

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 №3
«Виртуализация. Эмуляция. Гипервизоры.»

Практическая работа
по дисциплине «Современные технологии программирования»
студента 1 курса группы ПИ-б-о-231(2)
Аметов Кемран Ленверович
направления подготовки 09.03.04 «Программная инженерия»

Симферополь, 2024

Цель:

изучить основы работы гипервизоров, получить навыки работы с гипервизором KVM ядра Linux, а также подготовить базу для выполнения следующих лабораторных и практических занятий.

Ход выполнения задания.

Изучите возможности команды `qemu-img`:

1. Создайте образ виртуального жёсткого диска в папке `/tmp/` размером 1.5GB в формате `vmdk` с именем `disk_base_$USER.vmdk` (`USER` это переменная среды окружения в которой хранится логин текущего пользователя).

```
kemran@kemran:~$ cd /tmp
kemran@kemran:/tmp$ qemu-img create disk_base_$USER.vmdk 1.5G
Formatting 'disk_base_kemran.vmdk', fmt=raw size=1610612736
kemran@kemran:/tmp$
```

2. Измените формат образа на `qcow2`, изменив также расширение файла.

```
kemran@kemran:/tmp$ sudo qemu-img convert -f vmdk disk_base_kemran.vmdk -O qcow2
disk_base_kemran.qcow2
kemran@kemran:/tmp$
```

3. Увеличьте размер образа диска до 7Gb.

```
kemran@kemran:/tmp$ sudo qemu-img resize disk_base_kemran.qcow2 7G
Image resized.
kemran@kemran:/tmp$
```

4. С помощью `qemu-img` создайте целевой (дочерний) образ диска базирующийся на образе диска, созданном на предыдущем этапе. Образ в формате `qcow2` должен называться `disk_$USER.qcow2` и располагаться в директории `/tmp/`. В случае нехватки места в `/tmp/` переместите образ в домашнюю директорию.

```
qemu-img: /tmp/disk_kemran.qcow2: backing file specified without backing format
kemran@kemran:/tmp$ qemu-img create -f qcow2 -b /tmp/disk_base_$USER.qcow2 -F qcow2
/tmp/disk_$USER.qcow2
Formatting '/tmp/disk_kemran.qcow2', fmt=qcow2 cluster_size=65536 extended_l2=off c
ompression_type=zlib size=7516192768 backing_file=/tmp/disk_base_kemran.qcow2 backi
ng_fmt=qcow2 lazy_refcounts=off refcount_bits=16
kemran@kemran:/tmp$
```

2. Определите поддерживается ли гипервизор KVM на вашем оборудовании. Если KVM поддерживается, в дальнейшем используйте его при работе с ВМ.
3. Запустите виртуальную машину QEMU с необходимыми параметрами:
 1. Количество процессоров: `1`
 2. Оперативная память: `1.5G`
 3. Тип эмулируемой видеокарты: `std`
 4. Образ жёсткого диска образ: уже созданный образ (целевой) Вами на предыдущем этапе лабораторной работы
 5. Файл CD-ROM: [tel.iso](#)
 6. Сеть: *пользовательская*
 7. Проброс портов: порт хост-компьютера = `8080` на порт виртуальной машины = `80`
 8. Включите отображение меню выбора устройства для загрузки

9. Таймаут отображения меню: *10 секунд*

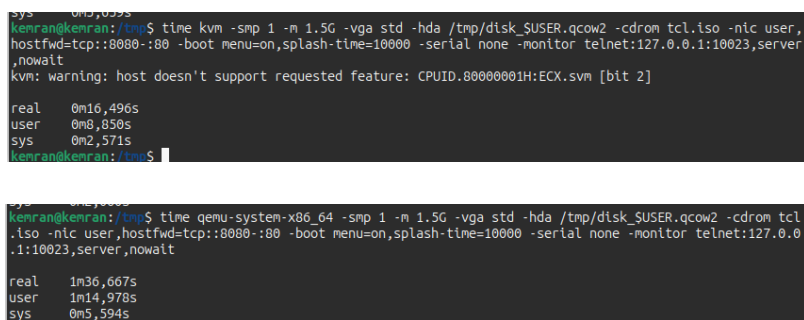
10. Дополнительные опции:

```
-serial none -monitor telnet:127.0.0.1:10023,server,nowait
```

4. Если на вашем оборудовании поддерживается гипервизор KVM, то:

- Произведите измерение времени полной загрузки системы в режиме эмуляции и в режиме виртуализации.
- Произведите измерение времени исполнения следующей команды в режиме эмуляции и в режиме виртуализации (измерить точное время исполнения можно с помощью команды `time`).

```
ssh-keygen -M generate -O bits=512 /tmp/1.txt
```



```
sys 0m2,029s
kenran@kenran:~$ time kvm -smp 1 -m 1.5G -vga std -hda /tmp/disk_$USER.qcow2 -cdrom tel.iso -nic user,
hostfwd=tcp::8080-:80 -boot menu=on,splash-time=10000 -serial none -monitor telnet:127.0.0.1:10023,server
,nowait
kvm: warning: host doesn't support requested feature: CPUID.80000001H:ECX.svm [bit 2]

real    0m16,496s
user    0m0,850s
sys     0m2,571s
kenran@kenran:~$

kenran@kenran:~$ time qemu-system-x86_64 -smp 1 -m 1.5G -vga std -hda /tmp/disk_$USER.qcow2 -cdrom tel
.iso -nic user,hostfwd=tcp::8080-:80 -boot menu=on,splash-time=10000 -serial none -monitor telnet:127.0.0
.1:10023,server,nowait

real    1m36,667s
user    1m14,978s
sys     0m5,594s
```

5. Взаимодействие с работающей ВМ через монитор.

1. Для этого ВМ должна работать. Загрузите гостевую ОС.
2. Подключитесь к монитору ВМ по протоколу `telnet` (отключиться от монитора можно сочетанием клавиш `Ctrl+]`, `Ctrl+d`).
3. Получите информацию о:
 - процессорах
 - регистрах процессоров
 - сети
 - блочных устройствах

```
kemran@kemran: /tmp
File Edit View Search Terminal Help
(qemu) info cpus
* CPU #0: thread_id=22821
(qemu) info registers

CPU#0
EAX=000058fc EBX=c099e280 ECX=00200002 EDX=f57c4cbc
ESI=f57d12c1 EDI=f57d12d9 EBP=c0995f30 ESP=c0995f2c
EIP=c07a2031 EFL=00200212 [---A---] CPL=0 II=0 A20=1 SMM=0 HLT=1
ES =007b 00000000 ffffffff 00c0f300 DPL=3 DS [-WA]
CS =0060 00000000 ffffffff 00c09b00 DPL=0 CS32 [-RA]
SS =0068 00000000 ffffffff 00c09b00 DPL=0 DS [-WA]
DS =007b 00000000 ffffffff 00c0f300 DPL=3 DS [-WA]
FS =00d8 34c8f000 ffffffff 00809300 DPL=0 DS16 [-WA]
GS =0000 00000000 ffffffff 00c00100
LDT=0000 00000000 ffffffff 00c00000
TR =0080 ff806000 0000407b 00008b00 DPL=0 TS532-busy
GDT= ff801000 000000ff
IDT= ff800000 000007ff
CR0=00050033 CR2=093a823c CR3=019b3000 CR4=000006d0
DR0=0000000000000000 DR1=0000000000000000 DR2=0000000000000000 DR3=0000000000000000
DR6=00000000ffff0fff DR7=0000000000000400
EFER=0000000000000000
FCW=037f FSW=0000 [ST=0] FTW=00 MXCSR=00001f80
FPR0=0000000000000000 0000 FPR1=0000000000000000 0000
FPR2=0000000000000000 0000 FPR3=0000000000000000 0000
FPR4=0000000000000000 0000 FPR5=0000000000000000 0000
FPR6=0000000000000000 0000 FPR7=0000000000000000 0000
XMM00=0000000000000000 0000000000000000 XMM01=0000000000000000 0000000000000000
XMM02=0000000000000000 0000000000000000 XMM03=0000000000000000 0000000000000000
XMM04=0000000000000000 0000000000000000 XMM05=0000000000000000 0000000000000000
XMM06=0000000000000000 0000000000000000 XMM07=0000000000000000 0000000000000000
(qemu) info network
e1000.0: index=0,type=nic,model=e1000,macaddr=52:54:00:12:34:56
\ #net078: index=0,type=user,net=10.0.2.0,restrict=off
(qemu) info block
ide0-hd0 (#block179): /tmp/disk_kemran.qcow2 (qcow2)
Attached to: /machine/unattached/device[23]
Cache mode: writeback
Backing file: /tmp/disk_base_kemran.qcow2 (chain depth: 1)
```

4. Удалите существующий проброс портов:
 - порт хост-компьютера = 8080 на порт виртуальной машины = 80.
5. Добавьте новый проброс портов к виртуальной машине:
 - порт хост-компьютера = 2222 на порт виртуальной машины = 22.
6. Выполните сохранение текущего состояния ВМ с тегом "running_state".
7. Перезагрузите виртуальную систему (аналог нажатия кнопки "Reset" на системном блоке).
8. Принудительно завершите работу ВМ (равноценно отключению питания для обычного компьютера).
9. Получите информацию об образах виртуальной машины, которые вы создавали и использовали во время работы ВМ. Какой объём они занимают в данный момент? Какие снимки состояния в них хранятся?
10. Восстановите работу ВМ из сохранённого снимка состояния. Обратите внимание, что работа ВМ мгновенно должна продолжиться в точности с момента сохранения.

```
(qemu) hostfwd_remove tcp::8080

(qemu) hostfwd_add ::2222::22
(qemu) savevm running_state
(qemu) system_reset
(qemu) system_powerdown
(qemu) loadvm running_state
(qemu)
```

Ответы на вопросы.

1. KVM (Kernel-based Virtual Machine) - это гипервизор, встроенный в ядро Linux. Он позволяет запускать несколько виртуальных машин на одном физическом сервере.

2. QEMU (Quick Emulator) - это программное обеспечение для эмуляции и виртуализации. Оно может эмулировать аппаратное обеспечение, такое как процессоры и устройства ввода-вывода, а также виртуализировать гостевые операционные системы.

3. KVM можно использовать, если ваш процессор поддерживает аппаратную виртуализацию (Intel VT-x или AMD-V), и если вы используете 64-битную версию Linux. Кроме того, ваша операционная система должна быть ядром Linux 2.6.20 или более поздней версии.

4. Монитор QEMU - это интерфейс командной строки, который позволяет управлять виртуальной машиной, запущенной в QEMU. Он может использоваться для управления процессором, памятью, устройствами ввода-вывода и другими аспектами виртуализации.

5. Некоторые основные команды монитора QEMU включают ``info`` (для получения информации о состоянии виртуальной машины), ``cont`` (для продолжения выполнения виртуальной машины), ``stop`` (для остановки виртуальной машины), ``savevm`` (для сохранения состояния виртуальной машины) и ``loadvm`` (для загрузки сохраненного состояния виртуальной машины).

6. Эмуляция - это процесс, при котором программное обеспечение эмулирует аппаратное обеспечение, такое как процессоры и устройства ввода-вывода. Это позволяет запускать программное обеспечение, разработанное для одной архитектуры, на аппаратуре с другой архитектурой.

7. Виртуализация - это процесс, при котором гипервизор позволяет нескольким виртуальным машинам работать на одном физическом сервере. Каждая виртуальная машина имеет свою собственную операционную систему и приложения, и они могут работать независимо друг от друга.

Вывод:

Изучил основы гипервизоров, получил навыки работы с KVM-гипервизором ядра Linux, а также подготовил основу для последующих лабораторных и практических занятий.