**МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ**

**БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ**

**Факультет прикладной математики и информатики**

**Кафедра информационных систем управления**

Веренич Владислав Николаевич

Командная оболочка bash: перенаправление ввода-вывода, сценарии и другие возможности

Отчет по лабораторной работе № 2

«Программирование мобильных и встраиваемых систем»

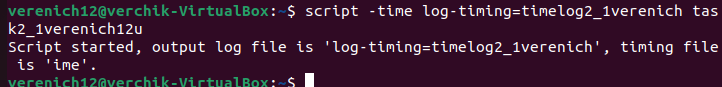
студентки 3 курса 12 группы

**Минск 2022**

**Задание 2.1. Перенаправление ввода-вывода**

Изучите примеры задания 2.1 и выполните их в ОС Ubuntu и Raspberry PI ( ранее Raspbian).

Включите ведение протокола командой script с журналом меток времени. Протокол назвать по следующему шаблону — taskXФамилияNM, где X — номер выполняемого задания, Фамилия — заменить на вашу фамилию латиницей и строчными буквами, N — номер группы, например 12 или 13, M — r — для Raspberry PI, u — для Ubuntu. Журнал меток назвать по следующему шаблону — timelogXФамилияNM, где X — номер выполняемого задания, Фамилия — заменить на вашу фамилию латиницей и строчными буквами, N — номер группы, например 12 или 13, M — r — для Raspberry PI, u — для Ubuntu.



**Примеры к заданию 2.1**

**Выполнить примеры ниже в Raspberry PI**

1. Создайте в текстовом редакторе два файла для дальнейшего использования в лабораторной работе. Файл packages1.txt должен содержать следующих восемь строк:

amanda

galeon

metacity

mozilla

postgresql

procinfo

rpmfind

squid

Файл packages2.txt должен содержать следующих шесть строк:

anaconda

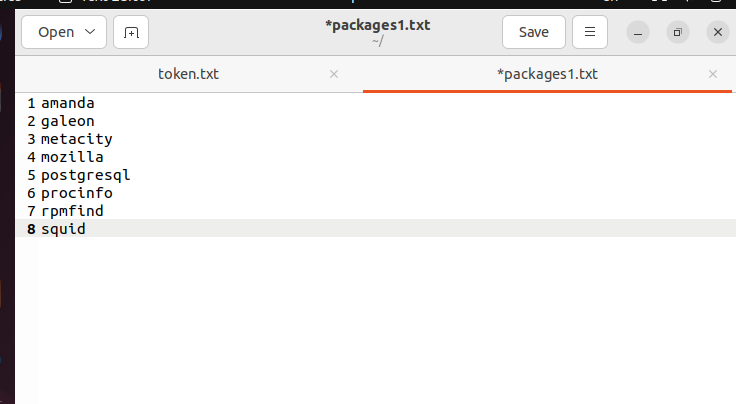
openssh

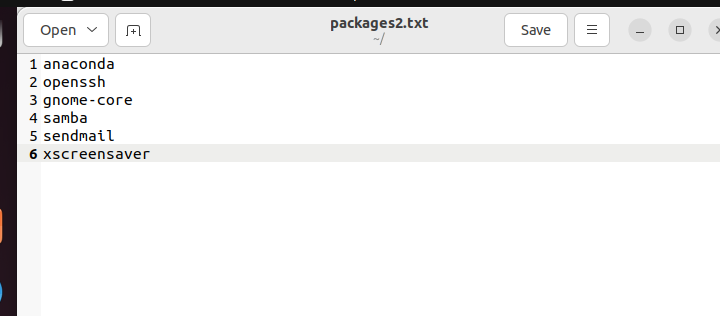
gnome-core

samba

sendmail

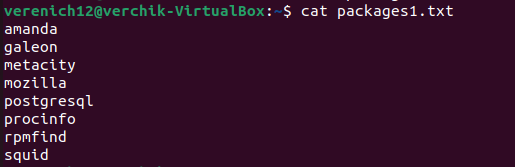
xscreensaver





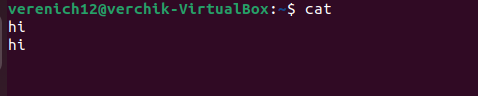
1. Для просмотра содержимого файлов предназначена команда cat. В простейшем случае эта команда принимает ввод из файла или стандартного канала ввода STDIN и посылает его в стандартный канал вывода STDOUT.

$ cat packages1.txt



1. Если команда cat не имеет аргументов, то ожидается, что она получает данные из стандартного канала ввода STDIN. Команда читает данные из канала стандартного ввода до нажатия на клавиатуре клавиш :

$ cat Type some sample text, then press return.



1. Большинство команд Linux работают как фильтры, т. е. Получают исходные данные из STDIN, делают обработку этих данных и выводят результаты в STDOUT. Например, команда tr (translate):

$ tr ’aeiou’ ’AEIOU’ Type some sample text, then press return.



1. Используя директиву ’>’, перенаправим стандартный вывод из одного файла в другой файл:

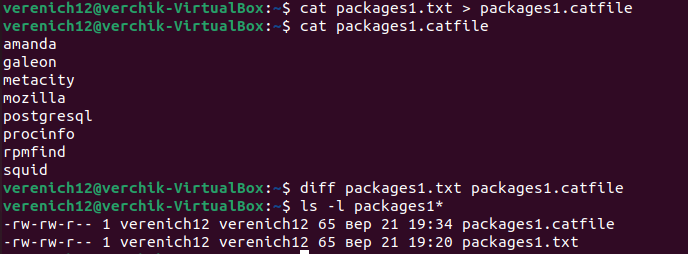
$ cat packages1.txt > packages1.catfile

$ cat packages1.catfile

Затем сверим исходный и полученный файлы:

$ diff packages1.txt packages1.catfile

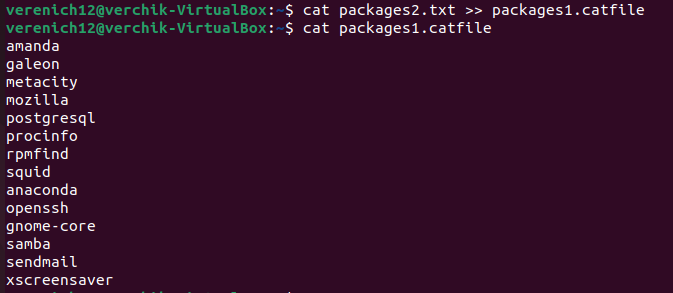
$ ls -l packages1\*



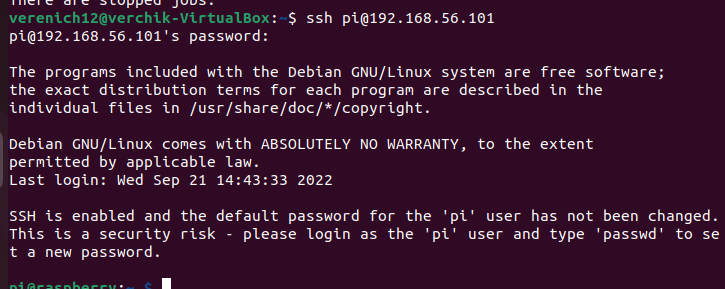
1. Для присоединения существующего файла к другому файлу предназначена директива ’>>’:

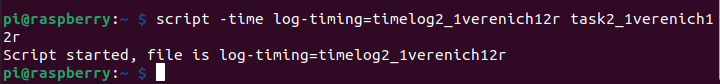
$ cat packages2.txt >> packages1.catfile

$ cat packages1.catfile



**Подключиться по ssh из Raspberry PI и выполнить примеры ниже в Ubuntu**

****

****

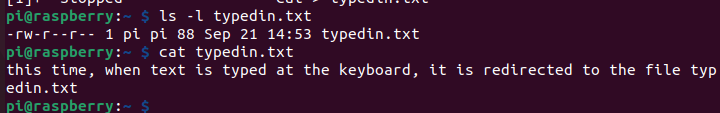
1. Если команде cat не передается аргумент и стандартный ввод перенаправлен в файл, то весь ввод с клавиатуры до нажатия клавиш будет перенаправлен в файл.

$ cat > typedin.txt

This time, when text is typed at the keyboard, it is redirected to the file typedin.txt.

$ ls -l typedin.txt

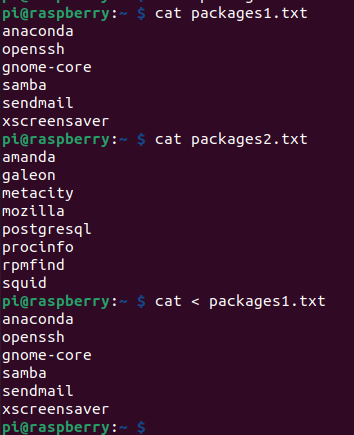
$ cat typedin.txt



1. Команда cat принимает в качестве аргумента имя файла или стандартный ввод, перенаправленный из файла. Проверьте это при помощи следующих двух команд:

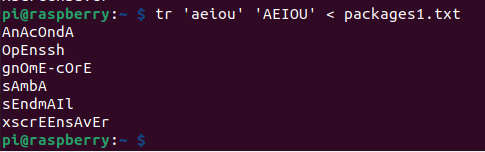
$ cat packages1.txt

$ cat < packages1.txt



1. Однако команда tr принимает ввод только из стандартного канала.

$ tr ’aeiou’ ’AEIOU’ < packages1.txt

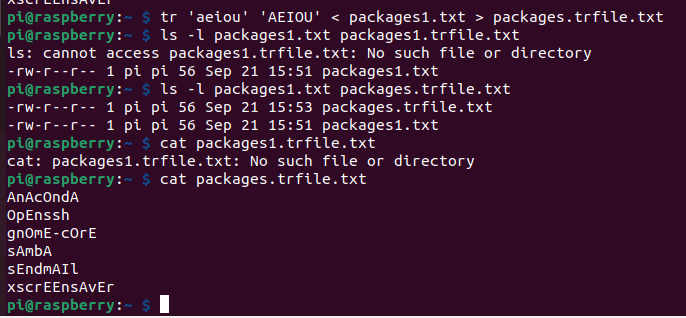


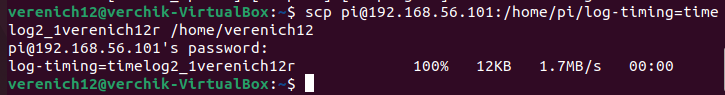
1. В следующем примере стандартный ввод и вывод одновременно перенаправляются

$ tr ’aeiou’ ’AEIOU’ < packages1.txt > packages1.trfile.txt

$ ls -l packages1.txt packages1.trfile.txt

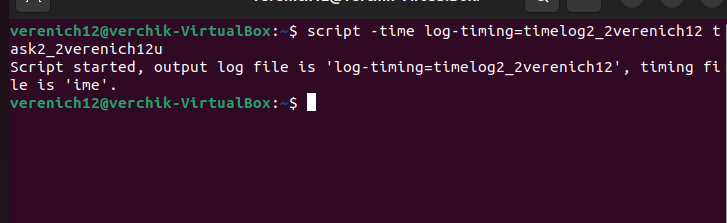
$ cat packages1.trfile.txt





**Задание 2.2. Утилита awk примеры для изучения**

Изучите примеры задания 2.2 и выполните их в ОС Ubuntu и Raspberry PI. Выполнить примеры 1-4 в Raspberry PI, подключиться по ssh к Ubuntu и выполнить примеры 5-14.



Примеры к заданию 2.2

**Пример 1.**

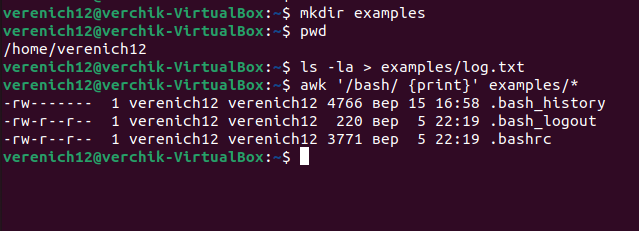
Предварительно создадим файл для обработки. Для этого выполним следующие действия:

1. Создаем каталог examples в домашнем каталоге пользователя. $ mkdir examples

2. Убедитесь, что Вы находитесь в домашнем каталоге, выполнив команду $ pwd

3. Перенаправим вывод команды ls расширенном формате в файл, например log.txt $ ls -la > examples/log.txt

4. И выполним команду для поиска данных в созданном файле или наборе файлов. Будем искать строки, содержащие слово «bash» $ awk '/bash/ {print}' examples/\*



Ищутся все файлы в каталоге examples, который находится в домашнем каталоге пользователя и в которых встречается слово bash. Результаты направляются на стандартное устройство вывода — действие {print}. В результатах выводятся строки, содержащие шаблон. Действие направления на стандартное устройство вывода {print} обычно задается как действие по умолчанию, поэтому команду примера можно переписать так: $ awk '/config/' examples/\* Результат при этом не изменится.

**Пример 3.**

1. Создайте в каталоге examples текстовый файл myfile следующего содержания:

This is a test.

This is the second test.

This is the third test.

This is the fourth test.

1. Просмотрите файл: $ cat myfile
2. Одна из основных функций awk заключается в возможности манипулировать данными в текстовых файлах. Делается это путём автоматического назначения переменной каждому элементу в строке. По умолчанию awk назначает следующие переменные каждому полю данных, обнаруженному им в записи:

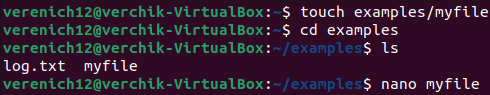
• $0 — представляет всю строку текста (запись).

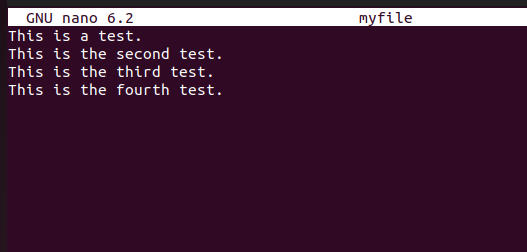
• $1 — первое поле.

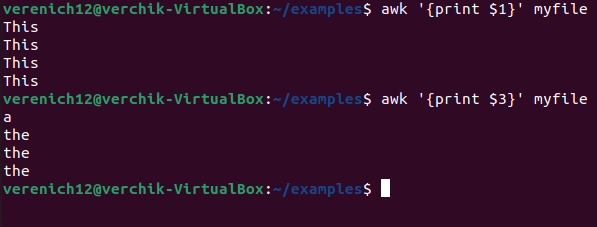
• $2 — второе поле.

• $n — n-ное поле.

2. Выполните вывод первого элемента каждой строки: 18 $ awk '{print $1}' myfile 3. Выполните вывод третьего элемента каждой строки: $ awk '{print $3}' myfile







**Пример 4.**

1. awk позволяет изменять регистр символов. В примере ниже необходимо ввести команду и две строки текста, одна из которых будет преобразована в строку в верхнем регистре:

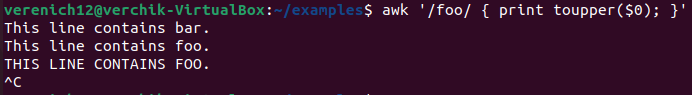
$ awk '/foo/ { print toupper($0); }'

This line contains bar.

This line contains foo.

THIS LINE CONTAINS FOO.

1. Для завершения ввода нажмите комбинацию клавиш Ctrl+C.



**Пример 5.**

1. Создать в каталоге examples текстовый файл list\_students, отсортированный по алфавиту, каждая строка которого включает: фамилию, имя, факультет, курс, рейтинговая оценка, и состоящий из следующих строкзаписей:

Асташко Иван ПИ 1 6

Бузун Евгений КБ 1 9

Кравченя Леонид ПМ 1 4

Кошкин Владимир ИН 1 7

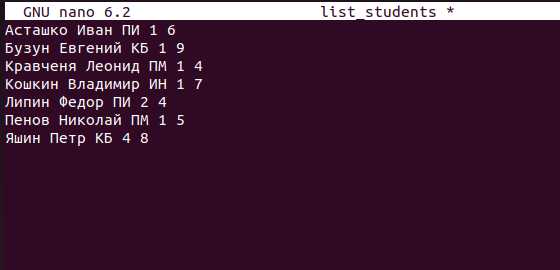
Липин Федор ПИ 2 4

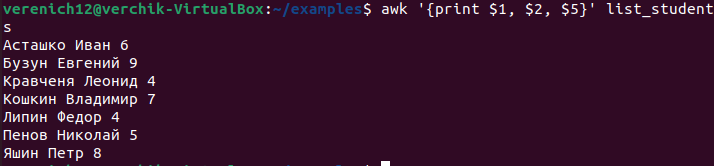
Пенов Николай ПМ 1 5

Яшин Петр КБ 4 8

1. Вывести список, состоящий из фамилий, имен и рейтинговых оценок: $ awk '{print $1,$2,$5}' list\_students







**Пример 6.**

Выбрать из исходного файла list\_students студентов с рейтинговой оценкой 8 и распечатать все значения полей для выбранных записей:

$ awk '/8/ {print $0}' list\_students

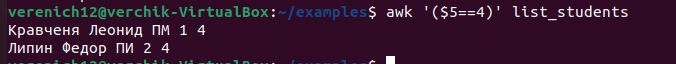
В случае, если оценка встречается в других полях, например, курс 4 и оценка 4, то условия выборки недостаточны.



**Пример 7.**

Выбрать из списка студентов, рейтинговая оценка которых 4:

$ awk '($5==4)' list\_students



**Пример 8.**

Выбрать из списка студентов специальности КБ, 1 курса:

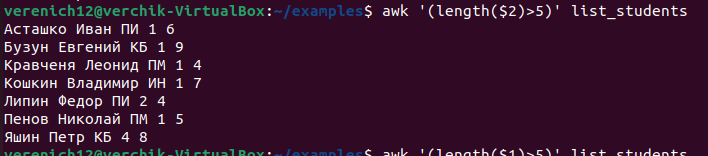
$ awk '($3=="КБ")&&($4==1)' list\_students



**Пример 9.**

Выбрать из файла со списком студентов все имена с длиной >5

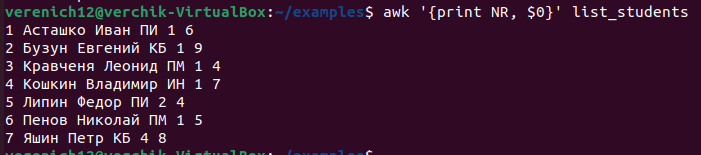
$ awk '(length($2)>5) {print}' list\_students



**Пример 10.**

Вывести данные исходного файла в виде таблицы и пронумеровать строки:

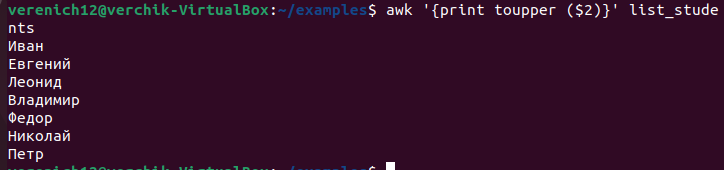
$ awk '{print NR, $0}' list\_students



**Пример 11.**

Вывести все имена студентов в верхнем регистре:

$ awk '{print toupper ($2)}' list\_students



**Пример 12.**

Посчитать суммарный балл оценок студентов:

awk '{sum += $5} END {print("SUM=",sum)}' list\_students



**Пример 13.**

Присутствует ли фамилия Леонов в списке файла list\_students?

$ awk '($1 ~ /Кравченя/)' list\_students



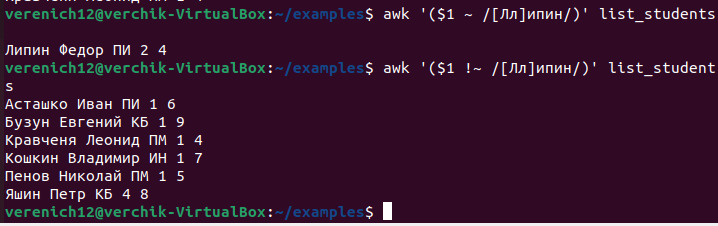
**Пример 14.**

Проверить, есть ли запись студента Липина в списке файла list\_students, причем фамилия может быть написана с прописной или строчной буквы:

$ awk '($1 ~ /[Лл]ипин/)' list\_students

Вывести все записи кроме студента Липин:

$ awk '($1 !~ /[Лл]ипин/)' list\_students



**Пример 15.**

1. Создайте текстовый файл colours.csv вида:

name,color,amount

apple,red,4

banana,yellow,6

strawberry,red,3

grape,purple,10

apple,green,8

plum,purple,2

kiwi,brown,4

potato,brown,9

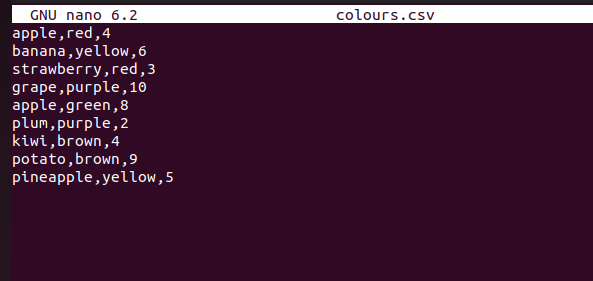
pineapple,yellow,5

1. Выберите все записи из файла colours.csv, количество которых больше 5, выполнив команду:

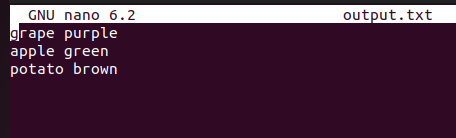
$ awk -F”,” '$3>6 {print $1, $2}' colours.csv > output.txt

В команде выше явно был указан разделитель данных, т.е. “,”. В результате выполнения команды будет создан файл, содержащий записи согласно условию.

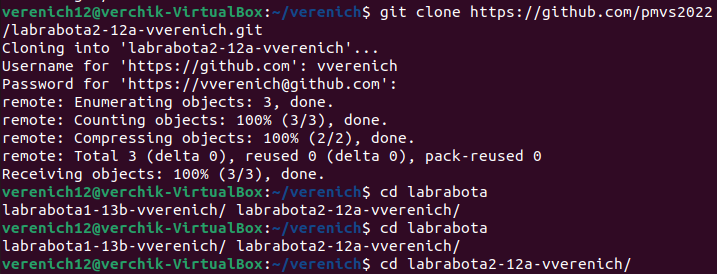


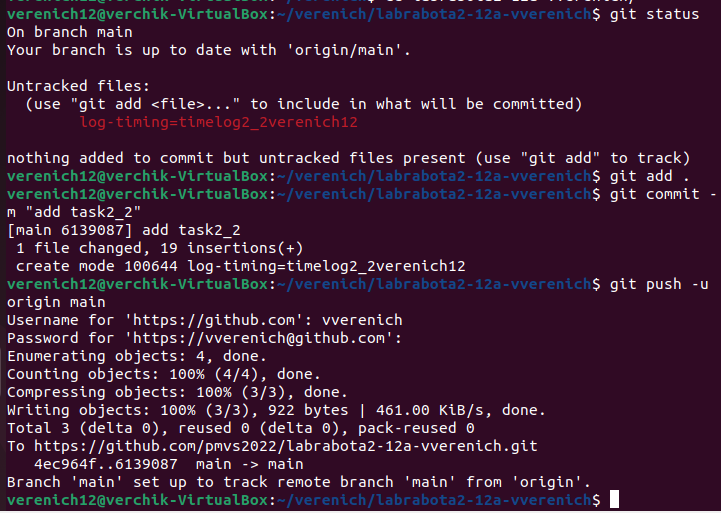






Загружаем скрипт в удаленный репозиторий:

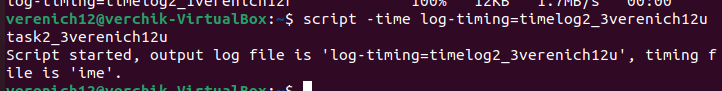




**Задание 2.3. Обработка строк файла с применением потокового редактора sed**

Изучите примеры задания 2.3 и выполните их в ОС Ubuntu и Raspberry PI. Выполнить примеры 1-4 в Raspberry PI, скопировать файл books в Ubuntu, подключиться по ssh к Ubuntu и выполнить примеры 5-8.

Включите ведение протокола командой script с журналом меток времени. Протокол назвать по следующему шаблону — taskXФамилияNM, где X — номер выполняемого задания, Фамилия — заменить на вашу фамилию латиницей и строчными буквами, N — номер группы, например 12 или 13, M — r — для Raspberry PI, u — для Ubuntu. Журнал меток назвать по следующему шаблону — timelogXФамилияNM, где X — номер выполняемого задания, Фамилия — заменить на вашу фамилию латиницей и строчными буквами, N — номер группы, например 12 или 13, M — r — для Raspberry PI, u — для Ubuntu.



**Примеры к заданию 2.3 1.**

Сохраните данные ниже в файл с именем books:

Book one.

The second book.

The third.

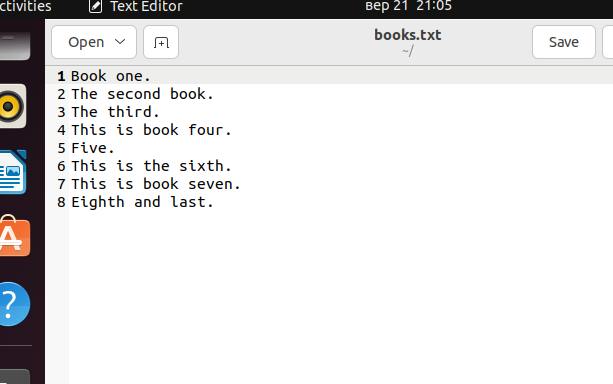
This is book four.

Five.

This is the sixth.

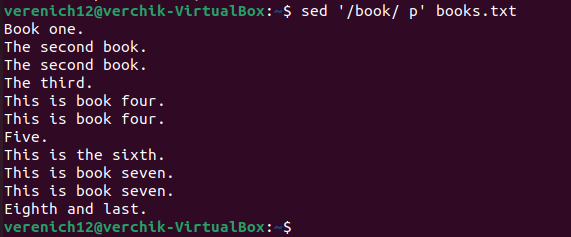
This is book seven.

Eighth and last.



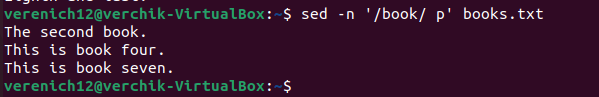
Чтобы вывести все строки файла и продублировать строки согласно шаблону, т. е. содержащие слово book выполните команду

$ sed '/book/ p' books



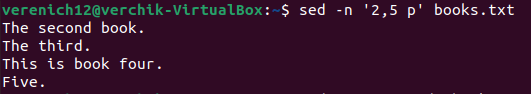
1. Чтобы выбрать определенные строки, например содержащие слово book, выполните команду

$ sed -n '/book/ p' books



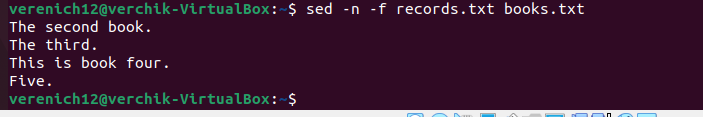
1. Чтобы вывести часть файла, например, строки с 2 по 5, выполните команду

$ sed -n '2,5 p' books



1. Для выполнения более сложных и длинных инструкций можем использовать файл программы для sed. Выполним команду из примера 4, указав параметры в файле records. Содержимое файла records: 2,5 p Пример команды из примера 4 с выполнением инструкций из файла:

$ sed -n -f records books



1. В данном примере выбираем строку 3 и используем инструкцию Добавить, чтобы добавить разделитель строк и текст «My favorite book.» к третьей строке:

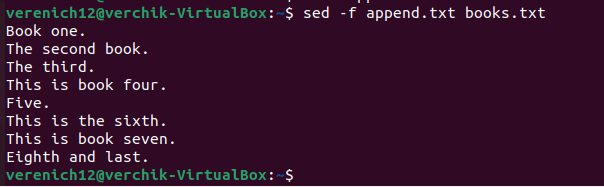
Содержимое файла appends

3 a\

My favorite book.

Пример команды с добавлением строки:

$ sed -f appends books



1. В данном примере требуется вставить разделитель строк и текст «SKARBONKA.» перед строками, в которых содержится слово «This»:

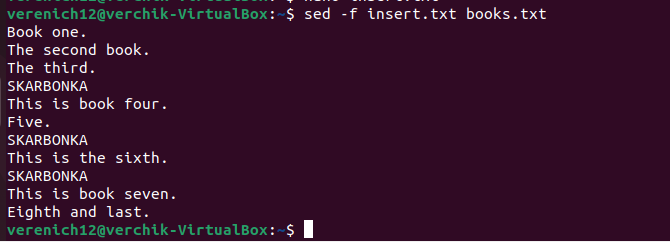
Содержимое файла insert

/This/ i\

SKARBONKA.

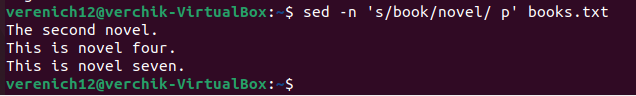
Пример команды с со вставкой строк:

$ sed -f insert books



1. В следующем примере выполняется замена слова book на novel

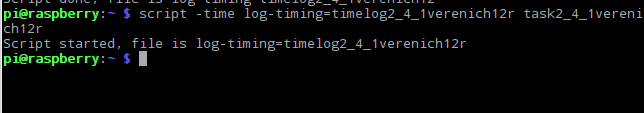
$ sed -n 's/book/novel/ p' books



**Задание 2.4. Исполнительная среда оболочки Bash**

Изучите примеры задания 2.3 и выполните их в ОС Ubuntu и Raspberry PI. Выполнить примеры 1-4 в Raspberry PI, скопировать файл books в Ubuntu, подключиться по ssh к Ubuntu и выполнить примеры 5-8.

Включите ведение протокола командой script с журналом меток времени. Протокол назвать по следующему шаблону — taskXФамилияNM, где X — номер выполняемого задания, Фамилия — заменить на вашу фамилию латиницей и строчными буквами, N — номер группы, например 12 или 13, M — r — для Raspberry PI, u — для Ubuntu. Журнал меток назвать по следующему шаблону — timelogXФамилияNM, где X — номер выполняемого задания, Фамилия — заменить на вашу фамилию латиницей и строчными буквами, N — номер группы, например 12 или 13, M — r — для Raspberry PI, u — для Ubuntu.



* + 1. **Примеры для изучения: Исполнительная среда оболочки**

**Выполнить примеры 1-3 в Raspberry PI, подключиться по ssh к Ubuntu и выполнить примеры 4-12**

1. Войдите в систему под своей учетной записью.

2. Выполните следующие команды, которые используют переменные a, full\_name и short\_name командной оболочки Bash.

$ a=879

$ echo "The value of \"a\" is $a."

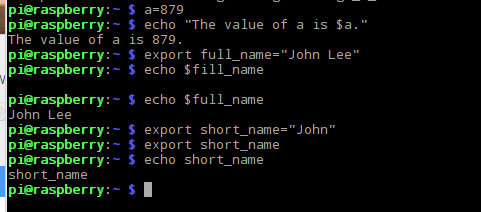
$ export full\_name="John Lee"

$ echo $full\_name

$ export short\_name="John"

$ export short\_name

$ echo $ short\_name



1. Выполните следующие команды, которые иллюстрируют операции над переменными

$ export foo=""

$ echo ${foo:-one}

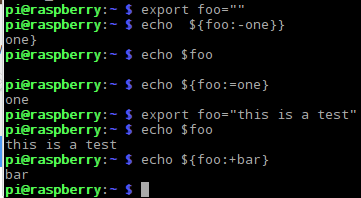
$ echo $foo

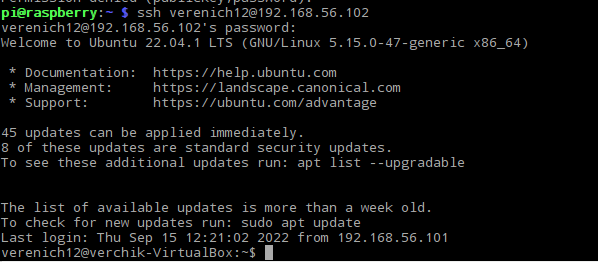
$ echo ${foo:=one}

$ export foo="this is a test"

$ echo $foo

$ echo ${foo:+bar}





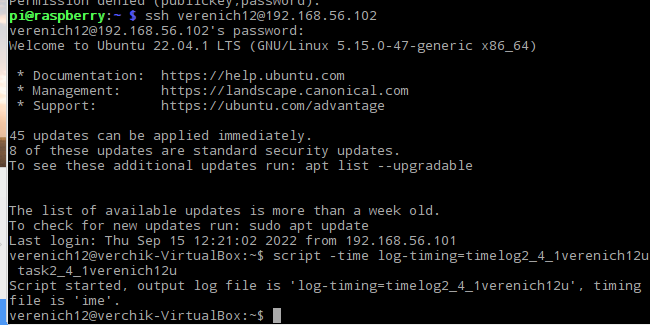
1. Выполнить следующие команды, которые создают переменные массивы.

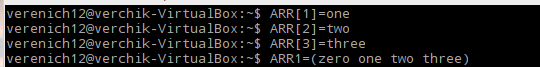
$ ARR[1]=one

$ ARR[2]=two

$ ARR[3]=three

$ ARR1=(zero one two three)





1. Выполнить следующую команду, которая выводит на консоль значения элементов массива.

$ echo ${ARR[0]} ${ARR[1]}



1. Выполните следующие команды для удаления элемента ARR[1] массива ARR и для удаления всего массива ARR:

$ unset ARR[1]

$ unset ARR



1. Создайте псевдоним для команды clear, выполнив следующую команду:

$ alias c=’clear’

Просмотрите созданные псевдонимы.

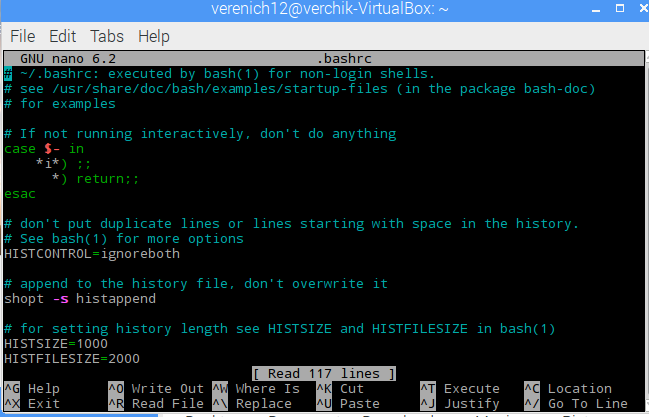
$ alias

$ c



1. Этот псевдоним будет утерян, если вы выйдите из командной оболочки, а затем войдете в нее вновь. Чтобы обеспечить сохранность псевдонима при каждом входе в командную оболочку пользователя student, нужно выполнить следующие действия. Откройте файл .bashrc в текстовом редакторе, например vim

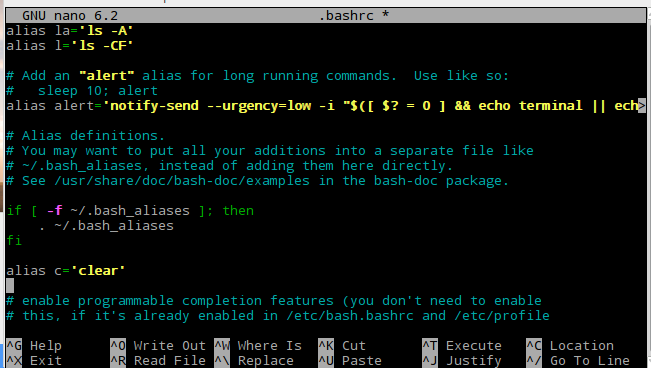
$ vim .bashrc



Найдите в файле bashrc строку, которая содержит текст: # User specific aliases and functions. Добавьте после этой строки следующую строку:

alias c=’clear’

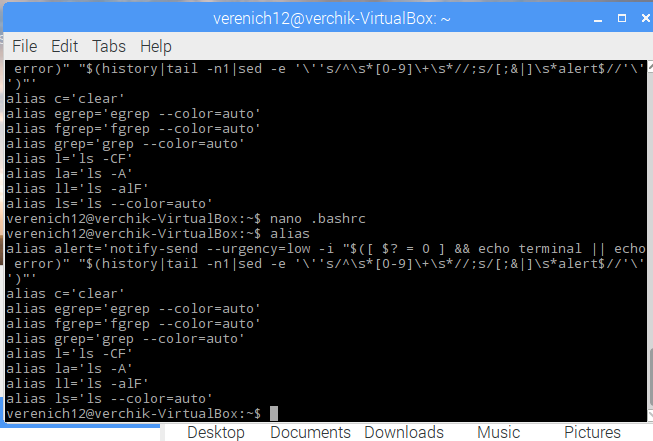
Сохраните файл и выйдите из текстового редактора.



1. Чтобы проверить сохранение псевдонима, сначала выйдите из командной оболочки, а затем снова войдите в неё под именем student и выполните команду

$ alias

$ c



1. Отобразите в терминале текущее значение вашей строки приглашения, выполнив команду

$ echo $PS1

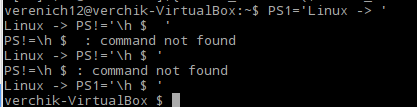


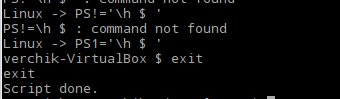
1. Измените вашу строку приглашения, выполнив следующую команду:

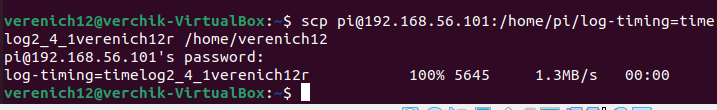
$ PS1=’Linux -> ’

1. Восстановите традиционную строку приглашения, выполнив следующую команду:

$ PS!=’\h $ ’







**2.4.2Примеры для изучения: Исполнение команд**

****

**Выполнить примеры 1-3 в Raspberry PI, подключиться по ssh к Ubuntu и выполнить примеры 4-7.**

1. Выполните следующую команду, которая иллюстрирует расширение командным интерпретатором фигурных скобок.

$ echo sp{el,il,al}l

spell spill spall



1. Выполните следующую команду, которая иллюстрирует расширение командным интерпретатором символа тильда на домашний каталог пользователя.

$ cat ~/message.txt

1. Выполните следующую команду, которая иллюстрирует расширение командным интерпретатором переменной.

$ echo $SHELL /bin/bash

1. Выполните следующую команду, которая иллюстрирует расширение командным интерпретатором команд.

$ echo $(date)

1. Объявите переменные X и Y и задайте им некоторые значения. Выполните следующую команду, которая иллюстрирует расширение командным интерпретатором арифметического выражения.

$ echo Area: $[$X \* $Y]

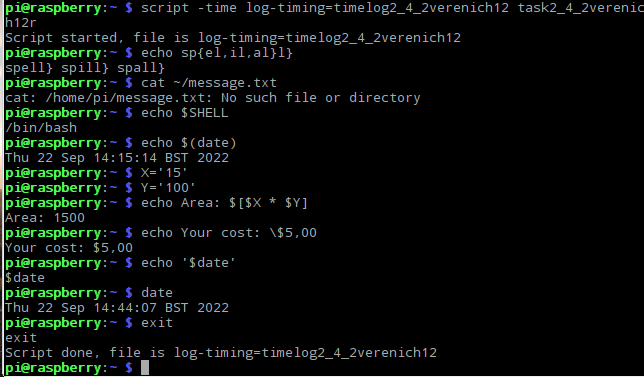
1. Выполните следующую команду, которая иллюстрирует запрещение расширения символа, который следует за символом обратный слэш.

$ echo Your cost: \$5,00

1. Выполните следующую команду, которая иллюстрирует запрещение расширения символов, заключенных в одинарные кавычки.

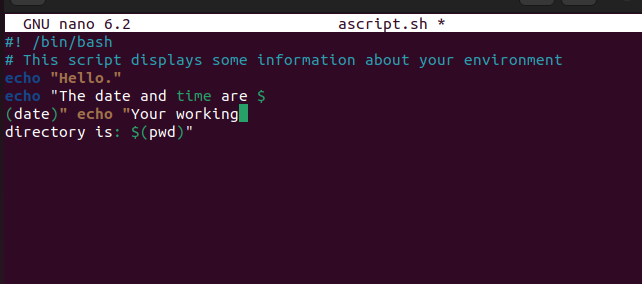
$ echo ’$date’

$ date

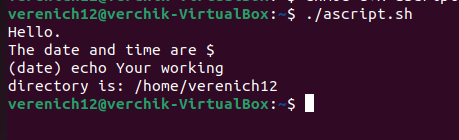


**2.4.3. Сценарии Bash**

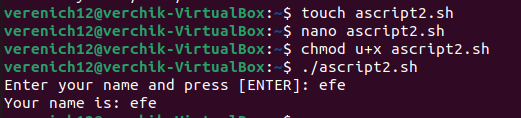
Пример 1.

****

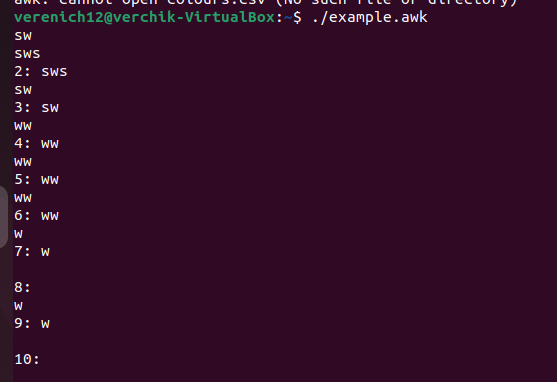
Пример 2.



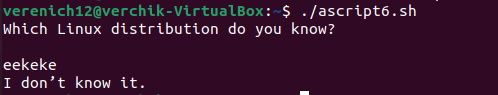
Пример 3.



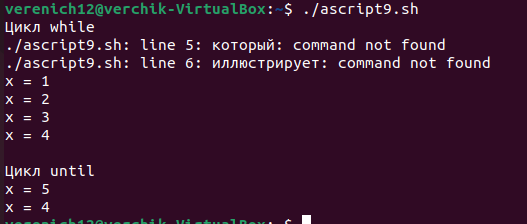
Пример 4.



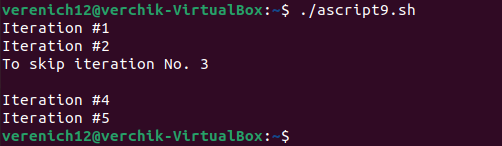
Пример 6.



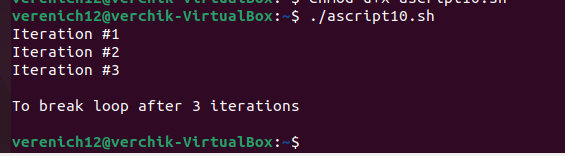
Пример 8.



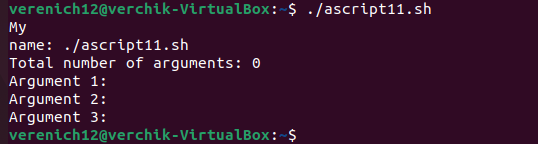
Пример 9.

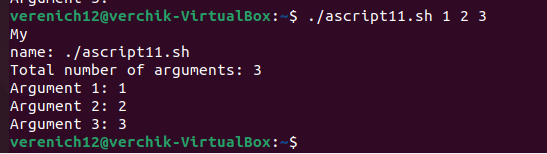


Пример 10.

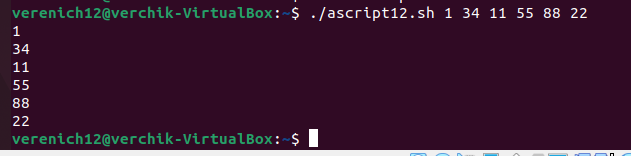


Пример 11.

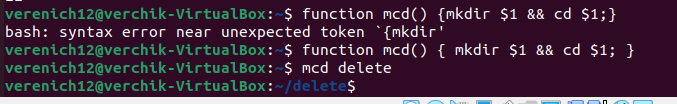




Пример 12.



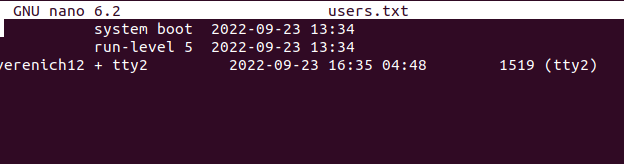
Пример 13.

****

**Вариант 4.**

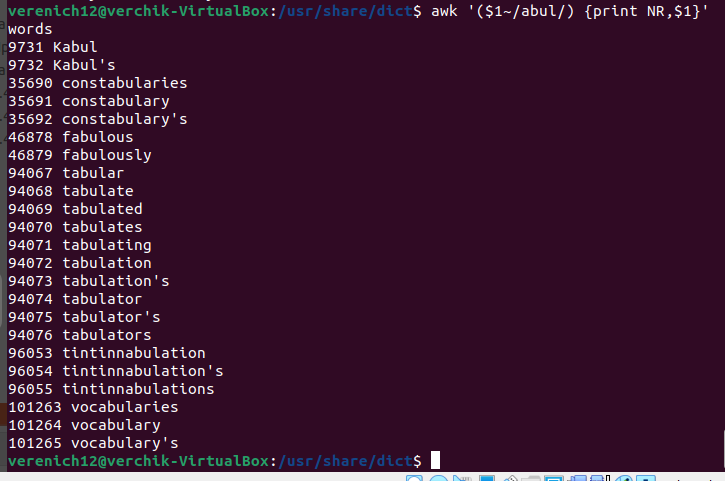
Задача 1.

Используя команду who, определите пользователей, работающих в системе и отобразите данные о дате и времени загрузки системы, текущий уровень, на котором выполняются процессы пользователя, и сведения о пользователе. Вывод команды who направьте в файл users.txt. Просмотрите содержимое файла users.txt.



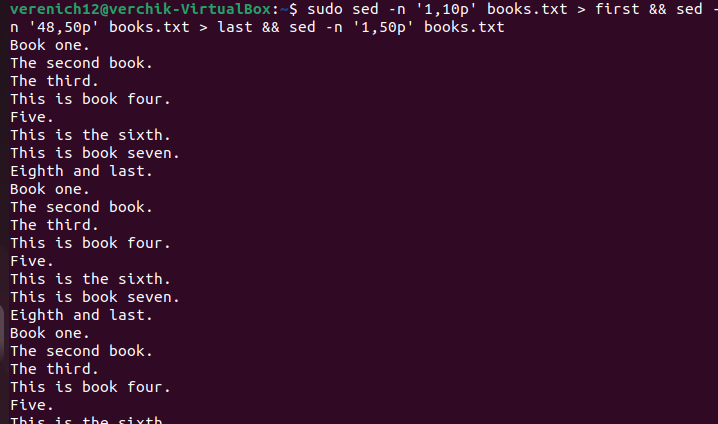
Задание 2.

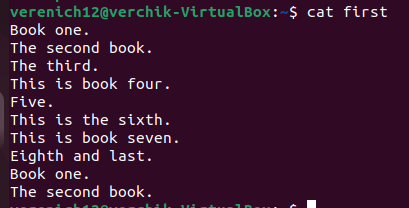
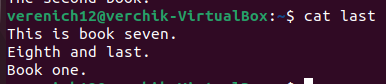
Используйте AWK для определения, сколько строк в файле /usr/share/dict/words (проверить на Ubuntu) содержат подстроку abul, а так же выводите их нумеруя. Определите самую длинную строку. Если данный файл отсутствует выполнять задание для любого текстового файла, который можно найти в подкаталогах каталога /usr/share.



Задание 3.

Напишите sed-программу по имени div , которая копирует файл из 50 строк на стандартный вывод, причем копирует строки 5-10 в файл по имени first и копирует последние 12 строк в файл по имени last.





Задача 4.

Напишите скрипт, который выполняет следующие действия:

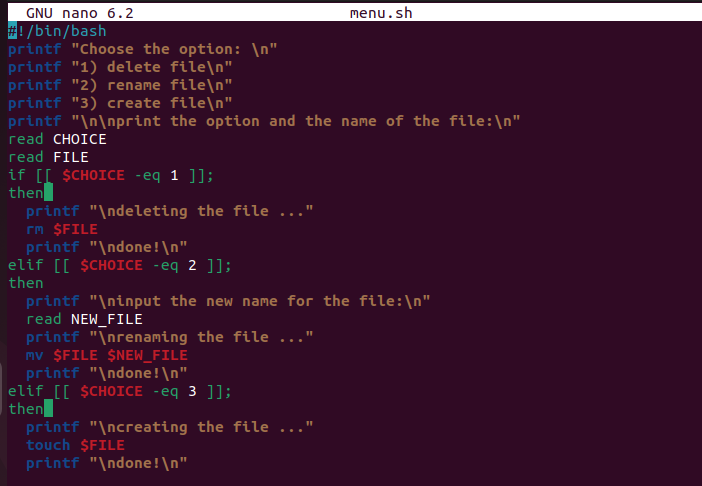
● Выводит на терминал меню, которое предлагает выбор из следующих действий:

○ удалить файл;

○ переименовать файл;

○ создать файл.

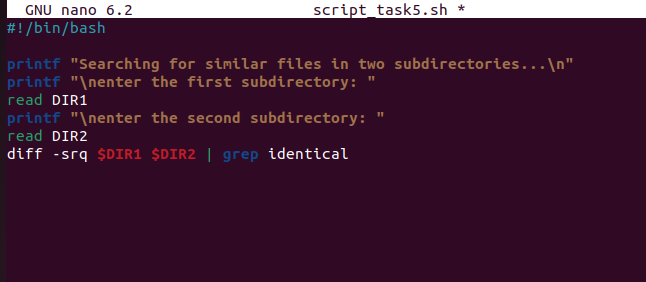
● запрашивает имена файлов, для выбранного действия; выполняет выбранное действие.

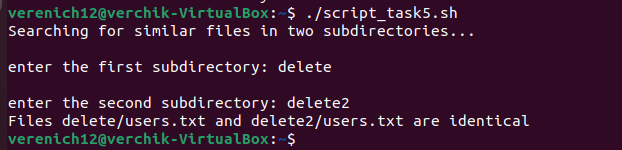




Задача 5.

Написать скрипт поиска одинаковых по их содержимому файлов в двух каталогах, например Dir1 и Dir2. Пользователь задаёт имена Dir1 и Dir2 в качестве первого и второго аргумента командной строки. В результате работы программы файлы, имеющиеся в Dir1, сравниваются с файлами в Dir2 по их содержимому. На экран выводятся число просмотренных файлов и результаты сравнения. Проверить работу программы для каталога /usr (Dir1)и любого каталога в каталоге /home (Dir2).





Задача 6.

Написать скрипт, который при вызове делает фото с веб-камеры утилитой fswebcam и сохраняет в заданную папку, которую задать в скрипте. В скрипт добавить проверку, если утилита fswebcam не установлена, то установить ее