

1 如何执行一个未设置x权限的Shell脚本？

参考答案

以下三种方式任选其一：

- sh 脚本文件路径
- source 脚本文件路径
- . 脚本文件路径

2 自定义Shell变量时，变量名有什么规则？

参考答案

- 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
- 赋值时等号两边不要有空格
- 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

3 编写一个Shell脚本程序，用来报告当前用户的环境信息。

新建脚本文件report.sh，执行后能够输出当前的主机名、登录用户名、所在的文件夹路径。效果如下所示：

当用户root执行的时候：

01. [root@svr5 ~] # /opt/report.sh
02. 当前的主机名是：svr5.tarena.com
03. 登录用户是：root
04. 当前位于 /root 文件夹下

当普通用户mike执行的时候：

01. [mike@svr5 ~] \$ /opt/report.sh
02. 当前的主机名是：svr5.tarena.com
03. 登录用户是：mike
04. 当前位于 /home/mike 文件夹下

参考答案

```

01. [ root@svr5 ~] # cat /opt/report.sh
02.  #!/bin/bash
03.  echo "当前的主机名是：$HOSTNAME"
04.  echo "登录用户是：$LOGNAME"
05.  echo "当前位于 $PWD 文件夹下"
06.  [ root@svr5 ~] # chmod +x /opt/report.sh

```

4 简述预定义变量\$\$、\$?、\$0、\$#、\$*、\$!的作用。

参考答案

- \$\$ 保存当前运行进程的（PID）号
- \$? 保存命令执行结果的（返回状态）
- \$0 保存当前运行的（进程名）或（脚本名）
- \$# 保存位置变量的（个数）
- \$* 保存所有位置变量的（值）
- \$! 保存后台（最后一个进程）的PID号

5 编写一个Shell脚本程序，能够快速添加用户账号。

新建脚本文件quick_uad.sh，执行时指定新用户名作为参数，能够快速添加对应的用户账号，并将用户的登录密码设为123456。效果如下所示：

```

01. [ root@svr5 ~] # ./quick_uad.sh xixi //添加用户xixi
02. 更改用户 xixi 的密码。
03.  passwd：所有的身份验证令牌已经成功更新。
04. [ root@svr5 ~] # id xixi
05.  uid=525(xixi) gid=525(xixi) 组=525(xixi)
06.
07. [ root@svr5 ~] # ./quick_uad.sh haha //添加用户haha
08. 更改用户 haha 的密码。
09.  passwd：所有的身份验证令牌已经成功更新。
10. [ root@svr5 ~] # id haha
11.  uid=526(haha) gid=526(haha) 组=526(haha)

```

参考答案

```
01. [ root@svr5 ~] # vim quick_uad.sh
02.  #! /bin/bash
03.  useradd $1
04.  echo 1234567 | passwd --stdin $1
05.
06. [ root@svr5 ~] # chmod +x quick_uad.sh
```

6 简述三种定界符在变量赋值操作中的特点。

参考答案

- 双引号 " "：允许扩展，以 \$ 引用其他变量
- 单引号 ' '：禁用扩展，即便 \$ 也视为普通字符
- 反撇号 ``：将命令的执行输出作为变量值