

# Unix 课程辅导材料

## 一. unix 系统

1.user → shell → kernel → HardWare

Shell 解析器和翻译器

Kernel 是管理设备、激活进程

2.根目录下分为了 8 个目录，其中一个不可以使用

opt 放第三方软件或程序

etc 配置文件

dev 操作系统命令

usr 操作系统命令

kernel 内核信息

var 日志信息、临时文件

export 各个用户的工作目录

3.在 unix/linux 中都是以文件存储的

## 二. 基本操作

1.登录到系统中需要提供用户名和密码,这些信息存在/etc/passwd 和/etc/shadow 中

/etc/passwd 中存储的→登录名:口令占位符: uid: 组 id : 注释字段: 起始目录: shell  
程序

/etc/shadow 中存储的+→登录名:加密口令

passwd 改变密码

2.clear

pwd 显示绝对路径

3. cd

cd /user/yangxp

cd dir1 ↔ cd ./dir1

cd ../..

cd

cd ~

**注意:** ~回车, 显示当前工作目录

~username 回车, 显示其他人的工作目录

4. ls

ls -l 除名称外, 显示详细信息

文件类型-拥有者权限-同组人权限-其他人权限 硬连接数 拥有者 拥有组 大小 最后修改时间 文件名

ls -a 在 ls 基础上显示隐藏文件, 即以.开头的文件

ls -t 在 ls 基础上以时间排序, 最近的在最上面。不写这一项是以 ascii 码排序

ls -F 在名称后面加显示文件类型: @是二进制文件, /是目录, \*是可执行文件, 文本文件什么都显示

ls -R 在 ls 基础上显示目录及目录下的文件

5.file \* 查看所有的文件的类型

6.chmod 一般用法:chmod 755 a.txt

7.touch 创建新的文件, 如果文件已经存在, 则改变文件的时间戳

8.mkdir 创建目录

mkdir dir1/dir2 前提是 dir1 必须存在

mkdir dir1 dir2 创建多个目录

mkdir ~/dir1 在主目录下创建 dir1

mkdir -p dir1/dir2 创建新的目录层次

9.cp 将一个或多个档案复制成另一个档案或者是将其复制到另一个目录去。

cp f1 f2: 将档名为 f1 的档案复制一份为档名为 f2 的档案。

cp f1 f2 f3 ... dir: 将档案 f1 f2 f3 ... 都以相同的档名复制一份放到目录 dir 里面。

cp -r dir1 dir2: 将 dir1 的全部内容全部复制到 dir2 里面。

cp 也有一些参数, 如下:

-i: 交互是否覆盖

-r: 此参数是用来做递归复制用, 可将一整个子树都复制到另一个目录中。

**注意:** 将父目录拷入子目录中会造成死循环

10.mv 剪切文件/目录、将一文件改名。如同 cp, 它也有三种格式:

mv f1 f2: 将档名为 f1 的档案变更成档名为 f2 的档案。

mv dir1 dir2: 将档名为 dir1 的目录变更成档名为 dir2 的目录。

mv f1 f2 f3 ... dir: 将档案 f1 f2 f3 ... 都移至目录 dir 里面。

mv 的参数有两个, -f 和 -i, 其中 -i 的意义与 cp 中的相同, 均是 interactive 询问之意。而 -f 为强迫( force ), 就是不管有没有同名的档案, 反正我就是要搬过去, 所有其他的参数遇到 -f 均会失效。

12.rm & rmdir 用来删除的指令。

rm 的格式如下:

rm f1 f2 f3 .....

而 rm 的参数比较常用的有几个: -f, -i, 与 -r

-f: 将会使得系统在删除时, 不提出任何警告讯息。

-i: 交互是否真要删除。

-r: 递归式的删除。

**注意:** 小心不要随便使用 rm -rf, 否则有一天你会"欲哭无泪".....

### 13.ln

`ln a.txt b.txt` 为 `a.txt` 建立硬连接 `b.txt`，目标文件事先不能存在。链接是文件/目录的别名

`ln -s d5 d6` 为目录 `d5` 建立软连接，若 `cd d6` 然后 `pwd`，显示的是 `d5` 目录

软链接	硬链接
是产生一个特殊的文件，该文件的内容是指向另外一个文件的位置	是指一个文件可以有多个名称 存储空间不变，只是建立了一个指针； 而 <code>copy</code> 需要开辟新的 <code>data block</code>
<code>ln -s dir1 dir2</code>	<code>ln a.txt b.txt</code>
可以跨越不同的文件系统	只能存在于一个文件系统中
inode 不同	inode 相同
针对目录	针对文件
直接删源文件时，link 还存在，但是打不开了	删除一个链接，只是将 <code>link number -1</code> ，除非全部删除 <code>link</code> ，才真正删除； 无所谓哪个是 <code>link</code> 或源文件

查看节点号加 `-i`, `ls -il` / `ls -i` 可看到 `inode`

### 14.man 命令/函数

`man -k keyword` (一般是函数)

回车-----下一行

空格-----下一页

q-----退出

/string 查找关键字

## 三. 元字符

1.\* 匹配 0 到多个字符

2.? 匹配一个字符

3.[] 匹配一组或一个范围的字符

`ls [a-f]h`

`ls [dD]f`

4.; 连接多条命令

5.> 输出重定向到非标准输出设备（终端），以覆盖的方式

6.< 输入重定向从非标准输入设备（键盘）

7.>> 输出重定向以追加的方式，若源文件不存在则创建新的

\*8.| 管道是将前面的输出作为下一个命令的输入

**注意：**①现在有一个目录为 `d2`，要模糊查找这个目录时，用 `ls *2`，只能显示出来 `d2` 目录下面的内容；这种情况下要使用 `find *2`，才能显示 `d2` 的名称等

②lp 打印机打印

例：`cal 1 2007 | lp`

③如果需要输出分屏，则在命令后面加 “ | more ”

④tee 实现即在屏幕上输出又在文件中输出

```
cal | tee a.txt
```

```
cal 1 2007 | tee a.txt 覆盖原文件内容
```

```
cal 1 2007 | tee -a a.txt 追加
```

## 四. 文本操作

1. cat 查看文件内容、合并文件

全部显示到终端或加>重定向到文件中；-n 加行号显示

cat 适合查看小文件，一般用作将很多小文件连成一个大文件：cat file\* >newFile

2. more 是分屏显示，适合于看大文件

3. head -3 a.txt 显示最上面的 3 行。缺省是 10 行

4. tail -3 a.txt 显示最下面的 3 行，一般用作查看日志信息

tail -f a.txt 可以动态查看文件的最新内容

**注意：**①/dev/null 表示空设备，cal > /dev/null 表示将输出的内容扔掉

②cat /etc/passwd | awk -F: '{print \$1 "\t" \$6}' \

| sort >~/userinfo

awk 做格式化输出

-F 格式定义，定义在后面的 '{ }' 中

Print 打印到屏幕上

\$1 第一列。每列以: 分隔

"\t" 制表符，tab 键

\换行符，表示命令换行

Sort 排序，默认是将 ASCII 码排序

目的：将/etc/passwd 文件中的第一个字段和第六个字段提取出来并排序，存入~/userinfo 文件中

③

**总结：**①创建文件的方式：

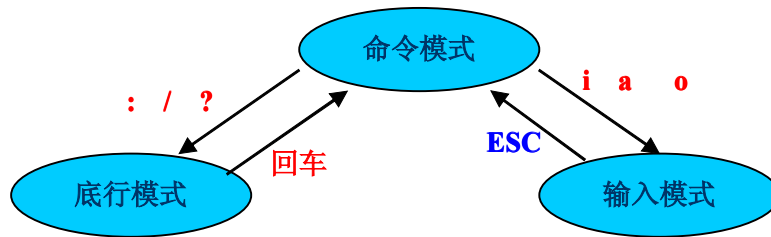
vi / touch / cat /> />>

②显示

cat / more / head / tail

## \*五. Vi

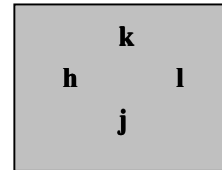
VI 的使用方法:



\$ 到行尾 shift +4  
^ 行首 shift+6  
G 文件尾 shift+g  
1G 到第一行  
20G 到第 20 行  
L 屏尾  
H 屏首  
M 屏中央

**:set number** 加行号  
**:set nonu** 去行号  
Ctrl B 前翻页  
Ctrl F 后翻页

退出: **ESC**  
**:q!**



**:w** 存盘  
**:w newfile** 存成新文件  
**:wq** 存盘退出  
**:q!** 不存盘强行退出

x 删除一个字符  
dw 删除当前词  
3dw 删除三个词  
dd 删除当前行  
5dd 删除 5 行  
**:5,10d** 冒号模式,  
从 5~10 行, 删!

r 替换一个字符  
cw 替换一个单词  
cc 替换一行  
c 替换从光标到行尾

-----  
yw 拷贝词  
yy 拷贝行 (Y)  
p 当前行下粘贴  
**:1,2 co 3** 拷贝行 1, 行 2 在  
行 3 之后  
**:4,5 m 6** 移动 4, 5 在 6 之后

~ 改变大小写  
J 把当前行和下一行连接  
u 废除 (撤消) undo  
**:21** 光标停在 21 行  
**21G** 光标停在指定行  
/串 从当前行向下查找  
?串 从当前行向上查找  
n 继续查找  
**:r file2** 在光标所在位置插入另一个文件 合并文件  
**:1,\$s/旧串/新串/g** 替换全文  
(或 **%s/旧串/新串/g**)  
**:1,\$ s/hiloo/xueh1/g** 当一行出现多次时加/g 全部替  
换, 否则只替换第一个  
**:1,\$ s/^/#/g** 每行加个#号

## 六. 查找

### 1. id

Id 显示自己的 uid 和 gid

id root 显示用户 root 的 uid 和 gid

id -a root 显示用户 root 在各个组里的别名

### 2. users 当前系统由哪些用户登录了

### 3. who 看登录用户的详细信息, who | more

登录名 (可重复) 伪终端号(只能唯一) 登录时间 登录 ip

一个用户是可以多次登录的, 但是每次登录的终端号是不可能重复的

w 比上述的命令更详细

用户名 终端号 登入时间 闲置时间。。。。

### 4. whoami 显示登陆本机的用户信息, 相当于 id, 但是 su 之后, 显示的是 root

### 5. who am i 显示当前有效用户身份, 比如 su 改变身份后, 仍然显示真正的用户

### 6. find 找文件, find 路径 查询条件;

Find 是精确匹配: find . -name file1

如果要模糊匹配: find . -name "file\*"

-name 文件名

-mtime 10 修改时间距今 10 天那天的文件, -10 表 10 天以内, +10 表超过 10 天的

-user 0 表 userid=0 的文件, 即 root 的文件

-size +400 表文件大小超过 400 个 blok, 一个 blok 是 512 字节的文件, 即 200K

-print 打印

-perm 777 权限是 777 的文件

-type f 只查普通文件,

-atime +365 访问时间超过一年的

-exec rm {} 执行删除操作, {} 表示将前面的结果作为 rm 的参数

### 7. grep 过滤

-v 不包含

### 8. wc 对单词进行统计

-l 以行为单位统计

-w 以单词为单位统计

-c 以字母为单位统计

### 9. du 统计磁盘容量

df -k 是对文件系统进行统计

du 单位是 512 字节

du -k 单位是 K, 即 1024 字节

### 10.ps 列出当前在系统中运行的进程, 只列出跟终端相关的进程

-e 列出系统中运行的所有的进程

-f 列出完整信息, 并格式化输出

ps -ef :

uid pid ppid c stime tty time cmd

→ 用户 id 进程 id 父进程 id 启动时间 终端号 (? 指不跟任何终端有关系, 一般是 daemon 进程) 累计使用 CPU 的时间 进程名

进程树: bash→csh→in.telnetd→inetd→init

prstat 实时显示当前系统中的所有进程,q/C 即可退出

11. su username 改变用户身份, exit 退出

12. kill 终止一个进程, 即发信号给进程

kill -15 终止进程, 可以忽略

kill -9 强制杀死进程

control+c 终止前台运行的进程

sleep 300.vs sleep 300&→前台运行 后台运行

control+z 挂起→jobs -l→bg %任务号

前台 ←=====→ 后台

Fg %任务号 ←jobs -l←stop 进程号

pkill sleep 可以直接跟进程名 (awk 提取进程号)

## 七. 网络

不同的网段之间不能通信

要想通讯则使用路由器

基本配置: ip、子网掩码、DNS 服务器地址

Dhcp 动态分配 IP 地址

ifconfig -a: 查看本机 IP, 在 pc 机上用/sbin/ifconfig

netstat -nr: 查看网络信息 (路由表):

目的地: default 是缺省路由, 即 0.0.0.0

ping:

telnet:远程登录

rlogin: 在一台服务器上重新登录到一个其他服务器上

不指定用户名则以现有的用户名登录, 否则用-l 指定用户名

ftp

## \*八. 环境变量

1. 查看:

①全局变量:

env 查看所有的环境变量。大写, 用: 分隔

echo LANG

echo \$LANG 显示变量 LANG 的系统缺省值, 或者用 echo "\$LANG"

echo \$HOME

②局部变量

echo \$path 查看, 小写, 用空格分隔

2. 设置:

①全局变量:

(C shell:) setenv variable value , 只对当前 shell 有效

例: CC=/usr/local/bin/gcc

```
CCC = /usr/local/bin/g++
C shell : .cshrc
B shell : .profile
Bash : .bash_profile
修改全局变量文件
然后运行文件 source .bash_profile
②局部变量
set path=(... ..)
```

**注意：**怎么在本机配置 ifconfig 命令

3. history

set history=20;(C shell 中必须要先 set,history 才能生效)

history→!n

history 5 只显示最后 5 条

!! 运行最后一条

4. aliases 别名 命令名（只对当前 shell 有效）

5. which 显示命令所在的路径

6. whereis 显示命令存在的所有目录，path 中定义的命令所在的目录顺序有关系，应该将最常用的放在前面

7. prompt 设置提示符

set prompt=value

## 九. 脚本

1. vi a.sh→chmod 755 a.sh→a.sh

注释用#标识

2. crontab

-l 查看日程表

-e 定制日程表，默认是 vi 编辑（setenv EDITOR ed）

分钟（0~59） 小时（0~23） 日 月 周（0~6） 脚本名称

-r 删除日程表