## *Получение данных от пользователя*

*Ключи и параметры командной строки — это отличный способ получить данные от того, кто пользуется скриптом, однако в некоторых случаях нужно больше интерактивности.*

*Иногда сценарии нуждаются в данных, которые пользователь должен ввести во время выполнения программы. Именно для этой цели в оболочке bash имеется команда read.*

*Эта команда позволяет принимать введённые данные со стандартного ввода (с клавиатуры). После получения данных, эта команда помещает их в переменную:*

*#!/bin/bash*

*echo -n "Enter your name: "*

*read name*

*echo "Hello $name, welcome to my program."*

*Обратите внимание на то, что команда echo, которая выводит приглашение, вызывается с ключом -n. Это приводит к тому, что в конце приглашения не выводится знак перевода строки, что позволяет пользователю скрипта вводить данные там же, где расположено приглашение, а не на следующей строке.*

*давайте не будем голословными, а попробуем убрать ключ -n и снова запустим файл.*

Потренироваться, не домашка:

Сделать файл скрипта, который выполнит:

Создаст файл и запишет в него информацию из команды df  -h о дисковом пространстве, но оставит в информации для записи лишь последний столбец в последней строчке, где заменит слово shm на ваше имя.(можно сделать последнюю строчку двумя способами через grep -w и через tail)

Выведет на экран содержимое созданного файла.

Что понадобится:

awk

sed

Решение:

#!/bin/bash

df -h | tail -1 | awk '{print $6}' | sed 's/shm/ivan/g' > test.txt

cat test.txt

Хакерство:

echo -e '#!/bin/bash\n date\n echo "it works!"' > script2.sh

**Циклы**.

Если вы загуглите использование циклов в bash, то вам вывалится огромное количество вариантов.

Давайте я покажу самый просто из вариантов и если вы его запомните - уже будет здорово.

Вы можете использовать разные стили написания. Например стиль Си. Ведь, если вы помните, все, что мы сейчас вводим и пишем выросло из языка Си.

Но есть чисто BASH стиль.

Что такое циклы? Это повторяющиеся действия.

Любой уважающий себя язык программирования должен содержать **циклы**. **Цикл** - это такая последовательность, которая позволяет выполнять определенный участок кода необходимое количество раз. С помощью **циклов** вы можете очень сильно сократить количество строк кода, которые необходимо написать для однотипных операций.

То, что мы сейчас рассмотрим, нормально работает как с массивами, так же и со списками файлов в директории.

#!/bin/bash

CITY=" Berlin"

echo Hello from $CITY

date

for run in {1..10}

do

echo $run

done

Что мы делаем. Мы объявляем переменную с маленькими буквами run, вначале мы ее нигде не заводили и ничего не присваивали.

Мы прям сразу на ходу ее заводим, называем ее run и {1..10} это стандартная конструкция минимального-максимального значения массива.

Где мы говорим, что мы заводим новую переменную, которая называется run и она принимает значение от 1 до 10.

Если мы пишем от 1 до 10, то с шагом 1 будет выполнятся 10 раз. И начало может быть не обязательно с 1(спойлер на домашку).

Тут мы говорим с какого начинаем и каким заканчиваем. И не обязательно это 1-10.

Еще раз - эта запись нам говорит, что массив будет с приростом в единицу от и до.

То есть у нас есть переменная run, она может быть от 1 до 10 и будет крутиться с шагом в единичку.

Все, что между do и done будет повторяться количество раз, которое указано в {1..10}

**В данном случае 10 раз.**

Давайте выйдем с сохранением и запустим:

localhost:~# ./script.sh

Hello from Berlin

Tue Feb 15 16:00:47 UTC 2022

1

2

3

4

5

6

7

8

9

10

localhost:~# cat script.sh

#!/bin/bash

CITY=" Berlin"

echo Hello from $CITY

date

for run in {1..10}

do

echo $run

done

У нас есть переменная run, к которой мы здесь же и обращаемся, говоря “а выведи нам значение”.

И тут у нас первая итерация:

Итерация ([лат.](https://ru.wikipedia.org/wiki/%2525D0%25259B%2525D0%2525B0%2525D1%252582%2525D0%2525B8%2525D0%2525BD%2525D1%252581%2525D0%2525BA%2525D0%2525B8%2525D0%2525B9_%2525D1%25258F%2525D0%2525B7%2525D1%25258B%2525D0%2525BA) [iteratio](https://ru.wiktionary.org/wiki/iteratio%2523%2525D0%25259B%2525D0%2525B0%2525D1%252582%2525D0%2525B8%2525D0%2525BD%2525D1%252581%2525D0%2525BA%2525D0%2525B8%2525D0%2525B9) «повторение») — повторение какого-либо действия.

run = 1 и мы говорим - выведи его.

Оболочка выводит. Смотрит на конечное число, сколько раз нужно повторить этот цикл и делает еще раз. Только run уже равно 2 и так далее.

После чего мы закончили.

Но у нас отработало все настолько быстро, что мы не успели ничего заметить. А как же драма? Давайте ее добавим! (=

localhost:~# nano script.sh

#!/bin/bash

CITY=" Berlin"

echo Hello from $CITY

date

for run in {1..10}

do

echo $run

sleep 1

done

Как мы видим, я добавил sleep 1, а это означает, что система будет ждать 1 секунду после выполнения одного действия.

Обратите внимание, как теперь выполняется наш код. Прям как в фильмах про хакеров.

Так же тут нормально работают дробные значения. Можно указать менее секунды.

**Если вдруг необходимо прервать процесс** - мы нажимает ctrl+c

**Условие задания:**

1. В рабочей папке на сервере создать файл bashscript.sh
2. Скрипт должен выполнить:

* создаст файл output.txt в вашей рабочей папке
* дозапишет в него информацию из файла по пути /var/spool/mail/ec2-user (только последние 10 строк)
* сделает счётчик и выведет на экран значения от a до j интервалом в 2 секунды

touch /opt/ВАША\_ГРУПА/ВАШЕ\_ИМЯ/bashscript.sh

файл скрипта:

\*

#!/bin/bash

cat /var/spool/mail/ec2-user | tail >> /opt/ВАША\_ГРУПА/ВАШЕ\_ИМЯ/bashscript.sh

for run in {a..j}

do

echo $run

sleep 2

done

\*