

Shell的编程过程

在Linux中，Shell不但能与用户进行逐个命令交互执行，也可以通过脚本语言进行编程。通过对Shell进行编程，可使大量任务自动化，对于**系统管理、系统维护**方面特别有用。本章将介绍Shell编程的入门知识，主要包括以下知识点：

- 了解使用Shell编程的过程。
- 了解Shell的变量定义、输入/输出方法。
- 掌握Shell进行条件测试的方法。
- 掌握脚本程序中流程控制的语句。
- 掌握函数定义和参数处理方法。
- 掌握Shell程序的调试方法。

复习与Shell相关的命令

1. 现在无论使用哪种Linux操作系统，**默认的Shell总是/bin/bash。**
2. 查看系统中SHELL变量的值：

```
echo $SHELL
```

3. 查看系统内置了哪些Shell，有以下两种方法：

```
① cat /etc/shells
```

```
② chsh -list
```

1. 创建Shell程序

与MS-DOS系统中的批处理文件类似，Linux中的脚本（Shell程序）是一个文本文件，包含各类Linux的Shell命令。通过脚本文件将这些命令汇集在一起，

可连续执行shell命令。

举例：编写一个最简单的“Hello World!”程序。

步骤：

(1) 创建一个用于保存文件的目录：`mkdir /usr/program`

(2) 在program目录下新建一个文本程序first。

(3) 打开first文件，输入Shell的脚本。

```
1:  #!/bin/bash
```

```
2:  var= "Hello World!"
```

```
3:  echo $var
```

说明：

第1行告诉系统，该文件后面的代码将用/bin/bash来执行。这是一般Shell程序第1行中必须包含的内容。

第2行为变量var定义了一个字符串值。

第3行将变量var的值显示输出到终端。

2. 设置执行权限

创建好first文件后，使用`ls -l`命令可以看到文件的所有者（这里是root）对该文件只有读和写的权限，**没有执行权限**。

执行`chmod u+x /usr/program/first`为脚本文件设置所有者的执行权限。当然，也可以执行`chmod a+x /usr/program/first`设置所有用户都可以运行该脚本程序。

注意：与DOS或Windows中的批处理程序不同，Linux中的脚本文件不是按扩展名来识别，而是通过**为文件添加执行权限来允许其执行**，用户可以为脚本文件

设置任意扩展名。

3. 执行Shell程序

两种方法：

(1) 使用Shell程序执行

```
#bash /usr/program/first
```

使用这种方法时，不需要设置脚本程序的可执行权限。

(2) 直接执行

```
#/usr/program/first
```

使用这种方法时，需首先设置该文件的执行权限。

如果执行当前目录下的脚本文件，通常也应该使用一个点表示当前目录，如执行当前目录下的脚本文件first，可使用命令：**# ./first**