Отчёт по выполнению 3 этапа внешнего курса

Операционные системы

Бережной Иван Александрович

Содержание

1	Цель работы	6
2	Задание	7
3	Выполнение работы	8
4	Выводы	36
Сп	37	

Список иллюстраций

3.1	Задание 1
3.2	Задание 2
3.3	Задание 3
3.4	Задание 4
3.5	Задание 5
3.6	Задание 6
3.7	Задание 7
3.8	Задание 8
3.9	Задание 9
3.10	Задание 10
3.11	Задание 11
	Задание 12
3.13	Условие задания 13
3.14	Задание 13
	Задание 14
3.16	Задание 15
	Задание 16
	Условие задания 17
3.19	Задание 17
3.20	Условие задания 18
	Задание 18
3.22	Задание 19
3.23	Задание 20
3.24	Задание 21
3.25	Задание 22
3.26	Задание 23
3.27	Задание 24
3.28	Задание 25
3.29	Задание 26
3.30	Задание 27
3.31	Задание 28
3.32	Залание 29

3.33	Задание 30	•													32
3.34	Задание 31														33
3.35	Задание 32														34
3.36	Задание 33														34
3.37	Задание 34														35

Список таблиц

1 Цель работы

Научиться работать с операционной системой Linux

2 Задание

Выполнить тестовые задания на основе теории

3 Выполнение работы

Выполняем третий этап: (рис. 3.1, 3.2, 3.3, 3.4, 3.5, 3.6, 3.7, 3.8, 3.9, 3.10, 3.11, 3.12, 3.13, 3.14, 3.15, 3.16, 3.17, 3.18, 3.19, 3.20, 3.21, 3.22, 3.23, 3.24, 3.25, 3.26, 3.27, 3.28, 3.29, 3.30, 3.31, 3.32, 3.33, 3.34, 3.35, 3.36, 3.37).

Проверим, какой из вариантов сработает в редакторе и выберем его.

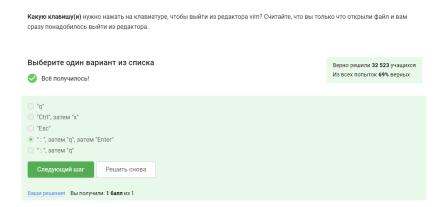


Рис. 3.1: Задание 1

Создаём текстовый файл vim tmp.txt, копируем туда строку "Strange_ TEXT is_here. 2=2 YES!" и проверяем каждый вариант.

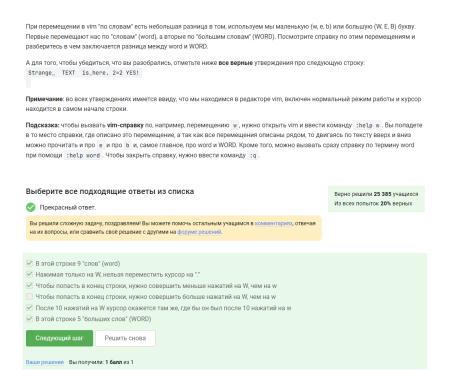


Рис. 3.2: Задание 2

Проверим каждый набор клавиш и выберем верные варианты ответа.

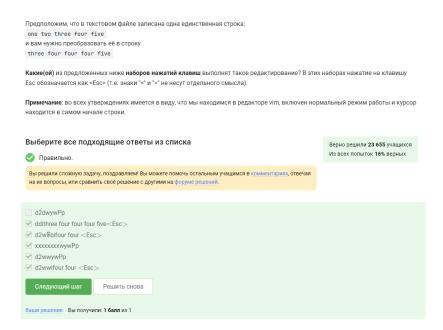


Рис. 3.3: Задание 3

Поиск и замена в Vim осуществляется командой :substitute, однако куда удобнее использовать для нее аббревиатуру :s. Общий синтаксис этой команды примерно такой:

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

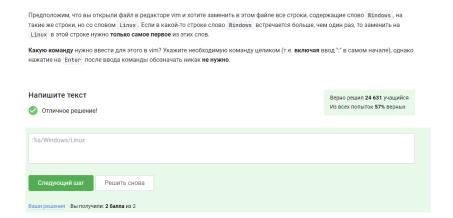


Рис. 3.4: Задание 4

Опять же, на практике проверим каждое утверждение.

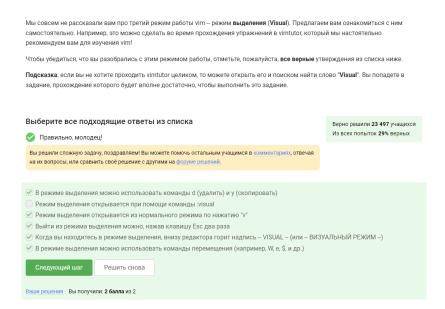


Рис. 3.5: Задание 5

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

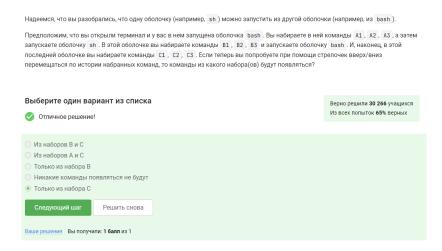


Рис. 3.6: Задание 6

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

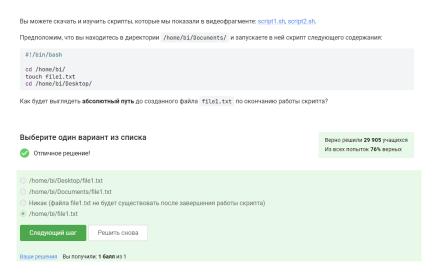


Рис. 3.7: Задание 7

В имени могут быть только буквы, цифры и подчеркивание.

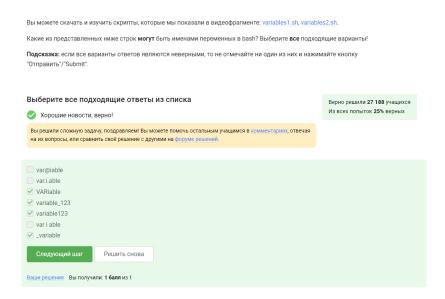


Рис. 3.8: Задание 8

\$ 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'"Argumentsare:\$1 =var1\$2=\$var2" - строка печати.

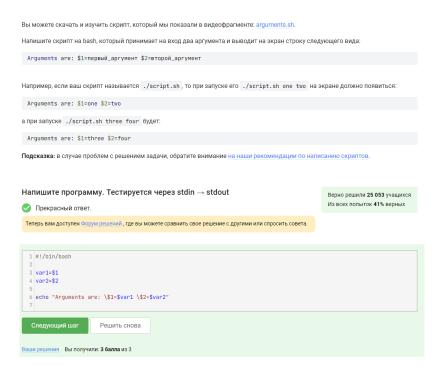


Рис. 3.9: Задание 9

\$0 #имя скрипта

- -де, # больше или равно
- -gt, (>) # больше
- -s <path> # разм файла больше 0 -le, # меньше или равно

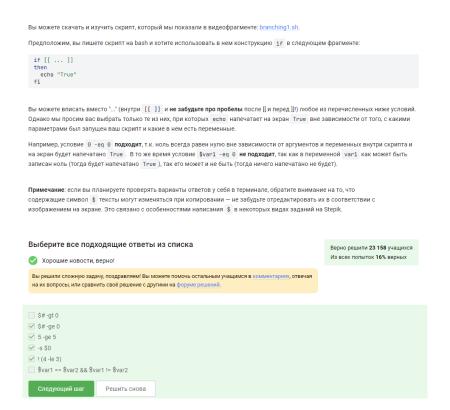


Рис. 3.10: Задание 10

- 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

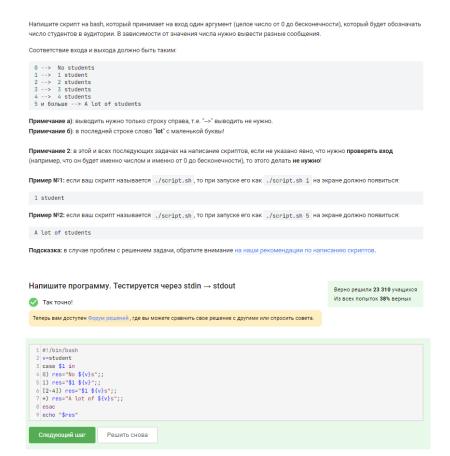


Рис. 3.11: Задание 11

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

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 echo "start"
if [[\$str > "c"]]
then
continue
fi 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 Верно решили 24 582 учащихся Из всех попыток 45% верных Здорово, всё верно. ○ 3 pasa "start" и ни pasy "finish" 5 раз "start" и 4 раза "finish" 🤍 5 раз "start" и 2 раза "finish 5 раз "start" и 5 раз "finish" Следующий шаг Решить снова Ваши решения Вы получили: 1 балл из 1

Рис. 3.12: Задание 12

Напишите скрипт на bash, который будет определять в какую возрастную группу попадают пользователи. При запуске скрипт 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 <rpynna>", где <rpynna> определяется на основе возраста по следующим правилам:

```
• младше либо равно 16: "child",
```

от 17 до 25 (включительно): "youth",

• старше 25: "**adult**".

Nº1

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:

```
./script.sh
enter your name:
Egor
enter your age:
16
Egor, your group is child
enter your name:
Elena
enter your age:
0
bye
```

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

Рис. 3.13: Условие задания 13

Рис. 3.14: Задание 13

- 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)

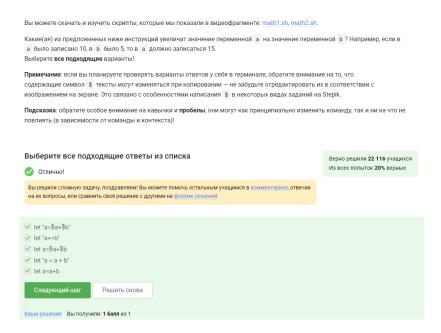


Рис. 3.15: Задание 14

ргодгамм в condition выполняет стандартный вывод в терминал. Если программа не должна выводить в терминал rm ср etc... то она возвращает true || false. Нужно перенаправить вывод (для тех, которые выводят) и ргодгамм возвращает true || false как в случае с теми прогами которые не выводят.

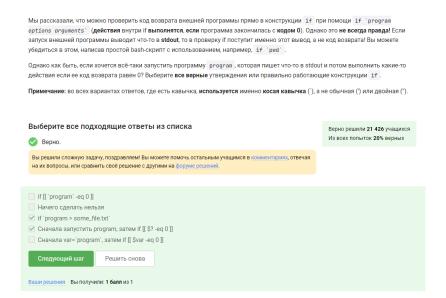


Рис. 3.16: Задание 15

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.

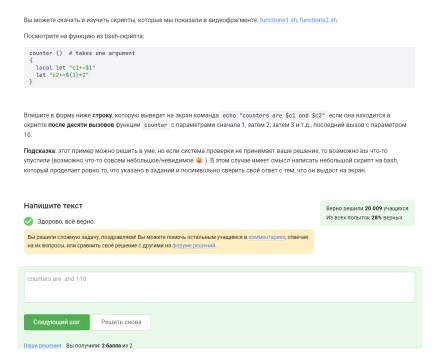


Рис. 3.17: Задание 16

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

Условие задания 17

Рис. 3.18: Условие задания 17

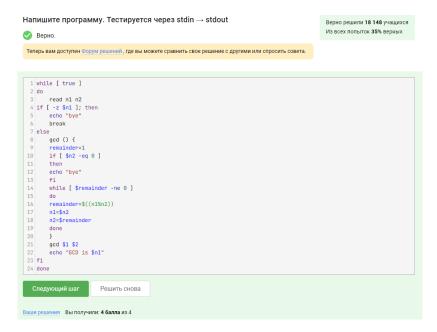


Рис. 3.19: Задание 17

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

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.
- 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 режими ожидания новой команды.
- 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:

то на экране будет: -2 error

```
10 + 1
2 ** 10
exit

то на экране будет:

11
1024
bye

Если же на вход поступит следующий файл:

3 - 5
2/10
exit
```

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

Рис. 3.20: Условие задания 18

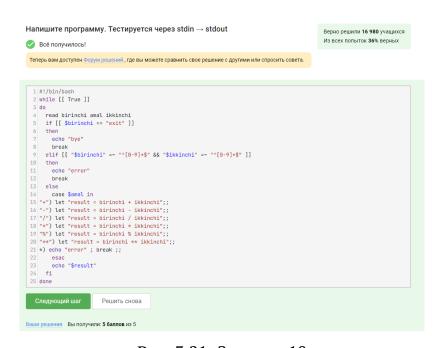


Рис. 3.21: Задание 18

-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.



Рис. 3.22: Задание 19

Проверим каждое утверждение на практике и найдём нужные ответы.

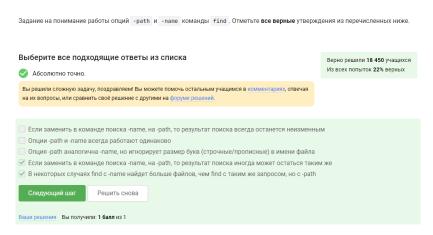


Рис. 3.23: Задание 20

Текущий каталог - это depth=1

/home/bi -> depth=1 /home/bi/dir1 -> depth=2 /home/bi/dir1/dir2 -> depth=3

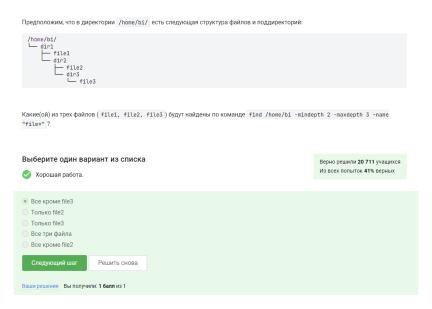


Рис. 3.24: Задание 21

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.



Рис. 3.25: Задание 22

Снова проверяем на практике.

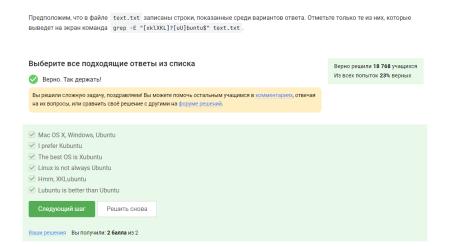


Рис. 3.26: Задание 23

Напишем без -п и посмотрим.



Рис. 3.27: Задание 24

аббревиатура ABBA отличается от двух других аббревиатур тем, что справа он неё стоит запятая без пробела: "ABBA,". 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.

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 !" ие: 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 использова 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 а+? (ноль или один раз по одной или более букве "а") нужно записать как (а+)? (при этом запись (а)+?, конечно же не поможет). Напишите текст Из всех попыток 34% верных Прекрасный ответ 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. kihkihkihikknk sed 's/[A-Z]\{2,\} /abbreviation /g' input.txt > edited.txt Следующий шаг Решить снова

Рис. 3.28: Задание 25

Перебором посмотрим и отметим нужный вариант.

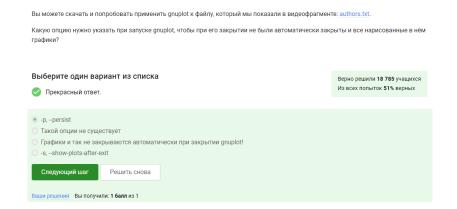


Рис. 3.29: Задание 26

Формат CSV содержит строки, где столбцы разделены запятой.

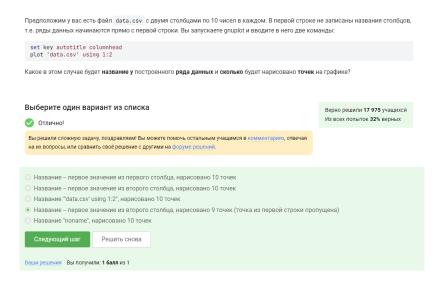


Рис. 3.30: Задание 27

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 переменной с координатой.

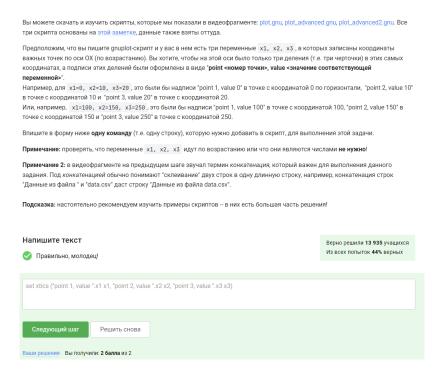


Рис. 3.31: Задание 28

- 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. Строка "pause 0.2" ставит выполнение на паузу на определенный промежуток времени. В задании сказали перерисовывать чаще, значит пауза должна быть меньше.

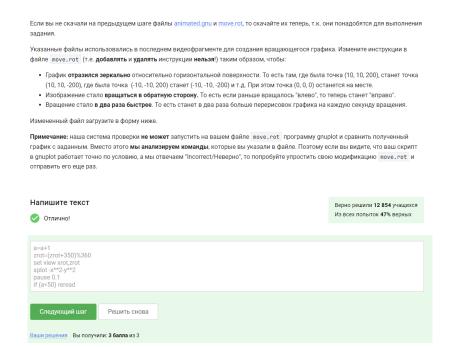


Рис. 3.32: Задание 29

• r - чтение; • w - запись; • x - выполнение; • s - выполнение от имени суперпользователя; • u - владелец файла; • g - группа файла; • о - все остальные пользователи; • 0 - никаких прав;

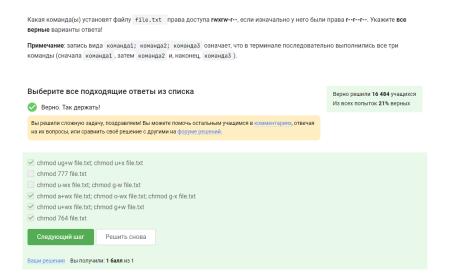


Рис. 3.33: Задание 30



Рис. 3.34: Задание 31

• 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

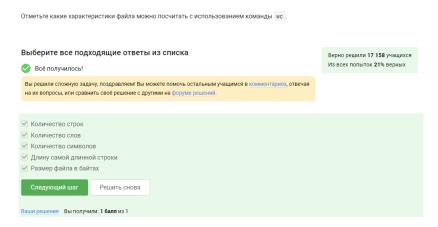


Рис. 3.35: Задание 32



Рис. 3.36: Задание 33

Создаётся три директории: dir1, dir2 и dir3.

Впишите в форму ниже максимально короткую команду (т.е. в которой минимально возможное число символов), которая позволит создать в текущей директории 3 поддиректории с именами dir1, dir2, dir3.

Если вы придумали команду, которая выполняет эту задачу, а система проверки сообщает вам "Incorrect"/"Неверно", то скорее всего вы придумали не самую короткую команду из возможных!

Напишите текст

© Отличное решение!

■ Верно решили 16 720 учащихся Из всех польтох 40% верных

■ Максительных верных

■ Максительных верных верных верных верных выполучили: 2 балла из 2

Рис. 3.37: Задание 34

4 Выводы

В ходе выполнения работы мы просмотрели курс и узнали/вспомнили возможности операционной системы Linux

Список литературы

::: Stepik