

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。Linux命令→BASH(linuxshell)→01(convertedbinarycommand)→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 命令可改变默认 登录 shell ，-t 选项显示系统可用的 shell Secho \$SHELL /bin/bash\$tcsh %ps -Ctrl-D 或 exit 命令可返回到原来的 shell 或默认的 shell shell 的启动文件 /etc/passwd 记录每个用户的 shell 程序： root:x: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、pwd：当前工作目录。 SECONDS：当前 shell 开始后所流逝的秒数	<p>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' 别名 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 为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/leafior -Ctrl-C：终止当前的命令 或 程序 -Ctrl-D：结束输入，或退出 linux 系统，或从上层 shell 返回 -Ctrl-Z：暂停当前命令执行</p> <p>文本编辑器 vi filename(wm filename) 命令模式+插入模式 vi filename 模式： 退出：ESC 或者 Ctrl+q 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</p> <p>文件系统结构 主目录：(登录目录) echo \$HOME /home/leafior 当前工作目录 “.” 目录本身；“.” 父目录 绝对路径 (/usr/src) 相对路径 (~/*course...)</p> <p>文件系统结构 mount [-t fstype] [-o options] /dev/xy/N dirmame 如：mount -t vfat /dev/hda1 /mnt/c 类型 设备文件 挂在目录 挂载 U 盘：mount -t vfat /dev/sda1 /mnt/usb fstype: iso9660/vfat/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 yy：同种设备的顺序，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/fdo (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 /doss 卸载</p> <p>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显示 -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，则以下的文件亦依序列出 ls -l [文件列表] 替换命令 PS1='cmd' 创建命令间的管道 cmd1 cmd2 分割脚本执行的命令 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 \ 历史动作/记录列表中的命令和当前命令 ll,ls \ TC shell 的提示符，或者指定一个任务号/时作为起始符 % 或者 %3 如：\$ls [0-9][a-zA-Z].html</p> <p>常用命令 gcc [options] [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 打开编译警告 (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 passwd; info passwd; ls -help [more]; man -52 open; man 2 open</p> <p>whatis：得到任何 UNIX 命令的更短的描述 whoomis：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</p>
whatis ：得到任何 UNIX 命令的更短的描述	
whoomis ：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	
whatis ：得到任何 UNIX 命令的更短的描述	
whoomis ：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	

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 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>。
-nN 若 N 前加 “+” 号表示显示从文件第 N 行开始的所有行；
显示显示文件的最后 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 物 leor 移动文件
mv [option] file-list directory 把文件列表中所有文件移到目录 d
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]
如果输入 -d 替代 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
-f 快速比较 (此时不得使用 -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。
L1a1,L21,3：在 F1 第 L1 行后添加 F2 的 L2 行到 L3 行
L1,L2d1,L4：把 file1 中的 L1 行到 L2 行替换成 file2 中的 L3 行到 L4 行
L1,L2d1,3：删除 file1 中的 L1 行到 L2 行
zdiff 和 zcmp 可以用来自比较两个压缩文件
unq [options] [+N] [input-file] [output-file]
删除已经排好的文件 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
lpstat 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 开头的 string
x\$ 以 x 结束的 string
^* *

(xy)+ 或 (xy)+ xy,xyxy,xyxyxy, ...
xy? x, xy
xy* x, xy, xyxy, xyxyy
xy+ xy, xyxy, xyxyy...
[.][*] [/Hh]ello /[/^A-KM-Z]love/ 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。
-f 将文件附加在备份文件后面。
-r filename 用 archive 作为存档或恢复文件的备份文件名；默认是 /dev/mto。如果 archive 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)。
-u 把把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 -xvf 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。
-i 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。
-p rpm-filename 删除指定的软件包。
-f rpm-filename 更新指定的软件包。
-l rpm-filename 安装指定的软件包。
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，如果它们都没有排好序，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 4 -r -b -s students 姓氏第 2 个字段开始作为比较键值
Sort -k 4 -r -b -s students 用用户作为比较键值，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
ln 执行历史命令列表中的第 n 个命令
ln -n 当前前的第 n 个命令
lstrng 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 权限等如 oct1 八进制数字，如 777) 的文件
-size size [N]C 搜索文件大小为 N 的文件
-type [t]c 搜索文件类型为 N 的文件
-user name 搜索所有权为 name 的文件
\{expr\} 当表达式为真结果时：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 a 标明需要匹配的表达式，在 \() 和 \< 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 只搜索源代码文件
-s 只搜索帮助文件
grep [选项] [模式] [文件列表] 中间 最慢，最灵活
egrep [选项] [字符集] [文件列表]
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 仅输出匹配的行数
-i 在匹配的过程中忽略字母的大小写
-l 仅输出有匹配文件的文件名
-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 -A / /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 命令来把目录更改成当前目录。
usr(u) / group(g) / other(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 [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：写权限 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
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 0 的位。
文件访问权限 = 默认访问权限 - 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 本文文件为 666
通过文件链接关系
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：指向文件索引节点的指针，inode 相同；不可跨越文件系统；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：代码本身是 shell 进程的一部分。· alias、bg、cd、continue、echo、exec、exit、fg、jobs、pwd、sd、set、shift、test、time、umask、unset 和 wait
外部命令：文件内容可以是以二进位代码或者 shell 脚本。通常使用的一些外部命令如 grep、more、cat、mkdir、rm、ls、sort、ftp、telnet、lp 和 ps
Shell 执行二进位命令：
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 执行脚本命令：
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
-c 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
-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
字符串输入：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+L \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
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
命令 ps aux 查看所有进程，包括守护进程

top：实时监控 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式地控制这些进程
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 命令可以挂起前台 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运行系统管理任务。
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 (在当前的 shell 的子 shell 中)
\$(shell:echo Hello,World)
Ctrl-C 终止前台进程
命令的条件执行
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 “%” 符号开始。
kill -1 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 号信号或者其他信号的进程，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 -s pstree -a | more
ulimit -u Bash 下显示用户可同时执行的最大进程数