



**Федеральное государственное бюджетное образовательное учреждение  
высшего образования**

**Московский государственный технический университет**

**им. Н. Э. Баумана**

**Национальный исследовательский университет (МГТУ им. Н.Э. Баумана)**

**Практическое задание:**

**«Основы виртуализации»**

Выполнил: Шарафутдинов Э. М.

Группа: РЛ6-61Б

Москва 2023 г.

## **Содержание**

<b>Цель работы</b>	<b>3</b>
<b>Краткие теоретические сведения</b>	<b>3</b>
<b>Порядок выполнения работы</b>	<b>4</b>
<b>Выводы</b>	<b>6</b>

## **Цель работы**

Ознакомиться с системой виртуализации VirtualBox, научиться настраивать виртуальную машину, совершать простые операции с ней, устанавливать гостевую операционную систему на ВМ.

## **Краткие теоретические сведения**

Виртуальная машина мало чем отличается от физических компьютеров. У нее есть ЦП, память, диски для хранения файлов и возможность подключения к Интернету.

Виртуализация — это процесс создания программной, или "виртуальной" версии компьютера с выделенными ресурсами ЦП, памяти и хранилища, которые "заимствуются" у физического компьютера и (или) удаленного сервера. Виртуальная машина — это компьютерный файл, который действует как обычный компьютер. Она может работать в окне в качестве отдельной вычислительной среды или даже как целая система, как это часто реализуется на рабочем компьютере, которым пользуются несколько человек.

Хотя виртуальные машины работают как отдельные компьютеры с отдельными операционными системами и приложениями, главное их преимущество в том, что они совершенно не зависят друг от друга и от физического компьютера, на котором размещены. Специальный программный компонент, который называется гипервизором или диспетчером виртуальных машин, позволяет одновременно запускать разные операционные системы на разных виртуальных машинах. Благодаря этому можно запускать виртуальные машины Linux, например, в ОС Windows или запускать более раннюю версию Windows на более поздней.

А поскольку виртуальные машины не зависят друг от друга, они чрезвычайно портативны. Можно практически мгновенно перемещать виртуальные машины с одного гипервизора на другой гипервизор на другом компьютере.

## Порядок выполнения работы

### 1. Выбор и скачивание образа ОС-Debian на сайте debian.

Name	Last modified	Size
<a href="#">Parent Directory</a>		-
<a href="#">SHA256SUMS</a>	2023-06-10 15:56	302
<a href="#">SHA256SUMS.sign</a>	2023-06-10 15:58	833
<a href="#">SHA512SUMS</a>	2023-06-10 15:56	494
<a href="#">SHA512SUMS.sign</a>	2023-06-10 15:58	833
<a href="#">debian-12.0.0-amd64-netinst.iso</a>	2023-06-10 14:04	738M
<a href="#">debian-edu-12.0.0-amd64-netinst.iso</a>	2023-06-10 14:04	747M
<a href="#">debian-mac-12.0.0-amd64-netinst.iso</a>	2023-06-10 14:04	736M

Рисунок 1 - образы диска для amd64

### 2. Скачивание и установка Oracle VM VirtualBox.

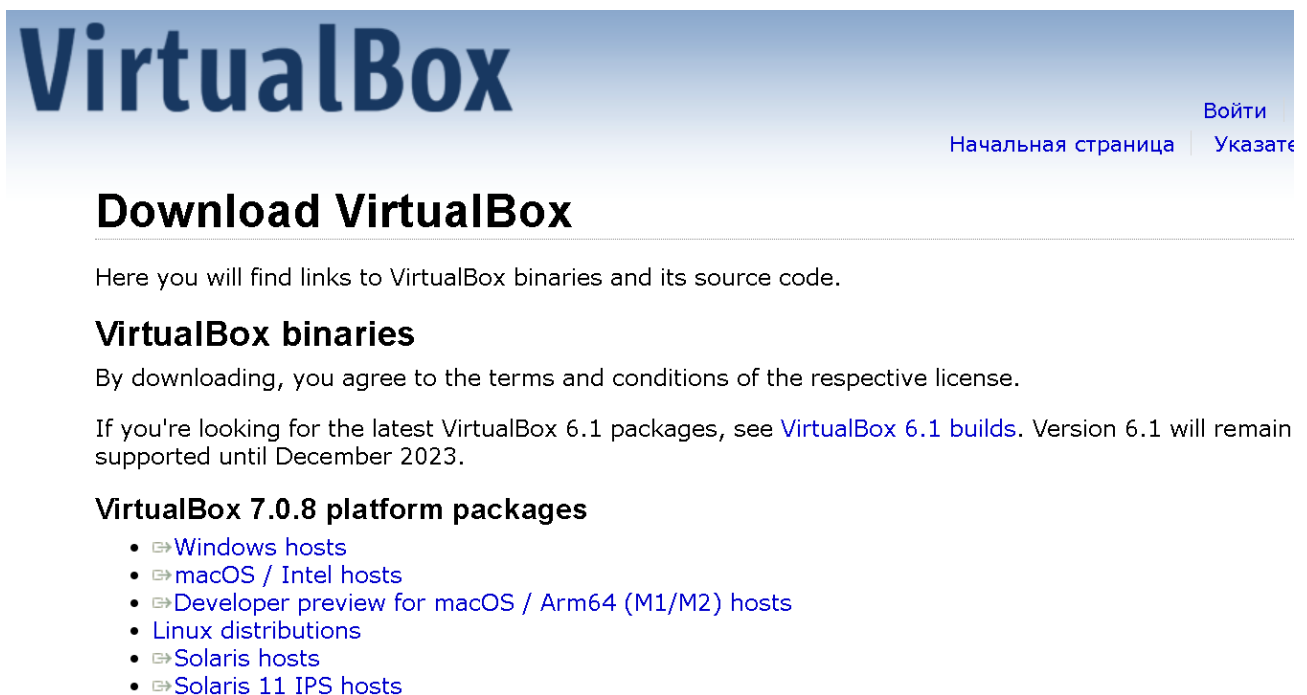


Рисунок 2 - скачивание Oracle VM VirtualBox

### 3. Создание виртуальной машины.

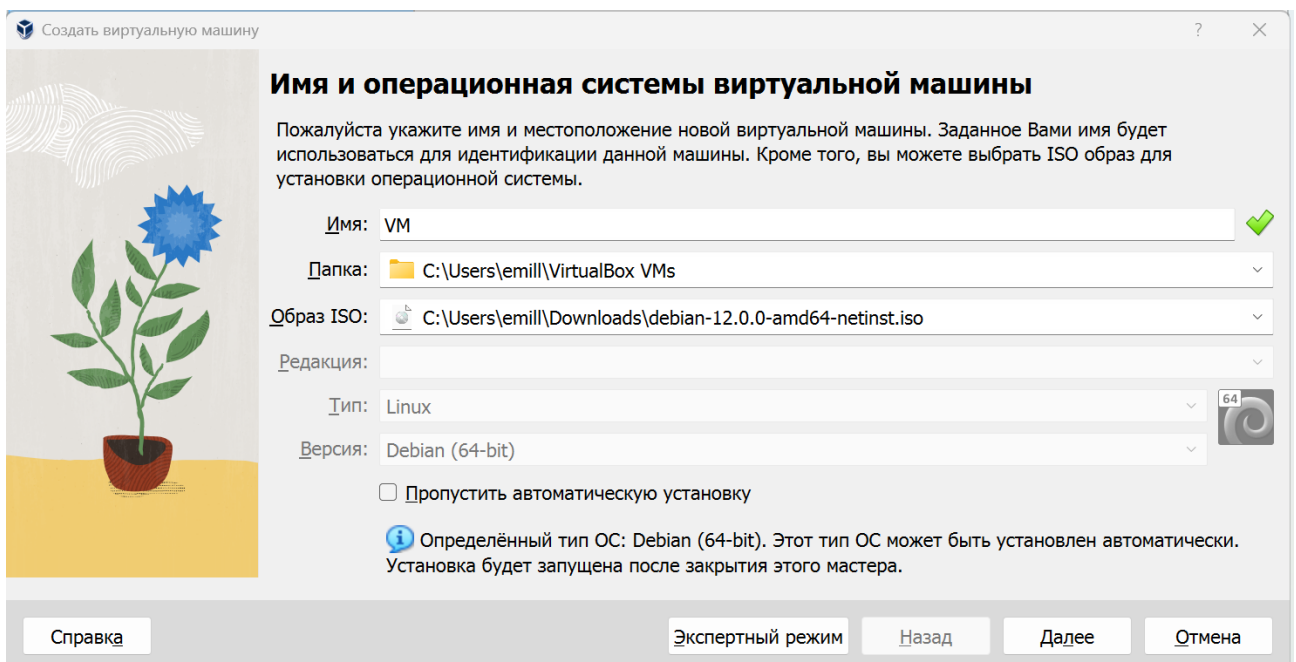


Рисунок 3 - создание виртуальной машины

4. Установка ОС Debian на жёсткий диск виртуальной машины.
5. Установка графического интерфейса ОС
6. Ввод в консоль команд free, df, lscpu, ip route, ip addr

```
emil@debian:~$ free
              total        used        free      shared  buff/cache   available
Mem:           1000112      477156      196828         18016        326128        365688
Swap:          998396       98136       900260

emil@debian:~$ df
Файловая система 1K-блоков  Использовано  Доступно  Использовано%  Смонтировано в
udev                475384           0     475384           0%  /dev
tmpfs               100012           912     99100           1%  /run
/dev/sda1           9232860      5988524     2753740        69%  /
tmpfs               500056           0     500056           0%  /dev/shm
tmpfs                5120            4        5116           1%  /run/lock
tmpfs              100008           72     99936           1%  /run/user/1000
```

Рисунок 4 - ввод команд free, df

```

emil@debian:~$ lscpu
Architecture:            x86_64
CPU op-mode(s):          32-bit, 64-bit
Byte Order:               Little Endian
Address sizes:            48 bits physical, 48 bits virtual
CPU(s):                   1
On-line CPU(s) list:      0
Thread(s) per core:       1
Core(s) per socket:       1
Socket(s):                1
NUMA node(s):             1
Vendor ID:                AuthenticAMD
CPU family:               23
Model:                    24
Model name:               AMD Ryzen 5 3500U with Radeon Vega Mobile Gfx
Stepping:                  1
CPU MHz:                  2096.000
BogoMIPS:                 4192.00
Hypervisor vendor:        KVM
Virtualization type:      full
L1d cache:                32 KiB
L1i cache:                64 KiB
L2 cache:                 512 KiB
L3 cache:                 4 MiB

```

Рисунок 5 - ввод команды lscpu

```

emil@debian:~$ ip route
default via 10.0.2.2 dev enp0s3 proto dhcp metric 100
10.0.2.0/24 dev enp0s3 proto kernel scope link src 10.0.2.15 metric 100
169.254.0.0/16 dev enp0s3 scope link metric 1000
emil@debian:~$ ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default ql
en 1000
    link/ether 08:00:27:96:0f:37 brd ff:ff:ff:ff:ff:ff
    inet 10.0.2.15/24 brd 10.0.2.255 scope global dynamic noprefixroute enp0s3
        valid_lft 84685sec preferred_lft 84685sec
    inet6 fe80::a00:27ff:fe96:f37/64 scope link noprefixroute
        valid_lft forever preferred_lft forever

```

Рисунок 6 - ввод команд ip route и ip addr

## Выводы

В ходе выполнения практической работы были успешно достигнуты поставленные цели. Мы ознакомились с системой виртуализации VirtualBox, которая позволяет создавать и управлять виртуальными машинами на компьютере. Это дает возможность работать с различными операционными системами и приложениями без необходимости установки на физический

компьютер, что существенно упрощает тестирование и разработку программного обеспечения.

В процессе работы мы научились настраивать виртуальную машину, задавая параметры, такие как объём оперативной памяти, количество процессорных ядер, размер жесткого диска и тип сетевого подключения.

Также мы освоили выполнение простых операций с виртуальной машиной, таких как запуск, остановка, сохранение состояния и восстановление.