



МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования

«МИРЭА – Российский технологический университет»

РТУ МИРЭА

Институт Информационных технологий

Кафедра Математического обеспечения и стандартизации информационных
технологий

Отчет по практическим работам № 13-15

по дисциплине «Системное программное обеспечение»

Выполнил:

Студент группы ИВБО-11-23

Туктаров Т.А.

Проверил:

Ст. преп. Овчинникова М.А.

МОСКВА 2025 г.

1 ПРАКТИЧЕСКАЯ РАБОТА 13

Задание

1. Загрузить на машину образ и создать контейнер с его помощью, согласно варианту: Alpine версии 3.20.5
2. Запустить контейнер с веб-сервером, отображающий содержимое корневой директории. В хостовой системе сервер должен открываться на порту 8821 (8800 + номер варианта).

Загружаем образ alpine версии 3.20.5:

```
timur@DESKTOP-C5JARVU:~$ sudo docker pull alpine:3.20.5
[sudo] password for timur:
3.20.5: Pulling from library/alpine
66a3d608f3fa: Pull complete
Digest: sha256:31687a2fdd021f85955bf2d0c2682e9c0949827560e1db546358ea094f740f12
Status: Downloaded newer image for alpine:3.20.5
docker.io/library/alpine:3.20.5
```

Запускаем образ на порте 8821:

```
timur@DESKTOP-C5JARVU:~$ sudo docker run -p8821:8000 -it python python -m http.server
Serving HTTP on 0.0.0.0 port 8000 (http://0.0.0.0:8000/) ...
172.18.0.1 - - [17/May/2025 15:53:56] "GET / HTTP/1.1" 200 -
172.18.0.1 - - [17/May/2025 15:53:56] code 404, message File not found
172.18.0.1 - - [17/May/2025 15:53:56] "GET /favicon.ico HTTP/1.1" 404 -
```

Смотрим результат на localhost с указанным портом:

Directory listing for /

- [.dockerenv](#)
- [bin@](#)
- [boot/](#)
- [dev/](#)
- [etc/](#)
- [home/](#)
- [lib@](#)
- [lib64@](#)
- [media/](#)
- [mnt/](#)
- [opt/](#)
- [proc/](#)
- [root/](#)
- [run/](#)
- [sbin@](#)
- [srv/](#)
- [sys/](#)
- [tmp/](#)
- [usr/](#)
- [var/](#)

<http://localhost:8821>

2 ПРАКТИЧЕСКАЯ РАБОТА 14

Задание:

1. Создать директорию data и в ней файл student.txt, содержащий ФИО, название группы и номер варианта.
2. Запустить контейнер с веб-сервером и примонтированной директорией data в директорию /mnt/files/ в контейнере. Контейнер должен отображать содержимое директории /mnt/files и открываться на порту (8800 + номер варианта).

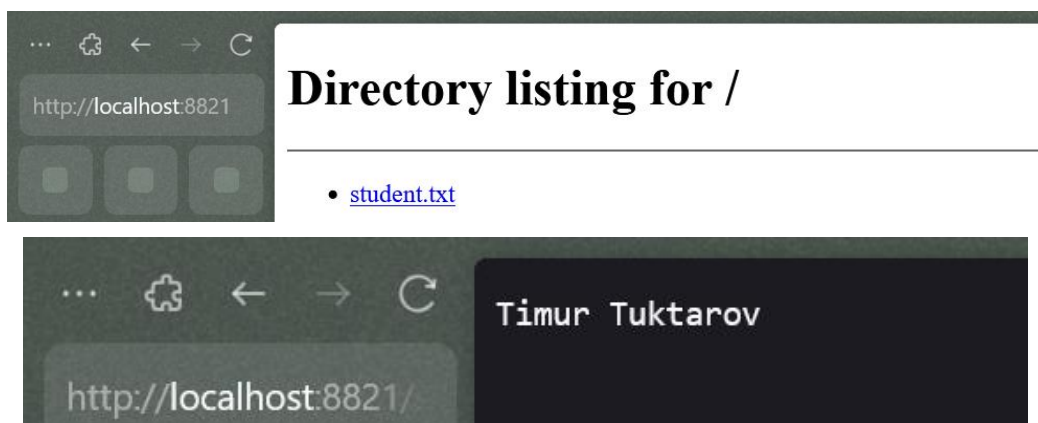
Создаем в корневой директории папку data и создаем в ней файл students.txt. Записываем в него ФИО.

```
timur@DESKTOP-C5JARVU:~$ mkdir data
timur@DESKTOP-C5JARVU:~$ cd data
timur@DESKTOP-C5JARVU:~/data$ touch student.txt
timur@DESKTOP-C5JARVU:~/data$ ls
student.txt
timur@DESKTOP-C5JARVU:~/data$ cat "Timur Tuktarov" > student.txt
cat: 'Timur Tuktarov': No such file or directory
timur@DESKTOP-C5JARVU:~/data$ echo "Timur Tuktarov" > student.txt
```

Запускаем контейнер с веб-сервером, и примонтированной директорией data.

```
timur@DESKTOP-C5JARVU:~$ sudo docker run -p8821:8000 --rm --name pyserver -d -v $(pwd)/data:/mnt/files python python -m
http.server -d /mnt/files
[sudo] password for timur:
9ce12a0497f16833c7564f3350f1a72e58e4c22a53fdc772ed98a656bdd2c47d
timur@DESKTOP-C5JARVU:~$ sudo docker exec -it pyserver bash
root@9ce12a0497f1:/# cd /mnt
root@9ce12a0497f1:/mnt# ls
files
root@9ce12a0497f1:/mnt# cd files
root@9ce12a0497f1:/mnt/files# ls
student.txt
root@9ce12a0497f1:/mnt/files# |
```

Вывод на localhost'e:



3 ПРАКТИЧЕСКАЯ РАБОТА 15

Задание

Написать Dockerfile, собрать образ, запустить контейнер (и записать команду для его запуска). Для монтирования создайте директорию data и в ней файл student.txt, содержащий ФИО, название группы и номер варианта.

Для установки пакетов использовать команду `apt install -y название-пакета`.

— необходимо использовать базовый образ `ubuntu:22.04`

— добавить директорию `data` в директорию `/mnt/files/` в контейнере.

Запустить веб-сервер, отображающий содержимое `/mnt/files`, в хостовой системе должен открываться на порту 8821 (8800 + номер варианта).

Установить пакет, согласно варианту: `patch`.

Создание Dockerfile'a:

```
timur@DESKTOP-C5JARVU:~$ touch Dockerfile
timur@DESKTOP-C5JARVU:~$ ls
Dockerfile  Tuktarov  ansible  data  iproute2  myfiles
```

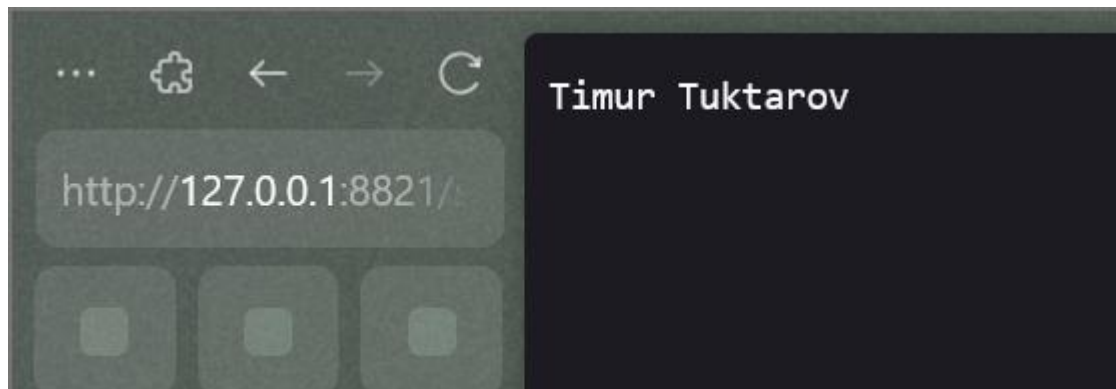
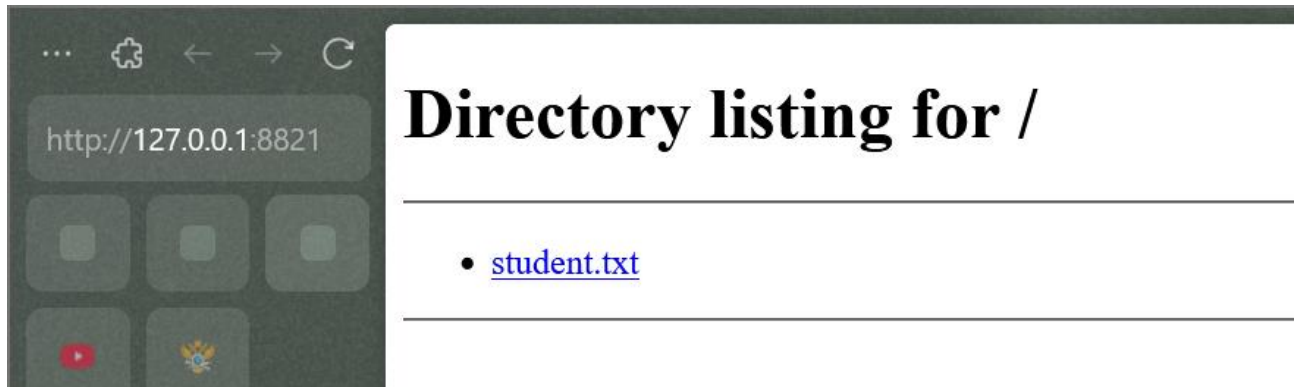
Пример Dockerfile:

```
FROM ubuntu:22.04
RUN apt update \
    && apt install -y python3 patch \
    && cd /usr/bin \
    && ln -s python3 python
ADD ./data /mnt/files/
EXPOSE 80
CMD ["python", "-m", "http.server", "-d", "/mnt/files", "80"]
```

Запуск Dockerfile'a

```
timur@DESKTOP-C5JARVU:~$ sudo docker build -t timimage .
[+] Building 2.1s (8/8) FINISHED
=> [internal] load build definition from Dockerfile
=> => transferring dockerfile: 247B
=> [internal] load metadata for docker.io/library/ubuntu:22.04
=> [internal] load .dockerignore
=> => transferring context: 2B
=> [1/3] FROM docker.io/library/ubuntu:22.04@sha256:67cadaff1dca1870f9fce41360d5a7eb6f7dcd3745e53c79ad5efd856311
=> [internal] load build context
=> => transferring context: 61B
=> CACHED [2/3] RUN apt update && apt install -y python3 patch && cd /usr/bin && ln -s python3 python
=> [3/3] ADD ./data /mnt/files/
=> exporting to image
=> => exporting layers
=> => writing image sha256:8d046013d24c3b8e73ecabce4c634f6e7c46a29b5f957681de016073a0df9d33
=> => naming to docker.io/library/timimage
timur@DESKTOP-C5JARVU:~$ sudo docker run --rm -it -p8821:80 timimage
```

Вывод localhost:



Проверка работы установленной утилиты patch:

```
root@125361a31e3e:/# usr/bin/patch --help
Usage: usr/bin/patch [OPTION]... [ORIGFILE [PATCHFILE]]

Input options:

-p NUM --strip=NUM Strip NUM leading components from file names.
-F LINES --fuzz LINES Set the fuzz factor to LINES for inexact matching.
-l --ignore-whitespace Ignore white space changes between patch and input.

-c --context Interpret the patch as a context difference.
-e --ed Interpret the patch as an ed script.
-n --normal Interpret the patch as a normal difference.
-u --unified Interpret the patch as a unified difference.

-N --forward Ignore patches that appear to be reversed or already applied.
-R --reverse Assume patches were created with old and new files swapped.

-i PATCHFILE --input=PATCHFILE Read patch from PATCHFILE instead of stdin.

Output options:

-o FILE --output=FILE Output patched files to FILE.
-r FILE --reject-file=FILE Output rejects to FILE.
```