

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ ФЕДЕРАЦИИ  
Федеральное государственное автономное образовательное учреждение  
высшего образования  
«КРЫМСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ им. В. И. ВЕРНАДСКОГО»  
ФИЗИКО-ТЕХНИЧЕСКИЙ ИНСТИТУТ  
Кафедра компьютерной инженерии и моделирования

**ОТЧЕТ ПО ПРАКТИЧЕСКОМУ ЗАДАНИЮ №4  
«ВИРТУАЛИЗАЦИЯ.ЭМУЛЯЦИЯ.ГИПЕРВИЗОРЫ»**

Практическая работа  
по дисциплине «Программная инженерия»  
студента 1 курса группы ПИ-б-о-241(1)  
Коробка Илья Леонидович

09.03.04 «Программная инженерия»

Симферополь, 2025

# Теоретическая база

## Основные понятия

**Виртуализация** - технология создания виртуальных (а не физических) версий компьютерных ресурсов: операционных систем, серверов, устройств хранения данных или сетевых ресурсов.

**Эмуляция** - полная имитация поведения одной системы с помощью другой системы. Эмулятор переводит каждую инструкцию гостевой системы в инструкции хост-системы.

**Гипервизор** - программа или аппаратно-программный комплекс, обеспечивающий одновременное выполнение нескольких операционных систем на одном физическом компьютере.

### Типы гипервизоров:

1. **Тип 1 (Bare Metal):** работают непосредственно на аппаратном обеспечении
  - Примеры: VMware vSphere/ESXi, Microsoft Hyper-V, Xen
1. **Тип 2 (Hosted):** работают поверх операционной системы
  - Примеры: VMware Workstation, VirtualBox, QEMU

### QEMU и KVM

**QEMU** (Quick Emulator) - гипервизор типа 2, способный работать в двух режимах:

- **Эмуляция:** полная программная эмуляция процессора и устройств
- **Виртуализация:** использование аппаратных возможностей процессора (с KVM)

**KVM** (Kernel-based Virtual Machine) - модуль ядра Linux, превращающий Linux в гипервизор типа 1. Использует аппаратные расширения виртуализации Intel VT-x или AMD-V.

# Пошаговое выполнение лабораторной работы

## 1. Работа с qemu-img

### 1.1 Создание образа диска в формате VMDK

`qemu-img create -f vmdk /tmp/disk-base_${USER}.vmdk 1.5G`

```
root@AvansPC /h/sec# qemu-img create -f vmdk /tmp/disk-base_${USER}.vmdk 1.5G
Formatting '/tmp/disk-base_root.vmdk', fmt=vmdk size=1610612736 compat6=off hwversion=undefined
root@AvansPC /h/sec# |
```

#### Разбор команды:

- `qemu-img create` - команда создания образа диска
- `-f vmdk` - формат образа (VMware Virtual Disk)
- `/tmp/disk-base_${USER}.vmdk` - путь и имя файла
- `1.5G` - размер диска

#### Форматы образов:

- **VMDK**: формат VMware, совместим с VMware продуктами
- **QCOW2**: формат QEMU, поддерживает снимки, сжатие, шифрование
- **VDI**: формат VirtualBox
- **RAW**: несжатый образ, максимальная производительность

### 1.2 Конвертация формата

`qemu-img convert -f vmdk -O qcow2 /tmp/disk-base_${USER}.vmdk /tmp/disk-base_${USER}.qcow2`

```
root@AvansPC /h/sec# qemu-img convert -f vmdk -O qcow2 /tmp/disk-base_${USER}.vmdk /tmp/disk-base_${USER}.qcow2
root@AvansPC /h/sec# |
```

#### Разбор команды:

- `convert` - команда конвертации
- `-f vmdk` - исходный формат
- `-O qcow2` - целевой формат
- Исходный и целевой файлы

### 1.3 Изменение размера образа

```
qemu-img resize /tmp/disk-base_${USER}.qcow2 7G
```

```
root@AvansPC /h/sec# qemu-img resize /tmp/disk-base_${USER}.qcow2 7G
Image resized.
root@AvansPC /h/sec# |
```

**Важно:** Это изменяет только размер образа, не файловую систему внутри!

### 1.4 Создание дочернего образа

```
qemu-img create -f qcow2 -b /tmp/disk-base_${USER}.qcow2 /tmp/disk_${USER}.qcow2
```

```
root@AvansPC /h/sec# qemu-img create -f qcow2 -F qcow2 -b /tmp/disk-base_${USER}.qcow2 /tmp/disk_${USER}.qcow2
Formatting '/tmp/disk_root.qcow2', fmt=qcow2 cluster_size=65536 extended_l2=off compression_type=zlib size=7516192768 backing_file=/tm
p/disk-base_root.qcow2 backing_fmt=qcow2 lazy_refcounts=off refcount_bits=16
root@AvansPC /h/sec# |
```

#### Разбор команды:

- -b - backing file (базовый образ)
- Дочерний образ хранит только изменения относительно базового

#### Преимущества дочерних образов:

- Экономия места на диске
- Быстрое создание множества ВМ с одной базой
- Возможность отката к базовому состоянию

## 2. Проверка поддержки KVM

# Проверка модуля KVM

```
lsmod | grep kvm
```

# Проверка поддержки аппаратной виртуализации

```
grep -E "(vmx|svm)" /proc/cpuinfo
```

```
root@AvansPC /h/sec# lsmod | grep kvm
kvm_amd                237568    0
kvm                    1384448    1 kvm_amd
irqbypass              12288     1 kvm
ccp                    184320    2 kvm_amd,amdtee
root@AvansPC /h/sec# grep -E "(vmx|svm)" /proc/cpuinfo
flags                : fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca
xt fxsr_opt pdpe1gb rdtscp lm constant_tsc rep_good nopl xtopology nonstop_
```

# Проверка устройства KVM  
ls -l /dev/kvm

```
root@AvansPC /h/sec# ls -l /dev/kvm
crw-rw-rw- 1 root kvm 10, 232 мая 29 16:20 /dev/kvm
root@AvansPC /h/sec# |
```

**Intel VT-x:** флаг vmx в /proc/cpuinfo **AMD-V:** флаг svm в /proc/cpuinfo

### 3. Запуск виртуальной машины

Для начала мне пришлось скачать qemu-system-x86 - программа для запуска гостевых ОС

```
root@AvansPC /h/sec [1]# sudo pacman -S qemu-system-x86
разрешение зависимостей...
проверка конфликтов...

Пакеты (14) capstone-5.0.6-1 dtc-1.7.2-4 edk2-ovmf-202411-1
libslirp-4.9.1-1 libtraceevent-1:1.8.4-1
libtracefs-1.8.2-1 libxdp-1.5.4-1 ndctl-79-1
qemu-common-10.0.0-5
qemu-system-x86-firmware-10.0.0-5 seabios-1.16.3-1
vde2-2.3.3-6 wolfssl-5.8.0-1
qemu-system-x86-10.0.0-5

Будет загружено:      18,49 MiB
Будет установлено:   104,48 MiB

:: Приступить к установке? [Y/n]
:: Получение пакетов...
```

А после и образ системы

```

root@AvansPC /h/sec [1]# wget http://tinycorelinux.net/15.x/x86_64
/release/TinyCorePure64-15.0.iso
--2025-05-29 16:38:26-- http://tinycorelinux.net/15.x/x86_64/rele
ase/TinyCorePure64-15.0.iso
Распознаётся tinycorelinux.net (tinycorelinux.net)... 128.127.66.77
Подключение к tinycorelinux.net (tinycorelinux.net)|128.127.66.77|
:80... соединение установлено.
HTTP-запрос отправлен. Ожидание ответа... 200 OK
Длина: 33554432 (32M) [application/octet-stream]
Сохранение в: «TinyCorePure64-15.0.iso»

TinyCorePure64-1 100%[=====>] 32,00M 5,10MB/s за 7,4s

2025-05-29 16:38:33 (4,30 MB/s) - «TinyCorePure64-15.0.iso» сохрान
ён [33554432/33554432]

```

```

qemu-system-x86_64 \
-smp 1 \
-m 1.5G \
-vga std \
-hda /tmp/disk_${USER}.qcow2 \
-cdrom tinycorelinux.iso \
-netdev user,id=net0,hostfwd=tcp::8080-:80 \
-device virtio-net-pci,netdev=net0 \
-boot menu=on,splash-time=10000 \
-enable-kvm \
-serial none \
-monitor telnet:127.0.0.1:10023,server,nowait

```

```

root@AvansPC /h/sec# qemu-system-x86_64 \
-smp 1 \
-m 1.5G \
-vga std \
-hda /tmp/disk_${USER}.qcow2 \
-cdrom TinyCorePure64-15.0.iso \
-netdev user,id=net0,hostfwd=tcp::8080-
:80 \
-device e1000,netdev=net0 \
-boot menu=on,splash-time=10000 \
-serial none \
-monitor telnet:127.0.0.1:10023,server,
nowait

VNC server running on ::1:5900
|

```



## Детальный разбор параметров:

- -smp 1: количество CPU (Symmetric MultiProcessing)
- -m 1.5G: объем оперативной памяти
- -vga std: тип видеокарты (std, cirrus, vmware, qxl, none)
- -hda: первый жесткий диск (IDE)
- -cdrom: CD-ROM образ
- -netdev user,id=net0,hostfwd=tcp::8080-:80:
- user - режим пользовательской сети (NAT)
- id=net0 - идентификатор сетевого устройства
- hostfwd=tcp::8080-:80 - проброс портов
- -device virtio-net-pci,netdev=net0: сетевая карта AMD
- -boot menu=on,splash-time=10000:
- menu=on - показать меню загрузки
- splash-time=10000 - таймаут в миллисекундах
- -enable-kvm: использовать KVM (аппаратная виртуализация)
- -serial none: отключить последовательный порт
- -monitor telnet:127.0.0.1:10023,server,nowait:
- монитор QEMU через telnet
- server - QEMU слушает подключения
- nowait - не ждать подключения для старта

Для подключения к ВМ пришлось скачать дополнительный софт

```
root@AvansPC /h/sec [7]# sudo pacman -Syu tigervnc
:: Обновление баз данных пакетов...
core is up to date
extra              7,8 MiB   4,89 MiB/s 00:02 [#####] 100%
multilib is up to date
gh0stzk-dotfiles is up to date
chaotic-aur         627,4 KiB   470 KiB/s 00:01 [#####] 100%
:: Запускается полное обновление системы...
разрешение зависимостей...
проверка конфликтов...

Пакеты (2) fltk-1.3.9-1  tigervnc-1.15.0-1

Будет загружено:      3,94 MiB
Будет установлено:   12,47 MiB
```

#### 4. Установка пакетов в TinyCore Linux

`tce-load -wi openssh`

**TinyCore Linux** - минималистичная Linux-система:

- `tce-load` - менеджер пакетов TinyCore
- `-w` - загрузить из интернета
- `-i` - установить пакет

#### 5. Измерение производительности

##### Время загрузки системы

##### С KVM (виртуализация):

`time qemu-system-x86_64 -enable-kvm [остальные параметры]`

```
-----
Executed in 364.75 secs    fish           external
   usr time    5.86 secs  568.00 micros    5.86 secs
   sys time    2.76 secs  345.00 micros    2.76 secs
```

##### Без KVM (эмуляция):

`time qemu-system-x86_64 [остальные параметры без -enable-kvm]`

```
t:127.0.0.1:10023,server,nowait
VNC server running on ::1:5900
q
^Cqemu-system-x86_64: terminating on signal 2

-----
Executed in 433.83 secs    fish           external
   usr time    83.97 secs    0.00 micros   83.97 secs
   sys time     4.79 secs  960.00 micros    4.78 secs
```

```
ot@AvansPC /h/sec# |
```



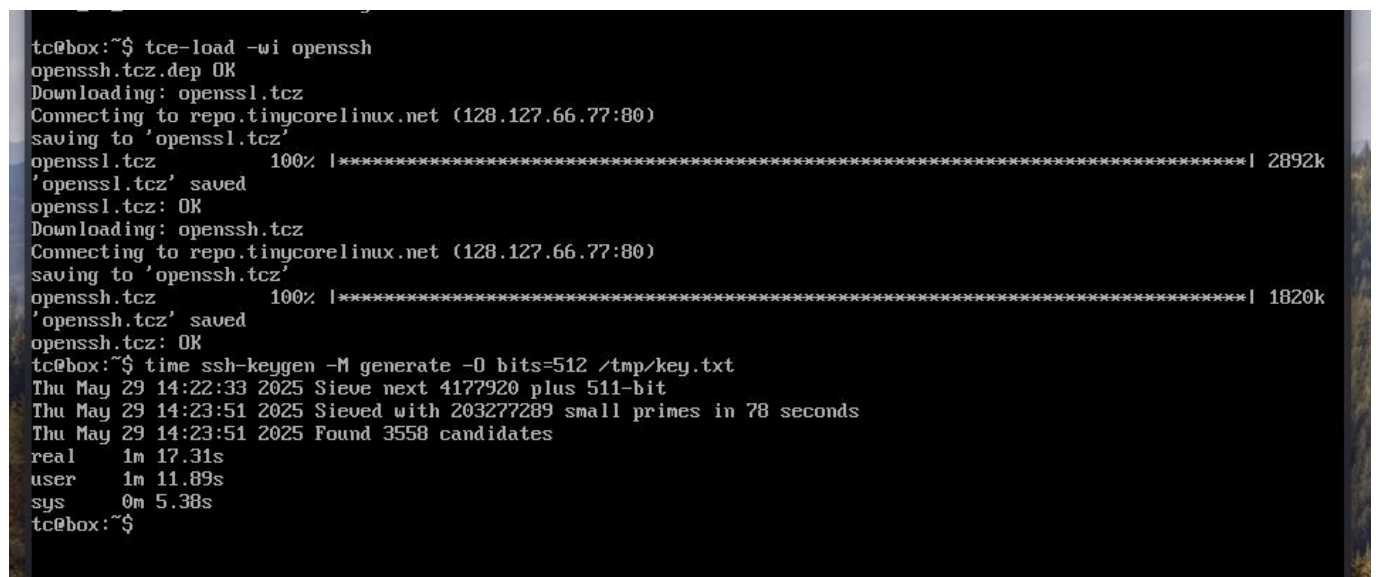
## Время выполнения команды

В виртуальной машине:

```
time ssh-keygen -M generate -O bits=512 /tmp/key.txt
```

## Разбор команды ssh-keygen:

- -M generate - генерировать модули для DH (Diffie-Hellman)
- -O bits=512 - размер ключа в битах
- /tmp/key.txt - выходной файл



```
tc@box:~$ tce-load -wi openssl
openssl.tcz.dep OK
Downloading: openssl.tcz
Connecting to repo.tinycorelinux.net (128.127.66.77:80)
saving to 'openssl.tcz'
openssl.tcz      100% |*****| 2892k
'openssl.tcz' saved
openssl.tcz: OK
Downloading: openssl.tcz
Connecting to repo.tinycorelinux.net (128.127.66.77:80)
saving to 'openssl.tcz'
openssl.tcz      100% |*****| 1820k
'openssl.tcz' saved
openssl.tcz: OK
tc@box:~$ time ssh-keygen -M generate -O bits=512 /tmp/key.txt
Thu May 29 14:22:33 2025 Sieve next 4177920 plus 511-bit
Thu May 29 14:23:51 2025 Sieved with 203277289 small primes in 78 seconds
Thu May 29 14:23:51 2025 Found 3558 candidates
real    1m 17.31s
user    1m 11.89s
sys     0m 5.38s
tc@box:~$
```

## 6. Работа с монитором QEMU

### Подключение к монитору

```
telnet 127.0.0.1 10023
```

**Отключение:** Ctrl+] затем Ctrl+d

### Основные команды монитора

#### Информация о процессорах:

```
info cpus
```

#### Информация о регистрах:

```
info registers
```

```

root@AvansPC /h/sec# telnet 127.0.0.1 10023
Trying 127.0.0.1...
Connected to 127.0.0.1.
Escape character is '^]'.
QEMU 10.0.0 monitor - type 'help' for more information
(qemu) ls
unknown command: 'ls'
(qemu) info cpus
* CPU #0: thread_id=32050
(qemu) info re
registers replay
(qemu) info registers

CPU#0
EAX=00000000 EBX=0016e614 ECX=0016e614 EDX=00000000
ESI=000003e8 EDI=00000000 EBP=000003e9 ESP=5ffaface
EIP=0010017c EFL=00000246 [---Z-P-] CPL=0 II=0 A20=1 SMM=0 HLT=1
ES =0028 00000000 ffffffff 00c09300 DPL=0 DS [-WA]
CS =0020 00000000 ffffffff 00c09b00 DPL=0 CS32 [-RA]

```

### Информация о сети:

info network

### Информация о блочных устройствах:

info block

```

(qemu) info network
e1000.0: index=0,type=nic,model=e1000,macaddr=52:54:00:12:34:56
\ net0: index=0,type=user,net=10.0.2.0,restrict=off
(qemu) info block
ide0-hd0 (#block193): /tmp/disk_root.qcow2 (qcow2)
  Attached to:      /machine/unattached/device[5]
  Cache mode:      writeback
  Backing file:     /tmp/disk-base_root.qcow2 (chain depth: 1)

ide1-cd0 (#block578): TinyCorePure64-15.0.iso (raw, read-only)
  Attached to:      /machine/unattached/device[6]
  Removable device: locked, tray closed
  Cache mode:      writeback

floppy0: [not inserted]
  Attached to:      /machine/unattached/device[15]
  Removable device: not locked, tray closed
(qemu) |

```

## Управление пробросом портов:

```
# Удалить существующий проброс
netdev_del net0
netdev_add user,id=net0,hostfwd=tcp::2222-:22
```

```
(qemu) netdev_del net0
(qemu) netdev_add
Error: Parameter 'id' is missing
(qemu) netdev_add user,id=net0,hostfwd=tcp::2222-:22
(qemu) |
```

## Управление снимками:

```
# Сохранить состояние
savevm running_state
```

```
# Загрузить состояние
loadvm running_state
```

```
# Список снимков
info snapshots
```

```
(qemu) savevm running_state
(qemu) loadvm running_state
(qemu) info snapshots
List of snapshots present on all disks:
ID      TAG          VM_SIZE          DATE          VM_CL
OCK     ICOUNT
--      running_state  137 MiB 2025-05-29 17:33:28  0000:04:50.
958     --
(qemu) |
```

## Управление ВМ:

# Перезагрузка (reset)  
system\_reset

# Выключение  
quit  
# или  
system\_powerdown

```
-----
Executed in 364.75 secs   fish           external
   usr time    5.86 secs  568.00 micros    5.86 secs
   sys time    2.76 secs  345.00 micros    2.76 secs
```

## 7. Работа с образами и снимками

### Информация об образе

qemu-img info /tmp/disk\_\$USER.qcow2

```
root@AvansPC /h/sec# qemu-img info /tmp/disk_$USER.qcow2
image: /tmp/disk_root.qcow2
file format: qcow2
virtual size: 7 GiB (7516192768 bytes)
disk size: 138 MiB
cluster_size: 65536
backing file: /tmp/disk-base_root.qcow2
backing file format: qcow2
Snapshot list:
ID      TAG              VM_SIZE   DATE           VM_CL
OCK     ICOUNT
1       running_state     137 MiB   2025-05-29 17:33:28  0000:04:50.
958     --
Format specific information:
  compat: 1.1
  compression type: zlib
  lazy refcounts: false
  refcount bits: 16
  corrupt: false
  extended l2: false
Child node '/file':
  filename: /tmp/disk_root.qcow2
  protocol type: file
  file length: 138 MiB (144441344 bytes)
  disk size: 138 MiB
root@AvansPC /h/sec# |
```



Покажет:

- Виртуальный размер
- Реальный размер на диске
- Формат
- Список снимков

## Просмотр снимков

`qemu-img snapshot -l /tmp/disk_$USER.qcow2`

```
root@AvansPC /h/sec# qemu-img snapshot -l /tmp/disk_$USER.qcow2
Snapshot list:
ID          TAG          VM_SIZE      DATE          VM_CLOCK      IC
OUNT
1          running_state    137 MiB 2025-05-29 17:33:28  0000:04:50.958
--
root@AvansPC /h/sec#
```

## Восстановление из снимка

В мониторе QEMU:

`loadvm running_state`

```
(qemu) loadvm running_state
Unknown savevm section or instance 'slirp' 1. Make sure that your current VM
setup matches your saved VM setup, including any hotplugged devices
Error: Error -22 while loading VM state
(qemu) netdev_del net0
(qemu) netdev_add user,id=net0,hostfwd=tcp::2222-:22
(qemu) loadvm running_state
(qemu) |
```



## Важные концепции

Различия между эмуляцией и виртуализацией

### Эмуляция:

- Полная программная имитация оборудования
- Медленнее (10-100x замедление)
- Не требует аппаратной поддержки
- Может эмулировать другие архитектуры

### Виртуализация:

- Использует аппаратные расширения процессора
- Близкая к нативной производительность (5-15% overhead)
- Требует VT-x/AMD-V
- Только та же архитектура

Copy-on-Write в QCOW2

- Базовый образ остается неизменным
- Дочерний образ хранит только изменения
- При чтении: сначала дочерний, потом базовый
- При записи: создается копия блока в дочернем образе

Типы сетей в QEMU

1. **User mode (SLIRP):** NAT, проброс портов
2. **TAP:** мостовое соединение
3. **Socket:** соединение между VM
4. **VDE:** Virtual Distributed Ethernet

### 1. В чем разница между QEMU и KVM?

- QEMU - эмулятор/гипервизор, KVM - модуль ядра для аппаратной виртуализации

### 2. Почему дочерние образы экономят место?

- Используют Copy-on-Write, хранят только изменения

### 3. Что происходит при savevm/loadvm?

- Сохраняется полное состояние VM: память, регистры, устройства

#### 4. Зачем нужен проброс портов?

- В режиме NAT VM недоступна извне, проброс открывает доступ к службам

#### 5. Что такое монитор QEMU?

- Интерфейс управления VM во время работы

# Просмотр справки

```
man qemu-system-x86_64
```

```
man qemu-img
```

```
qemu-img --help
```

```
qemu-system-x86_64 -help
```

# Список поддерживаемых форматов

```
qemu-img --help | grep "Supported formats"
```

# Список поддерживаемых устройств

```
qemu-system-x86_64 -device help
```