Министерство науки и высшего образования РФ

Федеральное государственное бюджетное образовательное учреждение

высшего образования

ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ СИСТЕМ УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ (ТУСУР)

Кафедра комплексной информационной безопасности электронно-вычислительных систем (КИБЭВС)

LINUX, BASH, DOCKER

Отчет по лабораторной работе №1

по дисциплине «Системное программирование»

|  |  |
| --- | --- |
|  | Выполнил:  Студент гр. 748  \_\_\_\_\_\_\_\_Подлеснова. \*.\*. (это я хз)  \_\_\_. \_\_\_2022    Принял:  м.н.с. ИСИБ  \_\_\_\_\_\_\_\_\_Калинин Е.О.  \_\_\_. \_\_\_2022 |
|  |  |

**1 Введение**

Целью лабораторной работы является: балдеееж.

Разработать скрипт работы с архивами, который:

* запрашивает тип действия: разархивировать или заархивировать
* для архивации: запрашивает каталог для архивации и имя архива, создаёт архив с этим именем
* для распаковки: спрашивает имя файла с архивом и распаковывает его

Для выполнения задания используйте утилиту tar.

**2 Ход работы**

**2.1 Написание скрипта**

Создадим папку для работы с лабораторными, и файл lab1.sh (рисунок 2.1).

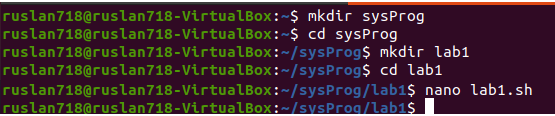


Рисунок 2.1 – Создание нового документа

Напишем код по варианту (рисунок 2.2).

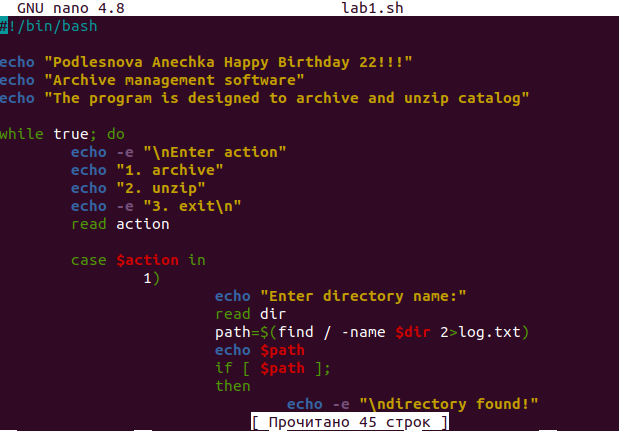


Рисунок 2.2 – Создание нового документа

Сделаем наш файл исполняемым, то есть разрешим его запускать всем пользователям (рисунок 2.3).



Рисунок 2.3 – Доступ к запуску

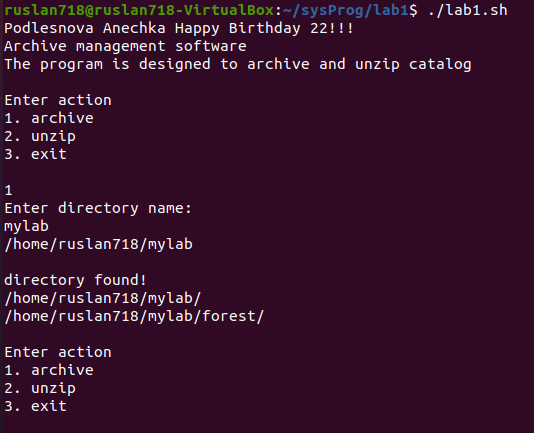


Рисунок 2.4 – Работа кода

**2.2 Установка DOCKER**

Далее установим все файлы для работы с докером. Сначала были выполнены команды для обновления пакетов (***sudo apt-get upgrade***)., и обновления индекса пакетов (***sudo apt-get update***). Результат представлен на рисунке 2.1.

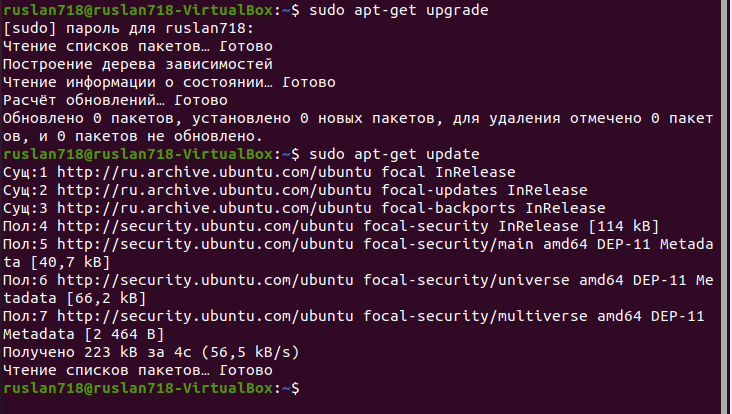


Рисунок 2.1 – Обновление индекса пакетов

Затем командой ***sudo apt install apt-transport-https ca-certificates curl software-properties-common*** были установлены пакеты, которые позволяют использовать пакеты по HTTPS (рисунок 2.2).

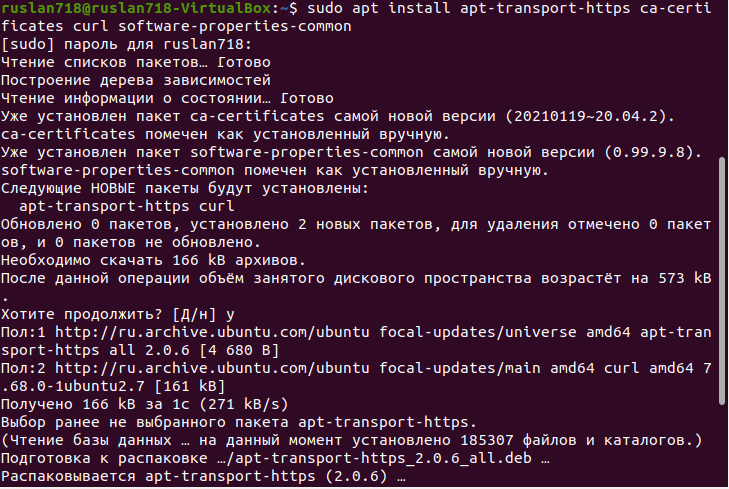
****

Рисунок 2.2 – Установка пакетов

Затем был добавлен ключ GPG для работы с фициальныи репозиторием Docker (***curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -).*** Результат представлен на рисунке 2.3.

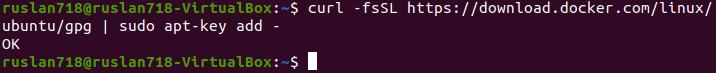


Рисунок 2.3 – Добавление ключа GPG

Добавлен репозиторий Docker в

список источников пакетов APT (рисунок 2.4). Команда: ***sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu bionic stable".***

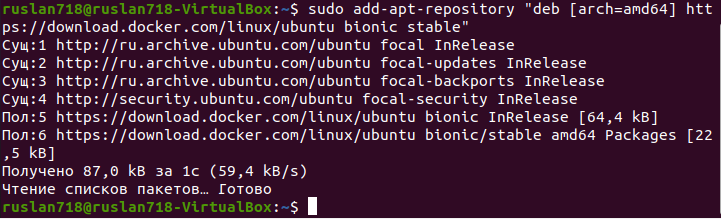


Рисунок 2.4 – Добавление Docker-репозитория

Затем была обновлена база данных пакетов информацией о пакетах Docker из вновь добавленного репозитория (рисунок 2.5).

Команда: ***sudo apt-get update.***

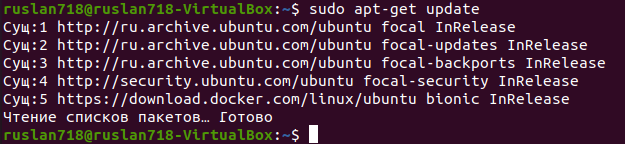


Рисунок 2.5 – Обновление базы данных пакетов

Убедимся, что устанавливаем Docker из репозитория Docker, а не из репозитория по умолчанию Ubuntu (***apt-cache policy docker-ce).*** Результат представлен на рисунке 2.6.

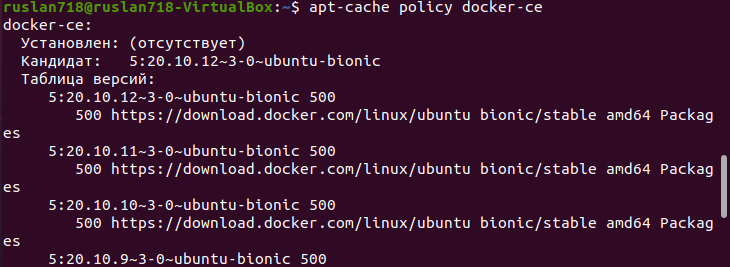


Рисунок 2.6 – Проверка репозитория

Далее установим Docker (рисунок 2.7). Команда: ***sudo apt install docker-ce.***

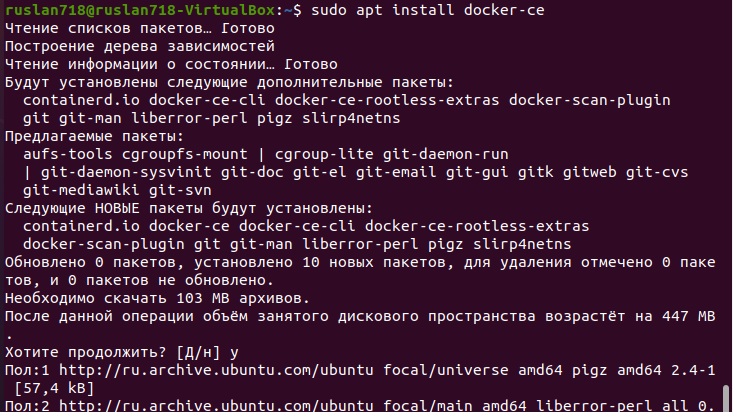
******

Рисунок 2.7 – Установка Docker

Убедимся, что процесс запущен (рисунок 2.8). Команда: ***sudo systemctl status docker.***

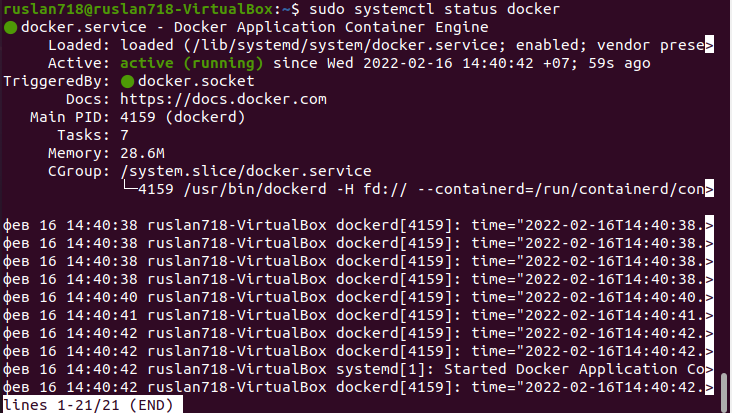
****

Рисунок 2.8 – Проверка работы Docker

Чтобы перейти обратно в режим написания команд, нажмем «ctrl+c». Далее добавим имя своего пользователя в группу docker c помощью команды ***sudo usermod -aG docker ${USER****}.* И теперь, чтобы изменения вступили в силу нужно либо выйти и снова залогиниться на удаленный сервер, или в случае локальной «машины», достаточно выполнить команду ***su - ${USER}.*** Проверим, в каких группах находится пользователь ***id -nG*** (рисунок 2.9).

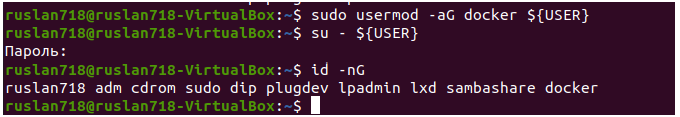
******

Рисунок 2.9 – Добавление пользователя в группу docker

Далее перейдем в ранее созданный каталог, после создадим Dockerfile с помощью команды «nano Dockerfile», в котором пропишем сценарий для работы нашего докера (рисунок 2.18 – 2.19).



Рисунок 2.10 – Создание докера

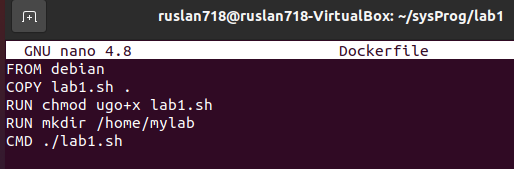
******

Рисунок 2.18 – Создание Dockerfile

После чего переходим в папку 1, и создаем образ командой ***doсker build -t anuta*** (рисунок 2.20).

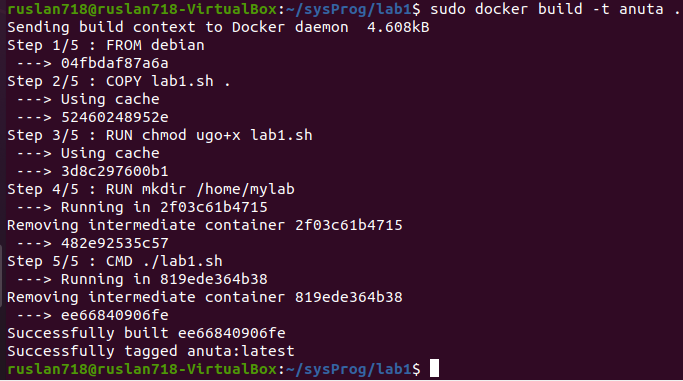
******

Рисунок 2.20 – Сборка образа

После установки «docker» и создание контейнера пропишем команду  
«docker images», чтобы посмотреть список образов, которые находятся на  
«docker» (рисунок 2.1).

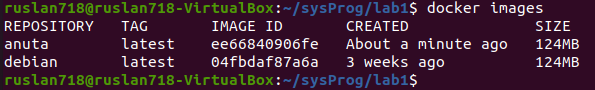


Рисунок 2.1 – Список образов

Запускаем созданный контейнер (рисунок 2.21).

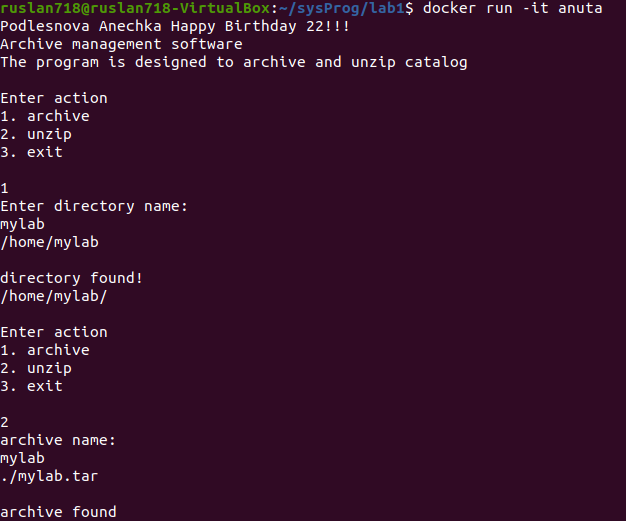
******

Рисунок 2.21 – Запуск образа

Далее нужно создать репозиторий на гитхабе, в который будем загружать контейнер (Рисунок 2.22).

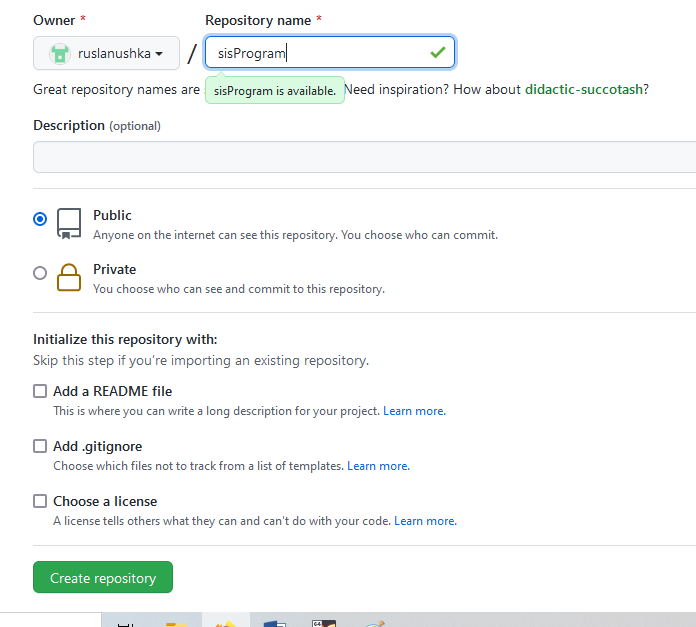


Рисунок 2.22 – Новый репозиторий

Подготовим файлы, которые будем отправлять на гитхаб (рисунок 2.23).

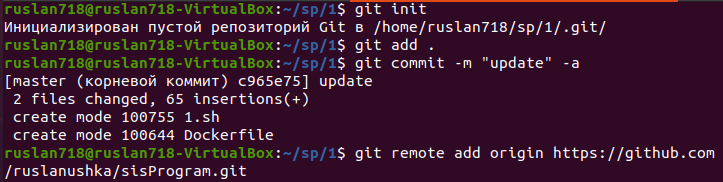
******

Рисунок 2.23 Подготовка файлов

Последним этапом отправим файлы на гитхаб (рисунок 2.24).

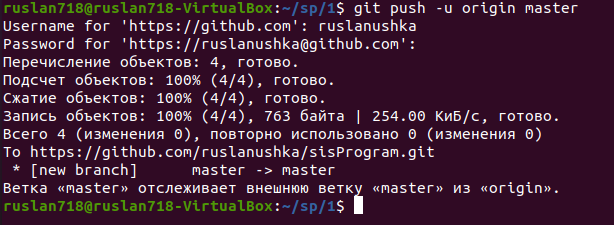


Рисунок 2.24 – загружаем файлы на гитхаб

**3 Заключение**

В ходе лабораторной работы были изучены возможности Docker для контейнеризации окружения программного обеспечения. Был написан скрипт по заданному варианту задания и написан Dockerfile для виртуализации заданной ОС. Файлы были отправлены на GitHub.

Приложение А

#!/bin/bash

echo "Podlesnova Anechka Happy Birthday 22!!!"

echo "Archive management software"

echo "The program is designed to archive and unzip catalog"

while true; do

echo -e "\nEnter action"

echo "1. archive"

echo "2. unzip"

echo -e "3. exit\n"

read action

case $action in

1)

echo "Enter directory name:"

read dir

path=$(find / -name $dir 2>log.txt)

echo $path

if [ $path ];

then

echo -e "\ndirectory found!"

tar -cvf $dir.tar --absolute-names $path 2>log.txt

else

echo -e "\ndirectory not found("

fi;;

2)

echo "archive name:"

read archive

path=$(find . -name $archive.tar 2>log.txt)

echo $path

if [ $path ];

then

echo -e "\narchive found"

tar -xvf $archive.tar

else

echo -e "\narchive not found("

fi;;

3)

echo "byee:3"

break;;

esac

done