1. Shell是一个命令解释器；
2. Shell还是一个功能相当强大的编程语言，易编写，易调试，灵活性较强。Shell是解释执行的脚本语言，在shell中可以直接调用Linux系统命令。

Shell的分类：

1. Bourne Shell ：从1979起unix就开始使用Bourne Shell，Bourne Shell的主文件名为sh。

Sh, ksh, Bash, psh, zsh

1. C Shell: C Shell主要在BSD版的unix系统中使用，其语法和C语言相类似而得名。

csh, tcsh

Bash:Bash与sh兼容

用alias命令的别名只能临时生效，重启之后就会没有，只有写入相应配置文件；让别名永久生效vi /root/.bashrc

正确输出和错误输出同时保存：

命令 >(>>) 文件 2>&1:覆盖和追加

命令 &>(&>>) 文件：同上

命令 >> 文件1 2>> 文件2：把正确的输出追加到文件1，把错误的输出追加到文件2中。

输入重定向：

wc [选项] [文件名]

-c 统计字节数

-w 统计单词数

-l 统计行数

多命令顺序执行：

； ：顺序执行，之间没有关系

&& ：顺序执行，只有前面命令都执行正确，后面命令才能执行

|| ：或的作用

命令 && echo yes || echo no:当命令正确执行后，逻辑与后面的echo yes会执行，逻辑或看见前面的命令正确执行，逻辑或的命令就不会执行；反之也是类似的。

管道符：

命令1 | 命令2

#命令1的正确输出作为命令2的操作对象

grep [选项] “搜索内容” 文件名

选项：

-i :忽略大小写

-n:输出行号

-v:反向查找

--color=auto:搜索出的关键字用颜色显示

通配符：

？：匹配一个任意字符

\*：匹配任意多个字符（也可以没有字符）

[]：匹配中括号中任意一个字符

[-]：匹配中括号中间的任意一个

[^]:逻辑非

变量分类：（set命令可以查看系统中所有环境变量）

用户自定义变量；：（也称本地变量）

环境变量；：export 变量名=变量值#声明变量

env#查询变量

unset 变量名#删除变量

位置参数变量；：位置环境变量是预定义变量中的一种，只是因为很多，所以单独拿出来，也就是说这两种变量的限制是一样的。

$n: n为数字，$0代表命令本身，$1-$9代表第一到第九个参数，十以上的参数需要用大括号包含，如${10}.

$\*: 这个变量代表命令行中所有的参数，$\*把所有的参数看出一个整体。

$@: 这个变量也代表命令行中的所有的参数，不过$@把每个参数区分别对待。

$#: 这个变量代表命令行中所有参数的个数。

预定义变量；

$?: 最后一次执行的命令的返回状态。如果这变量的值为0，证明上一个命令正确执行；如果这个变量的值为非0（具体是哪个数，由命令自己来决定），则证明上一个执行不正确了。（第一次使用）

$$: 当前进程的进程号（PID）

$!: 后台运行的最后一个进程的进程号（PID）

（命令 &）让一个命令放入后台执行。

PS1：定义系统提示符的变量

\d:显示日期，格式为“星期 月 日”

\h:

\t:

\T:

\A:

\u:

\w:

\W:

\#:

\$:

1. declear声明变量类型

declear [+/-][选项] 变量名

-：给变量设置类型属性

+：取消变量的类型属性

-i: 将变量声明为整数型

-x: 将变量声明为环境变量

-p: 显示指定变量的被声明的类型

2. expr或者let数值运算工具

aa=11

bb=22

dd=$(expr $aa + $bb):注意“+”号左右两侧必须有空格

1. “$((运算式))或”“$[运算式]”

aa=11

bb=22

ff=$(($aa+$bb))

gg=$[$aa+$bb]

bash的运算符（能看懂和查询就行）

环境变量的配置文件：

source 配置文件 或者 .配置文件（如果修改了配置文件的内容，必须重新登陆才能生效，要想现在生效，此命令强制让配置文件生效。）

默认环境变量：PATH，HISTSIZE,PS1,HOSTNAME等

/etc/profile#对所有的用户都是一样的

/etc/profile.d/\*.sh#对所有的用户都是一样的

~/.bash\_profile#对用户自己都是一样的

~/.bashrc#对用户自己都是一样的

/etc/bashrc#对所有的用户都是一样的