第十三章　学习 Shell Scripts

1、shell script 程序化脚本

shell script 是针对 shell 所写的脚本，也是批处理文件

这个程序是使用纯文本文件，将一些 shell 的语法和指令(含外部指令)写在里面，搭配正则表示法、管线命令与数据流重导向等功能，以达到我们所想要的处理目的。

2、shell script 速度比较慢，较少用在处理大量数值运算上，通常用于处理服务器的侦测而不是大量运算

3、由于 shell script 是纯文本文件，所以可以用 vim 编写

那如何执行呢，以/home/dmtsai/shell.sh 为例？

（1）直接指令下达：shell.sh 必须有可读和可执行(rx)权限，直接 /home/dmtsai/shell.sh

（2）用 bash 程序执行 bash /home/dmtsai/shell.sh

4、开始编写 script

```
#!bin/bash
PATH=/bin:/sbin:/usr/bin:/usr/sbin:/usr/local/bin:/usr/loca
export PATH
echo -e "Hello World!\a\n"
exit 0
```

```
[root@localhost scripts]# vi sh01.sh
[root@localhost scripts]# ll
总用量 4
-rw-r--r--. 1 root root 129 3月  29 00:07 sh01.sh
```

竟然是没有 x，就是说没有执行权限的

解析程序

（1）第一行 #!/bin/bash 在宣告这个 script 使用的 shell 名称是 bash

（2）第二行开始的那些#是指注释

（3）最好将一些重要的环境变量设置好，比如 PATH，方便直接下达一些外部指令

（4）exit 0 代表离开 script 并回传一个 0 给系统，$?=0，系统就知道 shell script 执行成功了

执行 script

```
[root@localhost scripts]# bash sh01.sh
Hello World!
```

还会发出声音，这是因为参数-e

如果把-e 去掉，就会是这样

```
[root@localhost scripts]# sh sh01.sh
Hello World!\a\n
```

原来-e 是必须的...

\a 好像没什么用...

5、获取用户输入

```
read -p "Please input your first name:" firstname
read -p "Please input your last name:" lastname
echo -e "\nYour full name is:$firstname $lastname"
exit 0
~
~
```

```
[root@localhost scripts]# sh sh02.sh
Please input your first name:zhong
Please input your last name:zhanhui

Your full name is:zhong zhanhui
```

6、随日期发化：利用 date 进行文件的建立

```
[root@localhost scripts]# ls
s1.sh  sh01.sh  sh02.sh  sh03.sh
[root@localhost scripts]# sh sh03.sh
I will use 'touch' command to create 3 files.
Please input your filename:date
[root@localhost scripts]# ls
date20180405  date20180406  date20180407  s1.sh  sh01.sh  sh02.sh  sh03.sh
```

```
echo -e "I will use 'touch' command to create 3 files."
read -p "Please input your filename:" fileuser

filename=${fileuser:-"filename"}

date1=$(date --date='2 days ago' +%Y%m%d)       #the day before yesterday
date2=$(date --date='1 days ago' +%Y%m%d)       #yesterday
date3=$(date +%Y%m%d)                           #today
file1=${filename}${date1}
file2=${filename}${date2}
file3=${filename}${date3}

touch "$file1"
touch "$file2"
touch "$file3"
~
```

实话说，两个星期没回顾 linux，都忘光了这些符号的意义了
连语句是什么意思都不懂

7、数值计算