

SHELL 模块要点总结

1. 什么是 Shell?

- 在 Linux 内核与用户之间的解释器程序，通常指/bin/bash
- 负责向内核翻译及传达用户/程序的指令
- 相当于操作系统的“外壳”

2. Shell 有哪些使用方式

--交互式：命令行

- 人工干预/智能化程度高
- 逐条执行，效率低

--非交互式：脚本

- 需要提前设计，智能化难度高
- 批量执行，效率高
- 方便在后台静悄悄执行

3. 自定义 shell 变量时，变量名有什么规则？

- 可以包括数字、字母、下划线，不能以数字开头
- 变量名区分大小写
- 赋值时等号两边不能有空格
- 尽量不要使用关键字和特殊字符
- 给同一个变量多次赋值时，最后一次的赋值生效

4. 变量的生命周期，和作用范围是？

- 在脚本里定义的变量，只有在脚本执行过程中有效
- 默认情况下，只在当前 shell 里可以使用，想要定义的 shell 在所有的 shell 里被使用，就要把变量定义为全局变量

5. 脚本的构成

- #！脚本声明
- #注释信息
- 可执行语句

6. 如何查看当前系统支持的 shell？（常见的 shell 程序种类）

查看 cat /etc/shell

常见种类：

```
/bin/sh      //Unix 默认的 shell
/bin/bash    //Linux 默认使用的 shell
/bin/nologin //非登录 shell
/bin/tcsh
/bin/csh
/bin/ksh
```

7. 查看变量、取消变量的命令

echo \$变量

unset \$变量

8. shell 变量类型,常用的预定义变量有那几个？

1)

- 系统环境变量:变量名通常都大写，由系统维护用来设置工作环境，只有个别变量用户可以直接更改
- 预定义变量:bash 内置，有特殊用途的变量，可直接调用，但不能直接赋值或修改

--位置变量: bash 内置，存储执行脚本时提供的参数

--自定义变量: 用户自主设置，修改及使用

2) \$? \$# \$* \$\$

9. 简述预定义变量\$、\$?、\$0、\$#、\$*、\$!的作用

\$: 当前运行进程的 PID 号

?: 命令执行结果的返回状态，0 表示正常，1 或其他表示异常

\$0: 当前运行的进程号或脚本号

\$#: 已加载的位置变量的个数

\$*: 所有位置变量的值

!: 后台最后一个进程的 PID 号

10. 请列出整数运算的三种运算方式

-- expr 表达式运算

-- \$[] 或 \$(()) 表达式运算

-- let 命令运算

11. 请分别说出 let 命令中“i++, i--, i+=2, i-=2, i*=2, i/=2”的含义

++ 自加一

-- 自减一

i+=2 i=i+2

i-=2 i=i-2

i*=2 i=i*2

i/=2 i=i/2

12. 说出“echo “scale=2;2/7” |bc” 的运算结果

0.28

解析：“scale=2”表示显示两位小数点(不四舍五入)，将 echo 输出的结果管道给 bc 计算器计算

13. 请说出字符串截取的三种方法及格式

方法一：使用 \${} 表达式

格式：\${var:起始长度:长度}

: 冒号隔开 从 0 开始算第一位

方法二：expr substr

格式：expr substr "\$var" 起始位置 长度

起始编号从 1 开始（第一种是从 0 开始）

方法三：使用 cut 分割工具

格式：echo \$var | cut -b 起始位置 结束位置

起始编号从 1 开始

“-” 为连续的

“,” 为不连续的

14. shell 上：0、1>、2>、>、2>&1、&>/dev/null 分别代表什么意思？

0 表示标准输入

1> 表示标准输出

2> 表示标准错误输出

> 默认为标准输出重定向，与 1> 相同

2>&1 意思是把标准错误输出，重定向到标准输出

&>/dev/null 意思是把标准输出和标准错误输出都重定向到“黑洞”

15. 用户配置文件有哪几个？分别的作用是什么？

~/.bashrc ~/.bash_profile

可以在里面定义变量，用户每打开一个终端时加载的文件，只针对当前用户有效

16. 系统配置文件有哪些？分别的作用是什么？

/etc/profile

/etc/bashrc

可以在里面定义变量，用户每打开一个终端时加载的文件，针对所有用户生效

17. 简述三种定界符在变量赋值操作中的特点

双引号 ""：允许扩展，以\$引用其他变量

单引号 ''：禁用扩展，即使\$也是为普通字符

反撇号 ``：将命令的执行输出作为变量值

18. shell 中单引号和双引号和不加引号分别代表什么意思？

--单引号：可以说是所见即所得，即将单引号的内容原样输出

--双引号：把双引号的内容输出，如果内容中有命令、变量等，会先把变量、命令解析出结果，然后再输出最终内容来

--无引号：把内容输出来，会将含有空格的字符串视为一个整体输出，如果内容中有命令、变量等，会先把变量、命令解析出结果，然后再输出内容来，如果字符串中带有空格等特殊字符，则不能完整地输出，需要改加双引号，一般连续的字符串，数字、路径等可以不加任何引号，不过最好用双引号括起来。

19. shell 中 export 的作用

设置或显示环境变量，将局部变量声明为全局变量

-f 代表[变量名称]中为函数名称。

-n 删除指定的变量。变量实际上并未删除，只是不会输出到后续指令的执行环境中。

-p 列出所有的 shell 赋予程序的环境变量。

20. shell 条件判断类型有哪些？

--文件状态

--数值比较

--字符比较

--逻辑比较

21. 列出常见的证书值比较操作，并说明各自作用

-eq 等于

-ne 不等于

-ge 大于或者等于

-le 小于或者等于

-gt 大于

-lt 小于

22. 各操作在 shell 中分别代表什么？

-n 非空为真

-z 字符串为空为真

-d 是目录且存在为真

-f 是普通文件且存在为真

-e 存在为真

-r 可读为真

-w 可写为真

-x 可执行为真

-l 是链接为真

!-z 不为空

23. 递归死循环(迅速耗尽系统资源)

.(){.|&};

24. if 循环分哪几种，它们的特点及语法结构分别是？

- 单分支（只能判断对的）

if [判断];then

命令

fi

- 双分支（只判断对与错，二选一）

if [判断];then

命令

else

命令

fi

- 多分支（多个判断及选择）

if [判断];then

命令

elif [判断];then

命令

elif [判断];then

命令

else

命令

fi

25. case 语句 的语法结构（简单的 if）

case 变量值 in

判断 1)

命令;;

判断 2)

命令;;

判断 3)

命令;;

else

命令;;

esac

26. While 循环的语法结构，及其死循环语法结构是？

while [判断] //为真，执行命令

do

命令

done

死循环：

while :

do

命令

done

27. for 循环的语法结构 (可镶嵌多个 if 语句)

```
for 变量 in 赋值
do
    命令
done
```

28. 简述 shell 环境常见的中断及推出控制指令

break:跳出当前所有的循环体，执行循环体后的语句

continue:跳过循环体内余下的语句，重新判断条件以执行下一次循环

exit:退出脚本，默认返回值是 0

(拓展)return:用在函数里,指返回值

(拓展)shift:用于迁移位置变量，将\$1~\$9 依次向左顺序移动

29. 说出“ A ; B ”,“ A && B ”, “A || B ” 的执行情况

“ A ; B ” 执行 A 后继续执行 B

“ A && B ” 执行 A 成功后再执行 B，A 不成功，B 不执行

“ A || B ” 执行 A 成功则不执行 B，A 不成功才执行 B

30. 在命令行输入'seq 1 2 5'输出结果是什么？

```
1
3
5
```

解析：初始值是 1，自增步长是 2，最大值是 5

31. dirname 和 basename 有什么作用？

dirname 获取脚本路径名

basename 获取脚本名

32. 什么是 expect 预期交互？触发依据是？有哪些实例？

--基于 TCL 编写的自动交互式程序，可以用在 shell 脚本中，为交互式过程自动输送预先准备的文本或指令，而无需人工干预

--触发依据是预期出现的特征提示文本

--实例:实现 SSH 自动登录，并远程执行指令

33. 命令行显示|不显示内容的命令

stty echo 显示

stty -echo 不显示

34. 正则表达式中的 ^、\$、.、+、?、* 分别表示什么含义？

^ 匹配行首

\$ 匹配行尾

. 单个字符

限制关键词的匹配次数，含义分别如下：

+：最少匹配一次，比如 a+ 可匹配 a、aa、aaa 等

?：最多匹配一次，比如 a? 可匹配零个或者一个 a

：匹配任意多次，比如 a 可匹配零个或者任意多个连续的 a

35. 什么是 sed？有什么特点？

sed 是一种可实现对文本的输出、删除、替换、复制、剪切、导入、导出等各种操作流式编辑器

特点：

-非交互

-逐行处理，默认不修改源文件

36. sed 常见命令选项有哪几个？基本处理动作有哪些

常见命令选项

- n 屏蔽默认输出（全部文本）
- i 直接修改文件内容
- r 启用扩展的正则表达式，若与其他选项一起用，应作为首个选项
- {} 可组合多个命令，以分号分隔

基本处理动作

- p (print) 输出
- d (delete) 删除
- s (substitute) 字符串替换

37. 正则表达式的“|”“\n”“\r”分别代表什么意思

“|”表示“或”的意思

“\n”表示换行

“\r”表示回车

38. echo -n 是什么意思？

输出内容不换行

39. awk 工具中“-F”“-v”分别有什么作用？

-F:指定分隔符

-v:调用外部 Shell 变量

40. 简述 awk 工具常用的内置变量、各自的作用

FS 保存或设置字段分隔符

\$n 指定分隔的第 N 个字段，即\$1/\$2/\$3

\$0 保存当前读入的整行文本内容

NF 记录当前处理行的字段个数（列数）

NR 记录当前已读入行的数量（行数）

FNR 保存当前处理行在原文本内的序号（行号）

（扩展）FILENAME 保存 awk 当前处理的文件名

（扩展）ENVIRON 调用 shell 环境变量 格式：ENVIRON["变量名"]

41. \$@和\$*的区别

\$@将命令行每个参数视为单独的字符串，等同用于"\$1""\$2""\$3"

\$*将所有的命令行所有参数视为单个字符串，等同于"\$1\$2\$3"