Linux Basic 12

TEL-RAN
by Starta Institute

scripts vol1

Повторение

TEL-RAN
by Starta Institute

- ssh
- Генерация ключа
- Просмотр ключа
- Важность
- Вход на учебный сервер
- Сервер
- Добавление новых пользователей на сервер
- Команда top
- Команда ps
- Команда ps-ef
- Разница top и ps
- kill и завершение процессов

Введение

TEL-RAN
by Starta Institute

- Скрипты
- Написание скриптов
- Оболочки
- Написание скриптов
- Права на файлы
- Запуск скрипта

Скрипты

TEL-RAN
by Starta Institute

Скрипт (англ. script - сценарий) - это небольшая программа, которая содержит последовательность действий, созданных для автоматического выполнения задачи.

Скрипты пишутся не только на разных языках программирования, но и в терминале bash в Linux.

Делается это для упрощения работы и удобства.

Одна из самых полезных возможностей bash-скриптов — это возможность извлекать информацию из вывода команд и назначать её переменным, что позволяет использовать эту информацию где угодно в файле сценария.

Написание скриптов

TEL-RAN
by Starta Institute

Открываем наши учебные терминалы.

Учебный терминал

Пишем:

nano script.sh

Сейчас неважно, где мы находимся, нам нужно понять сам принцип.

Файл, который мы создадим при помощи редактора nano создастся только тогда, когда мы будем выходить из редактора с сохранением.

А выходить с сохранением нужно ctrl+x

Зашли и открыли.

Написание скриптов

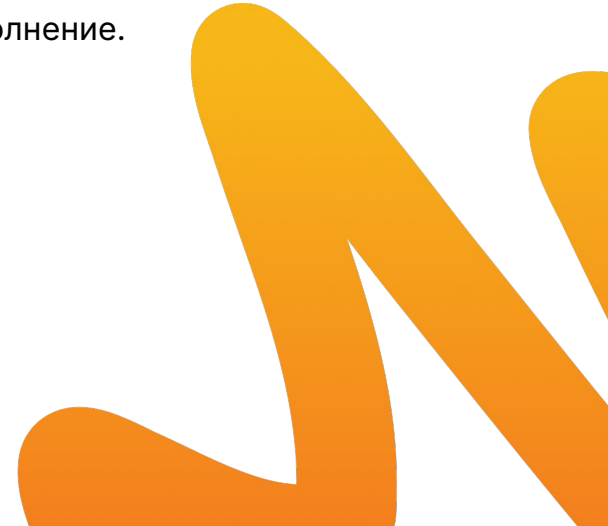
TEL-RAN
by Starta Institute

#! - эти два символа называют по-разному: решетка\восклицательный знак, диез\восклицательный знак и прочее.

В Linux ШЕБАНГ - он показывает, чем открывать файл.

- В Windows ассоциации файлов с программами с которыми они работают осуществляется при помощи расширения.
- В Linux файл является исполняемым, если у него есть права на исполнение. То есть если ему присвоили права на исполнение.

Чем открывать этот файл - передается первой строкой в этом файле.



Написание скриптов

TEL-RAN
by Starta Institute

Инструкция о том, чтобы система понимала, чем именно нужно открывать файл начинается с ШЕБАНГА.

В первой строчке мы видим, что система должна понять, что скрипт (или сценарий) будет запускаться при помощи bash, который находится в папке /bin.

Bash нужен для приема команд пользователя и их отправки операционной системе для последующей обработки.

Написание скриптов

TEL-RAN
by Starta Institute

Мы пишем команду, bash нам возвращает выполненную команду или результат действия на экран. Или не возвращает и просто отрабатывает.

В нашем случае: пишем в начале файла, что нам необходимо запустить все, что дальше в файле bash-ем.

Это будет набор инструкций, где каждая инструкция будет идти с новой строки.

Вместо /bash после /bin может идти /pyton чтобы все открывалось в интерпретаторе python. Либо java.

ПОЛЕЗНО ЗНАТЬ

TEL-RAN
by Starta Institute

Интерпрета́ция

построчный анализ, обработка и выполнение исходного кода программы или запроса, в отличие от компиляции, где весь текст программы, перед запуском анализируется и транслируется в машинный или байт-код без её выполнения.

Оболочки

TEL-RAN
by Starta Institute

Системе все-равно как и чем запускать код.

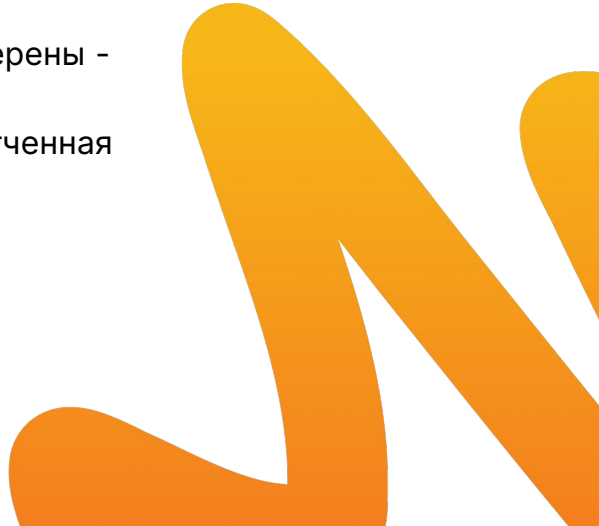
Мы будем рассматривать только bash оболочку.

Bash есть практически во всех современных дистрибутивах.

Как узнать в какой оболочке идет работа:

- Если мы нажимаем на TAB и нам подсказывается слово, то будьте уверены - перед вами bash - полная версия.
- Если вы нажимаете TAB и у вас нет подсказок, то перед вами облегченная версия оболочки. И оболочка может называться ash или sh.

cat /etc/os-release - проверить версию Linux



Написание скриптов

TEL-RAN
by Starta Institute

Вернемся к учебному терминалу и напишем:

```
#!/bin/bash

echo Hello

date
```

Команда echo выводит на экран текст, который мы напишем, а команда date - дату

Написание скриптов

TEL-RAN
by Starta Institute

Теперь выйдем и сохранимся:

ctrl+x

Проверим командой ls, что файл сохранился.

Вот, мы видим свой файл:

```
localhost:~# ls
```

```
bench.py    hello.c     hello.js    readme.txt  script.s
```

Написание скриптов

TEL-RAN
by Starta Institute

И чтобы убедиться, что у нас файл содержит команды давайте введем команду:

```
localhost:~# cat script.sh

#!/bin/bash

echo Hello

date
```

Написание скриптов

TEL-RAN
by Starta Institute

Теперь введем команду:

```
localhost:~# ls -l
total 20
-rw-r--r--    1 root     root           114 Jul  5  2020 bench.py
-rw-r--r--    1 root     root            76 Jul  3  2020 hello.c
-rw-r--r--    1 root     root            22 Jun 26  2020 hello.js
-rw-r--r--    1 root     root           151 Jul  5  2020 readme.txt
-rwxr--r--    1 root     root            12 Apr  4 15:30 script.sh
localhost:~#
```

Права на файлы

TEL-RAN
by Starta Institute

Файл, который мы создали не более чем текстовик.

Его можно только прочитать и записать. И то, только суперадмину.

Это еще не скрипт, а простой файл.

Что нам нужно сделать, чтобы он стал полноценным скриптом?

Помните, мы затрагивали тему прав на файлы? Что есть некая команда chmod, которая позволяет раздавать права направо и налево?

Итак, мы хотим добавить владельцу возможность исполнения.

```
localhost:~# chmod u+x script.sh
```

Запуск скрипта

TEL-RAN
by Starta Institute

Есть 3 способа запустить наш скрипт.

1. Если мы находимся в той же папке, что и наш скрипт, то:

```
localhost:~# ./script.sh
```

тут мы указываем, где взять в нашей папке то, что запустить.

Если мы введем название скрипта (файла), просто в терминал, то нас скажет, что файл не найден. Ведь его не ввели в переменные.

```
echo $PATH
```

```
/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bi
```

Помните мы говорили про переменные?

Так вот,запуская через точку слеш мы говорим системе, что нам необходимо в этой же папке взять наш файл и запустить.

Запуск скрипта

TEL-RAN
by Starta Institute

2. Запустить через полный путь:

```
localhost:~# pwd

/root

localhost:~# /root/script.sh

Hello

Tue Feb 15 15:17:46 UTC 2022
```

Запуск скрипта

TEL-RAN
by Starta Institute

3. Запустить через bash:

```
localhost:~# bash script.sh

Hello

Tue Feb 15 15:18:57 UTC 2021
```

Но третий не совсем правильный. Скорее запасной. Все дело в том, что мы делаем бесполезную ссылку на то, чем мы этот скрипт запускаем.

```
localhost:~# cat script.sh

#!/bin/bash

echo Hello

date
```

Запуск скрипта

TEL-RAN
by Starta Institute

Помните, вначале мы когда писали скрипт, мы говорили системе, чем нужно запускать скрипт?

И если мы в терминале пишем bash, то тем самым исключаем нашу первую строчку из текстового файла. А значит выполнили лишнее действие и потратили больше времени. Не совсем правильно.

Запуск скрипта

TEL-RAN
by Starta Institute

В этом виде интерпретатор будет относиться к текстовому файлу не как к скрипту, а как к простому текстовому файлу с двумя командами.

Однако есть и бонус в этой записи. Если, например, в нашем файле нету строки #!/bin/bash, то скрипт все равно отработает. И еще, если мы принудительно запускаем bash-ом, то файл может быть и не исполняемым. То есть есть свои плюсы.

Но пользоваться этим нужно осторожно и понимая когда и зачем.

Этим можно сэкономить время, но это не является полноценным скриптом.

Экспресс-опрос

TEL-RAN
by Starta Institute

- **Вопрос 1.**

  Какая первая строка в скрипте и для чего она предназначена?

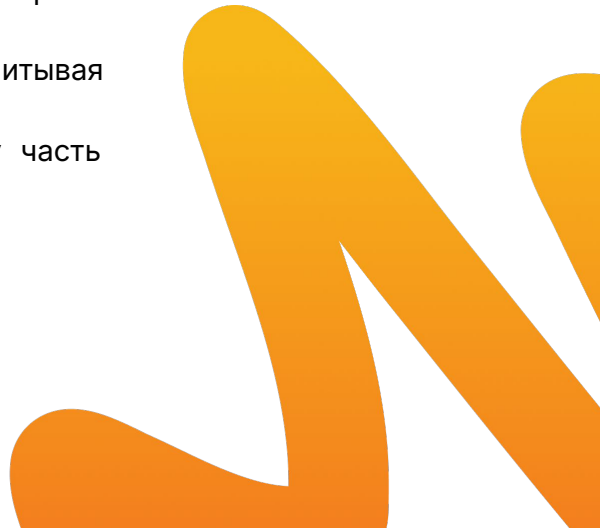
- **Вопрос 2.**

  Что будет, если команда написана с ошибкой? Будет ли выполняться скрипт далее или остановится на неправильной команде?

Домашнее задание

TEL-RAN
by Starta Institute

1. Создайте файл myfirstbashscript.sh где угодно.
2. Сделайте его исполняемым.
3. Заведите новую переменную USER и присвойте ей ваше имя в качестве значения
4. Добавьте в скрипт все необходимое, чтобы он :
5. Написал дату
6. Поприветствовал Вас по имени (hello $USER!)
7. Написал, из какой директории он работает
8. Добавил количество процессов одним числом (не забудьте, что первая строка выводит не сам процесс, а лишь заголовок).
9. Добавил количество процессов с именем bioset одним числом (не учитывая процесс grep)
10. Вывел права на файл /etc/passwd в формате (-rw-r--r--) - и только эту часть (потребуется awk)



Домашнее задание

TEL-RAN
by Starta Institute

Что вам понадобится:

echo, pwd, ps -ef, date, grep (флаг -v не забываем!), cat, wc -l, tail +n (строка), awk '{print $НОМЕР СТОЛБЦА}'

Заготовка для скрипта:

```
#!/bin/bash

#var and comments here
```

Полезные ссылки

TEL-RAN
by Starta Institute

- Коротко об SSH / Хабр