？可以baidu试一下。

登录和使用Linux

前提：拥有帐号(包括密码)，知道服务器的域名或ip
课程服务器的域名/ip是:training.hep.tsinghua.edu.cn
166.111.32.11

某用户的用户名为: yangzw

登录远程服务器一般用ssh方式：

1)从Linux终端登录

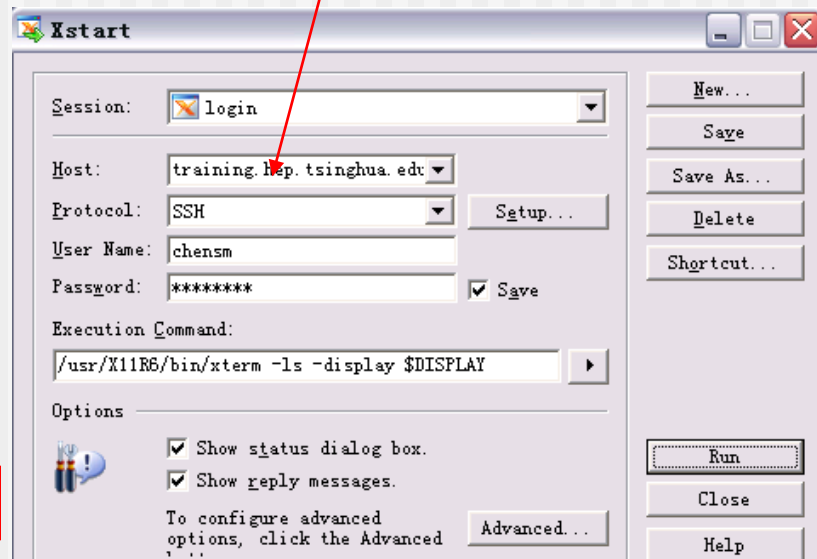
ssh yangzw@training.hep.tsinghua.edu.cn

2)Windows下用ssh客户端软件登录

putty, SecureCRT, Xmanager,...

登录之后就进入服务器的term终端，
然后可以在提示行中键入shell 指令

输入服务器域名，用户名，密码
选择ssh方式登录



Linux 简介(1)-内核和shell

UNIX/Linux 的任务可以简单地分为两部分：

承担操作系统与计算机之间的互动工作--内核(kernel),

承担操作系统与用户之间的互动工作--shell.

目前广泛应用的几大类shell (即指令集)：

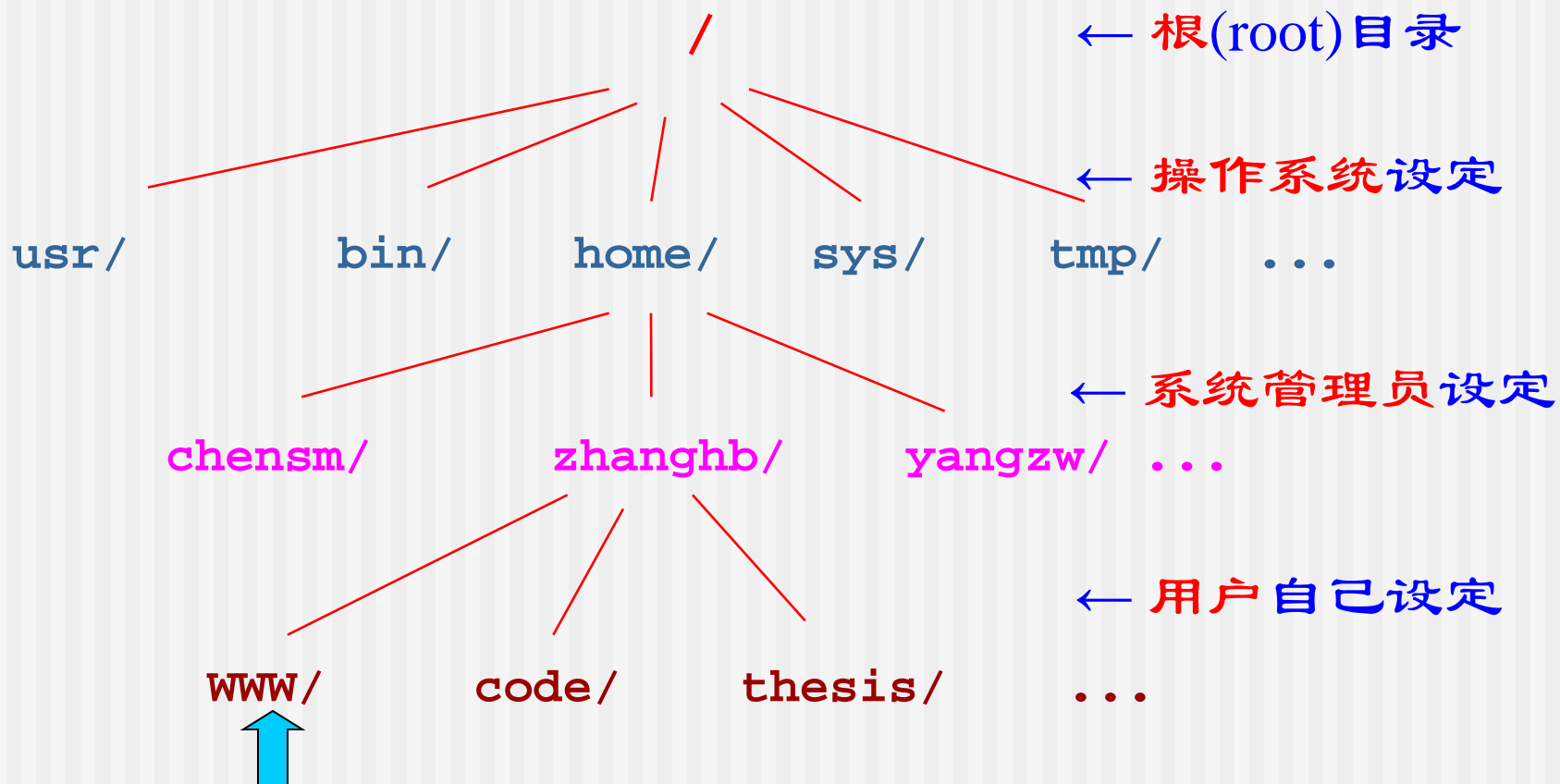
sh, csh, tcsh, bash, ...

目前一般系统默认的是bash，高能物理里常用的为tcsh。
语法稍微有些区别，大同小异。

用自己习惯的就可以，无所谓优劣。

Linux简介(2)-目录

UNIX/Linux 对文件与目录的管理,整体呈**树状**(tree)结构:



Linux: public_html/

Linux简介(3)-文件所有者和权限

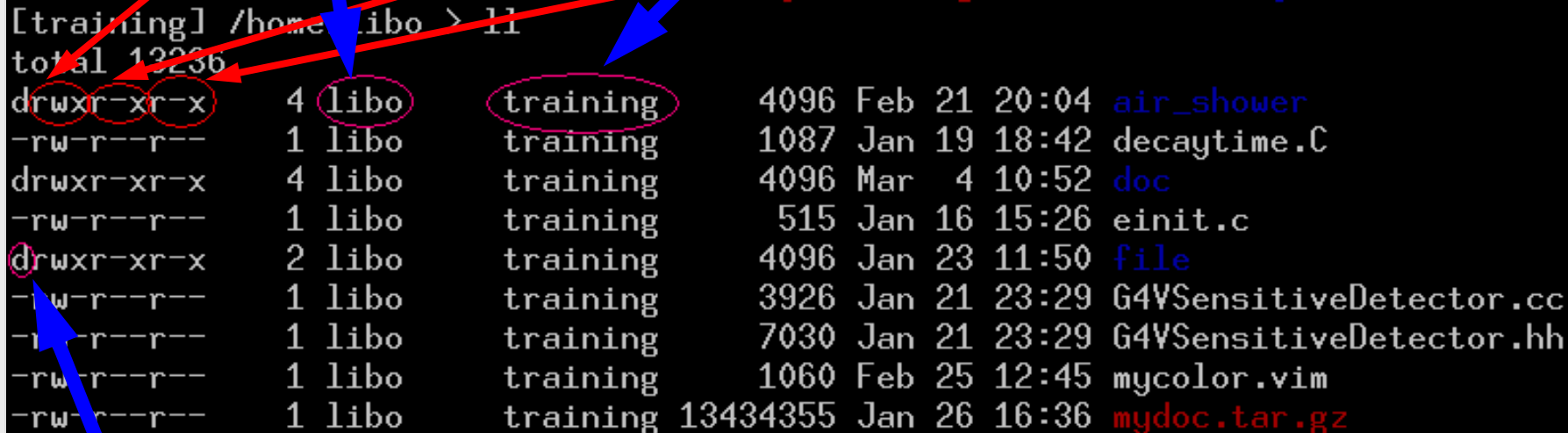
所有者与权限:

Linux文件都有特定的所有者和所有组。

文件具有3类权限: 可读(r)、可写(w)、可执行(x)

所有者有权规定3类用户对该文件的权限:

所有者自己(owner)、所有者同组成员(group)、其它用户(other)



The terminal screenshot displays the output of the 'ls -l' command in a directory. The first line shows the total size: 'total 13236'. Subsequent lines list files with their permissions, owner, group, size, date, and name. Annotations include red and blue arrows pointing from the text above to specific parts of the output: a red arrow points to the permissions 'drwxr-xr-x' of the first file; a blue arrow points to the owner 'libo' of the first file; a red arrow points to the group 'training' of the first file; and a blue arrow points to the permissions 'drwxr-xr-x' of the file 'file'.

Permissions	Link Count	Owner	Group	Size	Date	Time	File Name
drwxr-xr-x	4	libo	training	4096	Feb 21	20:04	air_shower
-rw-r--r--	1	libo	training	1087	Jan 19	18:42	decaytime.C
drwxr-xr-x	4	libo	training	4096	Mar 4	10:52	doc
-rw-r--r--	1	libo	training	515	Jan 16	15:26	einit.c
drwxr-xr-x	2	libo	training	4096	Jan 23	11:50	file
-rw-r--r--	1	libo	training	3926	Jan 21	23:29	G4VSensitiveDetector.cc
-rw-r--r--	1	libo	training	7030	Jan 21	23:29	G4VSensitiveDetector.hh
-rw-r--r--	1	libo	training	1060	Feb 25	12:45	mycolor.vim
-rw-r--r--	1	libo	training	13434355	Jan 26	16:36	mydoc.tar.gz

最前面的字母"d"表示这是个目录

"-"表明没有该位置对应的权限

2011-3-3

Linux简介(4)-绝对和相对路径

一个完整文件名应指定出其所处在的**路径** (path), **路径有相对路径和绝对路径**

绝对路径: `/home/chensm/geom/geom.dat` (都是以`"/`开始)

一个 `'~'` 符号可以用来简单表示home: (相对路径的例子)

`~/geom/geom.dat` **← 所登录的用户 (e.g. chensm)**

`~zhanghb/geom/result.dat` **← 不同的用户**

单个点 `'.'` 表示当前目录, 两个点 `'..'` 表示上一层目录

`/home/chensm/geom` **← 当前目录**

`../geom` **← 表示 /home/chensm/geom**

这种文件技巧对以后编写通用的 shell 脚本文件很有用。

Linux终端的常用命令(1)

<code>pwd</code>	显示当前目录(print working dir)
<code>passwd</code>	修改当前用户的密码
<code>ls [-lahrt]</code>	列出当前目录中的文件(list)
<code>cd [dir]</code>	进入指定目录或从当前目录回到用户的home目录 <code>cd, cd foo, cd ../ cd /home/zhanghb/</code>
<code>mkdir dfoo</code>	生成名为 <i>dfoo</i> 的子目录
<code>rm [-rf]foo</code>	删除文件 <i>foo</i> (参数 <code>rf</code> 表示强制删除文件夹, 慎用)
<code>rmdir foo</code>	删除名为 <i>foo</i> 的子目录(<i>foo</i> 应已经为空目录)
<code>cp foo bar</code>	拷贝文件 <i>foo</i> 到另一文件 <i>bar</i>
<code>mv foo bar</code>	更改文件 <i>foo</i> 的名称为 <i>bar</i>
<code>man <command></code>	显示 <code>command</code> 指令说明
<code>man -k <keyword></code>	寻找 “ <i>keyword</i> ”指令说明页
<code>history</code>	列出最近使用过的指令很有用
<code>du</code>	显示当前目录所用空间大小

注意指令字母大小写。实际上Linux区分一切大小写, 指令, 文件名, 目录名

Linux终端的常用命令(2)

more foo	显示名为foo的文件(按空格键换页)
less foo	与 more foo类似, 但可以往回翻页(按q 退出)
emacs foo &	用emacs 编辑名为 foo 的文件(& 为提交后台进程)
vi, pico, nano, ... 这些命令都以可编辑方式打开文件	
ps	显示正在运行的进程
kill 345	删除进程 345 (如果不行可尝试使用 kill -9)
./foo	在当前目录运行可执行文件 <i>foo</i>
ctrl-c	中断目前在前台执行的进程
ln -s source linkname	为source建立一个符号链接linkname
locate foo	在所有目录中寻找有文件名 <i>foo</i> 的路径
find . -name file1	在当前目录中寻找文件名为file1的路径
grep TH1F foo	显示文件 <i>foo</i> 中含 “TH1F” 的每一行
sed -e "s/str1/str2/g" foo > bar	将文件 <i>foo</i> 中字符串 “str1”改为 “str2”并将修改后的文件写到新文件 <i>bar</i> , <i>foo</i> 保持不变。

Linux终端的常用命令(3)

<code>chmod 755 <file></code>	更改文件file的属性,1:x 2:w 4:r 5:rx 7:rwx
<code>chmod ug+x foo</code>	使文件 <i>foo</i> 对用户与同小组成员增加执行权限
<code>diff file1 file2</code>	比较文件file1和file2的不同
<code>tar -zcvf 1.tgz file1 file2</code>	压缩file1, file2为1. tgz
<code>tar -zxvf 1.tgz</code>	解压缩1. tgz
<code>gcc test.c -o try1</code>	用C编译器编译test.c, 生成可执行文件try
<code>g++ test.cpp -o try2</code>	用C++编译器编译程序
<code>date</code>	显示系统当前时间
<code>sleep 10</code>	暂停10秒钟
<code>wc [-lw] file</code>	显示file的行数/字数等信息
<code>echo "Welcome to Linux World!"</code>	屏幕显示指定字符串
<code>file file1</code>	显示文件file1的属性

注: Linux有些特殊字符, 比如 `>`, `|`, `&`等符号

`ls > hehe.txt` 将ls的结果写入hehe.txt, 即重定向

`ps aux | grep yangzw` 显示跟用户yangzw有关的进程, 即通道

Emacs、Vi编辑器的基本指令

emacs: 很好很强大

打开文件 **emacs [filename]**

保存文件 **^x ^s**

退出文件 **^x ^c**

nano(pico): 简单文本编辑

打开文件 **nano [filename]**

保存文件 **^o**

退出文件 **^x**

vi(vim): 古老，不过也很强大

打开文件 **vi [filename]**

保存文件 **:w** (注意：是输入冒号然后输入w或q或q!)

退出文件 **:q**

不存退出 **:q!**

注：**vi**有两种模式，命令模式和输入模式

按小写字母"**i**"进入输入模式，按"**esc**"键进入命令模式

在命令模式中可以输入命令很方便的进行编辑修改

讲义最后列了一些**vi**的常用命令

熟练使用任何一种编辑器都可以极大提高工作效率，建议多多练习。

Shell、环境变量和脚本(1)

承担操作系统与用户之间的互动工作--shell.

常用的shell (即指令集): tcsh, bash, ...

chsh 该命令改变shell, 输入密码后输入/bin/tcsh或/bin/bash

shell中有很多环境变量, 有的是系统的环境变量, 有的是用户自己定义的环境变量, 为系统和用户程序服务。

环境变量一般用大些字母定义(有些类似于C语言的宏定义)

比如**PATH**, **PWD**, **USER**, **GROUP**等都是系统环境变量。

查看所有环境变量: **env**或者**printenv**

查看环境变量**PATH**的值: **echo \$PATH**或**printenv PATH**

定义环境变量

export YANGDIR=/home/yangzw (bash)

setenv YANGDIR /home/yangzw (tcsh)

取消环境变量 **unset YANGDIR**

Shell、环境变量和脚本(2)

什么是脚本(script):

脚本就是用于实现某种目的的命令集合。

这些命令集合放在一个文件中，由shell来解析执行。

为什么需要用脚本：

很多工作是重复性的，脚本可以让你更高效。

比如用脚本循环修改程序的某一部分，自动运行。

目 标：

- 1)知道什么是shell脚本(script)，如何写自己的脚本
- 2)可以看懂别人的脚本

执行shell脚本，比如有脚本test.sh:

./test.sh

注：运行前确保用户对test.sh有可执行权限，否则，

`chmod u+x test.sh`

Shell、环境变量和脚本

例：最简单的一个脚本
编写一个shell脚本test.sh:

```
#!/bin/bash
# This is a simple test shell script

echo "Hello everyone!"
ls /projects/$USER
date
echo $PWD
```

注：1) 标准的脚本都以“#!”开头，后面跟随bash/tcsh或其它脚本程序的路径(用which bash指令可以查看bash的路径)
2) 注释行以“#”开头(第一行的#!除外)
3) 需要执行的指令(一般每行一个指令)

Shell、环境变量和脚本

```
#!/bin/bash
# Another test shell script
```

####for循环####

```
for i in `ls /home/yangzw`
do
    echo $i
done
```

####while循环###

```
num=1
DIR="testDir"
while (( $num < 5 ))
do
    if [ -d $DIR$num ]; then
        echo "$DIR$num exist!!"
    else
        mkdir $DIR$num
    fi
    let num+=1
done
```

```
#!/bin/tcsh
# Another test shell script
```

####for循环####

```
foreach i `ls /home/yangzw`
    echo $i
end
```

####while循环###

```
num=1
DIR="testDir"
while ( $num < 5 )
    if ( -d $DIR$num ) then
        echo "$DIR$num exist!!"
    else
        mkdir $DIR$num
    endif
    set num=`expr $num + 1`
end
```

练习

1. 到/`projects`目录下以`$USER`为名建立自己的工作目录，在`$HOME`目录下建立一个符号链接“`workdir`”，链接到/`projects/$USER`（提示：可以用`echo`或`printenv`看看`USER`和`HOME`两个环境变量的值）
然后到`workdir`中创建目录`dir1`, `dir2`, `dir3`,
以及文件`file1`,`file2`,`index1.htm`,`index2.htm`,`test1.txt`
（提示：创建文件有多种方法，简单的方法是用`touch`命令）
设定以下两个环境变量`WORKDIR`和`MYNAME`，让它们分别等于
 /`projects/$USER`
 和 `${USER}_${GROUP}`
2. 编写脚本`myscript1.sh`，要求：
 - 1) 显示开始运行的时间；
 - 2) 在屏幕上打印出当前目录，当前用户名以及`SHELL`类型
 - 3) 显示`WORKDIR`与`MYNAME`的值
 - 4) 显示当前`$WORKDIR`目录所用磁盘空间
 - 5) 间隔5秒钟之后再显示出当前时间

练习

3. 编写脚本`looptest.sh`, 要求:
 - 1) 将练习1中所有文件属性改成只有自己可以读/写/执行, 其他人只能读
 - 2) 将`$WORKDIR`目录中`*.htm`文件改成`*.html`
(提示, 可以用`rename`命令, 也可以用`for`循环)
4. 将`/projects/yangzw/Lec_materials/test1.txt`拷贝到自己的工作目录, 再把`test1.txt`备份成`test1.txt.bak`.
练习用`sed`修改其中的字符串, 用`grep`查找其中的字符串,
比如:
 - 1) 查找其中包含“`Vim`”的行, 并将结果写入`final1.txt`
 - 2) 不区分大小写查找包含“`vim`”的行, 显示查找出多少行
 - 3) 用`sed`将其中的“`vim`”修改成“`Vim`”
 - 4) 比较`test1.txt`与`test1.txt.bak`的差别
 - 5) 删除`final1.txt`

.....
5. 练习使用其它常用的Linux命令

总结

- 介绍Linux操作系统以及常用命令
- Linux终端的常用编辑器(vi, emacs, pico, etc)
- 介绍shell、环境变量和脚本
- 介绍Linux下c/c++程序的编译指令

目的：了解Linux操作系统的基本概念
熟练掌握Linux操作系统的常用命令
了解Shell脚本编程的基本概念

参考资料

1. Linux从入门到精通, Phil Cornes著; 童寿彬等译
2. RedHat Linux 9系统管理, 李蔚泽编著
3. 精通Shell编程, Sriranga Veeraraghavan著; 卢涛译
4. Linux命令、编辑器与Shell编程, Mark Sobell著, 杨明军等译
5. <http://hep.tsinghua.edu.cn/~yangzw/CourseDataAna/>(一些Shell脚本编程的资料)
6. Google and/or Baidu !!!!

备用(vi常用指令)

vi 的常用技巧

- | | | | |
|--------------------|---------------------------------------|---|----------------------|
| 1. 显示行号
:se nu | 13. 删除光标至行尾
D | 22. 将3-9行的" Abc"替换为"ABC"
:3,9s/Abc/ABC/g | 33. 全文加亮光标当前变量
gd |
| 2. 移动光标到第5行
:5 | 14. 用某字母(如"k")替换光标所在字符
r k | 23. 将3-6行复制到第9行
:3,6 co 9 | 34. 保存文件
:w |
| 3. 移动光标到行首
^ | 15. 向下新增一行
o | 24. 将3-6行移动到第9行
:3,6 m 9 | 35. 保存退出
:wq |
| 4. 移动光标到行尾
\$ | 16. 向上新增一行
O | 25. 删除3-6行
:3,6 d | 36. 不保存退出
:q! |
| 5. 移动光标到文件头
gg | 17. 复制光标所在行
yy | 26. 3-6行行首加上"ABC"
:3,6s/^/ABC/g | 37. 进入输入模式
i |
| 6. 移动光标到文件尾
G | 18. 将复制的行粘贴到光标所在行下方
p | 27. 3-6行行首加上"//",即C++注释
:3,6s/^/\//g | 38. 进入命令模式
ESC |
| 7. 向后移动3个字
3w | 19. 将复制的行粘贴到光标所在行上方
P | 28. 3-6行行尾添加"ABC"
:3,6s/\$/ABC/g | 39. 在行首进入输入模式
I |
| 8. 向前移动4个字
4b | 20. 查找字符串"Abc"
/Abc | 29. 将光标的下一行连接到光标所在行
J | 40. 在行尾进入输入模式
A |
| 9. 删除光标所在字
dw | 21. 全局替换"Abc"为" ABC"
:%s/Abc/ABC/g | 30. 将光标所处字母变更大小写
~ | |
| 10. 删除光标所在字符
x | | 31. 取消操作(undo)
u | |
| 11. 删除行
dd | | 32. 重复操作(redo)
. | |
| 12. 删除光标后3行
3dd | | | |

2011-3-3