

博客园

首页

新随笔

联系

管理

订阅

XML

随笔- 745 文章- 1 评论- 60

十分钟学会写shell脚本

大家好！我是handsomecui，下面我为大家讲解一下shell脚本的写法，讲的不好的地方，欢迎大家留言拍砖。

1.在linux下会写shell脚本是非常重要的，下面我参照例子给大家展示几个脚本，顺带这学习shell 的语法：

什么时候helloworld是必不可少的，第一个脚本肯定与helloworld是离不开的：

```
#!/bin/sh
a="hello world!"
num=2
echo "a is : $a num is : ${num}nd"
```

运行结果： a is : hello world! num is : 2nd

shell脚本简单易懂，只要有一点点c语言的基础，这些都是小case；

2.下面带着大家学学if语句：

先抛出一个问题：写一个脚本，判断当前所用的shell

```
#!/bin/sh
#注意if的空格以及[]里面的空格这里错了半天啊
if [ "$SHELL" = "/bin/bash" ];then
echo "your login shell is the bash \n"
echo "SHELL is : $SHELL"
else
echo "your login shell is not bash but $SHELL"
fi
#[ -f "somefile" ] : 判断是否是一个文件
#[ -x "/bin/ls" ] : 判断/bin/ls是否存在并有可执行权限
#{ -n "$var" } : 判断$var变量是否有值
#[ "$a" = "$b" ] : 判断$a和$b是否相等
[ -f "/etc/shadow" ] && echo "This computer uses shadow passwords"
if [ -f "/etc/shadow" ];then
echo "电脑密码使用隐藏字符"
else
echo "并没有"
fi
```

#是注释的意思，相关部分已经在里面详细给大家注释解释了，注意if语句的格式

if[];then

elif[];then

else

fi

另外注意if语句还可以简单的用&&与||或语句来代替;另外如果想要深入了解if可以找男人test(在linux遇到问题要学会找男人解决，man)

3.为了让大家更加熟练的运用shell，那么练习下吧，问题：写一个脚本使其从一个文件里面读入有echo的语句，并把其写在本文件末尾；

```
#!/bin/sh
#-r代表此文件是否可读，具体见man test
mailfolder=/home/handsome/work/linux_learn/shell_learn/readme.txt
#[ -r "$mailfolder" ] || { echo "Can not read $mailfolder" ; exit 1;}
```

昵称: handsomecui

园龄: 2年7个月

粉丝: 54

关注: 12

+加关注

<

2017年12月

>

日	一	二	三	四	五	六
26	27	28	29	30	1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30
31	1	2	3	4	5	6

搜索

找我看

谷歌搜索

常用链接

我的随笔

我的评论

我的参与

最新评论

我的标签

随笔分类(795)

"Java编程"(26)

"Linux C编程"(5)

"设计模式"(2)

\$\$\$数据库(5)

\$\$\$数字图像(1)

***hash表(1)

***Kmp(12)

***STL(19)

***数据结构-单调队列(3)

...树-tire树(6)

...树-二叉树(11)

...树-线段树&&树状数组(33)

...树-最小生成树&&并差集(40)

...树-最优比率生成树(2)

...图论-二分图匹配(14)

...图论-强联通&&双联通(4)

...图论-拓扑结构(11)

...图论-网络流(5)

...图论-最短路(17)

。。。01分数规划(4)

。。。DP(61)

。。。DP-区间dp(3)

。。。背包(20)

。。。大数(8)

。。。二分||三分(34)

。。。母函数(7)

。。。数论--数学(45)

。。。数论-置换群(4)

。。。数学-暴力(10)

https://www.cnblogs.com/handsomecui/p/5869361.html

1/6

```
#echo "$mailfolder has mail from : "
#grep "^echo " $mailfolder

if [ -r "$mailfolder" ];then
    echo "$mailfolder has message from : "
    echo | grep '^echo ' $mailfolder >> readme2.txt
    chmod +r readme2.txt
    cat readme2.txt >> $mailfolder
    rm -f readme2.txt
else
    echo "Can not read $mailfolder"
    touch $mailfolder
    chmod +rw $mailfolder
    echo "echo 人生自古谁无死，六区蛋清找早情！ " >> $mailfolder
fi
```



这里由于重定向不能本文见输出到本文件输出，于是我就想了个办法，新建个文件，再重定向到这个文件，大神有别的简单方法可以下面留言拍砖；

4.接下来，我们练习下while的用法，老规矩具体问题来分析；

问题：写一个脚本（不建议使用for变量--for是根据空格取值）

- 1.设定变量FILE的值为/etc/passwd
- 2.依次向/etc/passwd中的每个用户问好，并且说出对方的ID什么（Hello,root, your UID is0.)
- 3.统计一个有多少个用户

```
#!/bin/bash
fl=/etc/passwd
count=`cat $fl | wc -l`
#下面是一个管道，下面循环读文件中的每一行
cat $fl |
while read line
do
    user=`echo $line|awk -F ':' '{print $1}'`
    #代表以 ： 分段$1就是取第1段
    uid=`echo $line|awk -F ':' '{print $3}'`
    echo "hello, $user Your UID is $uid"
done
echo "====User_count:$count===="
#前面求得的用户数
```



这里要注意到awk的用法，可以找linux的男人解决；

5.练习一下for语句吧

问题：写个脚本；

- 1.切换工作目录至/tmp

- 2.依次向/tmp目录中的每个文件或子目录问好（Hello,log)

- 3.统计/tmp目录下共有多个文件，并显示出来

```
1  #!/bin/bash
2  cd /tmp
3  for i in $(ls /tmp/*)
4  do
```

- 。。。 数学-概率&期望(4)
- 。。。 数学-规律(27)
- 。。。 数学-几何(13)
- 。。。 数学-几何-凸包(4)
- 。。。 数学-技巧题(3)
- 。。。 数学-模拟(30)
- 。。。 数学-容斥(8)
- 。。。 数学-思维(50)
- 比赛(10)
- 比赛bc
- 比赛cf(6)
- 比赛蓝桥杯(8)
- 周赛(7)
- 总结。。。。。。。心
得。。。。。。。 (16)
- +++manacher(5)
- +++博弈(5)
- +++模板(1)
- +++排序(9)
- +++贪心(14)
- +++字符串(11)
- ===递归(4)
- ===搜索(69)
- >>hdoj(2)
- >>poj以及其他oj(1)
- >>水题(27)
- 归并排序(4)
- 杂--各种语言Orz(3)
- 杂文&&handsomecui的文采世界(2)

随笔档案⁽⁷⁴⁵⁾

- 2017年12月 (1)
- 2017年11月 (1)
- 2017年8月 (3)
- 2017年7月 (3)
- 2017年6月 (1)
- 2017年5月 (4)
- 2017年3月 (11)
- 2017年1月 (1)
- 2016年12月 (8)
- 2016年11月 (13)
- 2016年10月 (6)
- 2016年9月 (5)
- 2016年8月 (1)
- 2016年7月 (2)
- 2016年6月 (9)
- 2016年5月 (52)
- 2016年4月 (44)
- 2016年3月 (63)
- 2016年2月 (31)
- 2016年1月 (47)
- 2015年12月 (61)
- 2015年11月 (100)
- 2015年10月 (68)
- 2015年9月 (47)
- 2015年8月 (83)
- 2015年7月 (57)
- 2015年6月 (7)
- 2015年5月 (16)

友情连接

积分与排名

积分 - 84660
排名 - 3655

最新评论

- 1. Re:bootstrap入门基础
正在看
- 2. Re:十分钟学会写shell脚本

--VbisoWen

```
5     echo "Hello , $i"
6 done
7 count=`ls -l|grep '^-'|wc -l`
8 echo "====file_count:$count===="
```

这个问题就相对简单了些，一个for语句就可以搞定，当然可以while搞搞

6.那么问题来了：

question：

传递两个整数给脚本，让脚本分别计算并显示这两个整数的和，差，积，商

```
#!/bin/bash
a=$1
b=$2
[ -z $a ] && echo "please \"$1 number1" && exit 1
[ -z $b ] && echo "please \"$2 number2" && exit 2
[ ! $# -eq 2 ] && echo "--please num1 num2" && exit 3
# $$传递到脚本的参数个数
# -eq相当于=, -ne相当于不等于, -ge >= , -gt >, -le <=, -lt <
echo "$a + $b = $((($a+$b))"
echo "$a - $b = $((($a - $b))"
echo "$a * $b = $((($a*$b))"
echo "$a / $b = $((($a/$b))"
```

这里要注意的是，需要传入两个参数;

具体用到的一些知识请看下面：

文件表达式	
if [-f file]	如果文件存在
if [-d ...]	如果目录存在
if [-s file]	如果文件存在且非空
if [-r file]	如果文件存在且可读
if [-w file]	如果文件存在且可写
if [-x file]	如果文件存在且可执行
整数变量表达式	
if [int1 -eq int2]	如果int1等于int2
if [int1 -ne int2]	如果不等于
if [int1 -ge int2]	如果>=
if [int1 -gt int2]	如果>
if [int1 -le int2]	如果<=
if [int1 -lt int2]	如果<
字符串变量表达式	
If [\$a = \$b]	如果string1等于string2 字符串允许使用赋值号做等号
if [\$string1 != \$string2]	如果string1不等于string2
if [-n \$string]	如果string 非空(非0)，返回0(true)
if [-z \$string]	如果string 为空
if [\$sting]	如果string 非空，返回0 (和-n类似)

默认情况下，总是有三个文件处于打开状态，标准输入(键盘输入)、标准输出（输出到屏幕）、标准错误（也是输出到屏幕），它们分别对应的文件描述符是0，1，2。那么我们来看看下面的几种重定向方法的区别：>/.....

- handsomecui
3. Re:第八届河南省赛C.最少换乘（最短路
建图）
Dijkstra
--popcjz
4. Re:蓝桥杯-三羊献瑞
1
--1阳
5. Re:java匿名内部类，多态，接口练习
巨巨带我
--上弦月307

阅读排行榜

1. 十分钟学会写shell脚本(22188)
2. 奇妙的算法之LCS妙解(5274)
3. 欧拉函数(4525)
4. 路径问题以及cookie详解(3589)
5. memset函数详解(2799)

评论排行榜

1. N种方法妙讲LIS算法(37)
2. C/S系统实现两数求和(非阻塞+epoll+心跳包检测用户在线状况+滚动日志+配置文件.)(4)
3. xmu1125 越野车大赛(三分)(4)
4. zoj2729 Sum Up(模拟)(4)
5. 为了肾六(dp)(2)

推荐排行榜

1. 浅谈并发并行异步同步(6)
2. C/S系统实现两数求和(非阻塞+epoll+心跳包检测用户在线状况+滚动日志+配置文件.)(6)
3. Buy Tickets(线段树)(2)
4. 欧拉函数(2)
5. N种方法妙讲LIS算法(1)

4.3 特定变量参数

既然已经知道了如何访问和使用shell脚本中的参数，多知道一点相关信息也是很有用的，有必要知道脚本时的一些相关控制信息，这就是特定变量的由来。共有7个特定变量：

`##` 传递到脚本的参数个数

`*#` 以一个单字符串显示所有向脚本传递的参数。与位置变量不同，此选项参数可超过9个

`$$` 脚本运行的当前进程ID号

`#!` 后台运行的最后一个进程的进程ID号


`@` 与`$` #相同，但是使用时加引号，并在引号中返回每个参数

`-` 显示shell使用的当前选项，与`set`命令功能相同


`?` 显示最后命令的退出状态。0表示没有错误，其他任何值表明有错误。

读到这里大家对上面的小脚本是不是感觉特别easy了，哈哈；

7. 写一个脚本:求1到100的和?




```
#!/bin/bash
sum=0
i=1
while [ $i -le 100 ]
do
    sum=$((sum + i))
    i=$((i+1))
done
echo "$sum"
```




这个脚本就so easy了；

8. 接下来，我们写个很实用的脚本吧，假设你现在在机房，你想要看看机房哪些ip在开着，方便咱。。。是吧。

我们可以 通过ping命令测试192.168.0.151到192.168.0.254之间的所有主机是否在线




```
#!/bin/bash
for i in `seq 2 254`
do
    ping -c1 -w1 192.168.51.$i > /dev/null 2>&1
    #-c可以指定ping的次数，-w测试的时间 -w 1就是1秒中无论成功失败都结束
    #-c -w后面都要输入参数所以都要带上1
    #$?代表最后命令退出状态，0代表没错，其他代表有错
    [ $? -eq 0 ] && echo "192.168.51.$i IP is UP!" || echo "192.168.51.$i IP is down!"
done
```



在这里2>&1代表不显示错误信息，其实就是把错误重定向到标准输出，然后重定向到/dev/null，/dev/null就是个黑洞，写入的任何东西都会丢失，其实简要说就是把正确或错误的输出扔到垃圾桶。

写到这里大家是不是感觉耳目一新呐，哈哈！

随笔练习：还有点bug



```
#!/bin/bash
i=1
sum=0
#while
while [ $i -le 100 ]
do
    ((sum=sum+i))
    ((i=i+1))
done
echo $sum
#for
sum=0
for ((j=1;j<=100;j++))
do
    ((sum=sum+j))
done
echo $sum
```

```
#select
select flag in $@
do
    case $flag in
        $1)echo $1;;
        $2)echo $2;;
        $3)echo $3;;
        $4)echo $4;;
        *)echo "input error"
        break ;;
    esac
done
#shift
echo -e "the all arg is\n"
while [ $1 ]
do
    echo $1
    shift
done
#if
num=`ls | wc -l`
[ num == 0 ] && echo -e "have no file\n" || echo -e "there are $num files\n"
for file in `ls`
do
    echo "this file $file is "
    [ -f $file ] && echo -e "ordinary file"
    if [ -d $file ];then
        echo -e "dictionary file";
    fi
done

#function
filename=""
rename(){
    filename=echo `date` | awk -F ' ' '{print $1 $2 $3}'
}
rename
echo -e "filename is $filename"
[ $? == 0 ] && mkdir $filename || echo -e "have error\n"
```



转载请注明地址: <http://www.cnblogs.com/handsomecui/>

另外欢迎大家访问我的网站: <http://www.handsomecui.top/>

分类: ["Linux C编程"](#)

好文要顶

关注我

收藏该文



[handsomecui](#)

[关注 - 12](#)

[粉丝 - 54](#)

[+加关注](#)

« 上一篇: [擅长排列的小明](#)

» 下一篇: [New Year Transportation\(水\)](#)

1

2

posted @ 2016-09-13 18:07 handsomecui 阅读(22190) 评论(1) 编辑 收藏

#1楼[楼主] 2017-06-22 15:41 | handsomecui

默认情况下，总是有三个文件处于打开状态，标准输入(键盘输入)、标准输出（输出到屏幕）、标准错误（也是输出到屏幕），它们分别对应的文件描述符是0，1，2。那么我们来看看下面的几种重定向方法的区别：

```
>/dev/null 2>&1
```

// 实际上，应该等同于这样：1>/dev/null 2>/dev/null，默认情况下就是1，标准输出，所以一般都省略。而&符号，后面接的是必须的文件描述符。不能写成2>1，这样就成了标准错误重定向到文件名为1的文件中了，而不是重定向标准错误到标准输出中。所以这里就是：标准输出重定向到了/dev/null，而标准错误又重定向到了标准输出，所以就成了标准输出和标准错误都重定向到了/dev/null

[支持\(0\)](#) [反对\(0\)](#)[刷新评论](#) [刷新页面](#) [返回顶部](#)

注册用户登录后才能发表评论，请 [登录](#) 或 [注册](#)，[访问](#)网站首页。

【推荐】50万行VC++源码: 大型组态工控、电力仿真CAD与GIS源码库

【推荐】加入腾讯云自媒体扶持计划，免费领取域名&服务器

【推荐】高性能云服务器2折起，0.73元/日节省80%运维成本

【新闻】H3 BPM体验平台全面上线



葡萄城报表
千万种报表 同一种选择

在线设计 报表

数据价值即刻体现

[立即了解](#)

最新IT新闻:

- 抱上阿里的大腿，A站能否扳回26倍的差距？
 - 谷歌再挖走苹果芯片工程师 自主移动芯片即将面世
 - 新融资让滴滴估值超过了Uber 新能源汽车业务将加速
 - 给我一个圣诞帽！几亿用户集体戏精的行为艺术
 - 马斯克要把特斯拉Roadster跑车发射到火星：史无前例
- » [更多新闻...](#)



阿里云 **告别高昂运维费用 云计算全面助力**

40+款核心产品免费半年 再+8000津贴任意采购

[立即申请](#)

最新知识库文章:

- [步入云计算](#)
 - [以操作系统的角度述说线程与进程](#)
 - [软件测试转型之路](#)
 - [门内门外看招聘](#)
 - [大道至简，职场上做人做事做管理](#)
- » [更多知识库文章...](#)

历史上的今天:

2015-09-13 [Ultra-QuickSort\(归并排序\)](#)