Переменные окружения

bash — это название shell (шелла), оболочки командной строки. Это та сущность, которая помогает общаться с операционной системой при помощи команд. Есть и другие шеллы, но мы будем в основном пользоваться bash, как самым распространённым.

Скрипт (ещё их называют сценариями) — это формально описанная последовательность действий, которая позволяет выполнять эти действия автоматически.

Переменные окружения

Как вы помните, для того чтобы запустить наш скрипт, нам требовалось указать до него путь (абсолютный или при помощи ./). Давайте поговорим об этом подробнее и, что самое интересное, о том, как от этого избавиться.

Проверить название шелла:

```
echo $SHELL
/bin/bash
```

Мы используем bash, и он находится в папке /bin.

Команда *echo*. Она просто позволяет вывести на стандартный вывод всё, что мы ей передаём. Например:

```
echo Hello World!
```

Или вот так, кавычки здесь не важны, но когда используются, команда их не выводит:

```
echo "Hello World!"
```

Как мы помним, стандартный вывод можно перенаправлять, поэтому при помощи команды echo мы можем создать файл с текстом при помощи однострочника:

```
echo "Some text here" > textfile
cat textfile
```

И дописать в конец файла больше текста:

```
echo "More text!" >> textfile
```

$SHELL — одна из переменных окружения.

Окружением мы называем тут некую виртуальную среду, которая создаётся при запуске консоли. Её определяет информация о пользователе, запустившем процесс, и о самой консоли. В переменных окружения хранится различная информация, которая может быть нужна другим программам, которые будут запускаться в этой же консоли.

Итак, есть переменная $SHELL (имя) и хранящаяся в ней информация `/bin/bash` (значение).
Это удобно: каждый раз, когда мы хотим получить информацию о том, какой шелл у нас используется, мы можем посмотреть значение переменной.

Знак доллара — это то, что указывает консоли на то, что сейчас будет не просто строка текста, а переменная. И команда echo не должна вывести “$SHELL”, а должна получить значение этой переменной.

Мы можем взаимодействовать с этой переменной не только при помощи команды *echo*, можем, например, узнать метаданные исполняемого файла bash:

```
stat $SHELL
```

При помощи команды *file*:

```
file $SHELL
```

Можем узнать, что открывать при помощи *cat* этот файл будет не лучшей идеей (напоминаю, что если всё-таки попробуете это сделать дома — просто подождите, а потом лучше закрыть окно терминала и открыть другое).

Есть и другие переменные окружения. Например:

```
$USER
$HOME
$PWD
```

Соответственно пользователь, домашняя директория, текущая директория.

Существует команда *pwd*. Тут можно предположить, что она берёт информацию о текущей директории из переменной окружения.

На самом деле, команд *pwd* две. Одна из них — программа, и одна — встроенная команда шелла. Чаще всего мы пользуемся именно встроенной командой, потому что именно она срабатывает, когда мы набираем в консоли:

```
pwd
```

Но есть и программа с таким же названием.

Как мы могли бы узнать об этом? Для того чтобы определить, команда шелла перед нами или программа, есть команда *type*:

```
type -a pwd
```

Pwd действительно существует в нескольких экземплярах!

type — встроенная команда шелла

```
type -a type
```

Для того чтобы воспользоваться программой pwd, нужно набрать её полный путь. И вот она не пользуется переменными окружения для определения текущей директории.

Переменная окружения PATH

Зачем указывать путь при запуске скрипта? Чтобы ответить на этот вопрос, нам нужно познакомиться с ещё одной переменной окружения.

```
$PATH
```

Давайте посмотрим, что в ней хранится:

```
echo $PATH
```

Это набор абсолютных путей, разделённых двоеточиями.

Каждый раз, когда вы вводите какую-либо команду, которая не является командой шелла, или команду без указания пути (то есть, например, *ca*t, как мы обычно и делаем) и нажимаете Enter, шелл сначала ищет такую команду среди встроенных, затем ищет в папках, перечисленных в PATH, исполняемый файл с таким названием. И, если находит, выполняет его. Если нет — выдаёт ошибку.

Если вызываемая команда — это встроенная команда шелла, мы сразу её выполняем. Предположим, мы используем не bash, а какой-нибудь другой шелл, в котором, например, нет встроенной команды *pwd*. В этом шелле мы пытаемся исполнить наш скрипт, использующий эту команду. Он продолжит работать, потому что есть ещё и исполняемый файл */bin/pwd*.
Таким образом, например, *echo*, *pwd* и некоторые другие команды существуют в двух ипостасях. И встроенная в шелл команда, и отдельная программа.

Итак, переменные окружения — это переменные, которые содержат различную информацию о запущенной консоли.

Создание переменной окружения

Это довольно просто:

```
export CITY=Moscow
echo $CITY
```

Обратите внимание, что, когда мы определяем переменную (то есть задаём её значение), мы используем её имя без дополнительных символов. А вот когда нам требуется получить значение переменной, мы используем знак доллара.

Для того чтобы увидеть список всех переменных окружения, используется команда env (или *printenv*):

```
env
```

Мы видим переменные, о которых мы говорили, и нашу новую переменную.

Переменные shell (шелла)

Помимо переменных окружения, существуют ещё и переменные шелла. Они доступны только для самого шелла. В них входят переменные окружения и довольно большой набор своих переменных. Если названия переменных окружения задаются исключительно в верхнем регистре, то названия переменных шелла обычно в нижнем.

Чтобы просмотреть все переменные шелла, мы используем команду *set*:

```
set  | less
```

Чтобы создать переменную шелла, мы просто задаём её имя и значение через знак равно в консоли:

```
SHELL_VAR="Variable"
```

```
set | grep SHELL_VAR
```

Среди переменных окружения, как и следовало ожидать, этой переменной нет:

```
env | grep SHELL_VAR
```

Чтобы она появилась, мы можем не переназначать её при помощи команды export, а экспортировать уже существующую переменную в переменные окружения:

```
export SHELL_VAR
env | grep SHELL_VAR
```

Однако если вы закроете консоль, эти изменения исчезнут. Чтобы эта переменная создавалась каждый раз, когда мы открываем консоль, мы можем записать её в специальный файл .bashrc.

Существуют файлы, которые начинаются с точки. Они не видны команде *ls* без ключа -a, и в них чаще всего содержится различная конфигурация. Например, в файле bashrc — конфигурация нашего шелла.

```
vim .bashrc
export CITY=Moscow
```

Создадим переменную окружения с названием города, чтобы не забыть, где находимся.
Закрываем файл с сохранением при помощи комбинации *wq* в командном режиме.

Теперь, если мы закроем терминал и откроем его заново, наша переменная CITY будет существовать:

```
env | grep CITY
```

А вот переменная SHELL_VAR, которая изначально была переменной шелла, а затем мы её экспортировали в переменную окружения, уже существовать не будет.

```
env | grep SHELL_VAR
```

Помимо создания новых переменных, мы можем переопределить значения старых, то есть изменить их значение полностью или взять то, что уже содержалось, и добавить что-то новое.
Например, мы можем взять содержимое переменной CITY и добавить к нему несколько слов.

```
export CITY="$CITY is the capital of Russia"
echo $CITY
Moscow is the capital of Russia
```