БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

Факультет прикладной математики и информатики

Кафедра технологий программирования

Никочник Даниил

Отчет по лабораторной работе №2

«Программирование мобильных и встраиваемых систем»

студентка 3 курса 12 группы

Преподаватель

Давидовская Мария Ивановна

Минск 2021

**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

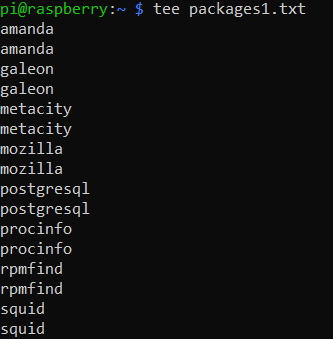
mozilla

postgresql

procinfo

rpmfind

squid



Файл packages2.txt должен содержать следующих шесть строк:

anaconda

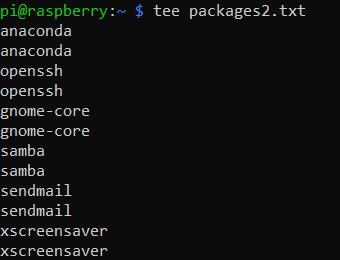
openssh

gnome-core

samba

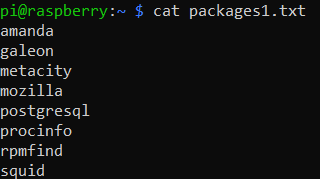
sendmail

xscreensaver



2. 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

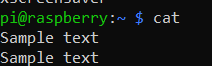


3. Если команда cat не имеет аргументов, то ожидается, что она получает данные из стандартного канала ввода STDIN. Команда читает данные из канала стандартного ввода до нажатия на клавиатуре клавиш :

$ cat

Type some sample text, then press return.

<Ctrl+d>

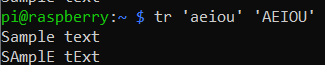


4. Большинство команд Linux работают как фильтры, т. е. Получают исходные данные из STDIN, делают обработку этих данных и выводят результаты в STDOUT. Например, команда tr (translate):

$ tr ’aeiou’ ’AEIOU’

Type some sample text, then press return.

<Ctrl+D>



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

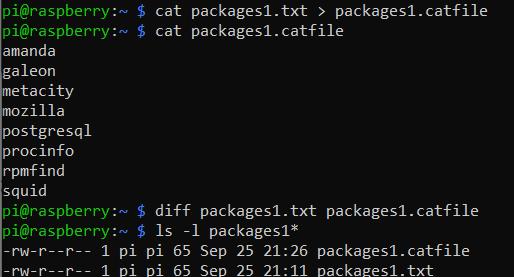
$ cat packages1.txt > packages1.catfile

$ cat packages1.catfile З

Затем сверим исходный и полученный файлы:

$ diff packages1.txt packages1.catfile

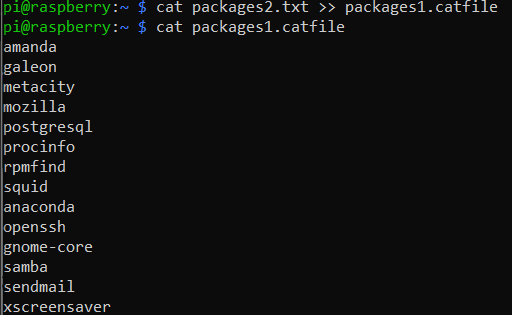
$ ls -l packages1\*



6. 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



7. 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

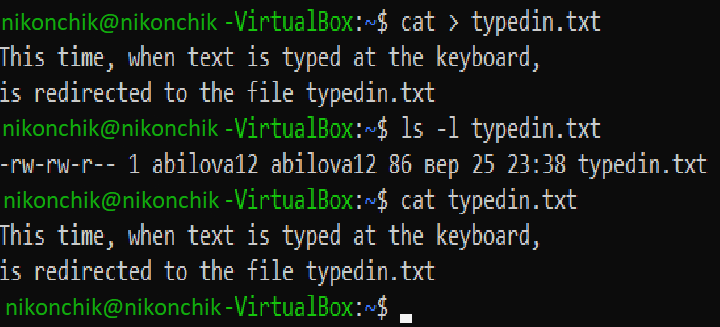
This time, when text is typed at the keyboard,

it is redirected to the file typedin.txt.

<Ctrl+d>

$ ls -l typedin.txt

$ cat typedin.txt



8. Повторите предыдущий шаг, подставив вместо команды cat команду tr.

$ tr ’aeiou’ ’AEIOU’ > trfile.txt

This time, when text is typed at the keyboard,

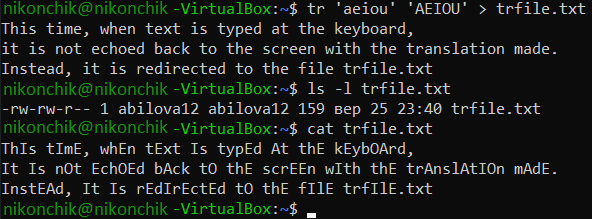
it is not echoed back to the screen with the translations made.

Instead, it is redirected to the file trfile.txt.

<Ctrl+d>

$ ls -l trfile.txt

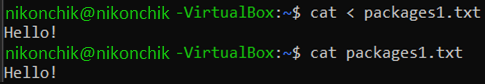
$ cat trfile.txt



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

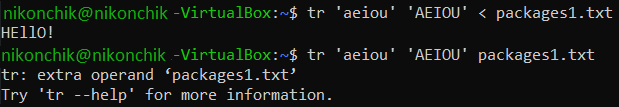
$ cat packages1.txt

$ cat < packages1.txt



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

$ tr ’aeiou’ ’AEIOU’ < packages1.txt

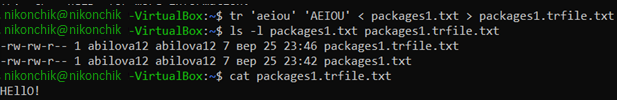


11. В следующем примере стандартный ввод и вывод одновременно перенаправляются

$ 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.

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 2.2

**Пример 1**. Предварительно создадим файл для обработки. Для этого выполним следующие действия:

1. Создаем каталог examples в домашнем каталоге пользователя.

$ mkdir examples

1. Убедитесь, что Вы находитесь в домашнем каталоге, выполнив команду

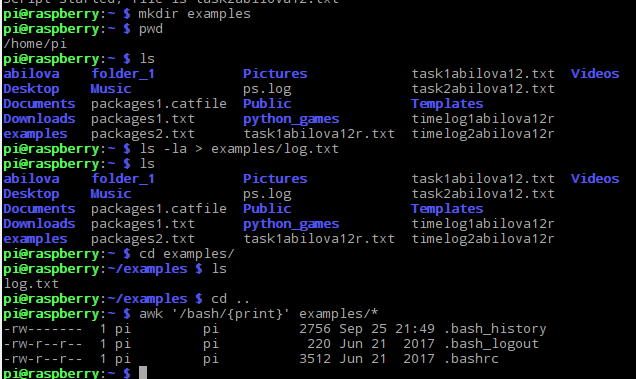
$ pwd

1. Перенаправим вывод команды ls расширенном формате в файл, например log.txt

$ ls -la > examples/log.txt

1. И выполним команду для поиска данных в созданном файле или наборе файлов. Будем искать строки, содержащие слово «bash»

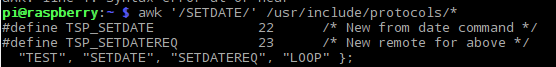
$ awk '/bash/ {print}' examples/\*



**Пример 2.** Найти по указанному пути файлы исходных текстов на языке С, включающие слова, например, QUIT или SETDATE:

$ awk '/QUIT|SETDATE/' /usr/include/protocols/\*





**Пример 3.**

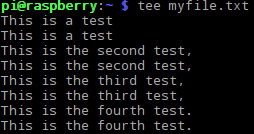
1. Создайте в каталоге examples текстовый файл myfile следующего содержания:

This is a test.

This is the second test.

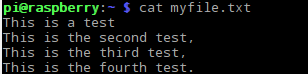
This is the third test.

This is the fourth test.



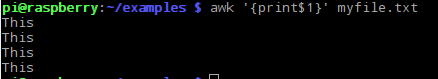
1. Просмотрите файл:

$ cat myfile



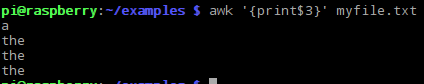
1. Выполните вывод первого элемента каждой строки:

$ awk '{print $1}' myfile



1. 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:

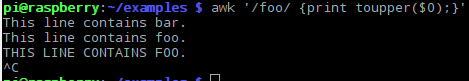
$ awk '/foo/ { print toupper($0); }'

This line contains bar.

This line contains foo.

THIS LINE CONTAINS FOO.

2. 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

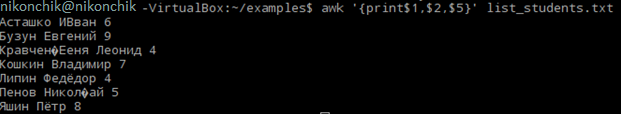
Липин Федор ПИ 2 4

Пенов Николай ПМ 1 5

Яшин Петр КБ 4 8

2. 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

Выполнил

Ничего не выводится из-за проблемы с кодировкой.

**Пример 9.**

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

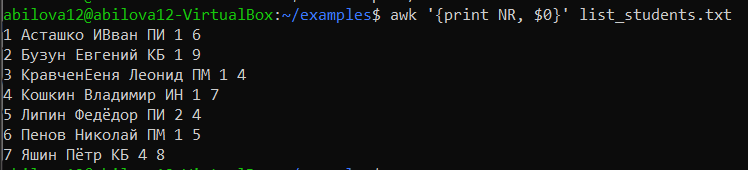
$ awk '(length($2)>5) {print}' list\_students

Выполнил

**Пример 10.**

Вывести данные исходного файла в виде таблицы и пронумеровать строки:

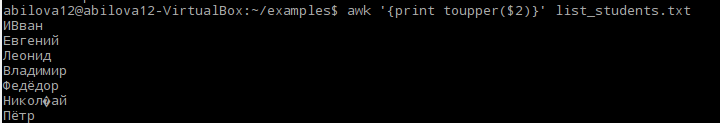
$ awk '{print NR, $0}' list\_students



**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

Выполнил

**Пример 13.**

Присутствует ли фамилия Леонов в списке файла list\_students?

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

Выполнил

**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

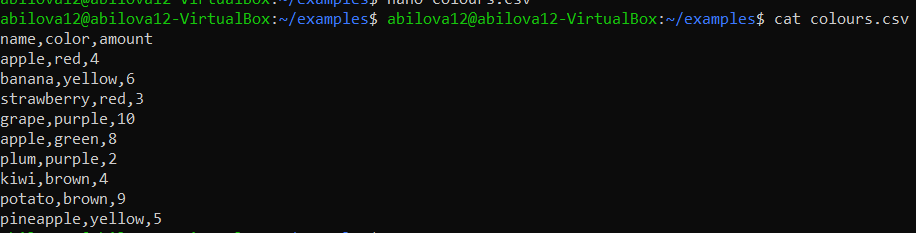
plum,purple,2

kiwi,brown,4

potato,brown,9

pineapple,yellow,5

Cat clours.csv



2. 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.

**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.

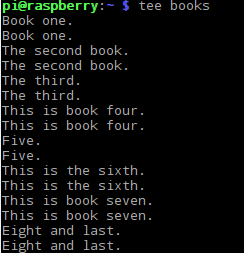
This is book four.

Five.

This is the sixth.

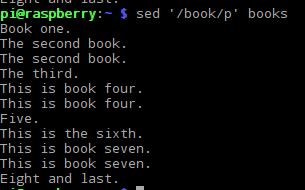
This is book seven.

Eighth and last.



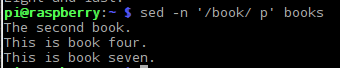
2. Чтобы вывести все строки файла и продублировать строки согласно шаблону, т. е. содержащие слово book выполните команду

$ sed '/book/ p' books



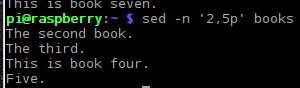
3. Чтобы выбрать определенные строки, например содержащие слово book, выполните команду

$ sed -n '/book/ p' books



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

$ sed -n '2,5 p' books



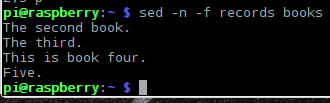
5. Для выполнения более сложных и длинных инструкций можем использовать файл программы для sed. Выполним команду из примера 4, указав параметры в файле records.

Содержимое файла records:

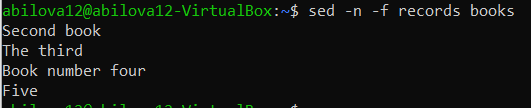
2,5 p

Пример команды из примера 4 с выполнением инструкций из файла:

$ sed -n -f records books



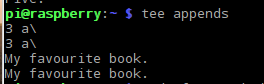
Sed -n -f records books



6. 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

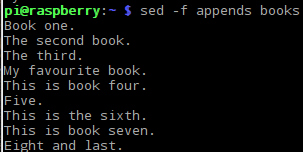
3 a\



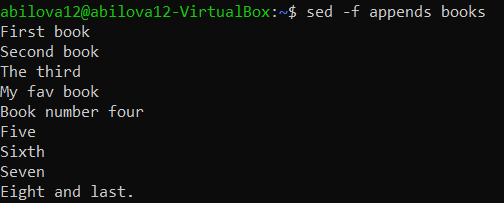
My favorite book.

Пример команды с добавлением строки:

$ sed -f appends books



sed -f appends books



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

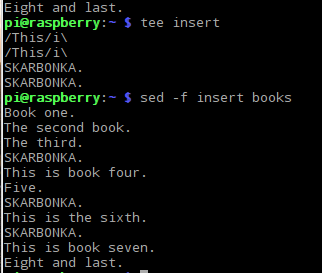
Содержимое файла insert

/This/ i\

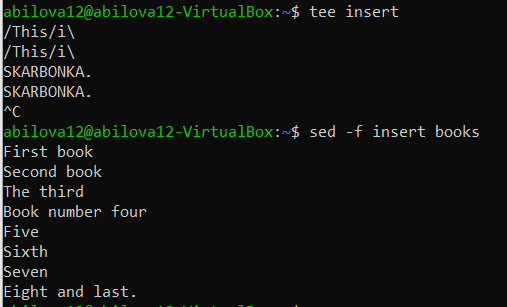
SKARBONKA.

Пример команды с со вставкой строк:

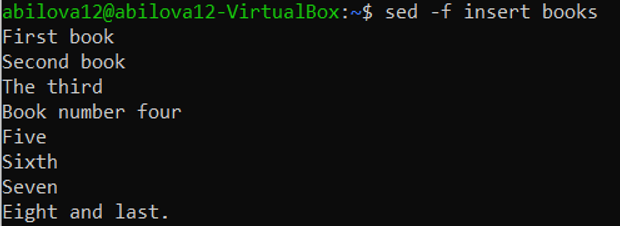
$ sed -f insert books



tee insert

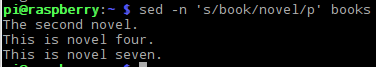


sed -f insert books

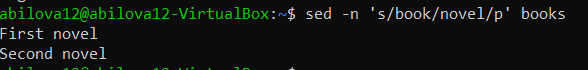


8. В следующем примере выполняется замена слова book на novel

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

****

**sed -n ‘s/book/novel/p’ books**

****

**Задание 2.4. Исполнительная среда оболочки Bash**

Изучите примеры задания 2.4 и выполните их в ОС Ubuntu и Raspberry PI.   
Имена скриптов могут быть любыми.

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.

Выаолнил

**2.4.1. 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

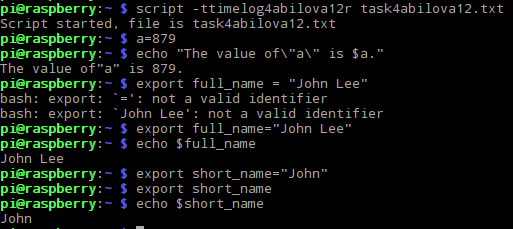
$a." $ export full\_name="John Lee"

$ echo $full\_name

$ export short\_name="John"

$ export short\_name

$ echo $ short\_name



3. Выполните следующие команды, которые иллюстрируют операции над переменными

$ export foo=""

$ echo ${foo:-one}

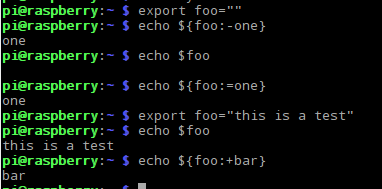
$ echo $foo

$ echo ${foo:=one}

$ export foo="this is a test"

$ echo $foo

$ echo ${foo:+bar}



4. Выполнить следующие команды, которые создают переменные массивы.

$ ARR[1]=one

$ ARR[2]=two

$ ARR[3]=three

$ ARR1=(zero one two three)

Выполнил

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

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

Выполнил

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

$ unset ARR[1]

$ unset ARR

Выполнил

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

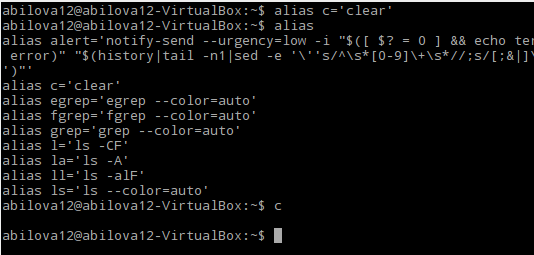
$ alias c=’clear’

Просмотрите созданные псевдонимы.

$ alias

$ c

alias c=’clear’



8. 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

Выполнил

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.

***В файле .bashrc не было строки, содержащей “# User specific aliases and functions ”***

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

$ alias

$ c

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

$ echo $PS1

Выполнил

11. Измените вашу строку приглашения, выполнив следующую команду:

$ PS1=’Linux -> ’

Выполнил

12. Восстановите традиционную строку приглашения, выполнив следующую команду:

$ PS!=’\h $ ’

Выполнил

**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.

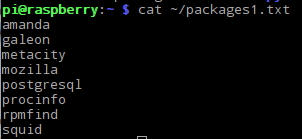
$ echo sp{el,il,al}l

spell spill spall



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

$ cat ~/message.txt



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

$ echo $SHELL

/bin/bash

Выполнил

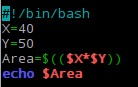
4. Выполните следующую команду, которая иллюстрирует расширение командным интерпретатором команд.

$ echo $(date)

Выполнил

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

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





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

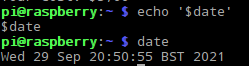
$ echo Your cost: \$5,00

Выполнил

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

$ echo ’$date’

$ date



**2.4.3. Примеры для изучения: Сценарии Bash**

Выполнить примеры 1-2 в Raspberry PI, подключиться по ssh к Ubuntu и выполнить примеры 3-12.

Выполнил

**Пример 1**.

1. Создайте текстовый файл ascript.sh, который содержит следующий текст.

#! /bin/bash

# This script displays some information about your

Environment

echo "Hello."

echo "The date and time are $(date)"

echo "Your working directory is: $(pwd)"

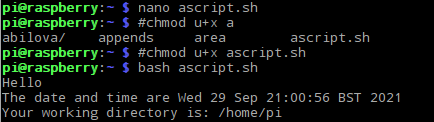
2. Сохраните файл в текущем каталоге.

3. Сделайте файл ascript.sh исполняемым, выполнив команду

# chmod u+x ascript.sh

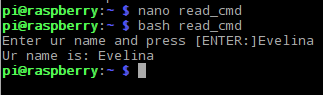
4. Запустите сценарий из командной строки, выполнив команду

# ./ascript.sh

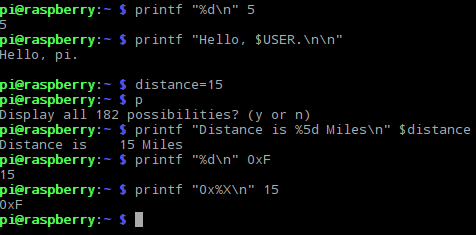


**Пример 2.**

1. Создайте и выполните следующий скрипт, который поясняет использование команды read для ввода данных с терминала.



2. Выполните следующие команды, которые иллюстрируют использование команды printf для форматированного вывода данных.



**Пример 4.**

1. Создайте и выполните следующий скрипт, который иллюстрирует использование условной инструкции if-then-else, сравнивая значения двух переменных.

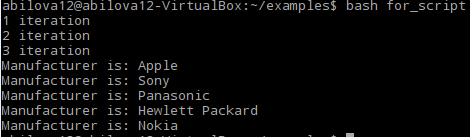
Выполнил

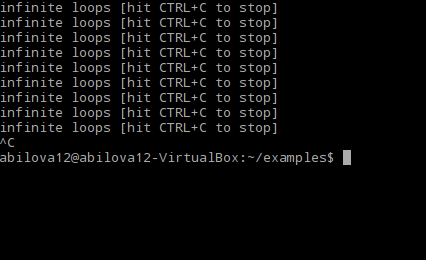
**Пример 5.**

1. Создайте и выполните следующий скрипт, который объясняет использование условной инструкции case для выбора нужного варианта значения переменной.

Выполнил

**Пример 6.**

Создайте и выполните следующий скрипт, который иллюстрирует использование инструкции цикла for для числовых и символьных значений, а также показывает, как программируются бесконечные циклы for. 



**Пример 7**

Создайте и выполните следующий скрипт, который иллюстрирует использование

Выполнил

**Пример 8.**

Создайте и выполните следующий скрипт, который иллюстрирует использование команды принудительного выхода из цикла continue.

Выполнил

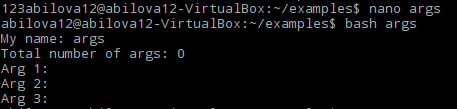
**Пример 9.**

Создайте и выполните следующий скрипт, который иллюстрирует использование команды принудительного завершения цикла break.

Выполнил

**Пример 10.**

Создайте и выполните следующий скрипт, который выводит свое имя, количество аргументов и значения первых трех аргументов, иллюстрируя тем самым обработку аргументов, которые передаются скрипту при вызове. При вызове этого скрипта передайте ему три произвольных аргумента



**Пример 11.**

Создайте и выполните следующий скрипт, который выводит все свои аргументы, иллюстрируя тем самым обработку произвольного количества аргументов. Вызовите этот скрипт, передав ему произвольное количество аргументов.

Выполнил

**Пример 12.**

В командной строке объявите и вызовите функцию mcd, которая создает каталог и делает его текущим.

$ function mcd() { mkdir $1 && cd $1; }

$ mcd temp

Выполнил

**Опубликовала лог-файлы в репозиторий:** Выполнил