###### bash环境--批处理脚本：

1. shell：外壳；
2. 解释器（提供了编程环境）：解释——词法分析（token），语法分析，语义分析；条件判断：选择分支；循环；
3. 过程式编程语言：顺序执行，选择执行，循环执行；
4. 脚本：使用bash提供的控制机制，将期望执行的命令罗列于文本文件中；bash没有库机制，整个程序几乎都要依赖于系统命令来完成；
5. 解释型vs编译型：
   1. 编译型：源程序（文本文件）-->预处理-->编译-->汇编-->链接；大多是静态语言
   2. 解释型：源程序（文本文件）-->解释器读取并执行；大多是动态语言，无法事先处理一些语法错误

###### Bash的重要特性：

1. 变量类别：
2. 本地变量：
   1. 只对当前shell进程有效的变量；对其他shell进程无效，包括当前shell进程的子进程；
   2. 进程的父子关系：init🡪shell🡪bash；查看进程树：export LANG=en pstree
   3. VAR\_NAME=VALUE——声明变量并赋值（向变量的存储空间保存数据）；
   4. ${VAR\_NAME}——引用变量：{}可去掉；前面加echo可打印；双引号—弱引用，里面的变量会被替换；单引用—里面的所有字符都是字面量，直接输出；
3. 环境变量：
4. 定义变量：export VAR\_NAME=value
5. 导出变量：export VAR\_NAME
6. 撤销变量：unset VAR\_NAME
7. 只读变量：readonly VAR\_NAME，使shell变量不可更改，不能再赋值或撤销
8. 对当前shell进程及其子shell有效，对其他shell进程无效；
9. 用户可自定义环境变量；bash有许多内置的环境变量，用户自定义会覆盖
10. 局部变量：
11. 对shell脚本中某代码片段有效，通常用于函数本地
12. local VAR\_NAME=VALUE
13. 位置变量：
14. $1,$2,…,${10},…
15. 特殊变量：
16. $?；
17. 查看当前shell进程的所有变量：set；
18. 查看当前shell进程的所有环境变量：export，print，env；
19. 变量命名：
20. 不能使用程序中的关键字（保留字）：if，case，for
21. 只能使用数字、字母和下划线且不能以数字开头；
22. 要见名知意（区分大小写）：userName；
23. 变量类型：
    1. 声明类型会决定存储机制、存储空间和参与的运算类型：
    2. 数值型：
       1. 精确数值型（整数）：
       2. 近似数值型（浮点型）：单精度浮点vs双精度浮点
    3. 字符型：char、string
    4. 布尔型：true，false
    5. 日期时间：
    6. 类型转换：显式转换、隐式转换
24. Bash是弱类型的语言：一切皆字符

###### bash的配置文件：

1. Profile类：为交互式登录的用户提供配置
   1. 全局配置——/etc/profile：基本的配置文件片段；/etc/profile.d/\*.sh：其他的配置文件的片段；
   2. 用户配置——~/.bash\_profile
   3. 功用：设定环境变量，运行命令或脚本
2. Bashrc类：为非交互书的用户提供配置
   1. 全局：/ect/bahsrc
   2. 用户：~/.bashrc
   3. 功用：设定本地变量，定义命令别名
3. 登录类型：
   1. 交互式：直接通过终端输入账号和密码登录,即使用su -l USERNAME 或 su - USERNAME
   2. 非交互式：su USERNAME；图形界面下打开的终端（伪终端）；执行脚本
4. 通过编辑配置文件修改的配置使其生效：
   1. 退出并重新登录
   2. 让bash重读此配置文件：. FILE 或 source FILE
5. 读取配置文件的顺序，后读取的生效：
   1. 交互登录的用户：/etc/profile 🡪 /etc/profile.s/\*.sh 🡪 ~/.bash\_profile 🡪 ~/.bashrc 🡪 /etc/bashrc
   2. 非交互式登录的用户：~/.bashrc 🡪 /etc/bashrc 🡪 /etc/profile.d/\*.sh