

# **Внешний курс**

**Продвинутые темы**

Богданюк Анна Васильевна НКАбд-01-23

# Содержание

<b>1</b>	<b>Цель работы</b>	<b>5</b>
<b>2</b>	<b>Задание</b>	<b>6</b>
<b>3</b>	<b>Выполнение лабораторной работы</b>	<b>7</b>
<b>4</b>	<b>Выводы</b>	<b>24</b>

# Список иллюстраций

3.1 Задание . . . . .	7
3.2 Задание . . . . .	8
3.3 Задание . . . . .	8
3.4 Задание . . . . .	9
3.5 Задание . . . . .	9
3.6 Задание . . . . .	10
3.7 Задание . . . . .	10
3.8 Задание . . . . .	11
3.9 Задание . . . . .	12
3.10Задание . . . . .	12
3.11Задание . . . . .	13
3.12Задание . . . . .	14
3.13Задание . . . . .	14
3.14Задание . . . . .	15
3.15Задание . . . . .	15
3.16Задание . . . . .	16
3.17Задание . . . . .	16
3.18Задание . . . . .	17
3.19Задание . . . . .	17
3.20Задание . . . . .	18
3.21Задание . . . . .	18
3.22Задание . . . . .	19
3.23Задание . . . . .	19
3.24Задание . . . . .	20
3.25Задание . . . . .	20
3.26Задание . . . . .	21
3.27Задание . . . . .	21
3.28Задание . . . . .	22
3.29Задание . . . . .	23

## **Список таблиц**

# 1 Цель работы

Целью работы является получение знания по продвинутым темам по Linux.

## **2 Задание**

1. Изучить теоритический материал
2. Пройти тесты

## 3 Выполнение лабораторной работы

q - quit. Чтобы выйти из vim (рис. 3.1).

Какую клавишу(и) нужно нажать на клавиатуре, чтобы выйти из редактора vim? Считайте, что вы только что открыли файл и вам сразу понадобилось выйти из редактора.

Выберите один вариант из списка

✓ Отличное решение!

Верно решили 32 523 учащихся  
Из всех попыток 69% верных

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

Следующий шаг Решить снова

[Ваши решения](#) Вы получили: 1 балл из 1

Рис. 3.1: Задание

В этой строке 5 “больших слов”, потому что в Strange\_ TEXT is\_here. 2=2 YES! их 5. После 10 нажатий на W курсор окажется там же, где бы он был после 10 нажатий на w, это можно проверить на практике (рис. 3.2).

При перемещении в vim "по словам" есть небольшая разница в том, используем мы маленькую (w, e, b) или большую (W, E, B) букву. Первые перемещают нас по "словам" (word), а вторые по "большим словам" (WORD). Посмотрите справку по этим перемещениям и разберитесь в чем заключается разница между word и WORD.

А для того, чтобы убедиться, что вы разобрались, отметьте ниже **все верные** утверждения про следующую строку:

```
Strange_ TEXT is_here. 2=2 YES!
```

**Примечание:** во всех утверждениях имеется ввиду, что мы находимся в редакторе vim, включен нормальный режим работы и курсор находится в самом начале строки.

**Подсказка:** чтобы вызвать **vim-справку** по, например, перемещению `w`, нужно открыть vim и ввести команду `:help w`. Вы попадете в то место справки, где описано это перемещение, а так как все перемещения описаны рядом, то двигаясь по тексту вверх и вниз можно прочитать и про `e` и про `b` и, самое главное, про word и WORD. Кроме того, можно вызвать сразу справку по термину word при помощи `:help word`. Чтобы закрыть справку, нужно ввести команду `:q`.

**Выберите все подходящие ответы из списка**

Верно решили **25 385** учащихся  
Из всех попыток **20%** верных

✓ Здорово, всё верно.

Вы решили сложную задачу, поздравляем! Вы можете помочь остальным учащимся в [комментариях](#), отвечая на их вопросы, или сравнить своё решение с другими на [форуме решений](#).

- ☐ В этой строке 12 "слов" (word)
- ☐ Чтобы попасть в конец строки, нужно совершить больше нажатий на W, чем на w
- ☒ В этой строке 5 "больших слов" (WORD)
- ☒ После 10 нажатий на W курсор окажется там же, где бы он был после 10 нажатий на w
- ☐ Нажимая только на w, нельзя переместить курсор на "."
- ☐ В этой строке 9 "больших слов" (WORD)

Следующий шаг    Решить снова

Рис. 3.2: Задание

Для замены во всем файле можно использовать символ '%'. Vim provides the `:s` (substitute) command for search and replace (рис. 3.3).

Предположим, что вы открыли файл в редакторе vim и хотите заменить в этом файле все строки, содержащие слово `Windows`, на такие же строки, но со словом `Linux`. Если в какой-то строке слово `Windows` встречается больше, чем один раз, то заменить на `Linux` в этой строке нужно **только самое первое** из этих слов.

**Какую команду** нужно ввести для этого в vim? Укажите необходимую команду целиком (т.е. **включая** ввод ":" в самом начале), однако нажатие на `Enter` после ввода команды обозначать никак **не нужно**.

**Напишите текст**

Верно решил **24 631** учащихся  
Из всех попыток **57%** верных

✓ Правильно.

`:%s/Windows/Linux`

Следующий шаг    Решить снова

[Ваши решения](#)    Вы получили: **2 балла** из 2

Рис. 3.3: Задание

В режиме выделения можно использовать команды `d` (удалить) и `y` (скопировать). Выйти из режима выделения можно, нажав клавишу Esc два раза. Режим выделения открывается из нормального режима



по нажатию “v” . В режиме выделения можно использовать команды перемещения (например, W, e, \$, и др.). Когда вы находитесь в режиме выделения, внизу редактора горит надпись – VISUAL – (или – ВИЗУАЛЬНЫЙ РЕЖИМ –) (рис. 3.4).

Мы совсем не рассказывали вам про третий режим работы vim – режим **выделения (Visual)**. Предлагаем вам ознакомиться с ним самостоятельно. Например, это можно сделать во время прохождения упражнений в vimtutor, который мы настоятельно рекомендуем вам для изучения vim!

Чтобы убедиться, что вы разобрались с этим режимом работы, отметьте, пожалуйста, **все верные** утверждения из списка ниже.

**Подсказка:** если вы не хотите проходить vimtutor целиком, то можете открыть его и поиском найти слово “Visual”. Вы попадете в задание, прохождение которого будет вполне достаточно, чтобы выполнить это задание.

Выберите все подходящие ответы из списка

Верно решили 23 497 учащихся  
Из всех попыток 29% верных

✓ Хорошая работа.

Вы решили сложную задачу, поздравляем! Вы можете помочь остальным учащимся в [комментариях](#), отвечая на их вопросы, или сравнить своё решение с другими на [форуме решений](#).

- ☐ Чтобы выйти из режима выделения, нужно ввести :q
- ☒ Режим выделения открывается из нормального режима по нажатию “v”
- ☐ Режим выделения открывается при помощи команды :visual
- ☒ В режиме выделения можно использовать команды перемещения (например, W, e, \$, и др.)
- ☒ В режиме выделения можно использовать команды d (удалить) и у (скопировать)
- ☐ Режим выделения открывается из любого другого режима по нажатию “v”

Следующий шаг

Решить снова

Ваши решения Вы получили: 2 балла из 2

Рис. 3.4: Задание

Только из набора C, т.к. у каждой оболочки свой буфер, который при выходе из нее буде записываться в файл истории (рис. 3.5).

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 попытаете при помощи стрелочек вверх/вниз перемещаться по истории набранных команд, то команды из какого набора(ов) будут появляться?

Выберите один вариант из списка

Верно решили 30 266 учащихся  
Из всех попыток 65% верных

✓ Всё получилось!

- ☐ Из наборов A и C
- ☐ Из наборов B и C
- ☐ Только из набора B
- ☒ Только из набора C
- ☐ Никакие команды появляться не будут

Следующий шаг

Решить снова

Ваши решения Вы получили: 1 балл из 1

Рис. 3.5: Задание

Можно проверить на практике, хотя все достаточно очевидно. file.txt создается в /home/bi/ (рис. 3.6).

Вы можете скачать и изучить скрипты, которые мы показали в видеофрагменте: [script1.sh](#), [script2.sh](#).

Предположим, что вы находитесь в директории `/home/bi/Documents/` и запускаете в ней скрипт следующего содержания:

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

Как будет выглядеть **абсолютный путь** до созданного файла `file1.txt` по окончании работы скрипта?

**Выберите один вариант из списка**

✓ Отличное решение!

Верно решили 29 905 учас  
Из всех попыток 76% верн

☐ /home/bi/Desktop/file1.txt

☐ Никак (файла file1.txt не будет существовать после завершения работы скрипта)

☒ /home/bi/file1.txt

☐ /home/bi/Documents/file1.txt

Следующий шаг Решить снова

[Ваши решения](#) Вы получили: 1 балл из 1

Рис. 3.6: Задание

Не может начинаться с цифры, но может начинаться с символа, точек быть не может (рис. 3.7).

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!

**Подсказка:** если все варианты ответов являются неверными, то не отмечайте ни один из них и нажимайте кнопку "Отправить"/"Submit".

**Выберите все подходящие ответы из списка**

✓ Хорошая работа.

Верно решили 27  
Из всех попыток 4

Вы решили сложную задачу, поздравляем! Вы можете помочь остальным учащимся в [комментариях](#), отвечая на их вопросы, или сравнить своё решение с другими на [форуме решений](#).

☒ variable123

☒ variable\_123

☒ \_\_variable

☒ variable

☒ \_variable

☐ var i able

☒ VARIable

Следующий шаг Решить снова

[Ваши решения](#) Вы получили: 1 балл из 1

Рис. 3.7: Задание

Две переменные (`var1=$1`, `var2=$2`) и печать сообщения `echo`. (рис. 3.8).

Вы можете скачать и изучить скрипт, который мы показали в видеофрагменте: [arguments.sh](#).

Напишите скрипт на `bash`, который принимает на вход два аргумента и выводит на экран строку следующего вида:

```
Arguments are: $1=первый_аргумент $2=второй_аргумент
```

Например, если ваш скрипт называется `./script.sh`, то при запуске его `./script.sh one two` на экране должно появиться:

```
Arguments are: $1=one $2=two
```

а при запуске `./script.sh three four` будет:

```
Arguments are: $1=three $2=four
```

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

Напишите программу. Тестируется через `stdin → stdout`

✓ Верно.

Вам доступен [Форум решений](#), где вы можете сравнить свое решение с другими или спросить совета.

Верно решили 25 053 учащихся  
Из всех попыток 41% верных

```
1 #!/bin/bash
2 var1=$1
3 var2=$2
4 echo "Arguments are: \${1=$var1} \${2=$var2}"
```

Рис. 3.8: Задание

Все подходят, проверено на практике (рис. 3.9).

Вы можете скачать и изучить скрипт, который мы показали в видеофрагменте: [branching1.sh](#).

Предположим, вы пишете скрипт на bash и хотите использовать в нем конструкцию `if` в следующем фрагменте:

```
if [[ ... ]]
then
  echo "True"
fi
```

Вы можете вписать вместо `"..."` (внутри `[[ ]]` и **не забудьте про пробелы** после `[[` и перед `]]`!) любое из перечисленных ниже условий. Однако мы просим вас выбрать только те из них, при которых `echo` напечатает на экран `True` вне зависимости от того, с какими параметрами был запущен ваш скрипт и какие в нем есть переменные.

Например, условие `0 -eq 0` **подходит**, т.к. ноль всегда равен нулю вне зависимости от аргументов и переменных внутри скрипта и на экран будет напечатано `True`. В то же время условие `$var1 -eq 0` **не подходит**, так как в переменной `var1` как может быть записан ноль (тогда будет напечатано `True`), так его может и не быть (тогда ничего напечатано не будет).

**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.

Выберите все подходящие ответы из списка

☒ Хорошая работа.

Верно решили **23 158** учащихся

Из всех попыток **16%** верных

Вы решили сложную задачу, поздравляем! Вы можете помочь остальным учащимся в [комментариях](#), отвечая на их вопросы, или сравнить своё решение с другими на [форуме решений](#).

- ☒ `$# -ge 0`
- ☒ `-e $0`
- ☒ `5 -ge 5`
- ☒ `-s $0`
- ☒ `-n $0`
- ☒ `-z ""`

Рис. 3.9: Задание

Проверено на практике, можно вручную посмотреть. Сначала four, потом four (рис. 3.10).

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`.

Выберите один вариант из списка

☒ Прекрасный ответ.

Верно решили **25 138** учащихся

Из всех попыток **64%** верных

- ☒ Сначала four, потом four
- ☐ Сначала two, потом one
- ☐ Сначала four, потом one
- ☐ Сначала one, потом two

Рис. 3.10: Задание

Запустила программу, результат 5 раз “start” и 4 раза “finish” (рис. 3.11).

Вы можете скачать и изучить скрипты, которые мы показали в видеофрагменте: [loops1.sh](#), [loops2.sh](#).

Посмотрите на фрагмент bash-скрипта:

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

Если запустить этот скрипт, то **сколько раз** на экран будет выведено слово “start”, а сколько раз слово “finish”?

Выберите один вариант из списка

☒ Всё получилось!

Верно реши  
Из всех поп

- ☐ 3 раза “start” и 3 раза “finish”
- ☒ 5 раз “start” и 4 раза “finish”
- ☐ 3 раза “start” и ни разу “finish”
- ☐ 3 раза “start” и 2 раза “finish”

Следующий шаг

Решить снова

[Ваши решения](#) Вы получили: **1 балл** из 1

Рис. 3.11: Задание

В последнем порядок =+, что неправильно, должно быть +=. В 3 и 4 нет let (рис. 3.12).

Вы можете скачать и изучить скрипты, которые мы показали в видеофрагменте: [math1.sh](#), [math2.sh](#).

Какие(ая) из предложенных ниже инструкций увеличат 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!

**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.

**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)!

Выберите все подходящие ответы из списка

Верно решили **22 116** учащихся  
Из всех попыток **20%** верных

✓ Хорошие новости, верно!

Вы решили сложную задачу, поздравляем! Вы можете помочь остальным учащимся в [комментариях](#), отвечая на их вопросы, или сравнить свое решение с другими на [форуме решений](#).

- ☒ `let a=a+b`
- ☒ `let a=$a+$b`
- ☐ `a=$a+$b`
- ☐ `a+= $b`
- ☐ `let "a+=b"`

Рис. 3.12: Задание

`/home/bi`, т.е. используется `pwd`, т.к. в коде написано “`pwd`” с косыми кавычками (рис. 3.13).

Вы можете скачать и изучить скрипт, который мы показали в видеофрагменте: [programs.sh](#).

Пусть вы находитесь в директории `/home/bi/Documents/` и запускаете в ней скрипт следующего содержания:

```
#!/bin/bash
cd /home/bi/
echo "`pwd`"
```

Что в этом случае выведет команда `echo` на экран?

Выберите один вариант из списка

Верно реш  
Из всех по

✓ Отличное решение!

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

Следующий шаг

Решить снова

[Ваши решения](#) Вы получили: **1 балл** из 1

Рис. 3.13: Задание

Проверено на практике (рис. 3.14).

Мы рассказали, что можно проверить код возврата внешней программы прямо в конструкции `if` при помощи `if `program` options arguments` (действия внутри `if` выполняются, если программа закончилась с кодом 0). Однако это **не всегда правда!** Если запуск внешней программы выводит что-то в `stdout`, то в проверку `if` поступит именно этот вывод, а не код возврата! Вы можете убедиться в этом, написав простой `bash`-скрипт с использованием, например, `if `pwd``.

Однако как быть, если хочется всё-таки запустить программу `program`, которая пишет что-то в `stdout` и потом выполнить какие-то действия если ее код возврата равен 0? Выберите **все верные** утверждения или правильно работающие конструкции `if`.

**Примечание:** во всех вариантах ответов, где есть кавычка, используется именно **косая кавычка** (```), а не обычная (`"`) или двойная (`'`).

Выберите все подходящие ответы из списка

Верно решили **21 426** учащихся  
Из всех попыток **20%** верных

☒ Верно.

Вы решили сложную задачу, поздравляем! Вы можете помочь остальным учащимся в [комментариях](#), отвечая на их вопросы, или сравнить своё решение с другими на [форуме решений](#).

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

Следующий шаг

Решить снова

[Ваши решения](#) Вы получили: **1 балл** из 1

Рис. 3.14: Задание

Первая переменная локальная, поэтому будет пустая строка, вторая – сумма арифметической прогрессии от 1 до 10 (55), но еще умноженная на 2, т.е. 110 (рис. 3.15).

Вы можете скачать и изучить скрипты, которые мы показали в видеофрагменте: [functions1.sh](#), [functions2.sh](#).

Посмотрите на функцию из `bash`-скрипта:

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

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.

**Подсказка:** 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.

Напишите текст

Верно решили **20 009** учащихся  
Из всех попыток **28%** верных

☒ Всё правильно.

Вы решили сложную задачу, поздравляем! Вы можете помочь остальным учащимся в [комментариях](#), отвечая на их вопросы, или сравнить своё решение с другими на [форуме решений](#).

counters are and 110

Рис. 3.15: Задание

`-iname` значит, что большая и маленькая буква - не важно. Также `star` должно стоять в начале названия файла (рис. 3.16).

Пусть в директории `/home/bi` 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/bi -iname "star*"`, но **НЕ найдет** команда `find /home/bi -name "star*"` ?

Выберите все подходящие ответы из списка

Верно решили 20 547 учащихся  
Из всех попыток 36% верных

☒ Здорово, всё верно.

☐ stardust.mpeg  
☐ star\_trek\_OST.mp3  
☒ Star\_Wars.avi  
☐ Eddard\_Stark\_biography.txt  
☒ STARS.txt

Следующий шаг    Решить снова

[Ваши решения](#)    Вы получили: 1 балл из 1

Рис. 3.16: Задание

В справочнике по `find` можно найти ответ. Если заменить в команде поиска `-name`, на `-path`, то результат поиска иногда может остаться таким же. В некоторых случаях `find` с `-name` найдет больше файлов, чем `find` с таким же запросом, но с `-path` (рис. 3.17).

Задание на понимание работы опций `-path` и `-name` команды `find`. Отметьте **все верные** утверждения из перечисленных ниже.

Выберите все подходящие ответы из списка

Верно решили 18 450 учащихся  
Из всех попыток 22% верных

☒ Хорошие новости, верно!

Вы решили сложную задачу, поздравляем! Вы можете помочь остальным учащимся в [комментариях](#), отвечая на их вопросы, или сравнить своё решение с другими на [форуме решений](#).

☒ Если заменить в команде поиска `-name`, на `-path`, то результат поиска иногда может остаться таким же  
☒ В некоторых случаях `find` с `-name` найдет больше файлов, чем `find` с таким же запросом, но с `-path`  
☐ Опция `-path` аналогична `-name`, но игнорирует размер букв (строчные/прописные) в имени файла  
☐ Если заменить в команде поиска `-name`, на `-path`, то результат поиска всегда останется неизменным  
☐ Опция `-path` используется только для поиска директорий, а `-name` только для поиска файлов

Следующий шаг    Решить снова

[Ваши решения](#)    Вы получили: 1 балл из 1

Рис. 3.17: Задание

Текущий каталог - это `depth=1`, а остальное считается просто: `/home/bi` -> `depth=1`, `/home/bi/dir1` -> `depth=2`, `/home/bi/dir1/dir2` -> `depth=3` (рис. 3.18).



Предположим, что в директории `/home/bi/` есть следующая структура файлов и поддиректорий:

```
/home/bi/  
├── dir1  
│   ├── file1  
│   └── dir2  
│       ├── file2  
│       └── dir3  
│           └── file3
```

Какие(ой) из трех файлов (`file1`, `file2`, `file3`) будут найдены по команде `find /home/bi -mindepth 2 -maxdepth 3 -name "file*" ?`

Выберите один вариант из списка

☒ Верно. Так держать!

Верно решили 20 711 учащихся  
Из всех попыток 41% верных

- ☐ Все кроме file1  
☐ Только file2  
☐ Только file1  
☐ Ни один файл найден не будет  
☒ Все кроме file3

Следующий шаг

Решить снова

Ваши решения Вы получили: 1 балл из 1

Рис. 3.18: Задание

-A num -after-context=num, -B num -before-context=num, -C num -num -context=num (рис. 3.19).

Задание на понимание работы опций `-A`, `-B` и `-C` команды `grep`. Пусть у вас есть файл `file.txt` из 10 строк, причем в каждой строке есть слово `"word"`. Если вы выполните на этом файле команды:

```
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
```

то какая(ие) из них создаст файл `results.txt` наибольшего размера?

Выберите один вариант из списка

☒ Правильно.

Верно решили 20 237 учащихся  
Из всех попыток 41% верных

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

Следующий шаг

Решить снова

Ваши решения Вы получили: 1 балл из 1

Рис. 3.19: Задание

Все, что заканчивается `buntu` подходит (рис. 3.20).

Предположим, что в файле `text.txt` записаны строки, показанные среди вариантов ответа. Отметьте только те из них, которые выведет на экран команда `grep -E "[xk\XKL]?[uU]buntu$" text.txt`.

Выберите все подходящие ответы из списка

✓ Прекрасный ответ.

Верно решили **18 768** учащихся  
Из всех попыток **23%** верных

Вы решили сложную задачу, поздравляем! Вы можете помочь остальным учащимся в [комментариях](#), отвечая на их вопросы, или сравнить свое решение с другими на [форуме решений](#).

- ☒ Hmm, XKLubuntu
- ☐ Mac OS X 10.9, Windows XP, Ubuntu 12.04
- ☒ The best OS is Xubuntu
- ☒ Linux is not always Ubuntu
- ☒ Ubuntu is better than Ubuntu
- ☒ Mac OS X, Windows, Ubuntu

Следующий шаг

Решить снова

[Ваши решения](#) Вы получили: **2 балла** из 2

Рис. 3.20: Задание

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. (рис. 3.21).

Что произойдет, если в команде `sed -n "[a-z]*/p" text.txt` не указывать опцию `-n` ?

Выберите один вариант из списка

✓ Отлично!

- ☒ Каждая строка будет выведена два раза
- ☐ Появится сообщение об ошибке
- ☐ На экран будет выведено всё содержимое файла `text.txt`
- ☐ На экран ничего не напечатается

Следующий шаг

Решить снова

[Ваши решения](#) Вы получили: **1 балл** из 1

Рис. 3.21: Задание

У `sed` опция `-r`, пробелы в `[]`, применить к 2 и более буквам, в конце `/g` (рис. 3.22).

Запишите в форму ниже инструкцию `sed`, которая заменит все "аббревиатуры" в файле `input.txt` на слово "abbreviation" и запишет результат в файл `edited.txt` (на экран при этом ничего выводить не нужно). Обратите внимание, что в инструкции должны быть указаны и сам `sed`, и оба файла!

Под "аббревиатурой" будем понимать слово, которое удовлетворяет следующим условиям:

- состоит только из больших букв латинского алфавита,
- состоит из хотя бы двух букв,
- окружено одним пробелом с каждой стороны.

При этом будем считать, что в тексте **не может быть две "аббревиатуры" подряд**. Например, текст `" YOU YOU and YOU!"` является **некорректным** (в нем есть две "аббревиатуры", но они идут подряд) и на таких примерах мы проверять вашу инструкцию **не будем**.

**Пример:** если у вас был текст `"Hi, I heard these songs by ABBA, TLA and DM !"`, то он должен быть преобразован в `"Hi, I heard these songs by ABBA, abbreviation and abbreviation !"`.

**Примечание:** после вашей замены "аббревиатуры" на слово "abbreviation" количество пробелов в тексте **не должно меняться!**

**Внимание!** Во время проверки **мы не запускаем команду**, которую вы ввели на реальном файле с "аббревиатурами" (это небезопасно, можно же ввести `rm -rf /*`)! Вместо этого мы сперва анализируем структуру вашей инструкции (например, что в ней использован именно `sed` и сделано это ровно один раз, что на вход подается `input.txt`, а результат будет записан в `edited.txt` и т.д.), а затем **запускаем её смысловую часть** (т.е. поиск по регулярному выражению и замена на "abbreviation") на тестовых примерах. К сожалению, наш запуск **не идеально повторяет** `sed`, но он очень близок к нему. Главная "несовместимость" заключается в том, что наша проверка не понимает идущие подряд символы, отвечающие за количество повторений (т.е. `*`, `+`, `?` и `{}`). Однако эту "несовместимость" легко исправить указав при помощи "(" и ")" какой из символов к чему относится! Например, регулярное выражения `a+?` (ноль или один раз по одной или более букве "a") нужно записать как `(a+)?` (при этом запись `(a)+?`, конечно же, не поможет).

Напишите текст

✓ Всё получилось!

Верно решили **16 632** учащихся  
Из всех попыток **34%** верных

Вы решили сложную задачу, поздравляем! Вы можете помочь остальным учащимся в [комментариях](#), отвечая на их вопросы, или сравнить своё решение с другими на [форуме решений](#).

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

Рис. 3.22: Задание

The `persist` option tells `gnuplot` to leave these windows open when the main program exits (рис. 3.23).

Вы можете скачать и попробовать применить `gnuplot` к файлу, который мы показали в видеофрагменте: [authors.txt](#).

Какую опцию нужно указать при запуске `gnuplot`, чтобы при его закрытии не были автоматически закрыты и все нарисованные в нём графики?

Выберите один вариант из списка

✓ Всё получилось!

Верно решили **18 785** учащихся  
Из всех попыток **51%** верных

- ☒ `-p, --persist`
- ☐ Такой опции не существует
- ☐ Графики и так не закрываются автоматически при закрытии `gnuplot`!
- ☐ `-raise`

Следующий шаг

Решить снова

Ваши решения Вы получили: **1 балл** из 1

Рис. 3.23: Задание

Название – первое значение из второго столбца, нарисовано 9 точек (точка из первой строки пропущена) (рис. 3.24).

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?

Выберите один вариант из списка

☒ Так точно!

Верно решили **17 975** учащихся  
Из всех попыток **32%** верных

Вы решили сложную задачу, поздравляем! Вы можете помочь остальным учащимся в [комментариях](#), отвечая на их вопросы, или сравнить свое решение с другими на [форуме решений](#).

- ☐ Название – первое значение из первого столбца, нарисовано 10 точек
- ☐ Название "no name", нарисовано 10 точек
- ☐ Название – первое значение из второго столбца, нарисовано 10 точек
- ☒ Название – первое значение из второго столбца, нарисовано 9 точек (точка из первой строки пропущена)
- ☐ Название "data.csv" using 1:2", нарисовано 10 точек

Следующий шаг

Решить снова

Ваши решения    Вы получили: **1 балл** из 1

Рис. 3.24: Задание

`set xtics ("point 1, value".x1 x1, "point 2, value".x2 x2, "point 3, value".x3 x3)` (рис. 3.25).

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 пишете `gnuplot`-скрипт и у вас в нем есть три переменные `x1`, `x2`, `x3`, в которых записаны координаты важных точек по оси OX (по возрастанию). Вы хотите, чтобы на этой оси было только три деления (т.е. три черточки) в этих самых координатах, а подписи 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!

Напишите текст

☒ Хорошая работа.

Верно решили **13 935** учащихся  
Из всех попыток **44%** верных

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

Рис. 3.25: Задание

`a=a+1 zrot=(zrot+350)%360 set view xrot,zrot splot -x2-y2 pause 0.1`  
`if (a<50) reread` (рис. 3.26).

Если вы не скачали на предыдущем шаге файлы [animated.gnu](#) и [move.rot](#), то скачайте их теперь, т.к. они понадобятся для выполнения задания.

Указанные файлы использовались в последнем видеофрагменте для создания вращающегося графика. Измените инструкции в файле `move.rot` (т.е. **добавлять** и **удалять** инструкции **нельзя!**) таким образом, чтобы:

- График **отразился зеркально** относительно горизонтальной поверхности. То есть там, где была точка (10, 10, 200), станет точка (10, 10, -200), где была точка (-10, -10, 200) станет (-10, -10, -200) и т.д. При этом точка (0, 0, 0) останется на месте.
- Изображение стало **вращаться в обратную сторону**. То есть если раньше вращалось "влево", то теперь станет "вправо".
- Вращение стало **в два раза быстрее**. То есть станет в два раза больше перерисовок графика на каждую секунду вращения.

Измененный файл загрузите в форму ниже.

**Примечание:** наша система проверки **не может** запустить на вашем файле `move.rot` программу gnuplot и сравнить полученный график с заданным. Вместо этого **мы анализируем команды**, которые вы указали в файле. Поэтому если вы видите, что ваш скрипт в gnuplot работает точно по условию, а мы отвечаем "Incorrect/Неверно", то попробуйте упростить свою модификацию `move.rot` и отправить его еще раз.

Напишите текст

✓ Здорово, всё верно.

Верно решили 12 854 учащихся  
Из всех попыток 47% верных

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

Рис. 3.26: Задание

Из `r-r-r-` получить `gwxgw-r-`. Первый будет `gwxgwxgwx`. Третий `r-r-r-` (рис. 3.27).

Какая команда(ы) установят файлу `file.txt` права доступа `gwxgw-r--`, если изначально у него были права `r--r--r--`. Укажите **все верные** варианты ответа!

**Примечание:** запись вида `команда1; команда2; команда3` означает, что в терминале последовательно выполнялись все три команды (сначала `команда1`, затем `команда2` и, наконец, `команда3`).

Выберите все подходящие ответы из списка

✓ Верно. Так держать!

Верно решили 16 484 учащихся  
Из всех попыток 21% верных

Вы решили сложную задачу, поздравляем! Вы можете помочь остальным учащимся в [комментариях](#), отвечая на их вопросы, или сравнить свое решение с другими на [форуме решений](#).

- ☐ `chmod o-wx file.txt; chmod g-x file.txt; chmod a+wx file.txt`
- ✓ ☒ `chmod a+wx file.txt; chmod o-wx file.txt; chmod g-x file.txt`
- ✓ ☒ `chmod 764 file.txt`
- ☐ `chmod u-wx file.txt; chmod g-w file.txt`
- ✓ ☒ `chmod ug+w file.txt; chmod u+x file.txt`
- ✓ ☒ `chmod u+wx file.txt; chmod g+w file.txt`

Следующий шаг

Решить снова

Ваши решения Вы получили: 1 балл из 1

Рис. 3.27: Задание

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

3.28).

доступа **rw-r-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--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--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 посмотреть 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 всё. Существует так называемый **sticky bit** (атрибут файла или директории), выставление которого меняет описанное выше поведение. Файлы (или директории) с таким атрибутом сможет удалить только их владелец вне зависимости от прав, установленных у директории, в которой эти файлы (или директории) лежат!

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

Выберите все подходящие ответы из списка

☒ Так точно!

Верно решили **14 683** учащихся  
Из всех попыток **15%** верных

Вы решили сложную задачу, поздравляем! Вы можете помочь остальным учащимся в [комментариях](#), отвечая на их вопросы, или сравнить своё решение с другими на [форуме решений](#).

- ☐ `chown user:group dir`
- ☐ `sudo chmod g+w dir`
- ☐ `sudo chown :group dir`
- ☒ `sudo chown user:group dir`
- ☒ `sudo chmod o+w dir`

Рис. 3.28: Задание

**wc** stands for word count. As the name implies, it is mainly used for counting purpose. It is used to find out number of lines, word count, byte and characters count in the files specified in the file arguments (рис. 3.29).

Отметьте какие характеристики файла можно посчитать с использованием команды `wc` .

Выберите все подходящие ответы из списка

✔ Отлично!

Вы решили сложную задачу, поздравляем! Вы можете помочь остальным учащимся в [комментариях](#), отвечая на их вопросы, или сравнить своё решение с другими на [форуме решений](#).

- ☐ Количество предложений
- ☒ Количество символов
- ☐ Количество определенных букв (например, количество букв "А")
- ☒ Количество строк
- ☒ Длину самой длинной строки

Следующий шаг

Решить снова

[Ваши решения](#) Вы получили: **1 балл** из 1

Рис. 3.29: Задание

## **4 Выводы**

Во время выполнения внешнего курса я получила знания по продвинутым темам по Linux.