

Отчет о прохождении 3 этапа внешних курсов

Введение

Джаллох Ишмаил НКАбд-05-24

Содержание

1 Цель работы	6
2 Задание	7
3 Теоретическое введение	8
4 Выполнение лабораторной работы	9
5 Сертификат	40
6 Выводы	41
Список литературы	42

Список иллюстраций

4.1 Задание 1	9
4.2 Задание 2	10
4.3 Задание 3	11
4.4 Задание 4	12
4.5 Задание 5	12
4.6 Задание 6	13
4.7 Задание 7	14
4.8 Задание 8	14
4.9 Задание 9	15
4.10 Задание 10	16
4.11 Задание 11	17
4.12 Задание 13	18
4.13 Задание 14	19
4.14 Задание 14	20
4.15 Задание 15	21
4.16 Задание 16	22
4.17 Задание 17	23
4.18 Задание 18	24
4.19 Задание 18	25
4.20 Задание 19	26
4.21 Задание 19	26
4.22 Задание 20	27
4.23 Задание 21	27
4.24 Задание 22	28
4.25 Задание 23	29
4.26 Задание 24	29
4.27 Задание 25	30
4.28 Задание 26	31
4.29 Задание 27	32
4.30 Задание 28	33
4.31 Задание 29	34
4.32 Задание 30	35
4.33 Задание 31	36
4.34 Задание 32	37
4.35 Задание 33	38
4.36 Задание 34	38

5.1 Сертификат	40
--------------------------	----

Список таблиц

1 Цель работы

Ознакомиться с функционалом операционной системы Linux.

2 Задание

Просмотреть видео и на основе полученной информации пройти тестовые задания.

3 Теоретическое введение

Линукс - в части случаев GNU/Linux – семейство Unix-подобных операционных систем на базе ядра Linux, включающих тот или иной набор утилит и программ проекта GNU, и, возможно, другие компоненты. Как и ядро Linux, системы на его основе, как правило, создаются и распространяются в соответствии с моделью разработки свободного и открытого программного обеспечения. Linux-системы распространяются в основном бесплатно в виде различных дистрибутивов – в форме, готовой для установки и удобной для сопровождения и обновлений, – и имеющих свой набор системных и прикладных компонентов, как свободных, так и проприетарных.

4 Выполнение лабораторной работы

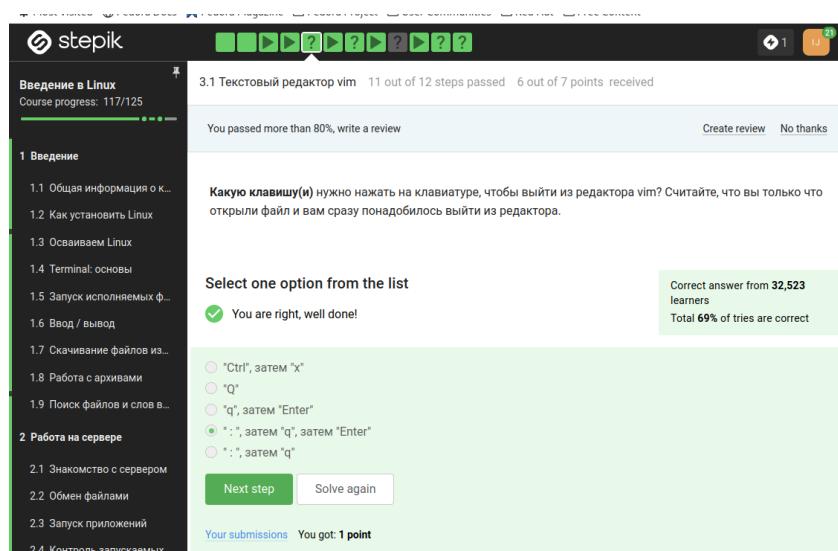


Рис. 4.1: Задание 1

Стоит упомянуть, что у редактора vim есть тьюториал, который позволяет разобраться с командами, необходимыми для стандартной работы. За выход из редактора отвечают следующие команды:

- ZQ - выйти без сохранения
- :q! - выйти без сохранения
- ZZ - записать файл и выйти (если файл не изменяли, то записываться он не будет)
- :wq - записать файл и выйти
- :x - записать файл и выйти
- :w - записать файл

- `:sav filename` - “сохранить как”
- `:w filename` - “сохранить как”
- `:w!` - записать файл

Как мы видим, вариантов много, при этом каждый сможет найти тот, который подойдёт под конкретную ситуацию.

Рис. 4.2: Задание 2

`Strange_ TEXT is_here. 2=2 YES!`

Точка считается “маленьким словом”, так что всего их 9: `Strange_, is_here, ., 2, =, 2, !` и два лишних пробела.

И если посчитать нажатия на `w` и на `W`, то действительно после 10 штук попадем в одно место. 10 нажатий на `W`, это то же самое, что и 10 нажатий на `w`,

Рис. 4.3: Задание 3

d2wwifour four <<Esc>>

d2wwywPp

d2w\$\$bifour four <<Esc>>

- \$ — в конец текущей строки;
- w — на слово вправо;
- b — на слово влево;
- i — начать ввод перед курсором;
- p — вставка содержимого неименованного буфера под курсором;
- P — вставка содержимого неименованного буфера перед курсором;
- yy (также Y) — копирование текущей строки в неименованный буфер;
- uu — копирование числа строк начиная с текущей в неименованный буфер;

Рис. 4.4: Задание 4

Поиск и замена в редакторе работают по следующей схеме:

`:{пределы}s/{что заменяем}/{на что заменяем}/{опции}`

Для замены во всем файле можно использовать символ %.

Рис. 4.5: Задание 5

Команда \$ – в конец текущей строки, W - до пробела вправо - то есть, перемещение.

Нажать Esc достаточно один раз, но да ладно.

Надпись visual - горит.

d – используется совместно с командами перемещения. Удаляет символы с текущего положения курсора до положения после ввода команды перемещения.

уу (также Y) – копирование текущей строки в буфер;

3.2 Скрипты на bash: основы 10 out of 10 steps passed 6 out of 6 points received

You passed more than 80%, write a review Create review No thanks

Надеемся, что вы разобрались, что одну оболочку (например, sh) можно запустить из другой оболочки (например, из bash). Предположим, что вы открыли терминал и у вас в нем запущена оболочка bash. Вы набираете в ней команды A1, A2, A3, а затем запускаете оболочку sh. В этой оболочке вы набираете команды B1, B2, B3 и запускаете оболочку bash. И, наконец, в этой последней оболочке вы набираете команды C1, C2, C3. Если теперь вы попробуете при помощи стрелочек вверх/вниз перемещаться по истории набранных команд, то команды из какого набора(ов) будут появляться?

Select one option from the list

Correct answer from 30,266 learners
Total 65% of tries are correct

All is correct.

Только из набора А

Из наборов В и С

Никакие команды появляться не будут

Только из набора С

Из наборов А и С

Next step Solve again

Рис. 4.6: Задание 6

Только из набора С потому что у каждой оболочки свой буфер, который при выходе из нее будет записываться в файл истории.

The screenshot shows a Stepik course interface for 'Введение в Linux'. The sidebar lists chapters 1 (Введение) and 2 (Работа на сервере). The main area displays a task titled '3.2 Скрипты на bash: основы' with a progress bar showing 10 out of 10 steps passed and 6 out of 6 points received. A message says 'You passed more than 80%, write a review'. Below is a code fragment:

```
#!/bin/bash
cd /home/bi/
touch file1.txt
cd /home/bi/Desktop/
```

A question asks: 'Как будет выглядеть **абсолютный путь** до созданного файла file1.txt по окончанию работы скрипта?' Below is a list of options:

- Никак (файла file1.txt не будет существовать после завершения работы скрипта)
- /home/bi/file1.txt
- /home/bi/Documents/file1.txt
- /home/bi/Desktop/file1.txt

Feedback indicates 'Correct answer from 29,905 learners' and 'Total 76% of tries are correct'.

Рис. 4.7: Задание 7

/home/bi/file1.txt - потому что именно в этой директории мы создаем новый файл, а уже после его создания мы переходим в другую папку.

The screenshot shows a Stepik course interface for 'Введение в Linux'. The sidebar lists chapters 1 (Введение), 2 (Работа на сервере), and 3 (Продвинутые темы). The main area displays a task titled '3.2 Скрипты на bash: основы' with a progress bar showing 10 out of 10 steps passed and 6 out of 6 points received. A message says 'You passed more than 80%, write a review'. Below is a question: 'Вы можете скачать и изучить скрипты, которые мы показали в видеофрагменте: variables1.sh, variables2.sh. Какие из представленных ниже строк **могут** быть именами переменных в bash? Выберите **все** подходящие варианты!' A note below says: 'Подсказка: если все варианты ответов являются неверными, то не отмечайте ни один из них и нажмите кнопку "Отправить"/"Submit".'

A message at the bottom says: 'You've solved a complex problem, congratulations! Now you can help other learners in comments by answering their questions, or compare your solution with others on solution forum.'

The task asks to select all correct options from a list:

- Yes!
- variable_123
- _variable
- 123variable
- var i able
- variabS\$le
- vari/able
- var-able

Buttons at the bottom are 'Next step' and 'Solve again'.

Рис. 4.8: Задание 8

Имя не может начинаться с цифры, содержать специальные символы или пробелы.

The screenshot shows a course titled 'Введение в Linux' with a progress of 117/125. The current step is '3.2 Скрипты на bash: основы'. The task instructions ask to write a bash script that takes two arguments and prints them. It provides examples for arguments like '\$1=первый_аргумент \$2=второй_аргумент' and '\$1=one \$2=two'. A note says 'Arguments are: \$1=three \$2=four' when run with three and four. A green box indicates 'Absolutely right.' and shows statistics: 'Correct answer from 25,053 learners' and 'Total 41% of tries are correct'. Below is a code editor with the following script:

```
1 #!/bin/bash
2 var1=$1
3 var2=$2
4
5 echo "Arguments are: \$1=$var1 \$2=$var2"
6
7
8
9
10
```

Рис. 4.9: Задание 9

\$ echo опции строка Эта команда печатает строки, которые передаются в качестве аргументов в стандартный вывод и обычно используется в сценариях оболочки для отображения сообщения или вывода результатов других команд.

var1=\$1 - обозначение переменных

var2=\$2

echo "Arguments are: \\$1=\$var1 \\$2=\$var2" - строка печати.

Рис. 4.10: Задание 10

- **\$0** - имя скрипта
- **\$#** - вернет количество аргументов
- **-ge** - больше или равно
- **-n** - не пустая строка.

Имя скрипта - это не пустая строка.

\$# Это число аргументов без учета имени скрипта, который всегда **\$0**. И число аргументов всегда будет или равно нулю, или больше него, тк просто не может скатиться в отрицательную сторону.

3.3 Скрипты на bash: ветвления и циклы 8 out of 9 steps passed 9 out of 10 points received

You passed more than 80%, write a review Create review No thanks

Вы можете скачать и изучить скрипты, которые мы показали в видеофрагменте:
branching2.sh, branching3.sh.

Посмотрите на фрагмент bash-скрипта:

```
if [[ $var -gt 5 ]]
then
    echo "one"
elif [[ $var -lt 3 ]]
then
    echo "two"
elif [[ $var -eq 4 ]]
then
    echo "three"
else
    echo "four"
fi
```

Какие строки и в какой последовательности он выведет на экран, если сначала этот скрипт запустили задав переменную var=3, а затем запустили еще раз, но уже с var=5.

Select one option from the list

Correct answer from 25,138 learners
Total 64% of tries are correct

Right.

Сначала two, потом one
 Сначала four, потом four
 Сначала two, потом four
 Сначала one, потом two

Next step Solve again

Рис. 4.11: Задание 11

- `-lt, (<)` - меньше
- `-gt` - больше
- `-eq` - равно

3 не больше 5, 3 не меньше 3, 3 не равно 4.

5 не больше 5, 5 не меньше 3, 5 не равно 4.

Оба раза выведет four.

The screenshot shows a course navigation sidebar on the left with sections like 'Введение в Linux' and '3 Продвинутые темы'. The main area displays a task titled '3.3 Скрипты на bash: ветвления и циклы'. It includes instructions, a code editor with examples, and a feedback section indicating the user passed the task.

Task Description:

Напишите скрипт на bash, который принимает на вход один аргумент (целое число от 0 до бесконечности), который будет обозначать число студентов в аудитории. В зависимости от значения числа нужно вывести разные сообщения.

Instructions:

Соответствие входа и выхода должно быть таким:

```

0 --> No students
1 --> 1 student
2 --> 2 students
3 --> 3 students
4 --> 4 students
5 и больше --> A lot of students

```

Notes:

Примечание 1: выводить нужно только строку справа, т.е. “-->” выводить не нужно.
 Примечание 2: в последней строке слово “lot” с маленькой буквы!

Example 1:

Пример №1: если ваш скрипт называется `./script.sh`, то при запуске его как `./script.sh 1` на экране должно появиться:

```
1 student
```

Example 2:

Пример №2: если ваш скрипт называется `./script.sh`, то при запуске его как `./script.sh 5` на экране должно появиться:

```
A lot of students
```

Feedback:

Correct answer from 23,310 learners
 Total 38% of tries are correct

Рис. 4.12: Задание 13

- (Start)
- a > c нет (Finish)
- (Start)
- , > c нет (Finish)
- (Start)
- b > c нет (Finish)
- (Start)
- , > c нет (Finish)
- (Start)
- c_d > c да

The screenshot shows a course titled "Введение в Linux" with a progress bar at 117/125. The current section is "3.3 Скрипты на bash: ветвления и циклы". A message says "You passed more than 80%, write a review". Below it, a note says "Вы можете скачать и изучить скрипты, которые мы показали в видеофрагменте: loops1.sh, loops2.sh.". A code snippet is shown:

```
for str in a , b , c_d
do
echo "start"
if [[ $str > "c" ]]
then
continue
fi
echo "finish"
done
```

A question asks: "Если запустить этот скрипт, то сколько раз на экран будет выведено слово "start", а сколько раз слово "finish"?". A note says "Correct answer from 24,582 learners Total 45% of tries are correct". Below are four options:

- 3 раза "start" и 3 раза "finish"
- 5 раз "start" и 2 раза "finish"
- 5 раз "start" и 4 раза "finish"
- 5 раз "start" и 5 раз "finish"

Buttons at the bottom are "Next step" and "Solve again".

Рис. 4.13: Задание 14

3.3 Скрипты на bash: ветвления и циклы 8 out of 9 steps passed 9 out of 10 points received

You passed more than 80%, write a review [Create review](#) [No thanks](#)

Напишите скрипт на bash, который будет определять в какую возрастную группу попадают пользователи. При запуске скрипта должен вывести сообщение "enter your name:" и ждать от пользователя ввода имени (используйте `read`, чтобы прочитать его). Когда имя введено, то скрипт должен написать "enter your age:" и ждать ввода возраста (опять нужен `read`). Когда возраст введен, скрипт пишет на экран <Имя>, your group is <группа>, где <группа> определяется на основе возраста по следующим правилам:

- младше либо равно 16: "child",
- от 17 до 25 (включительно): "youth",
- старше 25: "adult".

После этого скрипт опять выводит сообщение "enter your name:" и всё начинается по новой (бесконечный цикл). Если в какой-то момент работы скрипта будет введено пустое имя или возраст 0, то скрипт должен написать на экран "bye" и закончить свою работу (выход из цикла).

Примеры корректной работы скрипта:

№1

```
./script.sh
enter your name:
Egor
enter your age:
16
Egor, your group is child
enter your name:
Elena
enter your age:
8
bye
```

№2:

```
./script.sh
enter your name:
Elena Petrovna
enter your age:
25
Elena Petrovna, your group is youth
enter your name:
bye
```

Рис. 4.14: Задание 14

`child=16`

`adult=25`

`stdout=0`

```
while [[ $stdout != 1 ]] #конструкция типа while-True
```

`do`

```
    echo "enter your name: " #Пользователь вводит имя
```

`read name`

```
    if [[ (-z $name) || ($name = 0) ]] ;then #Если имя не по параметрам, простишь
```

`echo "bye"`

`stdout=1`

```
elif [[ -n $name ]]; then #А вот если имя нормальное
```

`while [[$stdout != 1]] ;do`

```

echo "enter your age: " #To пусть вводит возраст
read age #Считываем возраст
if [[ ($age -eq 0) || (-z $age) ]] ;then #Если возраст 0 или строка п
    echo "bye"
    stdout=1
elif [[ $age -le $child ]] ;then #Если меньше или равен ребенку, то р
    echo "$name, your group is child"
elif [[ $age -gt $adult ]] ; then #Больше взрослого - то взрослый
    echo "$name, your group is adult" ;else
    if [[ ($age -ge 17) && ($age -le 25) ]] ;then #Если от 17 до 25,
        echo "$name, your group is youth" ;fi
    fi ;break
done ;fi
done

```

done

The screenshot shows a Stepik course interface. The top navigation bar includes icons for back, forward, search, and help, along with a user icon and a notification badge. The main header displays the course title 'Введение в Linux' and 'Course progress: 117/125'. Below the header, a sidebar lists sections: '1 Введение', '2 Работа на сервере', and '3 Продвинутые темы'. The '3.4 Скрипты на bash: разное' section is highlighted. The main content area shows the challenge details: '3.4 Скрипты на bash: разное 10 out of 10 steps passed 14 out of 14 points received'. It also indicates that the user passed more than 80% and provides options to 'Create review' or 'No thanks'. A note says: 'Вы можете скачать и изучить скрипты, которые мы показали в видеофрагменте: [math1.sh](#), [math2.sh](#)'. A tip about variable expansion is provided: 'Какие(ая) из предложенных ниже инструкций увеличивают значение переменной a на значение переменной b? Например, если в a было записано 10, в b было 5, то в a должно записаться 15. Выберите все подходящие варианты!'. A note about command interpretation follows: 'Примечание: если вы планируете проверять варианты ответов у себя в терминале, обратите внимание на то, что содержащие символ \$ тексты могут изменяться при копировании — не забудьте отредактировать их в соответствии с изображением на экране. Это связано с особенностями написания \$ в некоторых видах заданий на Stepik.' A tip about quotes and spaces is given: 'Подсказка: обратите особое внимание на кавычки и пробелы, они могут как принципиально изменить команду, так и ни на что не повлиять (в зависимости от команды и контекста)!'. The challenge instructions ask to 'Select all correct options from the list'. A green checkmark indicates 'Great work!' and a message: 'You've solved a complex problem, congratulations! Now you can help other learners in comments by answering their questions, or compare your solution with others on solution forum.' A list of options is shown, with the last one checked: 'let a=\$a+\$b'. At the bottom are 'Next step' and 'Solve again' buttons.

Рис. 4.15: Задание 15

1. $a = \$a$
2. $a += b$ это то же самое, что и $a = a + b$, но с символами “ $=$ ” != “ $=+$ ”
3. если выражение не в скобках, но с пробелами - работать не будет. (`let a=a+b`
- сработает; `let a = a + b` - нет)

Рис. 4.16: Задание 16

programm выполняет стандартный вывод в терминал (если это принцип работы программы). И нам нужно настроить вывод в файл.

The screenshot shows a course interface on the Stepik platform. The main title is 'Введение в Linux' with a progress bar at 117/125. The sidebar lists sections: 1 Введение, 2 Работа на сервере, and 3 Продвинутые темы. The current step, '3.4 Скрипты на bash: разное', has been completed with 10 out of 10 steps passed and 14 out of 14 points received. A message says 'You passed more than 80%, write a review' with options 'Create review' and 'No thanks'. The main content area discusses bash script constructs like 'if' and 'if [\$var -eq 0]'. It includes a note about the difference between single quotes and double quotes. A green box highlights 'Select all correct options from the list' with a success message: 'Good news for you, correct!' and 'You've solved a complex problem, congratulations! Now you can help other learners in comments by answering their questions, or compare your solution with others on solution forum.' Below is a list of options for the question 'if [\$program -gt some_file.txt]':

- Ничего сделать нельзя
- if 'program > some_file.txt'
- Сначала var='program', затем if [[\$var -eq 0]]
- if ['\$program' -eq 0]
- Сначала запустить program, затем if [[\$? -eq 0]]

At the bottom are 'Next step' and 'Solve again' buttons.

Рис. 4.17: Задание 17

Первая переменная локальная, и это просто пустая строка, вторая переменная – это сумма арифметической прогрессии от 1 до 10, равна 55, но при умножении на 2 даст 110.

The screenshot shows a course page on the Stepik platform. The course is titled 'Введение в Linux' and has a progress bar indicating 117/125 steps completed. The main content area displays a task titled '3.4 Скрипты на bash: разное'. The task status is '10 out of 10 steps passed' and '14 out of 14 points received'. A message says 'You passed more than 80%, write a review' with options to 'Create review' or 'No thanks'. Below this, a note says 'Вы можете скачать и изучить скрипты, которые мы показали в видеофрагменте: functions1.sh, functions2.sh.' A code snippet for a Bash function 'counter' is shown:

```
counter () # takes one argument
{
    local let "c1+=${!1}"
    let "c2+=${!1}*2"
}
```

A note below the code says: 'Впишите в форму ниже строку, которую выведет на экран команда echo "counters are \$c1 and \$c2" если она находится в скрипте **после десяти вызовов** функции counter с параметрами сначала 1, затем 2, затем 3 и т.д., последний вызов с параметром 10.' A 'Подсказка:' note follows: 'этот пример можно решить в уме, но если система проверки не принимает ваше решение, то возможно вы что-то упустили (возможно что-то совсем небольшое/невидимое). В этом случае имеет смысл написать небольшой скрипт на bash, который проделает ровно то, что указано в задании и посимвольно сверить свой ответ с тем, что он выдаст на экран.'

The 'Write text answer' section shows a green checkmark and the text 'Great work!' followed by a note: 'You've solved a complex problem, congratulations! Now you can help other learners in comments by answering their questions, or compare your solution with others on solution forum.' To the right, it says 'Correct answer from 20,009 learners' and 'Total 28% of tries are correct'. The user's answer 'counters are and 110' is shown in a text input field.

Рис. 4.18: Задание 18

3.4 Скрипты на bash: разное 10 out of 10 steps passed 14 out of 14 points received

You passed more than 80%, write a review [Create review](#) [No thanks](#)

Напишите скрипт на bash, который будет искать наибольший общий делитель (**НОД**, greatest common divisor, GCD) двух чисел. При запуске ваш скрипт не должен ничего писать на экран, а просто ждет ввода двух натуральных чисел через пробел (для этого можно использовать `read` и указать ему две переменные – см. пример в видеофрагменте). После ввода чисел скрипт считает их НОД и выводит на экран сообщение "GCD is <посчитанное значение>", например, для чисел 15 и 25 это будет "GCD is 5". После этого скрипт опять входит в режим ожидания двух натуральных чисел. Если в какой-то момент работы пользователь ввел вместо этого пустую строку, то нужно написать на экран "bye" и закончить свою работу.

Вычисление НОД несложно реализовать с помощью [алгоритма Евклида](#). Вам нужно написать функцию `gcd`, которая принимает на вход два аргумента (назовем их **M** и **N**). Если аргументы равны, то мы нашли НОД – он равен **M** (или **N**), нужно выводить соответствующее сообщение на экран (см. выше). Иначе нужно сравнять аргументы между собой. Если **M** больше **N**, то запускаем ту же функцию `gcd`, но в качестве первого аргумента передаем (**M-N**), а в качестве второго **N**. Если же наоборот, **M** меньше **N**, то запускаем функцию `gcd` с первым аргументом **M**, а вторым (**N-M**).

Пример корректной работы скрипта:

```
./script.sh
10 15
GCD is 5
7 3
GCD is 1
bye
```

Примечание: в вызове функции из себя самой нет ничего страшного или неправильного, т.ч. смело вызывайте `gcd` прямо внутри `gcd`!

Примечание 2: для завершения работы функции в произвольном месте, можно использовать инструкцию `return` (все инструкции функции после `return` выполнятся не будут). В отличии от `exit` эта команда завершит только функцию, а не выполнение всего скрипта целиком. Однако в данной задаче можно обойтись и без использования `return`!

Подсказка: в случае проблем с решением задачи, обратите внимание [на наши рекомендации по написанию](#)

Рис. 4.19: Задание 18

Алгоритм нахождения НОД делением

1. Большее число делим на меньшее.
2. Если делится без остатка, то меньшее число и есть НОД (следует выйти из цикла).
3. Если есть остаток, то большее число заменяем на остаток от деления.
4. Переходим к пункту 1.

3.4 Скрипты на bash: разное 10 out of 10 steps passed 14 out of 14 points received

You passed more than 80%, write a review Create review No thanks

Напишите **калькулятор** на bash. При запуске ваш скрипт должен ожидать ввода пользователем команды (при этом на экран выводить ничего не нужно). Команды могут быть трех типов:

1. Слово "exit". В этом случае скрипт должен вывести на экран слово "bye" и завершить работу.
2. Три аргумента через пробел – первый операнд (целое число), операция (одна из "+", "-", "*", "/", "%", "**") и второй операнд (целое число). В этом случае нужно произвести указанную операцию над заданными числами и вывести результат на экран. После этого переходим в режим ожидания новой команды.
3. Любая другая команда из одного аргумента или из трех аргументов, но с операцией не из списка. В этом случае нужно вывести на экран слово "error" и завершить работу.

Чтобы проверить работу скрипта, вы можете записать сразу несколько команд в файл и передать его скрипту на stdin (т.е. выполнить `./script.sh < input.txt`). В этом случае он должен вывести сразу все ответы на экран.

Например, если входной файл будет следующего содержания:

```
10 + 1
2 ** 10
exit
```

то на экране будет:

```
11
1024
bye
```

Если же на вход поступит следующий файл:

```
3 - 5
2/10
exit
```

то на экране будет:

```
-2
error
```

Т.к. вторая команда была **некорректной** (в ней всего один аргумент, т.к. нет пробелов между числами и операцией, единственная допустимая команда из одного аргумента это "exit").

Рис. 4.20: Задание 19

3.5 Продвинутый поиск и редактирование 13 out of 13 steps passed 10 out of 10 points received

You passed more than 80%, write a review Create review No thanks

Пусть в директории `/home/b1` лежат файлы `Star_Wars.avi`, `star_trek_OST.mp3`, `STARS.txt`, `stardust.mpeg`, `Eddard_Stark_biography.txt`.

Отметьте все файлы, которые **найдет** команда `find /home/b1 -iname "star*"`, но **НЕ найдет** команда `find /home/b1 -name "star*"`?

Select all correct options from the list

You are right, well done!

Correct answer from 20,547 learners
Total 36% of tries are correct

STARS.txt
 Star_Wars.avi
 star_trek_OST.mp3
 stardust.mpeg
 Eddard_Stark_biography.txt

Next step Solve again

Your submissions You got: 1 point

Рис. 4.21: Задание 19

Калькулятор выглядит обычно – мы вводим два числа, пишем, что с ними надо

сделать, и потом, учитывая случаи ошибок, выводим результат.

The screenshot shows a Stepik course interface for 'Введение в Linux'. The sidebar on the left lists chapters 1 (Введение), 2 (Работа на сервере), and 3 (Продвинутые темы). Chapter 3 has 3.5 'Продвинутый поиск и редактирование' selected. The main area displays the step '3.5 Продвинутый поиск и редактирование' with a progress bar showing 13 out of 13 steps passed and 10 out of 10 points received. A message says 'You passed more than 80%, write a review'. Below it is a task: 'Select all correct options from the list'. One option is checked: 'Absolutely right.' A green box indicates 'Correct answer from 18,450 learners' and 'Total 22% of tries are correct'. The task asks to identify which of four statements about the 'find' command are true. The first statement is checked: 'В некоторых случаях find с -name найдет меньше файлов, чем find с таким же запросом, но с -path'. Other statements involve options like '-path' and '-name' and their interactions. At the bottom are 'Next step' and 'Solve again' buttons.

Рис. 4.22: Задание 20

-iname ищет без учета регистра, а -name в точности как в запросе. Звездочка стоит после слова – это значит после слова может быть сколько угодно символов.

This screenshot shows the same 'Введение в Linux' course interface as the previous one. Chapter 3 is selected. The main area displays step 3.5. A message says 'You passed more than 80%, write a review'. Below it is a task: 'Предположим, что в директории /home/b1/ есть следующая структура файлов и поддиректорий:' followed by a file tree diagram. The tree shows /home/b1/ containing dir1, which contains file1 and dir2; dir2 contains file2 and dir3, which contains file3. The task asks: 'Какие(ой) из трех файлов (file1, file2, file3) будут найдены по команде find /home/b1 -mindepth 2 -maxdepth 3 -name "file*"'? Below the tree, a list of five options is shown, with the last one checked: 'Все кроме file3'. A green box indicates 'Correct answer from 20,711 learners' and 'Total 41% of tries are correct'. At the bottom are 'Next step' and 'Solve again' buttons.

Рис. 4.23: Задание 21

`find [path] [expression]`

где: path - это путь к директории, в которой нужно выполнить поиск файлов (по умолчанию, поиск производится в текущей директории и всех ее поддиректориях);

expression - это выражение, которое определяет критерии поиска файлов.

-name: поиск файлов по имени. Например: `find /home/user -name myfile.txt`

The screenshot shows a Stepik course interface for 'Введение в Linux'. The current step is '3.5 Продвинутый поиск и редактирование'. The task description asks about the behavior of grep command options -A, -B, and -C when applied to a file containing the word 'word' multiple times. It provides the following command examples:

```
grep "word" file.txt > results.txt
grep -A 1 "word" file.txt > results.txt
grep -B 1 "word" file.txt > results.txt
grep -C 1 "word" file.txt > results.txt
```

The question asks which option creates the largest results.txt file. The correct answer is 'results.txt будет одинакового размера во всех случаях' (results.txt will be of the same size in all cases). The user has selected this option and completed the task successfully.

Below the task area, there is a sidebar with course navigation and a footer with social sharing icons.

Рис. 4.24: Задание 22

Текущий каталог - это `depth=1`, а остальное считается просто:

`/home/bi` -> `depth=1`

`/home/bi/dir1` -> `depth=2`

`/home/bi/dir1/dir2` -> `depth=3`

The screenshot shows a Stepik course interface for 'Введение в Linux'. The course progress is at 117/125. Step 3.5 is titled 'Продвинутый поиск и редактирование' (Advanced search and editing). The task description asks to find lines containing 'xkLXKL' or 'ubuntu\$' in 'text.txt' using grep. The user has passed more than 80% and can write a review. A green box indicates 'Great work!' and 'You've solved a complex problem, congratulations! Now you can help other learners in comments by answering their questions, or compare your solution with others on solution forum.' A sidebar shows statistics: 18,768 learners solved it, and 23% of tries were correct. The user's submission got 2 points. The sidebar also shows 786 likes and 246 dislikes.

Рис. 4.25: Задание 23

Из описания man: Print NUM lines of trailing context after/before matching lines
 “matching lines” - множественное число, строки в которых нашлось совпадение
 Т.е. если идут 2...10...100 строк подряд, в которых обнаружилось совпадение,
 контекст будет выведен до и после этой ГРУППЫ строк, а не до и после каждой
 строки в этой группе

The screenshot shows the same Stepik course and step 3.5. The task now involves sed and regular expressions. It asks what happens if the option '-n' is omitted from the command 'sed -n "/[a-z]*p" text.txt'. The user has passed more than 80% and can write a review. A green box indicates 'Totally right.' and 'You've solved a complex problem, congratulations! Now you can help other learners in comments by answering their questions, or compare your solution with others on solution forum.' A sidebar shows statistics: 19,784 learners solved it, and 39% of tries were correct. The user's submission got 1 point. The sidebar also shows 786 likes and 246 dislikes.

Рис. 4.26: Задание 24

Объяснение на втором скриншоте.

The screenshot shows a Stepik course interface for 'Введение в Linux'. The course progress is at 117/125. The current section is '3 Продвинутые темы' (Advanced Topics), which is highlighted in green. The challenge '3.5 Продвинутый поиск и редактирование' is selected. The challenge title is '3.5 Продвинутый поиск и редактирование' and it has 13 steps passed out of 13, with 10 out of 10 points received. A message says 'You passed more than 80%, write a review' with 'Create review' and 'No thanks' buttons. The challenge description asks to rewrite 'abbreviation' to 'abberviaturo' in 'input.txt' and save it to 'edited.txt'. It notes that instructions must include 'sed' and both files. It defines 'abberviaturo' as a word consisting of two abbreviations. An example is given: if the input is 'Hi, I heard these songs by ABBA, TLA and DM!', it should be changed to 'Hi, I heard these songs by ABBA, abberviaturo and abberviaturo!'. A note says to use 'grep -rf' instead of 'sed'. A warning about regex lookahead is present. The challenge has 16,632 correct answers from 13 learners. A green button at the bottom says 'All is correct.'

Рис. 4.27: Задание 25

The `-n` option disables the automatic printing, which means the lines you don't specifically tell it to print do not get printed, and lines you do explicitly tell it to print (e.g. with `p`) get printed only once.

The screenshot shows a Stepik course interface for 'Введение в Linux'. The course progress is 117/125. The current step is '3.6 Строим графики в gnuplot' with 10 out of 10 steps passed and 7 out of 7 points received. A message encourages the user to write a review. The task description asks how to use the '-p, --persist' option to keep plots open after gnuplot exits. Below is a list of options:

- s, --show-plots-after-exit
- raise
- Такой опции не существует
- p, --persist

Buttons at the bottom include 'Next step' and 'Solve again'. A stats box indicates 18,785 learners and 51% of tries are correct.

Рис. 4.28: Задание 26

аббревиатура АВВА отличается от двух других аббревиатур тем, что справа он неё стоит запятая без пробела: “АВВА,”.

При этом по условию аббревиатура должна выглядеть как [XX] или [XXX] (и ещё больше X). Следовательно, для этой проверки надо добавить пробел квадратными скобками [] слева и, соответственно, с права.

3.6 Строим графики в gnuplot 10 out of 10 steps passed 7 out of 7 points received

You passed more than 80%, write a review Create review No thanks

Предположим у вас есть файл `data.csv` с двумя столбцами по 10 чисел в каждом. В первой строке не записаны названия столбцов, т.е. ряды данных начинаются прямо с первой строки. Вы запускаете gnuplot и вводите в него две команды:

```
set key autotitle columnhead
plot 'data.csv' using 1:2
```

Какое в этом случае будет **название** у построенного **ряда данных** и сколько будет нарисовано **точек** на графике?

Select one option from the list

Great!

You've solved a complex problem, congratulations! Now you can help other learners in comments by answering their questions, or compare your solution with others on solution forum.

Correct answer from 17,975 learners
Total 32% of tries are correct

Название – первое значение из второго столбца, нарисовано 10 точек
 Название – первое значение из первого столбца, нарисовано 9 точек (точка из первой строки пропущена)
 Название "data.csv" using 1:2", нарисовано 10 точек
 Название – первое значение из первого столбца, нарисовано 10 точек
 Название – первое значение из второго столбца, нарисовано 9 точек (точка из первой строки пропущена)

Next step Solve again

Рис. 4.29: Задание 27

-persist lets plot windows survive after main gnuplot program exits.

3.6 Строим графики в gnuplot 10 out of 10 steps passed 7 out of 7 points received

You passed more than 80%, write a review Create review No thanks

Вы можете скачать и изучить скрипты, которые мы показали в видеофрагменте: [plot.gnu](#), [plot_advanced.gnu](#), [plot_advanced2.gnu](#). Все три скрипта основаны на [этой заметке](#), данные также взяты оттуда.

Предположим, что вы пишите gnuplot-скрипты и у вас в нем есть три переменные `x1`, `x2`, `x3`, в которых записаны координаты важных точек по оси ОХ (по возрастанию). Вы хотите, чтобы на этой оси было только три деления (т.е. три черточки) в этих самых координатах, а подпись этих делений были оформлены в виде `"point <номер точки>, value <значение соответствующей переменной>"`.

Например, для `x1=0`, `x2=10`, `x3=20`, это были бы надписи `"point 1, value 0"` в точке с координатой 0 по горизонтали, `"point 2, value 10"` в точке с координатой 10 и `"point 3, value 20"` в точке с координатой 20. Или, например, `x1=100`, `x2=150`, `x3=250`, это были бы надписи `"point 1, value 100"` в точке с координатой 100, `"point 2, value 150"` в точке с координатой 150 и `"point 3, value 250"` в точке с координатой 250.

Впишите в форму ниже **одну команду** (т.е. одну строку), которую нужно добавить в скрипт, для выполнения этой задачи.

Примечание: проверять, что переменные `x1`, `x2`, `x3` идут по возрастанию или что они являются числами не нужно!

Примечание 2: в видеофрагменте на предыдущем шаге звучал термин **конкатенация**, который важен для выполнения данного задания. Под конкатенацией обычно понимают "склеивание" двух строк в одну длинную строку, например, конкатенация строк "Данные из файла " и "data.csv" даст строку "Данные из файла data.csv".

Подсказка: настоятельно рекомендуем изучить примеры скриптов – в них есть большая часть решения!

Write text answer

Correct answer from 13,935 learners
Total 44% of tries are correct

Yes!

set xtics ("point 1, value ".x1 x1, "point 2, value ".x2 x2, "point 3, value ".x3 x3)

Рис. 4.30: Задание 28

`plot 'data.csv' using 1:2` даст ошибку:

```
warning: Skipping data file with no valid points ^ x range is
invalid
```

Скорее всего причиной такого поведения является тот факт, что формат CSV содержит строки, где столбцы разделены запятой? Содержимое файла:

1,21
2,22
3,23
4,24
5,25
6,26
7,27
8,28

9,29

10,30

The screenshot shows a course navigation bar on the left with sections like 'Введение', 'Работа на сервере', and 'Продвинутые темы'. The main content area displays a task titled '3.6 Строим графики в gnuplot'. It shows a progress bar at 10 out of 10 steps passed and 7 out of 7 points received. A message encourages users to write a review if they passed more than 80%. Below the message is a detailed description of what happened when files were moved. A 'Write text answer' section contains a correct answer from 12,854 learners, with a note that 47% of tries are correct. A code snippet is provided:

```
a=a+1  
zrot=(zrot+350)%360  
set view xrot,zrot  
splot -x**2-y**2  
pause 0.1  
if (a>50) reread
```

Рис. 4.31: Задание 29

Сначала идет команда установки подписей, а потом в скобках:
подпись - пробел - переменная с координатой - запятая
Повторяется это количество раз соответствующее числу переменных, и без
запятой (в случае с последней переменной)
А подпись в свою очередь получается конкатенацией текста из задания и пере-
менной с координатой.

The screenshot shows a Stepik course titled 'Введение в Linux'. The course progress is 117/125. The current step is '3.7 Разное' (3.7 Miscellaneous), which has 15 out of 15 steps passed and 7 out of 7 points received. A message says 'You passed more than 80%, write a review' with options 'Create review' and 'No thanks'. The task description asks for the command to set file permissions for 'file.txt' to 'rwxrw-r--'. It includes a note: 'Примечание: запись вида команда1; команда2; команда3 означает, что в терминале последовательно выполнились все три команды (сначала команда1, затем команда2 и, наконец, команда3).'. Below this, it says 'Select all correct options from the list'. The correct answer is 'chmod 764 file.txt'. Other options listed are: chmod o-wx file.txt; chmod g-x file.txt; chmod a+wx file.txt; chmod 467 file.txt; chmod u-wx file.txt; chmod g-w file.txt; chmod rwxrw-r-- file.txt; chmod 777 file.txt. A green checkmark is next to 'chmod 764 file.txt'. At the bottom, there are 'Next step' and 'Solve again' buttons, and a message 'Your submissions You got: 1 point'.

Рис. 4.32: Задание 30

1. График строится строкой “`splot x2+y2`”.
2. Вращение задается строкой “`zrot=(zrot+10)%360`”. Значит, смещение вперед (которое было изначально) можно также задать строкой “`zrot=(zrot+360+10)%360`” или иначе говоря “`zrot=(zrot+370)%360`”. А теперь посмотрим на наше требование - чтоб вращалось в другую сторону, значит, по аналогии, необходимо вместо перебора на 10 сделать недобор. “`zrot=(zrot+350)%360`”
3. Стока “`pause 0.2`” ставит выполнение на паузу на определенный промежуток времени. В задании сказали перерисовывать чаще, значит пауза должна быть меньше.

Введение в Linux
Course progress: 117/125

1 Введение

- 1.1 Общая информация о к...
- 1.2 Как установить Linux
- 1.3 Осваиваем Linux
- 1.4 Terminal: основы
- 1.5 Запуск исполняемых ф...
- 1.6 Ввод / вывод
- 1.7 Скачивание файлов из...
- 1.8 Работа с архивами
- 1.9 Поиск файлов и слов в...

2 Работа на сервере

- 2.1 Знакомство с сервером
- 2.2 Обмен файлами
- 2.3 Запуск приложений
- 2.4 Контроль запускаемых...
- 2.5 Многопоточные прило...
- 2.6 Менеджер терминалов...
- 2.7 Как установить Linux: р...

3 Продвинутые темы

- 3.1 Текстовый редактор vim
- 3.2 Скрипты на bash: осно...
- 3.3 Скрипты на bash: ветв...
- 3.4 Скрипты на bash: разн...
- 3.5 Продвинутый поиск и ...
- 3.6 Строим графики в gnu...
- 3.7 Оптимизация

Предположим вы использовали команду `sudo` для создания директории `dir`. По умолчанию для `dir` были выставлены права доступа `rwxr-xr-x` (владелец `root`, группа `root`). Таким образом никто кроме пользователя `root` не может ничего записывать в эту директорию, например, не может создавать файлы в ней.

После выполнения какой команды `user` из группы `group` всё-таки сможет создать файл внутри `dir`? Укажите **все верные** варианты ответов!

Примечание: считаем, что все команды выполняются от имени `user`, если явно не указано, что команда выполнена с `sudo`.

Примечание 2: мы выбрали пример с директорией, а не с файлом не случайно. Дело в том, что если создать при помощи `sudo` файл с правами `rw-r-t-r-` в директории, которая принадлежит пользователю, то возникнет любопытная ситуация. С одной стороны пользователь может удалить этот файл (т.к. ему разрешено удалять **все** файлы внутри его директории) и может прочитать его содержимое (т.к. право "r" у файла установлено для всех), с другой стороны он не может этот файл редактировать (т.к. право "w" у файла есть только для `root`). При этом некоторые "умные" редакторы, например, `vim` позволят даже редактировать этот файл, но сделают они это своеобразно: через удаление оригинала и создание копии уже с нужными правами (удалять мы можем, а раз можем читать, то и копию создать не сложно). Итого получается, что несмотря на права `rw-r-t-r-`, пользователь может сделать с этим файлом почти всё что угодно!

В случае же, когда речь идет о директории созданной `root`, ситуация будет проще: пользователь сможет смотреть её содержимое (у него есть право "r"), но удалять и создавать файлы в ней не сможет (права "w" у него нет).

Важно отметить, что директории в Linux это в каком-то смысле *файлы*. Содержимое такого "файла" – это записи о файлах и поддиректориях этой директории (грубо говоря их названия). Таким образом, право "r" у директории дает возможность просматривать "записи", т.е. просматривать её состав. Право "w" у директории дает возможность удалять/добавлять новые "записи", т.е. удалять/создавать файлы/поддиректории в ней.

На самом деле и это еще не все. Существует так называемый *sticky bit* (атрибут файла или директории), выставленное которого меняет описание выше поведения. Файлы (или директории) с таким атрибутом сможет удалять только их владелец вне зависимости от прав, установленных на директории, в которой эти файлы (или директории) лежат.

Отдельное спасибо слушателю курса **Alexey Antipovsky** за помощь в оформлении **Примечания 2**!

Select all correct options from the list

You're right!

Correct answer from 14,683 learners
Total 15% of tries are correct

You've solved a complex problem, congratulations! Now you can help other learners in

Рис. 4.33: Задание 31

- r - чтение;
- w - запись;
- x - выполнение;
- s - выполнение от имени суперпользователя (дополнительный);
- u - владелец файла;
- g - группа файла;
- o - все остальные пользователи;
- 0 - никаких прав;
- 1 - только выполнение;

- 2 - только запись;
- 3 - выполнение и запись;
- 4 - только чтение;
- 5 - чтение и выполнение;
- 6 - чтение и запись;
- 7 - чтение запись и выполнение.

Рис. 4.34: Задание 32

Решений два типа:

- Сменить права гостей, добавив W
- Сделать владельцем нужную группу или пользователя, в зависимости от того, у кого из них уже есть права на W
- Помнить, что root - владелец и остальные для него - others.

The screenshot shows a Stepik course interface for 'Введение в Linux'. The sidebar lists chapters 1 and 2. Chapter 3 has 15 steps passed out of 15, with 7 points received. A text input field contains 'du -h -s'. A green button says 'Totally right.' and a green box says 'Correct answer from 16,381 learners'. A note says 'Total 53% of tries are correct'.

Рис. 4.35: Задание 33

- **wc -l** вывести количество строк
- **wc -c** вывести количество байт
- **wc -m** вывести количество символов
- **wc -L** вывести длину самой длинной строки
- **wc -w** вывести количество слов

The screenshot shows a Stepik course interface for 'Введение в Linux'. The sidebar lists chapters 1 and 2. Chapter 3 has 15 steps passed out of 15, with 7 points received. A text input field contains 'mkdir dir(1..3)'. A green button says 'Great work!' and a green box says 'Correct answer from 16,720 learners'. A note says 'Total 40% of tries are correct'.

Рис. 4.36: Задание 34

`-h, --human-readable` print sizes in human readable format (e.g., 1K 234M 2G)

`-s, --summarize` display only a total for each argument

5 Сертификат

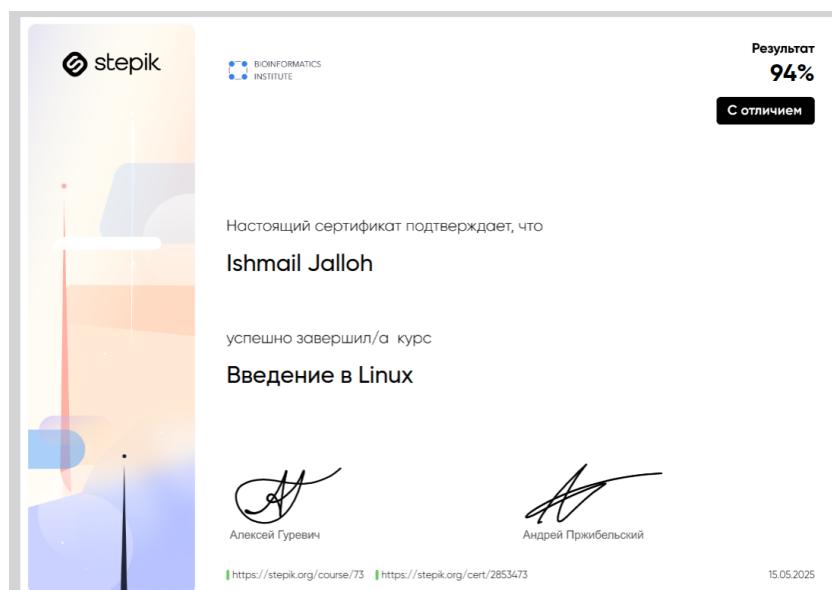


Рис. 5.1: Сертификат

6 Выводы

Я просмотрел курс и освежил в памяти навыки работы с более сложными командами в Линукс.

Список литературы

1. Введение в Linux