Написание скриптов на bash

Создадим наш первый скрипт файл

```
nano script1.sh
```

Создастся пустой файл, который мы можем редактировать, соответственно vim ждет пока мы его отредактируем.

Первая строчка будет выглядеть:

```
#!/bin/bash, но лучше #!/usr/bin/env bash
```

Первая строчка показывает каким интерпретатором обрабатывать код в вашем файле.

Начиная со второй строки # - закоментить строку.

Создадим наш первый скрипт файл

```
echo “вы вошли как пользователь”
whoami
echo “вы находитесь в директории”
pwd
echo “----------”
```

Сохраните, запустите файл.

bash script1.sh

Создадим наш первый скрипт файл

Удобнее запускать используя **./<название скрипта>**,

Но есть нюанс: файл нужно сделать исполняемым.

Вы это уже умеете.

Переменные. $

```
echo “home directory is $HOME”
echo “for this job I’ll have \$100”
pwd
echo “----------”
```

Переменные. Делаем свои

```
user=“Admin”
level=100500
echo “$user have skill level $level”
```

Переменные. Делаем свои

```
num1=5
num2=4
num3=$num1+$num2
echo “summary = $num3”
echo “--------”
```

Вроде логично да?

Переменные. Делаем свои

```
num1=5
num2=4
let num3=$num1+$num2
echo “summary = $num3”
echo “--------”
```

Но что бы сложилось корректно добавьте let, что бы посчитать арифметическое уравнение.

Условные операторы.

```
if grep rutuser /etc/passwd
then
     echo “user rutuser found”
else
     echo “user rutuser not found”
fi
```

fi - закрывает

Условные операторы. Удалим лишнее

```
if grep --quet rutuser /etc/passwd
then
     echo “user rutuser found”
else
     echo “user rutuser not found”
fi
```

Hello world

```
echo “how is your name”
read username
echo “hello $username”
```

Проверяем существование файлов

```
echo “choose filename:”
read filename
if [ -e $filename]; then
  echo “$filename exist
else
  echo “$filename not found”
fi
```

-e проверит наличие файла.

Циклы

```
for var in list
do
команды
done
```

```
#!/bin/bash
file="myfile"
for var in $(cat $file); do
  echo " $var"
done
```

Циклы

Тут надо учесть, что подобный подход, если ожидается построчная обработка данных, не сработает для файла более сложной структуры, в строках которого может содержаться по несколько слов, разделённых пробелами.

Цикл будет обрабатывать отдельные слова, а не строки.

- Пробел
- Знак табуляции
- Знак перевода строки

Если bash встречает в данных любой из этих символов, он считает, что перед ним — следующее самостоятельное значение списка.

Циклы

Для того, чтобы решить проблему, можно временно изменить переменную среды IFS. Вот как это сделать в bash-скрипте, если исходить из предположения, что в качестве разделителя полей нужен только перевод строки:

```
IFS=$’\n’
```

```
#!/bin/bash
file="/etc/passwd"
IFS=$'\n'
for var in $(cat $file); do
  echo " $var"
done
```

Давайте обойдем все файлы в некой директории и обработаем их

Напишите скрипт, который выводит на экран список файлов и директорий

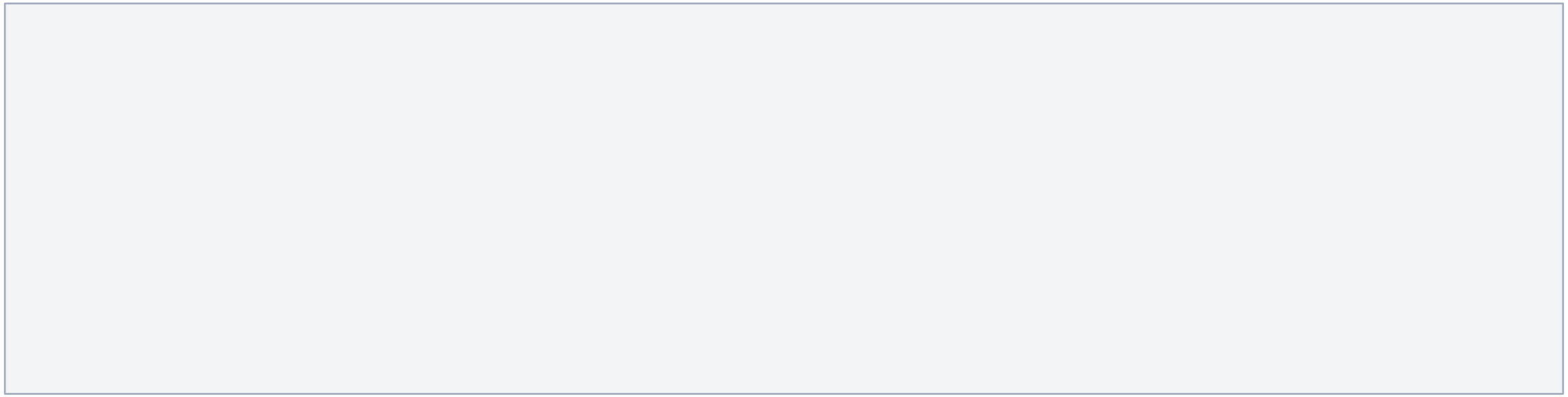
[-d filename] – проверит, существует ли файл, и является ли он директорией.
[-f filename] - проверит, существует ли файл, и является ли он файлом.

```
rutuser@server:~$ ./for.sh
/home/rutuser/123 is a directory
/home/rutuser/1231234 is a directory
/home/rutuser/1231234cxcxzc is a directory
/home/rutuser/123ed is a directory
/home/rutuser/12asdasd is a directory
/home/rutuser/1.txt is a file
/home/rutuser/for.sh is a file
/home/rutuser/script1.sh is a file
```

Давайте обойдем все файлы в некой директории и обработаем их

```
-d file Проверяет, существует ли файл, и является ли он директорией.
-e file Проверяет, существует ли файл.
-f file Проверяет, существует ли файл, и является ли он файлом.
-r file Проверяет, существует ли файл, и доступен ли он для чтения.
-s file Проверяет, существует ли файл, и не является ли он пустым.
-w file Проверяет, существует ли файл, и доступен ли он для записи.
-x file Проверяет, существует ли файл, и является ли он исполняемым.
file1 -nt file2 Проверяет, новее ли file1, чем file2.
file1 -ot file2 Проверяет, старше ли file1, чем file2.
-O file Проверяет, существует ли файл, и является ли его владельцем текущий
пользователь.
-G file Проверяет, существует ли файл, и соответствует ли его идентификатор
группы идентификатору группы текущего пользователя.
```

Давайте обойдем все файлы в некой директории и обработаем их



man

https://tldp.org/LDP/abs/html/