

Отчет о прохождении 3 этапа внешних курсов

Продвинутые темы

Рафи Кази ар, НКАбд-03-24

Содержание

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

3 Этап:

Select one option from the list

 Great!

- ☐ "q", затем "Enter"
- ☐ "q"
- ☒ " : ", затем "q", затем "Enter"
- ☐ "Q"
- ☐ "Ctrl", затем "x"

Next step

Solve again

[Your submissions](#) You got: **1 point**

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

начале), однако нажатие на **Enter** после ввода команды обозначает

Write text answer

✓ Great!

:%s/Windows/Linux

Next step

Solve again

[Your submissions](#) You got: **2 points**

Поиск и замена в редакторе работают по следующей схеме:

Для замены во всем файле можно использовать символ %.

Select all correct options from the list

✓ Great work!

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
learners
Total 29% of tries are

- ☒ Режим выделения открывается из нормального режима по нажатию "v"
- ☒ В режиме выделения можно использовать команды перемещения (например, W, e, \$, и др.)
- ☒ Выйти из режима выделения можно, нажав клавишу Esc два раза
- ☐ Чтобы выйти из режима выделения, нужно ввести :q
- ☒ Когда вы находитесь в режиме выделения, внизу редактора горит надпись -- VISUAL -- (или -- ВИЗУАЛЬНЫЙ РЕЖИМ)
- ☒ В режиме выделения можно использовать команды d (удалить) и y (скопировать)

Next step

Solve again

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.

yy (также Y) — копирование текущей строки в буфер;

Select one option from the list

 Great work!

- ☐ Только из набора A
- ☐ Только из набора B
- ☒ Только из набора C
- ☐ Из наборов B и C
- ☐ Из наборов A и C

Next step

Solve again

[Your submissions](#) You got: **1 point**

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

Select one option from the list



Absolutely right.

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

Next step

Solve again

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

- ☒ _variable
- ☒ variable_123
- ☒ variable
- ☒ VARiable
- ☒ variable123
- ☐ variab\$\$le
- ☐ var@iable

Next step

Solve again

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

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

[Next step](#)[Solve again](#)

[Your submissions](#) You got: **3 points**

1. Задаю общую часть в каждом выводе - слово "student": v=student
2. Выполняем команды для разных аргументов.
3. res - это результат для вывода
4. echo "\$res" - вывести результат

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

Next step

Solve again

[Your submissions](#)

You got: **1 point**



Great work!

Now you have access to the [Forum of Solutions](#) where you can discuss your solution.

You've solved a complex problem, congratulations! Now you can help other learners answering their questions, or compare your solution with others on [solution forum](#).

```
1 child=16
2 adult=25
3 stdout=0
4
5 while [[ $stdout != 1 ]]
6     do
7         echo "enter your name: "
8         read name
9         if [[ (-z $name) || ($name = 0) ]] ;then
10             echo "bye"
11             stdout=1
```

Рис. 1: 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, простимся
        echo "bye"
        stdout=1
    elif [[ -n $name ]]; then #А вот если имя нормальное
        while [[ $stdout != 1 ]] ;do
            echo "enter your age: " #То пусть вводит возраст
            read age #Считываем возраст
            if [[ ($age -eq 0) || (-z $age) ]] ;then #Если возраст 0
или строка пуста - прощаемся
                echo "bye"
                stdout=1
            elif [[ $age -le $child ]] ;then #Если меньше или равен
ребенку, то ребенок
```



```

        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, то подросток.
            echo "$name, your group is youth" ;fi
        fi ;break
    done ;fi
done

```

☒ let "a = a + b"

☒ let "a=\$a+\$b"

☒ let "a+=b"

☒ let a=\$a+\$b

☒ let a=a+b

Next step

Solve again

[Your submissions](#) You got: ...

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)

Select one option from the list

✓ Absolutely right.

- ☐ Код возврата команды pwd (0 в случае успешного выполнения и не 0 в случае ошибок)
- ☐ `pwd`
- ☐ pwd
- ☐ /home/bi/Documents
- ☒ /home/bi

Next step

Solve again

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

Write text answer

✓ Absolutely right.

You've solved a complex problem, congratulations! Now you can help other learners answering their questions, or compare your solution with others on [solution forum](#).

counters are and 110

Next step

Solve again

[Your submissions](#) You got: ...

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.

✓ Yes!

Now you have access to the [Forum of Solutions](#) where you can discuss your solution with others.

```
1 # put your shell (bash) code here
2 while [ true ]
3 do
4     read n1 n2
5     if [ -z $n1 ]; then
6         echo "bye"
7         break
8     else
9         gcd () {
10             remainder=1
11             if [ $n2 -eq 0 ]
12             then
13                 echo "bve"
```

✓ Correct.

- ☐ Eddard_Stark_biography.txt
- ☐ star_trek_OST.mp3
- ☐ stardust.mpeg
- ☒ Star_Wars.avi
- ☒ STARS.txt

Next step

Solve again

[Your submissions](#) You got: **1 point**

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](#).

- ☐ Опция -path используется только для поиска директорий, а -name только для поиска файлов
- ☐ Опции -path и -name всегда работают одинаково
- ☐ Опция -path аналогична -name, но игнорирует размер букв (строчные/прописные) в именах
- ☐ Если заменить в команде поиска -name, на -path, то результат поиска всегда останется тем же
- ☒ Если заменить в команде поиска -name, на -path, то результат поиска иногда может отличаться

Next step

Solve again

[Your submissions](#) You got: **1 point**

Калькулятор выглядит обычно - мы вводим два числа, пишем, что с ними надо сделать, и потом, учитывая случаи ошибок, выводим результат.

- ☐ Только file3
- ☐ Все кроме file1
- ☐ Все три файла
- ☐ Только file2
- ☒ Все кроме file3

Next step

Solve again

[Your submissions](#) You got: ...

-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.

Select one option from the list



Good job.

- ☐ `grep -C 1 "word" file.txt > results.txt`
- ☒ `results.txt` будет одинакового размера во всех случаях
- ☐ Все, кроме `grep "word" file.txt > results.txt`
- ☐ `grep -A 1 "word" file.txt > results.txt`
- ☐ `grep -A 1 "word" file.txt > results.txt` и `grep -B 1 "word" file.txt > results.txt`

Next step

Solve again

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

You've solved a complex problem, congratulations! Now you can help other learners in [community](#) answering their questions, or compare your solution with others on [solution forum](#).

```
sed 's/[A-Z]\{2,\} /abbreviation /g' input.txt > edited.txt
```

Next step

Solve again

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.

✓ Fabulous answer.

Correct answer
learners
Total 32% of

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](#).

- ☐ Название "noame", нарисовано 10 точек
- ☒ Название -- первое значение из второго столбца, нарисовано 9 точек (точка из первой строки пропущена)
- ☐ 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

Next step

Solve again

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.

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

Next step

Solve again

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

Write text answer



You are right, well done!

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

Next step

Solve again

[Your submissions](#) You got: **3 points**

- 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.

Write text answer

✓ Totally right.

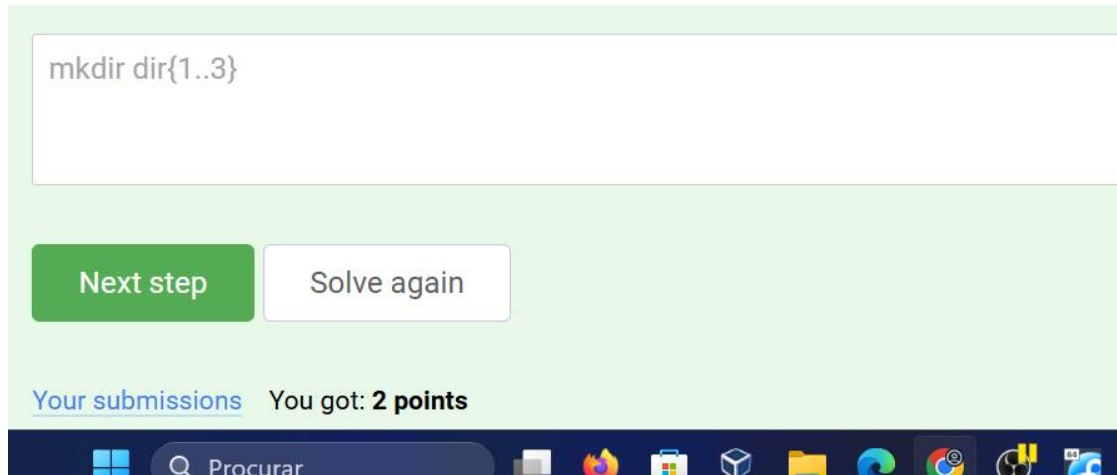


Рис. 2: Задание 32

5 Сертификат

Рис. 3: Сертификат

Рис. 3: Сертификат

6 Выводы

Я просмотрела курс и освежила в памяти навыки работы с более сложными командами в Линукс.

Список литературы

1. Введение в Linux