

shell广义上分为两类：

GUI：包括GNOME KDE XFACE等

GLI：sh csh ksh bash等(Linux

发行版本中，bash是默认使用的)

shell启动：当登陆后，系统会自动开启shell程序

进程：应用程序的副本，使用PID区分（在系统中，一个进程只认为自己存在）

shell：父shell程序和子shell程序（父子shell之间设置环境相互独立）

cat /etc shells：查看当前系统所支持的shell程序

yum -y install ksh csh：安装ksh、csh

exit：退出当前shell

pstree：查看进程树

bash的特性：

1、命令历史

history：查看linux之前执行的命令历史

-w：将命令历史保存到某个文件中

变量：PATH命令路径变量

HISTSIZE：命令历史大小变量

#echo \$HISTSIZE：查看HISTSIZE值

! n：执行命令历史编号的命令

! -n：倒数几行命令

! word：执行最近一次以word开始的命令

2、管道，重定向

计算机体系结构：

控制器：CPU，读取系统指令

运算器：CPU，运算

存储器：RAM（内部存储器：易失性）

输入输出设备：键盘，硬盘等

计算机总线：

地址总线：负责内存寻址

数据总线：负责传输数据

控制总线：负责控制指令

寄存器：CPU中内部临时存储空间

I/O设备：负责计算机内部存储设备和外部设备交互的设备（光盘，硬盘等）

程序：指令+数据

指令：有程序提供，负责加工数据

数据：数据可以有多种来源，如：来至变量，输入设备，文件等

三种数据流：

标准输入数据流，文件描述符为0

标准输出数据流，文件描述符为1

标准错误输出数据流，文件描述符为2

当在Linux中打开一个文件时，内核会反复调用，对于文件标识就很重要了，用文件描述符来标识文件，文件加载完成用数字标识

IO重定向：把输入输出数据来源，重新定向到其他的文件或设备

输出重定向：

>：覆盖重定向

>>：添加重定向

2>：错误覆盖输出重定向

2>>：错误添加输出重定向

&>：覆盖混合输出重定向

&>>：添加混合重定向

输入重定向：

<：输入重定向

管道：连接多条命令，把前一条命令的结果作为后一条命令的输入条件

commd1 | commd2 |

tr：实现字符转换，不修改源文件

语法：tr [command] [command]

例：tr 'a-z' 'A-Z'

tee：可以实现屏幕显示并保存到文件中

例：ls /usr/ | tee usr.out

ls /usr/ | tee usr.out | tr 'a-z' 'A-Z'

3、命令别名

alias：查看定义的别名

将 cd /etc/sysconfig/network-scripts/ 取个别名进行引用：alias cdnet='cd /etc/sysconfig/network-scripts/'

unalias command_alias：取消别名

使用：cdnet

4、命令行编辑

ctrl+a：快速跳转到命令行的行首

ctrl+e：快速到命令行的行尾

ctrl+u: 快速删除光标位置到命令行行首的字符

ctrl+k: 快速删除光标位置到命令行行尾的字符

ctrl+l: 清屏 (命令clear)

5、命令行展开

命令补齐: 在PATH变量中搜索命令补齐 (PATH变量必须正常, 输入要唯一, tab补齐)

路径和文件补齐:

5.2: 命令替换

就是将命令中的子命令替换成子命令执行结果的过程

echo "Dirctory is \$(pwd)." 其中\$(pwd)显示当前命令

6、命令行通配

*: 匹配任意长度的任意字符

? : 匹配任意单个字符

[]: 匹配指定范围内的任意单个字符

[^]: 匹配指定范围外的任意单个字符

[:space:]: 表示空格

[:punct:]: 表示所有的标点符号

[:lower:]: 表示所有的小写字母

[:upper:]: 表示所有大写字母

[:alpha:]: 表示所有字母

[:digit:]: 表示所有数字

[:alnum:]: 表示所有数字和字母

7、变量

8、编程