

Министерство образования и науки Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего профессионального образования
«ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ СИСТЕМ
УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ» (ТУСУР)
Кафедра комплексной информационной безопасности электронно-
вычислительных систем (БИС)

Linux, Docker, окружение для разработки

Отчет по лабораторной работе №1
по дисциплине «Системное программирование»
вариант

Студент гр. 738-1

_____ Г.К. Беляев

“ ” _____ 2022г.

Принял

Преподаватель кафедры КИБЭВС

Е.О. Калинин

1 Введение

Цель работы: Ознакомиться с операционной системой и основными командами ОС Unix/Linux и с возможностями Docker для контейнеризации окружения программного обеспечения и его зависимостей, средств разработки и сборки ПО.

2 Ход Работы

Для начала в операционной системе Linux Ubuntu терминал. Для того чтобы подготовить систему к установке Dockera необходимо обновить индекс пакетов (загружаем списки пакетов из репозитория и "обновляем" их, чтобы получить информацию о новейших версиях пакетов и их зависимостях) командой: `sudo apt-get update` (рисунок 2.1).

```
german@german-VirtualBox:~$ sudo apt-get update
[sudo] пароль для german:
Суц:1 http://ru.archive.ubuntu.com/ubuntu focal InRelease
Суц:2 http://ppa.launchpad.net/uroni/urbackup/ubuntu focal InRelease
Пол:3 http://ru.archive.ubuntu.com/ubuntu focal-updates InRelease [114 kB]
Пол:4 http://ru.archive.ubuntu.com/ubuntu focal-backports InRelease [108 kB]
Пол:5 http://security.ubuntu.com/ubuntu focal-security InRelease [114 kB]
Пол:6 http://ru.archive.ubuntu.com/ubuntu focal-updates/main amd64 Packages [1
608 kB]
Пол:7 http://ru.archive.ubuntu.com/ubuntu focal-updates/main i386 Packages [610
kB]
Пол:8 http://security.ubuntu.com/ubuntu focal-security/main amd64 Packages [1 2
75 kB]
Пол:9 http://ru.archive.ubuntu.com/ubuntu focal-updates/main Translation-en [30
8 kB]
Пол:10 http://ru.archive.ubuntu.com/ubuntu focal-updates/main amd64 DEP-11 Meta
data [279 kB]
Пол:11 http://ru.archive.ubuntu.com/ubuntu focal-updates/main amd64 c-n-f Metad
ata [14,8 kB]
Пол:12 http://ru.archive.ubuntu.com/ubuntu focal-updates/restricted amd64 Packa
```

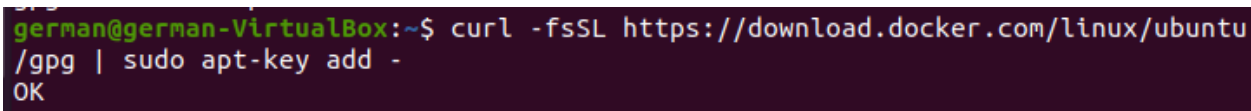
Рисунок 2.1 – Обновление индексов пакетов

Затем установим необходимые пакеты, которые позволяют apt использовать пакеты по HTTPS командой: `sudo apt install apt-transport-https ca-certificates curl software-properties-common` (рисунок 2.2).

```
german@german-VirtualBox:~$ sudo apt install apt-transport-https ca-certificat
e s curl software-properties-common
[sudo] пароль для german:
Чтение списков пакетов... Готово
Построение дерева зависимостей
Чтение информации о состоянии... Готово
Уже установлен пакет ca-certificates самой новой версии (20210119~20.04.2).
ca-certificates помечен как установленный вручную.
Уже установлен пакет curl самой новой версии (7.68.0-1ubuntu2.7).
curl помечен как установленный вручную.
Следующие НОВЫЕ пакеты будут установлены:
  apt-transport-https
Следующие пакеты будут обновлены:
  python3-software-properties software-properties-common
  software-properties-gtk
Обновлено 3 пакетов, установлено 1 новых пакетов, для удаления отмечено 0 пакет
ов, и 111 пакетов не обновлено.
Необходимо скачать 106 kB архивов.
После данной операции объем занятого дискового пространства возрастет на 176 kB
.
Пол:1 http://ru.archive.ubuntu.com/ubuntu focal-updates/universe amd64 apt-tran
sport-https all 2.0.6 [4.680 B]
```

Рисунок 2.2 – Установка пакетов

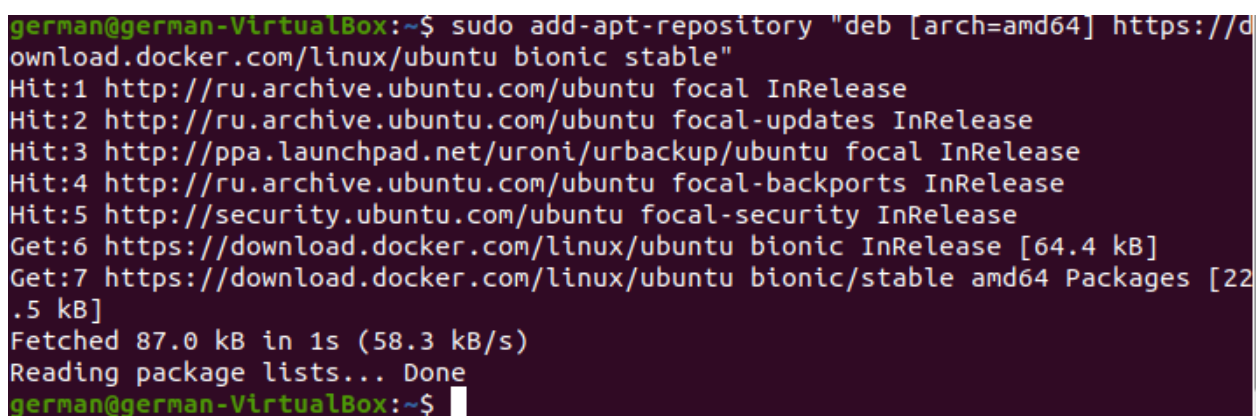
Затем добавим в свою систему ключ GPG официального репозитория Docker командой: `curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt key add` – (рисунок 2.3).



```
german@german-VirtualBox:~$ curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -
OK
```

Рисунок 2.3 – Добавление ключа

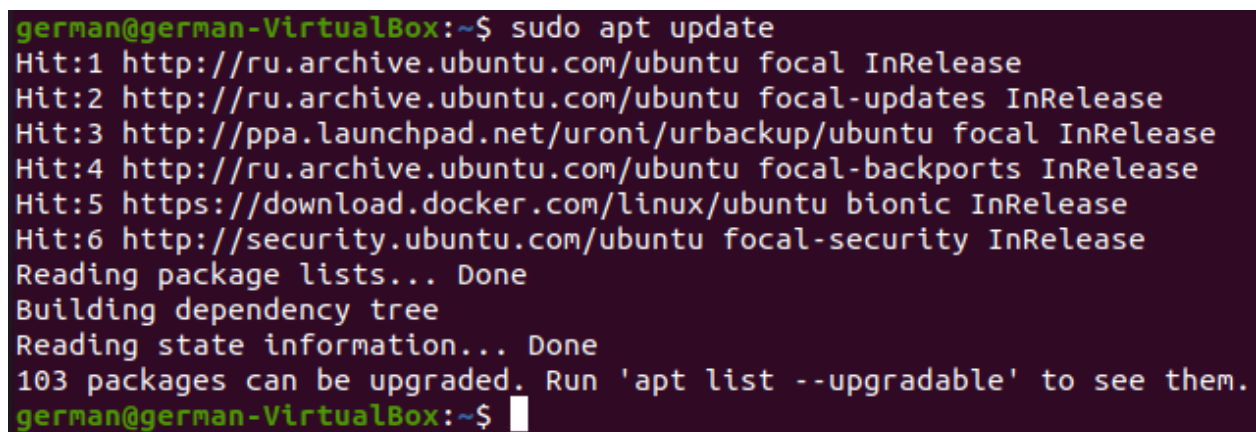
Далее добавим этот репозиторий в список источников пакетов APT командой: `sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu bionic stable"` (рисунок 2.4).



```
german@german-VirtualBox:~$ sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu bionic stable"
Hit:1 http://ru.archive.ubuntu.com/ubuntu focal InRelease
Hit:2 http://ru.archive.ubuntu.com/ubuntu focal-updates InRelease
Hit:3 http://ppa.launchpad.net/uroni/urbackup/ubuntu focal InRelease
Hit:4 http://ru.archive.ubuntu.com/ubuntu focal-backports InRelease
Hit:5 http://security.ubuntu.com/ubuntu focal-security InRelease
Get:6 https://download.docker.com/linux/ubuntu bionic InRelease [64.4 kB]
Get:7 https://download.docker.com/linux/ubuntu bionic/stable amd64 Packages [22.5 kB]
Fetched 87.0 kB in 1s (58.3 kB/s)
Reading package lists... Done
german@german-VirtualBox:~$
```

Рисунок 2.4 – Добавление репозитория Docker

Затем обновим базу данных пакетов информацией о пакетах Docker из вновь добавленного репозитория командой: `sudo apt-get update` (рисунок 2.5).



```
german@german-VirtualBox:~$ sudo apt update
Hit:1 http://ru.archive.ubuntu.com/ubuntu focal InRelease
Hit:2 http://ru.archive.ubuntu.com/ubuntu focal-updates InRelease
Hit:3 http://ppa.launchpad.net/uroni/urbackup/ubuntu focal InRelease
Hit:4 http://ru.archive.ubuntu.com/ubuntu focal-backports InRelease
Hit:5 https://download.docker.com/linux/ubuntu bionic InRelease
Hit:6 http://security.ubuntu.com/ubuntu focal-security InRelease
Reading package lists... Done
Building dependency tree
Reading state information... Done
103 packages can be upgraded. Run 'apt list --upgradable' to see them.
german@german-VirtualBox:~$
```

Рисунок 2.5 – Обновление базы данных пакетов

Следует убедиться, что устанавливаем Docker из репозитория Docker, а не из репозитория по умолчанию Ubuntu командой: `apt-cache policy docker-ce`. Вывод получится приблизительно следующий, откуда можно увидеть, что установка идет верно (рисунок 2.6).

```
german@german-VirtualBox:~$ apt-cache policy docker-ce
docker-ce:
  Installed: (none)
  Candidate: 5:20.10.12~3-0~ubuntu-bionic
  Version table:
     5:20.10.12~3-0~ubuntu-bionic 500
        500 https://download.docker.com/linux/ubuntu bionic/stable amd64 Pack
es
     5:20.10.11~3-0~ubuntu-bionic 500
        500 https://download.docker.com/linux/ubuntu bionic/stable amd64 Pack
es
     5:20.10.10~3-0~ubuntu-bionic 500
        500 https://download.docker.com/linux/ubuntu bionic/stable amd64 Pack
es
     5:20.10.9~3-0~ubuntu-bionic 500
        500 https://download.docker.com/linux/ubuntu bionic/stable amd64 Pack
es
     5:20.10.8~3-0~ubuntu-bionic 500
        500 https://download.docker.com/linux/ubuntu bionic/stable amd64 Pack
es
     5:20.10.7~3-0~ubuntu-bionic 500
        500 https://download.docker.com/linux/ubuntu bionic/stable amd64 Pack
```

Рисунок 2.6 – Проверка правильности установки

Теперь можно установить сам Docker командой: `sudo apt install docker-ce` (рисунок 2.7).

```
german@german-VirtualBox:~$ sudo apt install docker-ce
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  containerd.io docker-ce-cli docker-ce-rootless-extras docker-scan-plugin
  git git-man liberror-perl pigz slirp4netns
Suggested packages:
  aufs-tools cgroupfs-mount | cgroup-lite git-daemon-run
  | git-daemon-sysvinit git-doc git-el git-email git-gui gitk gitweb git-cvs
  git-mediawiki git-svn
The following NEW packages will be installed:
  containerd.io docker-ce docker-ce-cli docker-ce-rootless-extras
  docker-scan-plugin git git-man liberror-perl pigz slirp4netns
0 upgraded, 10 newly installed, 0 to remove and 103 not upgraded.
Need to get 103 MB of archives.
After this operation, 447 MB of additional disk space will be used.
Do you want to continue? [Y/n] Y
```

Рисунок 2.7 – Установка Docker

Теперь Docker установлен, демон запущен, и процесс будет запускаться при загрузке системы. Убедимся, что процесс запущен командой: `sudo systemctl status docker` (рисунок 2.9).

```
german@german-VirtualBox:~$ sudo systemctl status docker
● docker.service - Docker Application Container Engine
   Loaded: loaded (/lib/systemd/system/docker.service; enabled; vendor prese
   Active: active (running) since Sat 2022-03-05 16:21:25 +07; 2min 42s ago
   TriggeredBy: ● docker.socket
     Docs: https://docs.docker.com
    Main PID: 11482 (dockerd)
       Tasks: 7
      Memory: 30.1M
     CGroup: /system.slice/docker.service
            └─11482 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/co

мар 05 16:21:06 german-VirtualBox dockerd[11482]: time="2022-03-05T16:21:06.03>
мар 05 16:21:06 german-VirtualBox dockerd[11482]: time="2022-03-05T16:21:06.03>
мар 05 16:21:06 german-VirtualBox dockerd[11482]: time="2022-03-05T16:21:06.03>
мар 05 16:21:06 german-VirtualBox dockerd[11482]: time="2022-03-05T16:21:06.07>
мар 05 16:21:09 german-VirtualBox dockerd[11482]: time="2022-03-05T16:21:09.37>
мар 05 16:21:11 german-VirtualBox dockerd[11482]: time="2022-03-05T16:21:11.03>
мар 05 16:21:24 german-VirtualBox dockerd[11482]: time="2022-03-05T16:21:24.37>
мар 05 16:21:24 german-VirtualBox dockerd[11482]: time="2022-03-05T16:21:24.65>
мар 05 16:21:25 german-VirtualBox systemd[1]: Started Docker Application Conta>
мар 05 16:21:27 german-VirtualBox dockerd[11482]: time="2022-03-05T16:21:25.71>
lines 1-21/21 (END)
```

Рисунок 2.9 – Проверка процесса

Чтобы не вводить `sudo` каждый раз при запуске команды `docker`, добавим имя своего пользователя в группу `docker` командой: `sudo usermod -aG docker ${USER}`. Для применения этих изменений в составе группы необходимо разлогиниться и снова залогиниться на сервере или задать команду: `su - ${USER}`. Для продолжения работы необходимо ввести пароль пользователя. Убедиться, что пользователь добавлен в группу `docker` можно командой: `id -nG` (рисунок 2.10).

```
german@german-VirtualBox: ~
german@german-VirtualBox:~$ sudo usermod -aG docker german
[sudo] пароль для german:
german@german-VirtualBox:~$ su - german
Пароль:
german@german-VirtualBox:~$ id -nG
german adm cdrom sudo dip plugdev lpadmin lxd sambashare docker
german@german-VirtualBox:~$
```

Рисунок 2.10 – Добавление пользователя в группу Docker

Для просмотра всей информации о Docker используется команда: `docker info` (рисунок 2.11).

```
german@german-VirtualBox:~$ docker info
Client:
Context:    default
Debug Mode: false
Plugins:
  app: Docker App (Docker Inc., v0.9.1-beta3)
  buildx: Docker Buildx (Docker Inc., v0.7.1-docker)
  scan: Docker Scan (Docker Inc., v0.12.0)
Server:
Containers: 0
  Running: 0
  Paused: 0
  Stopped: 0
Images: 0
Server Version: 20.10.12
Storage Driver: overlay2
  Backing Filesystem: extfs
  Supports d_type: true
  Native Overlay Diff: true
  userxattr: false
Logging Driver: json-file
Cgroup Driver: cgroupfs
Cgroup Version: 1
Plugins:
  Volume: local
  Network: bridge host ipvlan macvlan null overlay
  Log: awslogs fluentd gcplogs gelf journald json-file local logentries splunk
```

Рисунок 2.11 – Просмотр информации о Docker

Для поиска необходимых образов на Docker Hub используется команда `docker` и подкоманда `search`. Чтобы найти образ Ubuntu, нужно ввести команду: `docker search ubuntu` (рисунок 2.12).

```
german@german-VirtualBox:~$ docker search ubuntu
NAME                STARS     OFFICIAL   AUTOMATED  DESCRIPTION
ubuntu              13821     [OK]       Ubuntu is a Debian-based Linux operating sys...
websphere-liberty   283       [OK]       WebSphere Liberty multi-architecture images ...
ubuntu-upstart      112       [OK]       DEPRECATED, as is Upstart (find other proces...
neurodebian         88        [OK]       NeuroDebian provides neuroscience research s...
open-liberty        50        [OK]       Open Liberty multi-architecture images based...
ubuntu-debootstrap  45        [OK]       DEPRECATED; use "ubuntu" instead
ubuntu/nginx        33        Nginx, a high-performance reverse proxy & we...
ubuntu/mysql        24        MySQL open source fast, stable, multi-thread...
ubuntu/apache2      23        Apache, a secure & extensible open-source HT...
ubuntu/prometheus   23        Prometheus is a systems and service monitori...
kasmweb/ubuntu-bionic-desktop 18        Ubuntu productivity desktop for Kasm Workspa...
ubuntu/squid        13        Squid is a caching proxy for the Web. Long-t...
```

Рисунок 2.12 – Поиск в Docker Hub

Скрипт просматривает Docker Hub и возвращает список всех образов, имена которых подходят под заданный поиск. Проще говоря, с помощью команды `search` был организован поиск, по ключевому слову `Ubuntu`, образов в Docker Hub. В столбце `OFFICIAL` строка `OK` показывает, что образ построен и поддерживается компанией, которая занимается разработкой этого проекта. После чего нужно выбрать необходимую версию и приступить к скачиванию командой: `docker pull ubuntu` (рисунок 2.13).

```
german@german-VirtualBox:~$ docker pull ubuntu
Using default tag: latest
latest: Pulling from library/ubuntu
7c3b88808835: Pull complete
Digest: sha256:8ae9bafbb64f63a50caab98fd3a5e37b3eb837a3e0780b78e5218e63193961f9
Status: Downloaded newer image for ubuntu:latest
docker.io/library/ubuntu:latest
```

Рисунок 2.13 – Скачивание образа

Для просмотра загруженных на компьютер образов нужно ввести команду: `docker images` (рисунок 2.14).

```
german@german-VirtualBox:~$ docker images
REPOSITORY    TAG       IMAGE ID       CREATED        SIZE
ubuntu        latest    2b4cba85892a   37 hours ago   72.8MB
german@german-VirtualBox:~$
```

Рисунок 2.14 – Просмотр образов

После загрузки образа можно запустить контейнер с загруженным образом с помощью подкоманды `run`. В качестве примера запустим контейнер с помощью последней версии образа `Ubuntu`. Комбинация параметров `-i` и `-t` обеспечивает интерактивный доступ к командному процессору контейнера. Команда: `docker run -it ubuntu`. Также сразу обновим индекс пакетов командой: `sudo apt-get update` (рисунок 2.15).


```

german@german-VirtualBox:~$ sudo docker run -it ubuntu
root@0e47708f8e83:/# apt-get update
Get:1 http://archive.ubuntu.com/ubuntu focal InRelease [265 kB]
Get:2 http://security.ubuntu.com/ubuntu focal-security InRelease [114 kB]
Get:3 http://archive.ubuntu.com/ubuntu focal-updates InRelease [114 kB]
Get:4 http://archive.ubuntu.com/ubuntu focal-backports InRelease [108 kB]
Get:5 http://archive.ubuntu.com/ubuntu focal/multiverse amd64 Packages [177 kB]
Get:6 http://security.ubuntu.com/ubuntu focal-security/universe amd64 Packages [844 kB]
Get:7 http://archive.ubuntu.com/ubuntu focal/restricted amd64 Packages [33.4 kB]
Get:8 http://archive.ubuntu.com/ubuntu focal/universe amd64 Packages [11.3 MB]
Get:9 http://security.ubuntu.com/ubuntu focal-security/restricted amd64 Packages [994 kB]
Get:10 http://security.ubuntu.com/ubuntu focal-security/main amd64 Packages [1592 kB]
Get:11 http://security.ubuntu.com/ubuntu focal-security/multiverse amd64 Packages [25.8 kB]
Get:12 http://archive.ubuntu.com/ubuntu focal/main amd64 Packages [1275 kB]
Get:13 http://archive.ubuntu.com/ubuntu focal-updates/multiverse amd64 Packages [29.4 kB]
Get:14 http://archive.ubuntu.com/ubuntu focal-updates/main amd64 Packages [2017 kB]
Get:15 http://archive.ubuntu.com/ubuntu focal-updates/restricted amd64 Packages

```

Рисунок 2.15 – Запуск контейнера и обновление индексов

Для выполнения работы необходимо установить текстовый редактор Nano. Nano - текстовый редактор для консоли. Используется в Unix-подобных ОС, распространяется свободно. Команда: `apt install nano` (рисунок 2.16).

```

root@0e47708f8e83:/# apt install nano
Reading package lists... Done
Building dependency tree
Reading state information... Done
Suggested packages:
  hunspell
The following NEW packages will be installed:
  nano
0 upgraded, 1 newly installed, 0 to remove and 0 not upgraded.
Need to get 269 kB of archives.
After this operation, 868 kB of additional disk space will be used.
Get:1 http://archive.ubuntu.com/ubuntu focal/main amd64 nano amd64 4.8-1ubuntu1 [269 kB]
Fetched 269 kB in 1s (329 kB/s)
debconf: delaying package configuration, since apt-utils is not installed
Selecting previously unselected package nano.
(Reading database ... 4127 files and directories currently installed.)
Preparing to unpack ../nano_4.8-1ubuntu1_amd64.deb ...
Unpacking nano (4.8-1ubuntu1) ...
Setting up nano (4.8-1ubuntu1) ...
update-alternatives: using /bin/nano to provide /usr/bin/editor (editor) in auto mode
update-alternatives: warning: skip creation of /usr/share/man/man1/editor.1.gz because associated file /usr/share/man/man1/nano.1.gz (of link group editor) doesn't exist
update-alternatives: using /bin/nano to provide /usr/bin/pico (pico) in auto mode

```

Рисунок 2.16 – Установка текстового редактора

Для создания текстового файла используем следующую команду: `nano «название файла».sh` (рисунок 2.17).

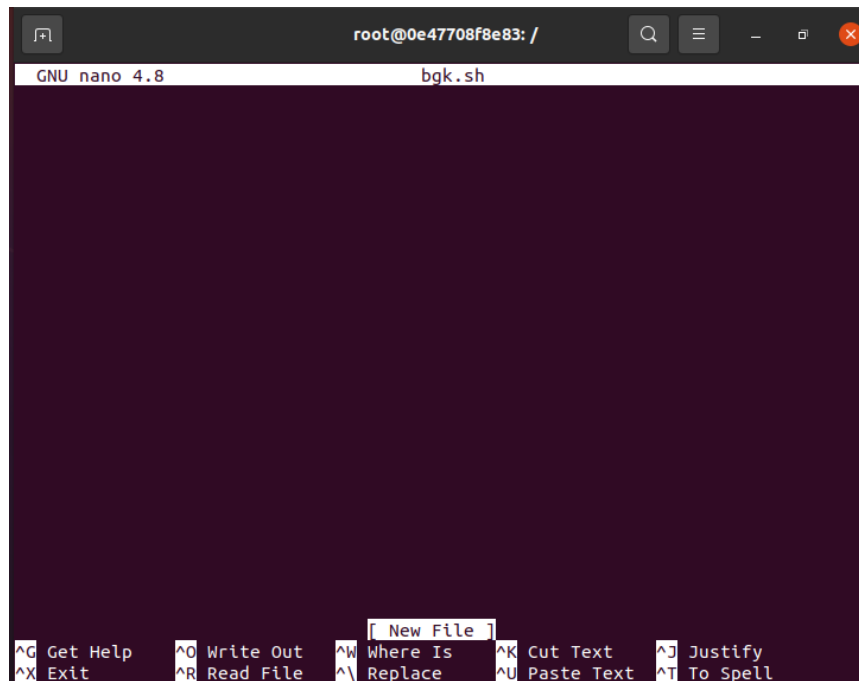


Рисунок 2.17 – Созданный текстовый файл

Проверим работу текстовых файлов с помощью команды `echo` (рисунок 2.18).



Рисунок 2.18 – Команда на вывод

Далее нужно проверить созданный файл с помощью команды `ls` (рисунок 2.19).

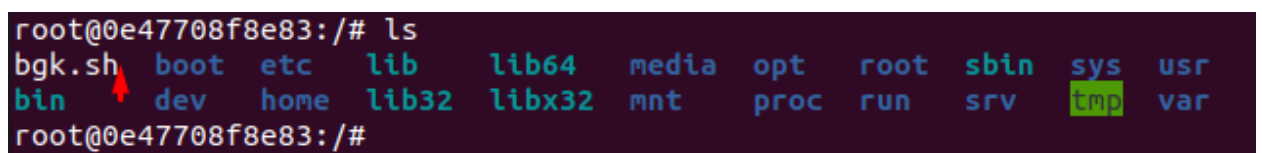


Рисунок 2.19 – Проверка созданного файла

Необходимо сделать данный файл исполняемым, для этого используем команду: `chmod ugo+x "название файла".sh` (рисунок 2.20).

```
root@0e47708f8e83:/# chmod ugo+x bgk.sh
root@0e47708f8e83:/# ls
bgk.sh  boot  etc  lib  lib64  media  opt  root  sbin  sys  usr
bin     dev  home lib32 libx32 mnt   proc run  srv   tmp  var
```

Рисунок 2.20 – Установка прав на файл

Проверив как работает скрипт командой: `sh "название файла".sh` (рисунок 2.21).

```
root@0e47708f8e83:/# sh bgk.sh
Hello 738-1
root@0e47708f8e83:/#
```

Рисунок 2.21 – Результат работы скрипта

Чтобы выйти и остановить контейнер, введем команду `exit` (рисунок 2.22).

```
root@0e47708f8e83:/# exit
exit
```

Рисунок 2.22 – Выход из контейнера

Чтобы посмотреть список всех контейнеров и их `coin id` (для подключения и запуска), нужно ввести команду: `docker ps -a` (рисунок 2.23).

```
german@german-VirtualBox:~$ docker ps -a
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS
0e47708f8e83	ubuntu	"bash"	7 minutes ago	Exited (0) About a minute ago
o	gracious banach			

Рисунок 2.23 – Просмотр контейнеров

Для запуска остановленного контейнера, нужно ввести команду: `sudo docker start "coin id"`. Подключение к существующему запущенному осуществляется через команду: `sudo docker attach "coin id"` (рисунок 2.24).

```
german@german-VirtualBox:~$ sudo docker start 0e47708f8e83
0e47708f8e83
german@german-VirtualBox:~$ sudo docker attach 0e47708f8e83
root@0e47708f8e83:/#
```

Рисунок 2.24 – Запуск контейнера

Далее создадим Dockerfile с помощью текстового редактора nano. После создания dockerfile откроется окно, где мы можем приступить к написанию кода (рисунок 2.25).

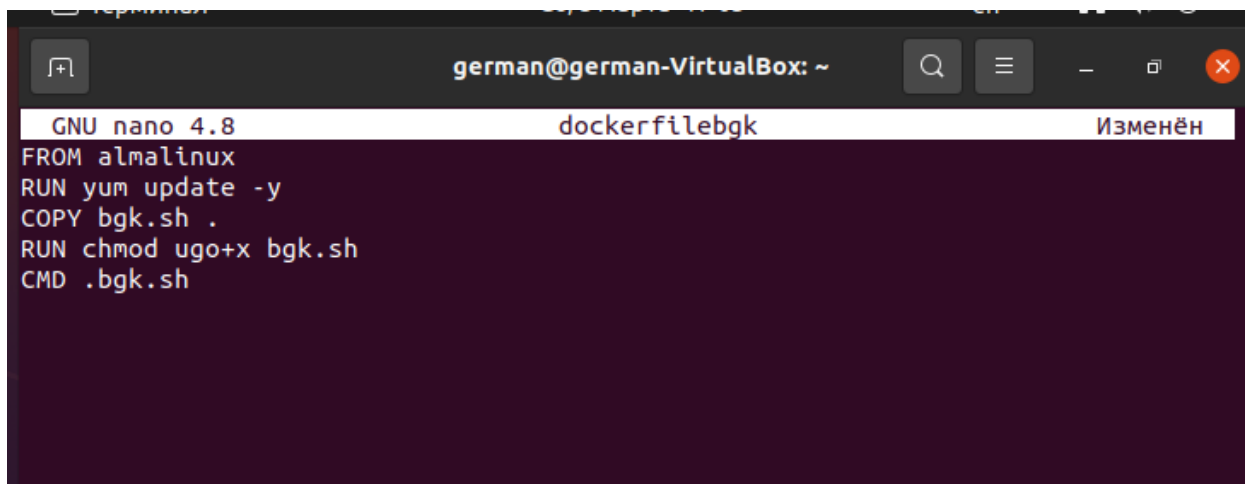


Рисунок 2.25 – DockerFile

После чего стоит собрать образ командой: `docker build -t container "имя создаваемого контейнера" -f "имя dockerfile"`. Команда: `docker run -it «имя файла»` запускает контейнер (рисунок 2.27).

(рисунок 2.26).

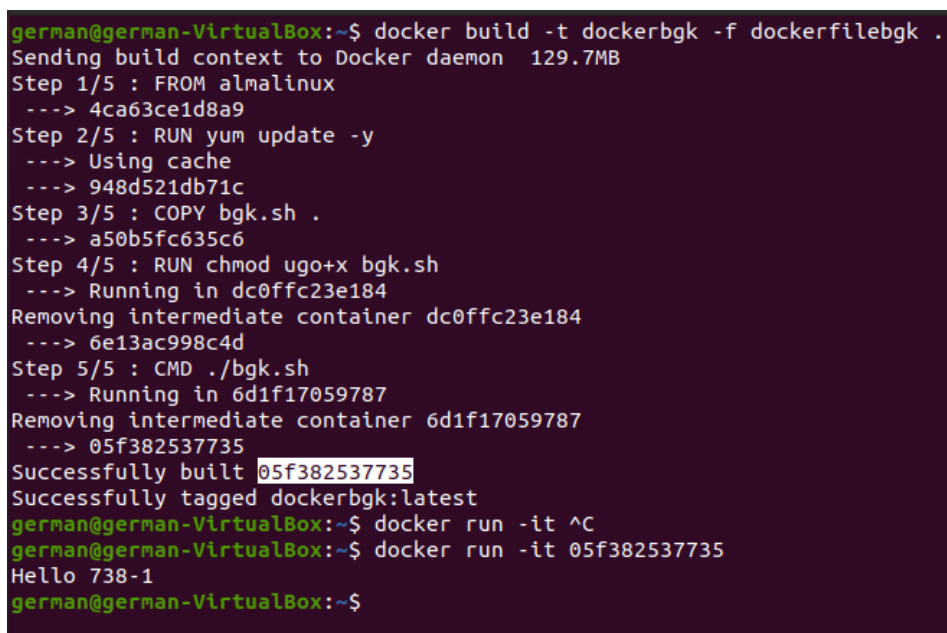


Рисунок 2.26 – Сборка образа и проверка работоспособности

Далее необходимо выполнить индивидуальное задание. В соответствии с вариантом №1 необходимо создать контейнер на операционной системе Alma Linux и написать скрипт, который:

- запрашивает у пользователя имя пакета;
- если указанный пакет установлен, то выводит информацию о пакете;
- если указанный пакет не установлен, то выводит сообщение об этом и производит поиск пакета в репозиториях;
- если пакет найден в репозиториях, то предлагает пользователю его установить или выйти из скрипта.

Рабочий скрипт и Dockerfile загружены на github (ссылка: <https://github.com/CoReShAdA>). Ниже представлен пример работы скрипта (рисунки 2.27-2.29).

```
german@german-VirtualBox:~$ docker run -it ed33328476da
Программа поиска пакетов
Программа предназначена для проверки наличия введенного пакета и его установки,
если пакет не найден
Разработчик: Беляев Герман 738-1
-----
Приятного пользования!!!
-----
Введите название пакета: g
Error: No matching Packages to list
Пакет не установлен
Last metadata expiration check: 22:52:21 ago on Sat Mar  5 10:04:22 2022.
Error: No matching Packages to list
Введенный вами не найден в репозиториях
Вы хотите продолжить? [д;Н]н
Введенное вами значение не корректно, введите д или Н
Вы хотите продолжить? [д;Н]д
Введите название пакета: nano
Error: No matching Packages to list
Пакет не установлен
Last metadata expiration check: 22:52:39 ago on Sat Mar  5 10:04:22 2022.
Available Packages
nano.x86_64                2.9.8-1.el8                baseos
Введенный вами nano был найден в репозиториях, хотите ли его установить? [д;Н]д
Last metadata expiration check: 22:52:45 ago on Sat Mar  5 10:04:22 2022.
Dependencies resolved.
=====
Package           Architecture      Version             Repository          Size
=====
```

Рисунок 2.27 – Результат работы скрипта

```

=====
Package           Architecture      Version           Repository        Size
=====
Installing:
  nano             x86_64            2.9.8-1.el8       baseos             580 k

Transaction Summary
=====
Install 1 Package

Total download size: 580 k
Installed size: 2.2 M
Downloading Packages:
nano-2.9.8-1.el8.x86_64.rpm           967 kB/s | 580 kB      00:00
-----
Total                                 374 kB/s | 580 kB      00:01
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
  Preparing      :                                1/1
  Installing     : nano-2.9.8-1.el8.x86_64        1/1
  Running scriptlet: nano-2.9.8-1.el8.x86_64        1/1
  Verifying      : nano-2.9.8-1.el8.x86_64        1/1

Installed:

```

Рисунок 2.28 – Результат работы скрипта

```

Installing      : nano-2.9.8-1.el8.x86_64          1/1
Running scriptlet: nano-2.9.8-1.el8.x86_64          1/1
Verifying      : nano-2.9.8-1.el8.x86_64          1/1

Installed:
  nano-2.9.8-1.el8.x86_64

Complete!
Вы хотите продолжить? [д;Н]д
Введите название пакета: nano
Installed Packages
 nano.x86_64                2.9.8-1.el8                @baseos
Last metadata expiration check: 22:53:08 ago on Sat Mar  5 10:04:22 2022.
Installed Packages
Name       : nano
Version    : 2.9.8
Release    : 1.el8
Architecture : x86_64
Size       : 2.2 M
Source     : nano-2.9.8-1.el8.src.rpm
Repository : @System
From repo  : baseos
Summary    : A small text editor
URL        : https://www.nano-editor.org
License    : GPLv3+
Description : GNU nano is a small and friendly text editor.

Вы хотите продолжить? [д;Н]Н
german@german-VirtualBox:~$

```

Рисунок 2.29 – Результат работы скрипта

3 Заключение

В ходе выполнения работы было произведено ознакомление с операционной системой и основными командами ОС Unix/Linux и с возможностями Docker для контейнеризации окружения программного обеспечения и его зависимостей, средств разработки и сборки ПО.