

Bach Scripting

Создание скрипта

Создаем файл с расширением
`script.sh`

В первой строке пишем
`#!/bin/bash`

Далее код
`echo 'My first script!'`

Выполнение скрипта

Для выполнения необходимо задать ему тип выполняемого файла.

```
shell>chmod  
u+x script.sh
```

И вызвать на выполнение

```
shell>./script.sh
```

Объявление переменных

#Присваивание

```
a=879
```

```
echo "Значение \"a\" – $a."
```

#Присваение значения

```
b=$a
```

```
echo $b
```

#Подстановка команд

```
a=Ãecho Hello!Ã
```

```
echo $a
```

Проверка условия

Результат выполнения true либо false

[условие]

Пример: есть ли такой файл

```
shell>[ -a file.txt ]
```

```
shell>echo $?
```

Управляющие конструкции

Шаблон:

```
if условие
then
    команды 1
else
    команды 2
fi
```

Пример

```
if cmp a b &> /dev/null
then
    echo "Files are identical."
else
    echo "Files a and b differ."
fi
```

Пример 2

```
if [ 0 ]  
then  
    echo "0 – истина."  
else  
    echo "0 – ложь."  
fi
```


Логические операторы

&&	–	и
	–	или
!	–	не

```
if [ $# -eq 0 ] && [ -z "$1" ]  
then;  
    echo «Нет входных аргументов»;  
else  
    echo «Входные аргументы $1»  
fi
```

Несколько проверок с помощью elif

```
if [ condition1 ]; then
    command1
elif [ condition2 ]; then
    command2
elif [ condition3 ]; then
    command3
else
    default command
fi
```

Пример

```
echo -n "Enter your login name:"  
read name  
if [ -z $name ] || [ "$name" = "" ];then  
    echo "You did not enter name"  
elif [ "$name" = "root" ]; then  
    echo "hello root"  
elif [ "$name" = "peter" ]; then  
    echo "hello peter"  
else  
    echo "You are not root but hi $name"  
fi
```

Задача

Пользователь вводит свой возраст.
Скрипт ему говорит к какой группе он
относится.

от 1 – 7 ребенок

7 – 18 школьник

18 – 25 студент

26 – 60 инженер

65 – 100 пенсионер

100 и выше - долгожитель

Оператор case

```
case значение in  
шаблон1 )  
command1  
;;  
шаблон2 )  
command2  
;;  
esac
```

Шаблонная часть может содержать метасимволы *, ?,
[...]

Пример 1

```
echo -n "Enter a number from 1 to 5:"  
read ans  
case $ans in  
1) echo "you select 1"  
;;  
2) echo "you select 2"  
;;  
3) echo "you select 2"  
;;  
*) echo "`basename $0`:This is not between  
1 and 5" > &2  
exit 1  
;;  
esac
```

Пример 2

```
#!/bin/sh
echo -n "Do you wish to proceed [y..n]:"
read ans
case $ans in
y|Y|yes|Yes) echo "yes is selected"
;;
n|N) echo "no is selected"
;;
*) echo "Abasename $0: Unknow response 5"
>&2
exit 1
;;
esac
```

Написать скрипт который стартуует,
перезагружает или останавливает
ряд служб.

Пример:

```
>./service stop
```

service – название скрипта

stop – параметр через аргумент

Циклы

Формат команды:

```
for имя_переменной in list
do
команда1
команда . .
done
```

Простой цикл

```
#!/bin/bash
for i in 1 2 3 4 5
do
    echo "Welcome $i times"
done
```

Простой цикл с диапазоном

```
#!/bin/bash  
for i in {1..5}  
do  
    echo "Welcome $i times"  
done
```

Цикл с диапазоном и шагом

```
#!/bin/bash
echo "Bash version $
{BASH_VERSION}..."
for i in {0..10..2}
do
    echo "Welcome $i times"
done
```

Цикл с перечислением

```
for VARIABLE in file1 file2 file3
do
    echo    $VARIABLE
done
```

Цикл с использованием команд

```
for file in `ls`  
do  
    echo $file  
done
```

Цикл в стиле C

```
#!/bin/bash
for ( ( i=1; i <= 10; i++ ) )
do
echo "number is $i"
done
```

Batch Processing Techniques

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

```
## Batch file processing script
```

```
for file in /path/to/files/*.txt; do
```

```
    if [ -f "$file" ]; then
```

```
        echo "Processing: $file"
```

```
        ## Add processing logic here
```

```
        grep "error" "$file" >> error_log.txt
```

```
    fi
```

```
done
```


Задание

Используя предыдущий пример напишите скрипт который выводит только каталоги и вычисляет размер каждого каталога!