**Shell命令**

**Shell** 是Linux系统的用户界面，提供了用户与内核进行交互操作的一种接口。它接收用户输入的命令并把它送入内核

Ls(commands)🡺BASH(linuxshell)🡺01(convertedbinarycommands)🡺linuxKernel

LINUX系统是多进程、多用户和交互式的计算环境。

**Shell 命令搜索路径**

Shell搜索的目录名字都保存在shell变量PATH（在TC shell中是path）中。

变量PATH中的目录名用符号分开。在bash中“:”

变量PATH保存在~/.profile或者~/.login中（~:主目录）

**chsh** 命令来改变默认**登录shell**, -l选项显示系统可用的shell

$echo $SHELL /bin/bash $tcsh %ps

<Ctrl-D> 或exit命令返回到原来的shell或默认的shell

**shell的启动文件**

/ect/passwd记录每个用户的shell程序：root:x:0:0:root:/root:/bin/bash

用户名：密码：UID：GID：身份描述：主目录：登录shell

**/bin/bash的启动**

继承getty设置的TERM、PATH等环境变量. 首先寻找/etc/profile脚本文件，并执行它。然后在~下寻找启动脚本文件.bash\_profile，执行它；如果不存在，则执行.bash\_login；如果.bash\_login也不存在，则执行.profile文件。然后bash将作为一个交互式shell执行.bashrc文件(注：“~”为登陆目录)

~/.bash\_out：退出时执行

TERM：终端类型

/etc/bashrc ：整个系统的别名和功能；

/etc/profile 整个系统的环境参数和启动程序；

~/.bashrc：用户的的别名和功能；

~/.bash\_profile：用户的环境参数和启动程序

~/.profile：用户登录环境配置文件(Bourne shell的)

**shell的环境变量**

显示shell的路径：echo $SHELL

查看环境变量值： set

设置环境变量值： 变量名=值

HISTFILE：用于贮存历史命令的文件

HISTSIZE：历史命令列表的大小

HOME：当前用户的主目录

OLDPWD：前一个工作目录

PATH：bash寻找可执行文件的搜索路径

PS1：命令行的一级提示符

PS2：命令行的二级提示符。

PWD、cwd：当前工作目录。

SECONDS：当前shell开始后所流逝的秒数

**Shell元字符**

" 引用多个字符,允许替换 "$file".bak

‘ 引用多个字符 ‘$100,000’

$ 一行的结束/显示变量的值 $PATH

& 让一个命令在后台执行 command &

( ) 在子shell中执行命令 (cmd1;cmd2)

{ } 在当前shell中执行命令 {cmd1;cmd2}

\* 匹配0个或者多个字符 chap\*.ps

? 匹配单个字符 lab.?

[ ] 插入通配符 [a-s],[1,5-9]

^ 一行的开始/否定符号 [^3-8]

` 替换命令 PS1=`cmd`

| 创建命令间的管道 cmd1|cmd2

; 分割顺序执行的命令 cmd1;cmd2

< 重定向命令的输入 cmd<file

> 重定向命令的输出 cmd>file

\ 转义字符/允许在下一行中继续shell命令

! 启动历史记录列表中的命令和当前命令 !!,!4

% TC shell的提示符，或者指定一个任务号时作为起始字符 %或者%3

如：$ls [0-90[a-zA-Z].html

**常用命令**

**gcc [option] [sourcefilename]**

-S 激活预处理、编译；生成汇编代码（.s文件）

-c 激活预处理、编译、汇编；生成目标文件（.o文件）

-o 生成目标

-Wall 打开编译告警（所有）

-g 嵌入调试信息，方便gdb调试

**passwd :**修改密码命令

**man, info**

如：man password; info password; ls –help|more; man –S2 **open**; man 2 open

**whatis**：得到任何LINUX命令的更短的描述

**whoami**：显示用户名leafior

**which**：当某个工具或程序有多个副本时，用which来识别哪个副本在运行

**who**：显示现在正在使用系统的用户的信息

**w**：比who更加详细地列出系统上用户的信息

**hostname**：显示登录上的主机的名字Ubuntu

**uname**：显示操作系统的信息Linux

PATH=~/bin:$PATH:. 搜索路径中增加~/bin和.目录

**cal** [[month] year]如cal 4 2011

**alias** [name[=string]…] 为name命令建立别名string。如alias more=’pg’

unalias 删除别名 alias ll=’ls –C’

uptime显示系统运行时间命令

**su** [-][-c <command>] [username]

-c< cmd >执行完指定的指令后，即恢复原身份。

-改变身份时，也同时变更工作目录，及HOME、SHELL、PATH变量等

Username指定要变更的用户名，默认为root

**pwd**： 显示当前的工作目录 /home/leafier

<Ctrl-C>: 终止当前的命令or程序

<Ctrl-D>: 结束输入，or退出linux系统，or从上层shell返回

<Ctrl-Z>: 暂停当前命令执行

**文本编辑器**

vi filename(vim filename) 命令模式+插入模式

进入插入模式：I | 退出：ESC或者Ctrl+i

:wq保存并退出 | :q!不保存退出

命令模式下在命令前加：！可以执行shell，如：！pwd

编辑器：vi(m), *emacs, gedit, kedit*, pico/nano

**文件系统结构**

主目录~（登陆目录）echo $HOME /home/leafior

当工作目录 “.”目录本身；“..”父目录

绝对路径（/usr/src）相对路径（~/course…）

**文件系统挂载**

**mount [-t fstype] [-o options] /dev/xxyN dirname**

如：mount –t vfat /dev/hda1 /mnt/c

类型 设备文件 挂在目录

挂接U盘：mount –t vfat /dev/sda1 /mnt/usb

fstype: iso9660/vfat/ntfs/msdos

/dev 保存所有设备文件的目录

xx：IDE硬盘为hd、SCSI硬盘和usb盘为sd，软盘fd

y：同种设备的顺序号，第一个硬盘为a

N：同一个设备编号，硬盘1-4为前面四个主分区，5开始为逻辑分区。

/dev/hda1（第一个硬盘的第一个分区）

/dev/hda5、/dev/hda6（逻辑分区）

/dev/fd0（软盘） /dev/hdc（光盘）

/dev/sda1（通常为移动硬盘或U盘的第一个分区）

mount –a：挂载在/etc/fstab中列出的所有设备

umount /dev/hda5 或umount /dosd卸载

**文件结构导航**

**echo**

-E不解释转义字符/-e解释(“\c”回车不换行)

-n不输出行尾换行符

**ls**

-r 将文件以相反次序显示(原定依英文字母次序)

-t 将文件依建立时间之先后次序列出

-i 显示inode号

-A 同-a ，但不列出“.”及“..”隐藏文件也来

-F 可执行文件则加“\*”, 目录则加“/”

-R 若目录下有文件，则以下的文件亦皆依序列出

ls –l [文件列表]

ls –ld [目录列表]

-rw-r--r-- 1 root root 44870 Dec 27 16:33 Makefile

(1) -普通文件，b块特殊文件，c字符特殊文件，d目录，l连接，p命名管道（FIFO）

所有者、组和其它用户的访问权限

(2)连接数 (3)所有者的登录名 (4)所有者的组名

(5)文件大小byte (6)(7)(8)修改时间 (9)文件名

ls ~/ [!0-9]\*.[c,C]ls [0-9][a-zA-Z].html #两个字符构成

**mkdir [options] dirnames**

-p 若上层目录目前尚未建立，则创建父目录

-m设置目录的权限。设置法与chmod 指令相同

$ mkdir –m 777 dir\_name

**rmdir [options] dirnames**

-p 删除指定目录之后，若该目录的上层目录已变成空目录，则将其一并删除

**touch** sample.h文件不存在，新建空；存在，修改时间

**file [options] file-list**：显示文件内容类型命令

**cat/** tac nl == cat -n

-E 每行尾显示符号$ -n 显示每一行的行号

$ cat 执行指令，不加任何参数

键入任何文字后，回车，系统回应一模一样的文字

$ cat file1 file2 > file3

file1和file2合成file3，若文件file3已经存在，覆盖；若用“>>”，新的内容就会附加在原有内容之后。

空格下一页，回车下一行，Q退出

cat sample

**pr**把文件分割成多页并在每页插入页眉，包含日期、当前时间、文件名和页号

df 命令可显示所有文件系统对i节点和磁盘块的使用情况。

该命令各个选项的含义如下：

-a 显示所有文件系统的磁盘使用情况，包括0块（block）的文件系统，如/proc文件系统。

-k 以k字节为单位显示。

-i显示i节点信息，而不是磁盘块。

-t 显示各指定类型的文件系统的磁盘空间使用情况。

-x 列出不是某一指定类型文件系统的磁盘空间使用情况（与t选项相反）。

-T 显示文件系统类型。

**more** 、pg 一次显示一个屏幕的内容

+/str 从包含str那行的前两行开始显示

-nN 每屏/页显示N行

+N从第N行开始显示文件内容

$more –c –10 file1 #显示文件file1的内容，每10行显示一次，而且在显示之前先清屏。

$ more sample phones

一次显示一屏幕的sample文件的内容，然后以同样的方式显示phones文件的内容

more –c -10 file1每10行显示一次，显示前先清屏

**less** 可上下翻页

-N 显示行号

-o file 当命令输入是管道的时候，除了显示屏，还把输出写到文件file里；若已经存在，会提示是否覆盖。

-p pattern 在文件中查找匹配“pattern”的第一处位置。

**head** 默认显示10行，-N控制显示开始的行

**tail** 默认显示10行，-f如果文件正在被追加，会继续显示追加的行，直到键入<Ctrl-C>。

-n N 若N前加“+”号表示显示从文件第N行开始的所有行；否则显示文件的最后N行

-r 逆序显示（最后一行先显示）

**cp**

-a 在备份中保持尽可能多的源文件结构和属性

-d 拷贝时保留链接

-f 删除已存在的目标文件不提示

-i 提示是否覆盖已存在的目标文件

-p 保持原先文件的所有者，组权限和时间标志

-r 递归复制目录，把所有非目录文件当普通文件复制

-R 递归复制目录

**mv**

mv [option] file1 file2 重命名file1为覅leor移动文件

mv [opion] file-list dirctory把文件列表中所有文件移到目录下

-i 在覆盖目标文件前提示用户 | -f 强制转移

**rm**

rm [option] file-list

-f 强制删除 | -i 删除前提示 | -r 递归删除

**wc** [options] file-list

显示文件大小，有行数、单词数和字符数（字节数）

-c 只显示字符数 -l 只显示行数 -w 只显示单词数

**diff** [options] [file1] [file2]

如果用-代替file1或file2，将从标准输入读取输入

-b 忽略行尾的空白

-e 生成并且显示一个脚本，编辑器ed可以用它来把文件file1转化成file2

-h 快速比较（此时不能使用-e选项）

若使用diff的时候不带任何选项，它会生成一系列指令，如果两个文件不同的话就可以使用这些指令把文件file1转化成file2：

L1aL2,L3：在F1第L1行后添加F2的L2行到L3行

L1,L2cL3,L4：把file1中的L1行到L2行替换成file2中的L3行到L4行

L1,L2dL3：删除file1中的L1行到L2行

**zdiff和zcmp**可以用来比较两个压缩文件

**uniq** [options] [+N][input-file] [output-file]

删除已经排序好的文件input-file中的所有重复行， 并把处理后不重复的行输出到output-file中

如果未指定output-file，输出到标准输出设备上

如果未指定input-file，从标准输入设备中输入

-c 在每行之前显示他们出现的次数

-d 只显示重复行 -u 只显示未重复行

**lpr** [options] file-list 打印

-# N 把file-list中的文件打印N份；默认只打印一份

-P ptr 把打印请求提交给打印机ptr

-T title 在标题页上打印标题title

-m 打印结束后发送电子邮件

-p 用命令pr来格式化输出

lpr提交一个需要打印的文件

lpq显示打印任务的状态

lprm从打印队列里清楚一个或多个打印任务

lpc激活打印控制程序

lptest生成测试打印机的波纹图案

**正则表达式**

支持工具：awk, ed, egrep, grep, sed, vi

x|y|z x或y或z

/L..e/ Love, Live, Lose, …

^x 以x开始的string

x$ 以x结束的string

\\* \*

(xy)+或\(xy\)+ xy,xyxy,xyxyxy, …

xy? x, xy

xy\* x, xy, xyy, xyyy

xy+ xy, xyy, xyyy, ..

[], [^] /[Hh]ello/ /[^A-KM-Z]ove/ Love

{n} 匹配n次

{n,} 匹配n或n+次

{n,m} >=n, <=m

**tar [options][filename-list]**

-c 建立新的备份文件。

-r 将文件附加在备份文件后面。

-f archname 用archname作为存档或恢复文件的备份文件名；默认是/dev/mto。如果archname是-，从标准输入读（对解压文件），或写到标准输出（对建立档案文件），这是当tar用作管道时的一个特性。

-t 以类似ls –l格式列出磁带上的内容（备份在磁带上的文件名）。

-u 将把比备份文件中更新的文件加入到备份文件中。

-x 从备份文件中取出文件。

-z 在tar创建备份文件时，使用gzip命令对它进行压缩；而从备份文件提取文件时，用gzip命令来解压备份文件。

-v 详细显示文件处理过程，用x选项解压文件的过程或存档文件的过程。

$ tar -cvf bash.help.tar \*.help

$ tar -zxvf linux-2.5.15.tar.gz

**rpm [options] [rpm-filename]**

**-v** 显示安装过程的详细信息。

**-h** 显示安装进度。

-a 查询所有安装的软件包。

-f filename 查询指定文件名的软件包。

-p rpm-filename 查询指定的软件包。

-d 只有列出文件。

**-i** 显示软件包信息，包含名称、版本和描述。

-l 列出软件包的文件。

-R 列出相关的软件包。

-s 显示软件包内文件的状态。

**-U** rpm-filename 升级指定的软件包。

-q 使用交互模式。

**-e** rpm-filename 删除指定的软件包。

-F rpm-filename 更新指定的软件包。

-i rpm-filename 安装指定的套件。

**sort [选项] 文件列表**

-b 忽略前导的空白

-d 按字典顺序排序，比较时仅字母、数字、空格和制表符有意义

-f 将小写字母与大写字母同等对待

-k pos1,pos2 指定一个或几个**字段**作为排序关键字，字段位置从pos1开始，到pos2为止。如不指定pos2，则关键字为从pos1到行尾。字段,字符的位置从1开始。

-r 按逆序输出排序结果

-c 检查给定文件是否已排好序，如果它们没有都排好序，则打印一个出错信息，并以状态值1退出。

-o 输出文件将排序输出写到输出文件中而不是标准输出，如果输出文件是输入文件之一，sort先将该文件的内容写入一个临时文件，然后再排序和写输出结果。

Sort –k 2 students 姓氏第2个字段开始作为比较键值

Sort –k 4 –r –b students用电户作为比较键值，忽略行首的空格和tab，逆序排列结果

**history [options] [filename]**显示或操作历史命令列表

历史命令保存在~/.bash\_history文件中

!! 执行最近一个命令

!n 执行历史命令列表中的第n个命令

!-n 当前之前的第n个命令

!string 最近用到的以string开始的命令

**find [目录列表][表达式]**

-exec CMD 如果命令(CMD)返回0（返回值为真），则该文件符合要求；命令必须以\; 结束

-ok CMD 和-exec相同，执行命令需要确认

-inum N 搜索inode为N的文件

-links N 搜索有N个链接的文件

-name pattern 搜索文件名匹配pattern的文件

-newer file 搜索修改时间在file之后的文件

-perm oct 权限等于oct(八进制数字，如777)的文件

-print 显示符合要求的文件路径和文件名

-size ±N[c] 搜索文件大小为N块。字符c用来确定块的大小，默认为512个字节。+N表示大小超过N块的，-N就是小于N 块的

-user name搜索所有权为name的文件

\( expr \) 当表达式为真结果为真；表达式可用OR和AND组合

! expr 取反，当表达式为假时结果为真

得到一个文件的所有硬链接：

$ find /usr . –inum 258072 –print

括号用来标明需要匹配的表达式，在\(和-o前后必须要有空格。这个命令没有提示直接删除匹配的文件；如果想要提示，用参数-ok替换-exec。

$ find . \( -name core –o –name ‘\*.ps’ –o –name ‘\*.p’ \) –print –exec rm {} \;

**whereis [options] [file-list]**查明系统上是否存在特定一个目录，若存在，给出路径whereis –b cat

-b 只搜索可执行文件

-m 只搜索帮助文件

-s 只搜索源代码

**grep [选项]模式[文件列表] 中间**

**egrep[选项] [字符串] [文件列表]** 最慢，最灵活

**fgrep[选项] [表达式] [文件列表]** 最快,限制多

如果没有文件列表，则从标准输入读入数据

-c 仅输出匹配的行的个数

-i 在匹配的过程中忽略字母的大小写

-l 仅输出有匹配行的文件名

-n 匹配时同时输出行号

-s 对shell脚本有用，成功返回0，失败返回非零值

-v 打印出不匹配的行 -w 全字匹配

$ grep '[a-z]\{8\}' students

包含至少连续8个小写字母单词的行

$ egrep “^J|^K” students

**权限**

对于目录：

r :列出目录的内容 w：建立，删除

x :允许用户搜索这个目录，如果你没有对目录的执行特权，那么就不能使用ls –l命令来列出目录下的内容或者是使用cd命令来把该目录变成当前目录。

usr(u) / group(g) / others(o)

1 bit表示每一种权限，用户有8种可能的操作权限

3 bits表示某一用户的权限；3种用户，9 bits

**chmod**

chmod [{ugoa}{+-=}{rwx}] [文件或目录]

u：所有者 g：所属组 o：其他人 a：所有人 +：增加权限 -：减少权限 =：赋予权限 r读权限 w写权 x执行权限

chmod [mode=421] [文件或目录]

r: 4 w: 2 x: 1 -R 递归修改文件夹

**umask [mask]** :新创建的文件或目录的访问特权都将设置为1，除了在参数mask中设置为1的对应位。

文件访问权限 = 默认的访问权限–mask

Umask 013 对于一个新建的可执行文件764

**默认访问权限:** 执行文件为777 文本文件为666

**通过文件连接共享**

**ln [options] existing-file new-file**

**ln [options] existing-file-list directory**

-f 强迫建立链接

-n 如果“new-file”已存在，不创建链接。

-s 建立一个符号链接而不是硬链接

-d 建立目录的硬链接

**硬链接**：一个指向文件索引节点的指针，inode相同；不可跨越文件系统；只有超级用户才可以建立目录硬链接；不占用空间(极少)

**软链接**：可跨越文件系统，甚至跨越网络(NFS)；如果链接指向的文件从一个目录移动到另一个目录，就无法通过符号链接访问它；占有少量空间，存inode

**进程**

进程创建和终止是LINUX系统处理外部命令所采用的唯一机制。

内部命令：代码本身就是shell进程的一部分。.、alias、bg、cd、continue、echo、exec、exit、fg、jobs、pwd 、set、shift、test、time、umask 、unset和wait

外部命令：文件内容可以是二进制代码或者shell脚本。通常使用的一些外部命令如grep、more、cat、mkdir、rmdir、ls、sort、ftp、telnet、lp和ps

Shell执行二进制文件：

1.Shell使用fork创建子进程；2.子进程执行exec，用命令对应的可执行文件覆盖自身；3.命令执行，bash等待命令结束。

Shell执行脚本文件：

创建一个子shell并让子shell依次执行脚本中命令，执行与从键盘输入的命令采用相同的方式

子shell为每一个要执行的命令创建一个子进程。

子shell执行脚本文件中的命令时，父shell等待子shell 结束。子shell遇到脚本文件的EOF终止。

子shell终止，父shell结束等待状态，开始重新执行。

**ps**

-a 显示所有终端上执行的进程信息，包括其他用户

-e/-A 显示所有系统中运行的进程的信息

-j 采用作业控制格式显示所有信息（包括父进程的PID、组ID、会话ID等）

-l 用长列表来显示状态报告信息。

-p 根据进程ID显示对应的信息。

-u ulist 显示在ulist列表中有对应的UID或者名称的用户的进程信息(UID或者用户名由逗号分开)。

-t tlist 选取列在tlist中的终端上的进程；如果没有tlist，显示不带参数ps命令执行的结果。

命令ps aux参看所有进程，包括守护进程

**top**：实时监视CPU状态。该命令显示系统中CPU密集型任务的状态并且允许你交互地控制这些进程

后台/作业

$cmd &

[作业号]进程PID——作业是一个不运行于前台的进程，并且只能在关联的终端上访问。这样的进程通常在后台执 行或者成为被挂起的进程。

**fg[%jobid]** 后台🡪前台

%|%+ 当前的作业 %- 以前的作业 %N作业号为N

%Name开头名字为Name %?Name 命令中含Name

**<Ctrl-Z>** 挂起前台进程

**bg** 挂起进程🡪后台，参数同fg（后台🡺前台）

**jobs** 显示所有挂起/停止的和后台进程的作业号

**suspend** 可以挂起当前shell进程

**守护进程(Daemons)**：运行于后台的系统进程。用于向用户提供各种类型的服务和执行系统管理任务。

smtpd (e-mail service) httpd (web browsing)

inetd (internet related services)

**;顺序执行 &并发执行 |重定向**

最后一个&和之前的&间所有命令用一个进程完成。

$ date & who; whoami ; uname; echo Hello, World! &

date用一个进程执行，其他用另外一个进程执行

**命令组：** 命令组中的所有命令都在一个进程中执行（在当前shell的子shell中）

$ (date;echo Hello,World! )

Ctrl+C 终止前台进程

**命令的有条件执行**

cmd1**&&**cmd2：1成功则2 cmd1 **| |** cmd2：1失败则2

**kill [-signal\_number] proc-list kill –l**

发送signal\_number信号到PID或者jobID在proc-list中的进程；jobID必须以“%”号开始。

kill –l返回所有信号的号码以及名字的列表

1 挂断:退出系统,用调制解调器使用系统时挂断电话

2 中断<Ctrl-C> 3 退出<Ctrl-\>

9 强制终止 15 终止进程（默认的信号号码）

为了终止一个忽略15号信号或者其它信号的进程，需要使用9号信号，即强制终止信号

进程号0可以指代所有在当前登录期间创建的进程。kill –9 0可以终止所有登录时产生的进程（即当前会话中的所有进程），这样，你就不得不退出系统。

**ps –e f或pstree**：用图的形式显示当前系统中执行进程的进程树——$ pstree –a | more

**ulimit –u：**Bash下显示用户可同时执行的最大进程数

**重定向**

command < input-file > output-file

command > output-file < input-file

>换成>>则追加文件，否则替换

**sdin— 0 stout — 1 sderr — 2**

cat lab1 lab2 1> output 2>&1

2> &1：使文件描述符2为文件描述符1 的拷贝，导致错误信息送往和该命令输出相同的地方

在命令行的解析中，文件的重定向顺序是从左到右

$ cat lab1 lab2 lab3 2>&1 1>output

标准出错先设置成显示器，标准输出才改为output

**tee [options] file-list**

从标准输入中得到输入，送到标准输出和file-list中

cmd1| tee file1…fileN|cmd2

cmd1标准输出作为tee的标准输入，tee输出送到文件file1到fileN中，同时作为cmd2的标准输入

**< &-** 关闭标准输入 **> &-** 关闭标准输出

**m<&或m>&-** 将文件描述符m关闭

<> file将标准输入和输出都分配给文件file

n <file将文件file设为文件描述符n

n> file将文件描述符n指向文件file

> &file标准输出和出错输出重定向到文件file

**输入输出**  
**1．I/O重定向**  
< ：输入重定向   
>; ：输出重定向(没有文件则创建，有则覆盖)   
>;>; ：输出重定向(没有则创建，有则追加到文件尾部)   
<< ：输入重定向(here文档)   
格式： command << label   
**2．字符串I/O操作**  
字符串输出：**echo**   
命令选项： -e：启动转义序列 -n：取消输出后换行   
转义序列： \a：Alt/Ctrl+G(bell) \b：退格Backspace/Ctrl+H   
\c：取消输出后换行 \f：Formfeed/Ctrl+J   
\r：Return/Ctrl+M \v：Vertical tab   
\n：八进制ASCII字符 \\：单个\字符 \t：Tab制表符   
字符串输入：**read**  
可以用于用户交互输入，也可以用来一次处理文本文件中的一行   
命令选项： -a：将值读入数组，数组下标从0开始   
-e：使用GNU的readline库进行读入，允许bash的编辑功能 -p：在执行读入前打印提示