

1) 如何向脚本传递参数 ?

```
./script argument
```

例子: 显示文件名称脚本

```
./show.sh file1.txt
```

```
cat show.sh
```

```
#!/bin/bash
```

```
echo $1
```

2) 如何在脚本中使用参数 ?

第一个参数 : \$1, 第二个参数 : \$2

例子 : 脚本会复制文件(arg1) 到目标地址(arg2)

```
./copy.sh file1.txt /tmp/
```

```
cat copy.sh
```

```
#!/bin/bash
```

```
cp $1 $2
```

3) 如何计算传递进来的参数 ?

```
$#
```

4) 如何在脚本中获取脚本名称 ?

```
$0
```

5) 如何检查之前的命令是否运行成功 ?

```
$?
```

6) 如何获取文件的最后一行 ?

```
tail-1
```

7) 如何获取文件的第一行 ?

```
head-1
```

8) 如何获取一个文件每一行的第三个元素 ?

```
awk '{print $3}'
```

9) 假如文件中每行第一个元素是 FIND, 如何获取第二个元素

```
awk '{ if ($1 == "FIND") print $2}'
```

10) 如何调试 bash 脚本

将 -xv 参数加到 #!/bin/bash 后

例子:

```
#!/bin/bash -xv
```

11) 举例如何写一个函数 ?

```
function example {
```

```
echo "Hello world!"
```

```
}
```

12) 如何向连接两个字符串 ?

```
V1="Hello"
```

```
V2="World"
```

```
V3=${V1}${V2}
```

```
echo $V3
```

输出

```
HelloWorld
```

13) 如何进行两个整数相加 ?

```
V1=1
V2=2
let V3=$V1+$V2
echo $V3
```

输出

3

据 @kashu 的意见，本题的更佳回答为：  
两个整数相加，还有若干种方法实现：

```
A=5
B=6
echo $((A+B)) # 方法 2
echo $[A+B] # 方法 3
expr $A + $B # 方法 4
echo $A+$B | bc # 方法 5
awk 'BEGIN{print "$A"+"$B"}' # 方法 6
```

**14) 如何检查文件系统中是否存在某个文件？**

```
if [ -f /var/log/messages ]
then
echo "File exists"
fi
```

**15) 写出 shell 脚本中所有循环语法？**

```
for 循环：
foriin$(ls);do
echo item:$i
done
while 循环：
#!/bin/bash
COUNTER=0
while [ $COUNTER -lt 10 ]; do
echo The counter is $COUNTER
let COUNTER=COUNTER+1
done
until 循环：
#!/bin/bash
COUNTER=20
until [ $COUNTER -lt 10 ]; do
echo COUNTER $COUNTER
let COUNTER-=1
done
```

**16) 每个脚本开始的 #!/bin/sh 或 #!/bin/bash 表示什么意思？**

这一行说明要使用的 shell。#!/bin/bash 表示脚本使用 /bin/bash。对于 python 脚本，就是

```
#!/usr/bin/python。
```

**17) 如何获取文本文件的第 10 行？**

`head -10 file|tail -1`

18) **bash** 脚本文件的第一个符号是什么

#

19) 命令: `[ -z "" ] && echo 0 || echo 1` 的输出是什么

0

20) 命令 “**export**” 有什么用 ?

使变量在子 shell 中可用。

21) 如何在后台运行脚本 ?

在脚本后面添加 “&”。

据 @kashu 的意见, 更好的答案是:

`nohup command&`

大部分时间我们可能是远程使用 Linux, 我碰到过由于网络断线使得在后台运行的 `command` & 没了...

22) “**chmod 500 script**” 做什么 ?

使脚本所有者拥有可执行权限。

23) “>” 做什么 ?

重定向输出流到文件或另一个流。

24) **&** 和 **&&** 有什么区别

**&** - 希望脚本在后台运行的时候使用它

**&&** - 当前一个脚本成功完成才执行后面的命令/脚本的时候使用它

25) 什么时候要在 `[ condition ]` 之前使用 “if” ?

当条件满足时需要运行多条命令的时候。

26) 命令: `name=John && echo 'My name is $name'` 的输出是什么

variable

27) **bash shell** 脚本中哪个符号用于注释 ?

#

28) 命令: `echo ${new:-variable}` 的输出是什么

variable

29) ‘ 和 ” 引号有什么区别 ?

’ - 当我们不希望把变量转换为值的时候使用它。

” - 会计算所有变量的值并用值代替。

30) 如何在脚本文件中重定向标准输出和标准错误流到 `log.txt` 文件 ?

在脚本文件中添加 “`exec >log.txt 2>&1`” 命令。

31) 如何只用 **echo** 命令获取字符串变量的一部分 ?

`echo ${variable:x:y}`

x - 起始位置

y - 长度

例子:

`variable="My name is Petras, and I am developer."`

`echo ${variable:11:6}` # 会显示 Petras

32) 如果给定字符串 `variable="User:123:321:/home/dir"`, 如何只用 **echo** 命令获取 `home_dir` ?

`echo ${variable##*:*:*:}`

或

```
echo ${variable###*}
```

33) 如何从上面的字符串中获取 “User” ？

```
echo ${variable%:*:*:*}
```

或

```
echo ${variable%%:*}
```

34) 如何使用 awk 列出 UID 小于 100 的用户 ？

```
awk -F: ' $3<100' /etc/passwd
```

35) 写程序为用户计算主组数目并显示次数和组名

```
cat /etc/passwd|cut -d: -f4|sort|uniq -c|while read c g
do
{ echo $c; grep :$g: /etc/group|cut -d: -f1;}|xargs -n 2
done
```

36) 如何在 bash shell 中更改标准的域分隔符为 “:” ？

```
IFS=":"
```

37) 如何获取变量长度 ？

```
${#variable}
```

38) 如何打印变量的最后 5 个字符 ？

```
echo ${variable: -5}
```

39) \${variable:-10} 和 \${variable: -10} 有什么区别？

\${variable:-10} - 如果之前没有给 variable 赋值则输出 10；如果有赋值则输出该变量

\${variable: -10} - 输出 variable 的最后 10 个字符

40) 如何只用 echo 命令替换字符串的一部分 ？

```
echo ${variable//pattern/replacement}
```

41) 哪个命令将命令替换为大写 ？

```
tr '[:lower:]' '[:upper:]'
```

42) 如何计算本地用户数目 ？

```
wc -l /etc/passwd|cut -d" " -f1 或者 cat /etc/passwd|wc -l
```

43) 不用 wc 命令如何计算字符串中的单词数目 ？

```
set ${string}
```

```
echo $#
```

44) “export \$variable” 或 “export variable” 哪个正确 ？

```
export variable
```

45) 如何列出第二个字母是 a 或 b 的文件 ？

```
ls -d ?[ab]*
```

46) 如何将整数 a 加到 b 并赋值给 c ？

```
c=$((a+b))
```

或

```
c=`expr $a + $b`
```

或

```
c=`echo "$a+$b"|bc`
```

47) 如何去除字符串中的所有空格 ？

```
echo $string|tr -d " "
```

48) 重写这个命令，将输出变量转换为复数：item="car"; echo "I like \$item" ？

```
item="car"; echo "I like ${item}s"
```

49) 写出输出数字 0 到 100 中 3 的倍数(0 3 6 9 ...)的命令 ?

```
for i in {0..100..3}; do echo $i; done
```

或

```
for (( i=0; i<=100; i=i+3 )); do echo "Welcome $i times"; done
```

50) 如何打印传递给脚本的所有参数 ?

```
echo $*
```

或

```
echo $@
```

51) [ \$a == \$b ] 和 [ \$a -eq \$b ] 有什么区别

[ \$a == \$b ] - 用于字符串比较

[ \$a -eq \$b ] - 用于数字比较

52) = 和 == 有什么区别

= - 用于为变量赋值

== - 用于字符串比较

53) 写出测试 \$a 是否大于 12 的命令 ?

```
[ $a -gt 12 ]
```

54) 写出测试 \$b 是否小于等于 12 的命令 ?

```
[ $b -le 12 ]
```

55) 如何检查字符串是否以字母 "abc" 开头 ?

```
[[ $string == abc* ]]
```

56) [[ \$string == abc\* ]] 和 [[ \$string == "abc\*" ]] 有什么区别

[[ \$string == abc\* ]] - 检查字符串是否以字母 abc 开头

[[ \$string == "abc\*" ]] - 检查字符串是否完全等于 abc

57) 如何列出以 ab 或 xy 开头的用户名 ?

```
egrep "^ab|^xy" /etc/passwd|cut -d: -f1
```

58) bash 中 \$! 表示什么意思 ?

后台最近执行命令的 PID.

59) \$? 表示什么意思 ?

前台最近命令的结束状态。

60) 如何输出当前 shell 的 PID ?

```
echo $$
```

61) 如何获取传递给脚本的参数数目 ?

```
echo $#
```

(LCTT 译注: 和第 3 题重复了。)

62) \$\* 和 \$@ 有什么区别\*

\$\* - 以一个字符串形式输出所有传递到脚本的参数

\$@ - 以 \$IFS 为分隔符列出所有传递到脚本中的参数

63) 如何在 bash 中定义数组 ?

```
array=("Hi" "my" "name" "is")
```

64) 如何打印数组的第一个元素 ?

```
echo ${array[0]}
```

65) 如何打印数组的所有元素 ?

```
echo ${array[@]}
```

66) 如何输出所有数组索引 ?

```
echo ${!array[@]}
```

**67) 如何移除数组中索引为 2 的元素 ?**

```
unset array[2]
```

**68) 如何在数组中添加 id 为 333 的元素 ?**

```
array[333]="New_element"
```

**69) shell 脚本如何获取输入的值 ?**

a) 通过参数

```
./script param1 param2
```

b) 通过 read 命令

```
read -p "Destination backup Server : " desthost
```

**70) 在脚本中如何使用 "expect" ?**

```
/usr/bin/expect << EOD
```

```
spawn rsync -ar ${line} ${desthost}:${destpath}
```

```
expect "*?assword:*
```

```
send "${password}\r"
```

```
expect eof
```

```
EOD
```