Документация пользователя эмулятора процессора QEMU

ПЕРЕВОД: Павел Марьянов <acid_jack@ukr.net>, март 2006

Содержание

---

1. Введение

1.1 Возможности

QEMU - это БЫСТРЫЙ! эмулятор процессора, использующий динамическую трансляцию для достижения хорошей скорости эмуляции.

QEMU имеет два режима работы:

•- Эмуляция всей системы. В этом режиме QEMU эмулирует полноценную систему (например ПК), включая процессор и различное периферийное оборудование. Его можно использовать для запуска различных операционных систем без необходимости перезагрузки ПК или для отладки кода системы.
•- Эмуляция пользовательского режима (только Linux-машины). В этом режиме QEMU может запускать процессы Linux, скомпилированные для одного процессора, на другом процессоре. Его можно использовать для запуска Wine - эмулятора API Windows (http://www.winehq.org), или для упрощённой кросс-компиляции и кросс-отладки.

QEMU может работать на системе-хозяине без драйвера, обеспечивая при этом удовлетворительную производительность.

При эмуляции целой системы поддерживается следующее оборудование:

•PC (процессор x86 или x86_64)
•PREP (процессор PowerPC)
•G3 BW PowerMac (процессор PowerPC)
•Mac99 PowerMac (процессор PowerPC, в процессе разработки)
•Sun4m (32-битный процессор Sparc)
•Sun4u (64-битный процессор Sparc processor, в процессе разработки)
•Malta board (32-битный процессор MIPS, в процессе разработки)

При эмуляции пользователя поддерживаются процессоры x86, PowerPC, ARM и Sparc32/64.

2. Установка

Если вы хотите самостоятельно скомпилировать QEMU, обратитесь к compilation.

2.1 Linux

Если для вашего дистрибутива есть пакет с уже откомпилированными бинарными файлами - вам нужно просто установить их. В противном случае см. compilation.

2.2 Windows

Экспериментальный бинарный инсталлятор можно загрузить с http://www.free.oszoo.org/download.php.

2.3 Mac OS X

Экспериментальный бинарный инсталлятор можно загрузить с http://www.freeoszoo.org/download.php.

3. Запуск QEMU в качестве эмулятора системы типа PC

3.1 Введение

Как эмулятор системы QEMU эмулирует следующее периферийное оборудование:

•- i440FX host PCI bridge и PIIX3 PCI to ISA bridge
•- PCI VGA-карту Cirrus CLGD 5446 или фиктивную VGA-карту с VESA-расширениями Bochs (аппаратный уровень, включая все нестандартные режимы).
•- PS/2-мышь и клавиатуру

- 2 PCI IDE-интерфейса с поддержкой жёстких и приводов CD-ROM
- дисковод
- сетевые PCI-адаптеры NE2000
- последовательные порты
- звуковую карту Soundblaster 16

QEMU использует PC BIOS проекта Bochs и Plex86/Bochs LGPL VGA BIOS.

3.2 Быстрый запуск

Загрузите и распакуйте образ linux (`linux.img'), а затем наберите:

```
qemu linux.img
```

Должен загрузиться Linux, предоставив вам строку приглашения.

3.3 Запуск

```
использование: qemu [опции] [образ_диска]
```

disk_image - это "сырой" образ жёсткого диска, соответствующий нулевому жёсткому IDE-диску.

Общие параметры:

`-fda файл'
`-fdb файл'

Здесь файл - образ дискеты для дисковода 0/1 соответственно (See Образы дисков.). Вы можете использовать дискету системы-хозяина, указав `/dev/fd0' в качестве файла.

`-hda файл'
`-hdb файл'
`-hdc файл'
`-hdd файл'

Здесь файл - образ жёсткого диска 0, 1, 2 или 3 соответственно (See Образы дисков.).

`-cdrom файл'

Здесь file - образ CD-ROM (вы не можете одновременно использовать `-hdc' и `-cdrom'). Вы можете использовать CD-ROM системы-хозяина, указав `/dev/cdrom' в качестве файла.

`-boot [a|c|d]'

Загрузка с дискеты (a), жёсткого диска (c) или CD-ROM (d). По умолчанию выполняется загрузка с жёсткого диска.

`-snapshot'

Запись во временный файл, а не в файл образа жёсткого диска. В этом случае образ "сырого" диска остаётся нетронутым. Однако вы можете принудительно включить запись, нажав клавиши C-a s (See Образы дисков.).

`-m megs'

Устанавливает объём виртуальной памяти в megs мегабайт. По умолчанию он составляет 128 МБ.

`-nographic'

Обычно QEMU использует SDL для вывода VGA-данных. Используя эту опцию, вы можете вообще отключить графический вывод, сделав, таким образом, QEMU простым консольным приложением. Эмулируемый последовательный порт перенаправляется в консоль. Зато вы так вы можете использовать QEMU для отладки ядра Linux, имея в своём распоряжении последовательную консоль.

`-k язык'

Использование раскладки клавиатуры для
данного языка (например ru для русского). Эта опция нужна только в тех случаях, когда трудно получить "сырые" коды клавиш (напр., на Mac'ах или в некоторых серверах X11). Вам не нужно использовать её на машинах PC/Linux или PC/Windows.

Доступны следующие раскладки:

```
ar  de-ch  es  fo     fr-ca  hu  ja  mk     no  pt-br  sv
da  en-gb  et  fr     fr-ch  is  lt  nl     pl  ru     th
de  en-us  fi  fr-be  hr     it  lv  nl-be  pt  sl     tr
```

По умолчанию используется раскладка en-us.

`-enable-audio'

По умолчанию эмуляция SB16 отключена, поскольку она может вызвать некоторые проблемы у Windows. Вы можете включить её вручную с помощью этой опции.

`-localtime'

Настраивает часы реального времени согласно локальному времени (по умолчанию в UTC). Эта опция нужна для обеспечения правильного времени в MS-DOS или Windows.

`-full-screen'

Запуск в полноэкранном режиме.

`-pidfile файл'

Сохраняет идентификатор процесса QEMU в файл. Полезно, если вы запускаете QEMU из скрипта.

`-win2k-hack'

Используйте эту опцию при установке Windows 2000, чтобы предотвратить ошибку переполнения диска. После установки Windows 2000, вам больше не нужно использовать эту опцию (она замедляет передачу данных по IDE).

Сетевые опции:

`-n скрипт'

Задаёт скрипт инициализации сети TUN/TAP [default=/etc/qemu-ifup]. Этот скрипт запускается для настройки сетевого интерфейса системы-хозяина (обычно tun0), соответствующего виртуальной сетевой карте NE2000.

`-nics n'

Эмуляция n-ого числа сетевых карт (по умолчанию 1).

`-macaddr адрес'

Задаёт mac-адрес первого интерфейса (в hex-формате типа aa:bb:cc:dd:ee:ff). Mac-адрес увеличивается для каждого нового сетевого интерфейса.

`-tun-fd fd'

Подразумевает, что fd общается с сетевым интерфейсом tap/tun системы-хозяина и использует его. Примеры использования опции смотрите на сайте http://bellard.org/qemu/tetrinet.html.

`-user-net'

Использование сетевого стека в режиме пользователя. Используется по умолчанию, если не найден скрипт инициализации сети tun/tap.

`-tftp префикс'

При использовании сетевого стека в режиме пользователя активирует встроенный сервер TFTP. Все файлы, начинающиеся с префикса, могут быть загружены с системы-хозяина в гостевую систему с помощью клиента TFTP. Клиент TFTP в гостевой системе должен быть настроен на двоичный режим передачи данных (используйте команду bin для TFTP-клиента для Unix). IP-адресом системы-хозяина обычно выступает 10.0.2.2.

`-smb каталог'

При использовании сетевого стека в режиме пользователя активирует встроенный сервер SMB, чтобы Windows-системы могли иметь прозрачный доступ к файлам системы-хозяина, находящимся в указанном `каталоге'.

В гостевой ОС Windows нужно добавить строку:

```
10.0.2.4 smbserver
```

в файл `C:\WINDOWS\LMHOSTS' (для Windows 9x/Me) или `C:\WINNT\SYSTEM32\DRIVERS\ETC\LMHOSTS' (Windows NT/2000).

Тогда доступ к `каталогу' можно получить через `\\smbserver\qemu'.

Учтите, что на системе-хозяине должен быть установлен сервер SAMBA как `/usr/sbin/smbd'. QEMU был успешно протестирован с smbd версии 2.2.7a из Red Hat 9 и версии 3.0.10-1.fc3 из Fedora Core 3.

`-redir [tcp|udp]:порт_хозяина:[гость_хозяина]:порт_гостя'

При использовании сетевого стека в режиме пользователя перенаправляет TCP- или UDP-подключения, входящие в порт_хозяина, на гостя_хозяина в порт_гостя. Если гость_хозяина не указан, используется 10.0.2.15 (адрес, выдаваемый по умолчанию встроенным сервером DHCP).

Например, чтобы перенаправить подключение X11 хозяина с экрана 1 на экран 0 гостя, используйте следующие команды:

```
# на хозяине
qemu -redir tcp:6001::6000 [...]
# теперь xterm хозяина должен открыться на гостевом сервере X11
```

```
xterm -display :1
```

Чтобы перенаправить telnet-подключение к порту 5555 хозяина на telnet'овский порт гостя, используйте следующие команды:

```
# на хозяине
qemu -redir tcp:5555::23 [...]
telnet localhost 5555
```

Далее, если вы запустите telnet localhost 5555, вы подключитесь к telnet-серверу гостя.

`-dummy-net'

Использование фиктивного сетевого стека: сетевыми картами не будет получено ни одного пакета.

Загрузка Linux. С помощью этих опций вы можете использовать заданное ядро Linux, не устанавливая его на образ диска. Это может быть полезным для раннего тестирования различных ядер.

`-kernel bzImage'

Использование bzImage в качестве образа ядра.

`-append cmdline'

Использование cmdline в качестве командной строки ядра.

`-initrd файл'

Использование файл в качестве исходного ram-диска.

Отладочные/экспертные опции:

`-serial dev'

Перенаправление виртуального последовательного порта в устройство dev системы-хозяина. Доступные устройства:

vc

виртуальная консоль

pty

[только для Linux] Pseudo TTY (новый PTY выделяется автоматически)

null

устройство void

stdio

[только для Unix] стандартный ввод/вывод

Стандартным устройством в графическом режиме является vc, а в неграфическом - stdio.

Эту опцию можно использовать многократно для эмуляции до четырёх последовательных портов.

`-monitor dev'

Перенаправление монитора в устройство dev системы-хозяина (те же самые устройства, что и для последовательных портов). Стандартным устройством в графическом режиме является vc, а в неграфическом - stdio.

`-s'

Ожидание подключения gdb к порту 1234 (See Использование GDB.).

`-p порт'

Изменение порта подключения gdb.

`-S'

Не запускать процессор при запуске (вы должны ввести 'c' в мониторе).

`-d'

Выводить журнал событий в /tmp/qemu.log.

`-hdachs c,h,s,[,t]'

Принудительно использовать физическую геометрию жёсткого диска 0 (1 <= c <= 16383, 1 <= h <= 16, 1 <= s <= 63) и опционально устанавливать принудительный режим передачи данных BIOS (t=none, lba или auto). Обычно QEMU может сам определить все эти параметры. Эта опция полезна для образов дисков старого MS-DOS.

`-isa'

Эмуляция системы только с ISA (по умолчанию эмулируется система с PCI).

`-std-vga'

Эмуляция стандартной VGA-карты с VBE-расширениями Bochs (по умолчанию эмулируется PCI VGA Cirrus Logic GD5446)

`-loadvm файл'

Запуск из сохранённого ранее состояния (loadvm в мониторе)

3.4 Клавиши

Во время графической эмуляции вы можете использовать следующие клавиши:

Ctrl-Alt-f

Переключение в полноэкранный режим.

Ctrl-Alt-n

Переключение в виртуальную. консоль 'n'. Стандартными консолями могут быть:

1

дисплей системы назначения

2

монитор

3

последовательный порт

Ctrl-Alt

Переключение захвата клавиатуры и мыши.

В виртуальных консолях вы можете использовать сочетания клавиш Ctrl-Up, Ctrl-Down, Ctrl-PageUp и Ctrl-PageDown для перемещения по журналу сообщений.

Во время эмуляции, если вы используете опцию `-nographic', нажмите используйте Ctrl-a h, чтобы можно было вызвать в терминале следующие команды:

Ctrl-a h

Вывод этой справки.

Ctrl-a x

Выход из эмулятора.

Ctrl-a s

Сохранение данных диска в файл (если используется -snapshot).

Ctrl-a b

Отправка break (magic sysrq в Linux).

Ctrl-a c

Переключение между консолью и монитором.

Ctrl-a Ctrl-a

Отправка комбинации клавиш Ctrl-a.

3.5 Монитор QEMU

Монитор QEMU используется для выполнения сложных команд в эмуляторе QEMU. Вы можете использовать его для:

- •- извлечения или вставки образов съёмных накопителей (таких как CD-ROM или дисководов);
- •- "замораживать"/"размораживать" виртуальную машину (ВМ) и сохранять или восстанавливать её состояние из файла на диске;
- •- изучать состояние ВМ без необходимости использования внешнего отладчика.

3.5.1 Команды

Доступны следующие команды:

`help или ? [cmd]'

Вывод справки по всем командам или только по команде cmd.

`commit'

Запись изменений в образ диска (если используется -snapshot).

`info суб-команда'

показывает различную информацию о состоянии системы:

`info network'

показывает состояние сети

`info block'

показывает блочные устройства

`info registers'

показывает регистры процессора

`info history'

показывает историю набранных команд

`q или quit'

Выход из эмулятора.

`eject [-f] устройство'

Извлечение съёмного носителя (используйте для принудительного извлечения -f).

`change устройство файла'

Смена съёмного носителя.

`screendump файла'

Сохранение содержимого экрана в виде PPM-изображения в файл.

`log объект1[,...]'

Включение журналирования указанных объектов в файл `/tmp/qemu.log'.

`savevm файл'

Сохранение состояния виртуальной машины в файл.

`loadvm файл'

Восстановление состояния виртуальной машины из файла.

`stop'

Остановка эмуляции.

`c или cont'

Возобновление эмуляции.

`gdbserver [порт]'

Запуск сеанса gdbserver (по умолчанию порт=1234).

`x/fmt адрес'

Дамп виртуальной памяти, начиная с указанного адреса.

`xp /fmt адрес'

Дамп физической памяти, начиная с указанного адреса.

fmt - это формат, сообщающий команде, как нужно форматировать данные. Его синтаксис: `/{количество}{формат}{размер}'

количество

это число объектов для дампа.

формат

может быть x (шестнадцатиричный), d (десятичный со знаком), u (десятичный без знака), o (восьмеричный), c (символьный) или i (инструкции asm).

размер

может быть b (8 бит), h (16 бит), w (32 бит) или g (64 бит). На платформе x86 размеры h или w могут быть указаны с форматом i для выбора размера инструкций для 16- или 32-битного кода соответственно.

Примеры:

- Дамп 10 инструкций относительно текущего указателя:
  ```
  (qemu) x/10i $eip
  ```

```
0x90107063:  ret
0x90107064:  sti
0x90107065:  lea    0x0(%esi,1),%esi
0x90107069:  lea    0x0(%edi,1),%edi
0x90107070:  ret
0x90107071:  jmp    0x90107080
0x90107073:  nop
0x90107074:  nop
0x90107075:  nop
0x90107076:  nop
```

- Дамп 80-ти 16-битных значений относительно начала видеопамяти:

```
(qemu) xp/80hx 0xb8000
0x000b8000: 0x0b50 0x0b6c 0x0b65 0x0b78 0x0b38 0x0b36 0x0b2f 0x0b42
0x000b8010: 0x0b6f 0x0b63 0x0b68 0x0b73 0x0b20 0x0b56 0x0b47 0x0b41
0x000b8020: 0x0b42 0x0b69 0x0b6f 0x0b73 0x0b20 0x0b63 0x0b75 0x0b72
0x000b8030: 0x0b72 0x0b65 0x0b6e 0x0b74 0x0b2d 0x0b63 0x0b76 0x0b73
0x000b8040: 0x0b20 0x0b30 0x0b35 0x0b20 0x0b4e 0x0b6f 0x0b76 0x0b20
0x000b8050: 0x0b32 0x0b30 0x0b30 0x0b33 0x0720 0x0720 0x0720 0x0720
0x000b8060: 0x0720 0x0720 0x0720 0x0720 0x0720 0x0720 0x0720 0x0720
0x000b8070: 0x0720 0x0720 0x0720 0x0720 0x0720 0x0720 0x0720 0x0720
0x000b8080: 0x0720 0x0720 0x0720 0x0720 0x0720 0x0720 0x0720 0x0720
0x000b8090: 0x0720 0x0720 0x0720 0x0720 0x0720 0x0720 0x0720 0x0720
```

`p или print/fmt выражение'

Вывод значения выражения. Используется только часть format из fmt.

`sendkey клавиши'

Отправляет клавиши в эмулятор. Используйте - для одновременного нажатия нескольких клавиш. Пример:

```
sendkey ctrl-alt-f1
```

Эта команда полезна для отправки нажатий клавиш, которые ваш графический интерфейс перехватывает на низком уровне, наподобие ctrl-alt-f1 в X Window.

`system_reset'

Сброс системы.

3.5.2 Целочисленные выражения

Монитор понимает целочисленные выражения, которые можно использовать вместо любых целочисленных аргументов. Вы можете использовать названия регистров для получения значений определённых регистров процессора, вставляя перед ними знак $.

3.6 Образы дисков

QEMU, начиная с версии 0.6.1, поддерживает несколько форматов образов дисков, включая образы растущего объёма (их размер увеличивается по мере записи непустых секторов), а также сжатые и зашифрованные образы дисков.

3.6.1 Быстрое создание образа диска

Вы можете создать образ диска с помощью команды:

```
qemu-img create образ.img размер
```

где образ.img - это имя файла образа диска, а размер - его размер в килобайтах. Вы можете добавить окончание M, чтобы указать размер в мегабайтах, и G для размера в гигабайтах.

Дополнительную информацию See Запуск qemu-img.

3.6.2 Режим snapshot

Если вы используете опцию `-snapshot', все образы дисков используются только для чтения. При записи секторов они записываются во временный файл, созданный в `/tmp'. Однако вы можете включить принудительную запись в "сырой" образ диска с помощью команды commit в мониторе (или клавиш C-a s в последовательной консоли).

3.6.3 Запуск qemu-img

```
usage: qemu-img команда [опции команды]
```

Поддерживаются следующие команды:

`create [-e] [-b базовый_образ] [-f формат] файл [размер]'
`commit [-f формат] файл'
`convert [-c] [-e] [-f формат] файл [-O выходной_формат] выходой_файл'
`info [-f формат] файл'

Параметры команд:

файл

> имя файла с образом диска

базовый_образ

> образ диска, доступный только для чтения, который используется в качестве базы для образа т.н. copy-on-write; в этом образе сохраняются только изменённые данные.

формат

это формат образа диска. В большинстве случаев он определяется автоматически. Поддерживаются следующие форматы:

raw

"Сырой" формат (по умолчанию). Преимуществом этого формата является то, что он прост и легко экспортируется в другие эмуляторы. Если ваша файловая система поддерживает дыры (holes) (например, в ext2 или ext3 в Linux), тогда место на диске будут занимать только записанные секторы. Используйте qemu-img info, чтобы узнать реальный размер, занимаемый образом, или ls -ls в Unix/Linux (или du -h - прим. переводчика).

qcow

Формат QEMU - наиболее универсальный формат. Он позволяет получить образы меньшего размера (полезно в том случае, если вы ваша файловай система не поддерживает "дыры", например, в Windows), опциональное шифрование AES и сжатие zlib.

cow

Формат User Mode Linux Copy On Write. Используется в QEMU для образов с увеличивающимся объёмом. Поддерживается только для совместимости с предыдущими версиями. Не работает в win32.

vmdk

Формат, совместимый с VMware 3 и 4.

cloop

Формат Linux Compressed Loop. Полезен только для повторного использования сжатых образов CD-ROM, использующихся, например, на LiveCD с Knoppix.

размер

размер образа диска в килобайтах. Возможно использование необязательных окончаний: M (мегабайт) и G (гигабайт).

выходной_файл

имя файла создаваемого образа диска.

выходной_формат

формат создаваемого диска.

-c

означает, что создаваемый образ должен быть сжат (только для формата qcow).

-e

означает, что создаваемый образ должен быть зашифрован (только для формата qcow).

Описание команды:

`create [-e] [-b базовый_образ] [-f формат] файл [размер]'

Создаёт образ нового диска в виде
указанного файла указанного размера и формата.

Если указан базовый_образ, тогда в образ будут записываться только изменения, относительно базового_образа. Базовый_образ не будет изменяться, только если вы не воспользуетесь в мониторе командой commit.

`commit [-f формат] файл'

Вносит в базовый образ изменения, записанные в файл.

`convert [-c] [-e] [-f формат] файл [-O выходной_формат] выходной_файл'

Преобразовывает образ диска файл в образ выходной_файл с использованием указанного формата. Опционально он может быть зашифрован (опция -e) или сжат (опция -c).

Шифрование и сжатие поддерживаются только в формате qcow. Сжатие работает в режиме "только для чтения", т.е. если перезаписывается сжатый сектор, то данные в него пишутся в несжатом виде.

При шифровании используется довольно безопасный алгоритм AES (со 128-битными ключами). Для обеспечения максимальной защиты используйте длинные пароли (порядка 16 символов).

Преобразование образов также полезно для уменьшения их размеров при использовании формата увеличивающегося размера, такого как qcow или cow: при этом выявляются пустые секторы и в полученном образе они отсутвуют.

`info [-f формат] файл'

Выводит информацию об образе диска filename. Используйте её для определения пространства, зарезервированного на жёстком диске под этот образ, поскольку оно может отличаться от показываемого размера.

3.7 Эмуляция сети

QEMU может эмулировать до 6-и сетевых карт (NE2000-типа). Каждая из карт может быть подключена к определённому сетевому интерфейсу системы-хозяина.

3.7.1 Использование сетевого интерфейса tun/tap

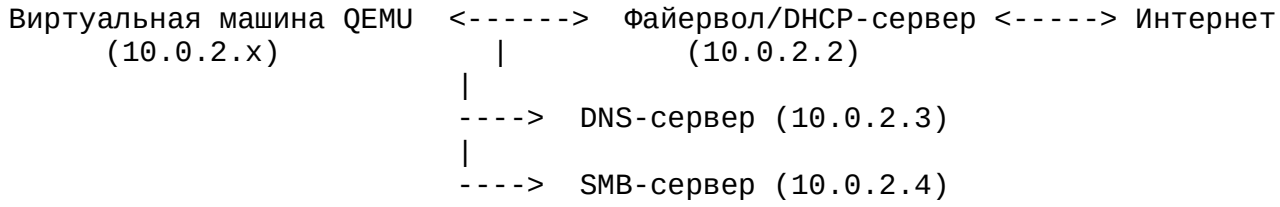
Это стандартный способ эмуляции сети. QEMU добавляет в систему-хозяина виртуальное сетевое устройство (под названием tun0), а затем вы можете настроить его так, как если бы это была настоящая ethernet-карта.

В качестве примера вы можете загрузить архив `linux-test-xxx.tar.gz', скопировать скрипт `qemu-ifup' в `/etc' и настроить соответствующим образом sudo, чтобы команда ifconfig, находящаяся в `qemu-ifup', могла быть выполнена с правами root'a. Сначала вы должны убедиться в том, что ядро системы-хозяина поддерживает сетевые интерфейсы TUN/TAP: должно существовать устройство `/dev/net/tun'.

В direct_linux_boot есть пример использования такой сети в дистрибутиве Linux.

3.7.2 Использование сетевого стека в режиме пользователя

При использовании опции `-user-net', или если у вас нет скрипта инициализации tun/tap, QEMU использует полностью пользовательский сетевой стек (вам не нужны права root'a для использования виртуальной сети). Конфигурация виртуальной сети представляет собой следующее:



ВМ QEMU ведёт себя так, как если бы она находилась за файерволом, блокирующим все входящие подключения. Вы можете использовать клиент DHCP для автоматической настройки сети в ВМ QEMU.

Чтобы проверить работу сети в режиме пользователя, пропингуйте адрес 10.0.2.2 и проверьте, получили ли вы адрес из диапазона 10.0.2.x от виртуального DHCP-сервера QEMU.

Обратите внимание, что ping не работает по отношению к Интернет-адресам, поскольку для этого требуются привилегии root'a. Это означает, что вы можете пропинговать только локальный маршрутизатор (10.0.2.2).

При использовании встроенного сервера TFTP, маршрутизатор также выступает в роли TFTP-сервера.

При использовании опции `-redir', TCP- или UDP-подключения к системе-хозяину могут быть перенаправлены в гостевую систему. Например, это позволяет перенаправлять X11-, telnet- или SSH-подключения.

3.8 Непосредственная загрузка Linux

В этом разделе рассказано о том, как запустить ядро Linux внутри QEMU без необходимости создания целого загрузочного диска. Это очень полезно для быстрого тестирования ядра Linux. Также разъяснена настройка сети в QEMU.

1. Загрузите архив `linux-test-xxx.tar.gz', содержащий ядро Linux и образ диска.
2. Необязательно: если вам нужна поддержка сети (например для запуска X11-приложений), вы можете скопировать скрипт `qemu-ifup' в `/etc' и настроить соответствующим образом sudo, чтобы команда ifconfig, находящаяся в `qemu-ifup', могла быть выполнена с правами root'a. Сначала вы должны убедиться в том, что ядро системы-хозяина поддерживает сетевые интерфейсы TUN/TAP: должно существовать устройство `/dev/net/tun'.

   Если сеть включена, между ядром системы-хозяина и эмулируемым ядром устанавливается виртуальное сетевое соединение. Из ядра хозяина эмулируемое ядро доступно по IP-адресу 172.20.0.2, а ядро хозяина доступно в эмулируемом по IP-адресу 172.20.0.1.

3. Запустите qemu.sh. Вы должны будете увидеть следующее:

```
> ./qemu.sh
Connected to host network interface: tun0
Linux version 2.4.21 (bellard@voyager.localdomain) (gcc version 3.2.2 2003022
BIOS-provided physical RAM map:
 BIOS-e801: 0000000000000000 - 000000000009f000 (usable)
 BIOS-e801: 0000000000100000 - 0000000002000000 (usable)
32MB LOWMEM available.
On node 0 totalpages: 8192
zone(0): 4096 pages.
zone(1): 4096 pages.
zone(2): 0 pages.
Kernel command line: root=/dev/hda sb=0x220,5,1,5 ide2=noprobe ide3=noprobe id
ide_setup: ide2=noprobe
ide_setup: ide3=noprobe
```

```
ide_setup: ide4=noprobe
ide_setup: ide5=noprobe
Initializing CPU#0
Detected 2399.621 MHz processor.
Console: colour EGA 80x25
Calibrating delay loop... 4744.80 BogoMIPS
Memory: 28872k/32768k available (1210k kernel code, 3508k reserved, 266k data,
Dentry cache hash table entries: 4096 (order: 3, 32768 bytes)
Inode cache hash table entries: 2048 (order: 2, 16384 bytes)
Mount cache hash table entries: 512 (order: 0, 4096 bytes)
Buffer-cache hash table entries: 1024 (order: 0, 4096 bytes)
Page-cache hash table entries: 8192 (order: 3, 32768 bytes)
CPU: Intel Pentium Pro stepping 03
Checking 'hlt' instruction... OK.
POSIX conformance testing by UNIFIX
Linux NET4.0 for Linux 2.4
Based upon Swansea University Computer Society NET3.039
Initializing RT netlink socket
apm: BIOS not found.
Starting kswapd
Journalled Block Device driver loaded
Detected PS/2 Mouse Port.
pty: 256 Unix98 ptys configured
Serial driver version 5.05c (2001-07-08) with no serial options enabled
ttyS00 at 0x03f8 (irq = 4) is a 16450
ne.c:v1.10 9/23/94 Donald Becker (becker@scyld.com)
Last modified Nov 1, 2000 by Paul Gortmaker
NE*000 ethercard probe at 0x300: 52 54 00 12 34 56
eth0: NE2000 found at 0x300, using IRQ 9.
RAMDISK driver initialized: 16 RAM disks of 4096K size 1024 blocksize
Uniform Multi-Platform E-IDE driver Revision: 7.00beta4-2.4
ide: Assuming 50MHz system bus speed for PIO modes; override with idebus=xx
hda: QEMU HARDDISK, ATA DISK drive
ide0 at 0x1f0-0x1f7,0x3f6 on irq 14
hda: attached ide-disk driver.
hda: 20480 sectors (10 MB) w/256KiB Cache, CHS=20/16/63
Partition check:
 hda:
Soundblaster audio driver Copyright (C) by Hannu Savolainen 1993-1996
NET4: Linux TCP/IP 1.0 for NET4.0
IP Protocols: ICMP, UDP, TCP, IGMP
IP: routing cache hash table of 512 buckets, 4Kbytes
TCP: Hash tables configured (established 2048 bind 4096)
NET4: Unix domain sockets 1.0/SMP for Linux NET4.0.
EXT2-fs warning: mounting unchecked fs, running e2fsck is recommended
VFS: Mounted root (ext2 filesystem).
Freeing unused kernel memory: 64k freed

Linux version 2.4.21 (bellard@voyager.localdomain) (gcc version 3.2.2 2003022

QEMU Linux test distribution (based on Redhat 9)

Type 'exit' to halt the system
```

```
sh-2.05b#
```

4. Затем вы можете поиграться с ядром, используя виртуальную последовательную консоль. Например, вы можете выполнить команду ls. Введите Ctrl-a h, чтобы получить справку о клавишах, которые вы можете использовать в виртуальной последовательной консоли. В частности используйте Ctrl-a x для выхода из QEMU и Ctrl-a b в качестве "магической" клавиши системного вызова (Magic SysRq).
5. Если сеть включена, запустите в эмуляторе скрипт `/etc/linuxrc' (не забудьте точку в начале):

```
. /etc/linuxrc
```

Затем разрешите X11-подключения к своему ПК из эмулируемого Linux'a:

```
xhost +172.20.0.2
```

Теперь вы можете запустить `xterm' или `xlogo', чтобы убедится в том, что перед вами действительно виртуальная система Linux!

ЗАМЕЧАНИЯ:

1. В состав архива также входит ядро версии 2.5.74. Чтобы попробовать его, просто замените им bzImage в qemu.sh.
2. Чтобы корректно завершить работу с qemu, вы можете выполнить в нём команду shutdown. qemu автоматически закроется по завершении работы Linux.
3. Вы можете немного увеличить скорость загрузки, отключив опрос отсутствующих интерфейсов IDE. Для этого добавьте следующие параметры в командную строку ядра:

```
ide1=noprobe ide2=noprobe ide3=noprobe ide4=noprobe ide5=noprobe
```

4. Этот образ диска является изменённой версией образа, созданного Кевином Лоутоном (Kevin Lawton) для проекта plex86 (www.plex86.org).

3.9 Использование GDB

QEMU поддерживает упрощённую работу с gdb. Таким образом вы можете нажать 'Ctrl-C' во время работы виртуальной машины и просмотреть её состояние.

Чтобы можно было использовать gdb, запустите qemu с опцией '-s'. При этом он будет ожидать подключения gdb:

```
> qemu -s -kernel arch/i386/boot/bzImage -hda root-2.4.20.img -append "root=/dev/hd
```

```
Connected to host network interface: tun0
Waiting gdb connection on port 1234
```

Затем запустите gdb, передав ему в качестве параметра исполняемый файл 'vmlinux':

```
> gdb vmlinux
```

В gdb подключитесь к QEMU:

```
(gdb) target remote localhost:1234
```

Затем вы можете использовать gdb обычным образом. Например, введите 'c', чтобы запустить ядро:

```
(gdb) c
```

Вот несколько полезных подсказок по использованию gdb при работе с системным кодом:

1. Используйте info reg для вывода всех регистров процессора.
2. Используйте x/10i $eip для вывода кода в разряде PC.
3. Используйте set architecture i8086 дампа 16-битного кода. Затем воспользуйтесь x/10i $cs*16+*eip для получения дампа кода в разряде PC.

3.10 Дополнительная информация по эмулируемой ОС

3.10.1 Linux

Чтобы иметь доступ к графическим режимам SVGA в X11, используйте драйвер vesa или cirrus. Для получения оптимальной производительности используйте 16-битную глубину цвета и в гостевой системе, и в системе-хозяине.

При использовании в гостевой системе ядра Linux версии 2.6 вам следует добавить параметр clock=pit в командную строку ядра, потому что ядра Linux серии 2.6 по умолчанию выполняют очень ограниченную проверку часов реального времени, которую QEMU не может в точности сэмулировать.

Так же при использовании в гостевой системе ядра Linux версии 2.6 убедитесь, что не задействован патч 4G/4G, потому что с этим патчем QEMU работает медленнее. Модуль ускорения QEMU (QEMU Accelerator Module) в этом случае также работает гораздо медленнее. Этот патч использовался по умолчанию в ранних версиях ядер Linux в Fedora Core 3 (< 2.6.9-1.724_FC3). В более новых версиях его нет.

3.10.2 Windows

Если ваша система довольно медленная, лучше эмулировать Windows 95, поскольку она работает быстрее всех. Также неплохим вариантом будет Windows 2000.

3.10.2.1 Поддержка графических режимов SVGA

QEMU эмулирует видеокарту Cirrus Logic GD5446. Все версии Windows, начиная с Windows 95, должны нормально распознавать и работать с этой видеокартой. Для получения оптимальной производительности используйте 16-битную глубину цвета и в гостевой системе, и в системе-хозяине.

3.10.2.2 Снижение загрузки процессора

Windows 9x некорректно использует инструкцию CPU HLT. В результате процессор системы-хозяина продолжает использоваться даже при простое. Для устранения этой проблемы вы можете установить специальную утилиту http://www.user.cityline.ru/~maxamn/amnhltm.zip. Обратите внимание, что для NT, 2000 и XP такая утилита не нужна.

3.10.2.3 Проблема с переполнением диска в Windows 2000

В Windows 2000 есть ошибка, приводящая к проблеме переполнения диска во время установки. Поэтому при её установке используйте в QEMU опцию `-win2k-hack', чтобы устранить эту проблему. После установки Windows 2000 вам больше не нужно использовать эту опцию (она замедляет передачу данных по IDE).

3.10.2.4 Завершение работы Windows 2000

Windows 2000 не может автоматически завершить свою работу в QEMU в отличие от Windows 98. Это связано с тем, что Windows 2000 по умолчанию не использует драйвер автоматического управления питанием (APM), предоставляемого BIOS'ом.

Чтобы исправить это, выполните следующее (спасибо Струану Барлетту (Struan Bartlett)): откройте Control Panel => Add/Remove Hardware & Next => Add/Troubleshoot a device => Add a new device & Next => No, select the hardware from a list & Next => NT Apm/Legacy Support & Next => Next (снова) несколько раз. Теперь драйвер установлен и Windows 2000 может корректно сообщать QEMU о завершении своей работы в нужный момент.

3.10.2.5 Использование общих каталогов для Unix и Windows

Смотрите sec_invocation об использовании опции `-smb'.

3.10.2.6 Проблемы с безопасностью в Windows XP

Некоторые сборки Windows XP нормально устанавливаются, однако во время загрузки выдают сообщение об ошибке безопасности:

```
A problem is preventing Windows from accurately checking the
license for this computer. Error code: 0x800703e6.
```

Единственным способом обойти эту проблемы является загрузка с безопасном режиме с отключенной поддержкой сети.

В следующих релизах QEMU эта проблема скорее всего будет устранена.

3.10.3 MS-DOS и FreeDOS

3.10.3.1 Снижение загрузки процессора

DOS некорректно использует инструкцию CPU HLT. В результате процессор системы-хозяина продолжает использоваться даже при простое. Для устранения этой проблемы вы можете установить специальную утилиту http://www.vmware.com/software/dosidle210.zip.

4. Запуск QEMU в качестве эмулятора системы типа PowerPC

Используйте исполняемый файл `qemu-system-ppc', чтобы эмулировать целую систему типа PREP или PowerMac PowerPC.

QEMU эмулирует следующее периферийное оборудование PowerMac:

- UniNorth PCI Bridge
- PCI VGA-совместимую карту с VESA-расширениями Bochs
- 2 PCI IDE-интерфейса с поддержкой жёстких и приводов CD-ROM
- сетевые PCI-адаптеры NE2000
- энергонезависимую оперативную память
- VIA-CUDA с клавиатурой и мышью ADB.

QEMU эмулирует следующее периферийное оборудование PREP:

- PCI Bridge
- PCI VGA-совместимую карту с VESA-расширениями Bochs
- 2 PCI IDE-интерфейса с поддержкой жёстких и приводов CD-ROM
- дисковод
- сетевые PCI-адаптеры NE2000
- последовательные порты
- энергонезависимую оперативную память PREP
- PC-совместимую клавиатуру и мышь

В QEMU используется Open Hack'Ware Open Firmware Compatible BIOS, доступный на сайте http://site.voila.fr/jmayer/OpenHackWare/index.htm.

Вы можете прочитать главу об эмуляции системы типа PC, чтобы получить большее представление об использовании QEMU.

Следующие опции касаются только эмуляции PowerPC:

`-prep'

Эмуляция системы PREP (по умолчанию это PowerMAC)

`-g ШхВ[хГЛУБИНА]'

Исходный графический режим VGA. По умолчанию используется 800x600x15.

Дополнительная информация доступна по адресу http://jocelyn.mayer.free.fr/qemu-ppc/.

5. Запуск QEMU в качестве эмулятора системы типа Sparc32

Используйте исполняемый файл `qemu-system-sparc', чтобы эмулировать JavaStation (архитектура sun4m). Достигнута почти полная эмуляция.

QEMU эмулирует следующее периферийное оборудование sun4m:

- IOMMU
- видеобуфер TCX
- сетевую карту Lance (Am7990)
- энергонезависимую оперативную память M48T08
- Slave I/O: таймеры, контроллеры прерываний, последовательные порты Zilog, клавиатуру и логический power/reset
- ESP SCSI-контроллер с поддержкой жёстких и приводов CD-ROM
- дисковод

В этой архитектуре количество оборудования является фиксированным.

В QEMU используется Proll - замена PROM, доступная на сайте http://people.redhat.com/zaitcev/linux/. Необходимые патчи для QEMU включены в архив с исходными текстами.

Образец ядра Linux серии 2.6 и образ ram-диска доступны на веб-сайте QEMU. Обратите внимание, что в настоящий момент не работают ядра Linux серии 2.4, NetBSD и OpenBSD.

Следующие опции касаются только эмуляции Sparc:

`-g ШхВ'

Исходный графический режим TCX. По умолчанию используется 1024x768.

6. Запуск QEMU в качестве эмулятора системы типа Sparc64

Используйте исполняемый файл `qemu-system-sparc64', чтобы эмулировать vfibye Sun4u. Пока что эмулятор ни для чего другого не подходит.

QEMU эмулирует следующее периферийное оборудование sun4u:

- •- UltraSparc IIi APB PCI Bridge
- •- PCI VGA-совместимую карту с VESA-расширениями Bochs
- •- энергонезависимую оперативную память M48T59
- •- PC-совместимые последовательные порты

7. Запуск QEMU в качестве эмулятора системы типа MIPS

Используйте исполняемый файл `qemu-system-mips', чтобы эмулировать машину MIPS. Эмулятор начинает запускать ядро Linux.

8. Запуск QEMU в качестве эмулятора пользовательского пространства

8.1 Быстрый запуск

Чтобы запустить процесс Linux, QEMU нужен сам исполняемый файл процесса и все используемые им динамические библиотеки (x86).

•Для архитектуры x86 вы можете просто попробовать запустить любой процесс, используя родные библиотеки:

```
qemu-i386 -L / /bin/ls
```

Опция -L / означает, что динамический компоновщик x86 необходимо искать с префиксом `/'.

•Поскольку QEMU также является процессом linux, вы моете запустить qemu с помощью qemu (ЗАМЕЧАНИЕ: вы сможете сделать это только в том случае, если вы сами скомпилировали QEMU из исходных кодов):

```
qemu-i386 -L / qemu-i386 -L / /bin/ls
```

•На не x86 процессорах вам нужно сначала загрузить как минимум glibc для x86 (`qemu-runtime-i386-XXX-.tar.gz' на веб-сайте QEMU). Убедитесь, что не установлена переменная окружения LD_LIBRARY_PATH:

```
unset LD_LIBRARY_PATH
```

Затем вы можете запустить исполняемый файл `ls', скомпилированный под x86:

```
qemu-i386 tests/i386/ls
```

Вы можете взглянуть на `qemu-binfmt-conf.sh', чтобы узнать, как QEMU автоматически запускается ядром Linux, когда вы пытаетесь запустить исполняемые файлы x86. Для этого в ядро Linux должен быть загружен модуль binfmt_misc.

•Версия QEMU для x86 также присутствует. Вы можете попробовать следующие вещи:

```
qemu-i386 /usr/local/qemu-i386/bin/qemu-i386 /usr/local/qemu-i386/bin/ls-i386
```

8.2 Запуск Wine

•Убедитесь, что у вас есть работающий QEMU и дистрибутив glibc для x86 (см. предыдущий раздел). Чтобы проверить это, выполните следующее:

```
qemu-i386 /usr/local/qemu-i386/bin/ls-i386
```

•Загрузите бинарный инсталлятор Wine для x86 (`qemu-XXX-i386-wine.tar.gz' на веб-сайте QEMU).
•Настройте свою учётную запись. Для этого взгляните на скрипт `/usr/local/qemu-i386/bin/wine-conf.sh'. Ваш старый каталог ${HOME}/.wine будет сохранён в ${HOME}/.wine.org.
•Теперь попробуйте запустить, например, `putty.exe':

```
qemu-i386 /usr/local/qemu-i386/wine/bin/wine /usr/local/qemu-i386/wine/c/Progr
```

8.3 Опции командной строки

```
использование: qemu-i386 [-h] [-d] [-L путь] [-s размер] программа [аргументы...]
```

`-h'

Вывод справки.

`-L path'

Префикс интерпретатора x86 elf (по умолчанию используется /usr/local/qemu-i386)

`-s размер'

Размер стека x86 в байтах (по умолчанию 524288)

Опции отладки:

`-d'

Включить журналирование (logfile=/tmp/qemu.log)

`-p размер_страницы'

Работает так, как если бы размер страницы системы-хозяина составлял указанное значение.

9. Компиляция из исходного кода

9.1 Linux/Unix

9.1.1 Компиляция

Сначала распакуйте архив с исходными текстами:

```
cd /tmp
tar zxvf qemu-x.y.z.tar.gz
cd qemu-x.y.z
```

Затем настройте QEMU и соберите его (обычно никакие опции не нужны):

```
./configure
make
```

Затем выполните под root'ом:

```
make install
```

чтобы установить QEMU в `/usr/local'.

9.1.2 Версии проверенных утилит

Чтобы успешно скомпилировать QEMU, очень важно, чтобы у вас были правильные версии утилит. Самым важным является gcc. Я не могу гарантировать работу QEMU, если вы не используете проверенную версию gcc. Взгляните на содержимое файлов 'configure' и 'Makefile', если вы хотите заставить работать другую версию gcc.

```
хост      gcc      binutils      glibc    linux       дистрибутив
-------------------------------------------------------------------
x86       3.2      2.13.2        2.1.3    2.4.18
         2.96     2.11.93.0.2   2.2.5    2.4.18      Red Hat 7.3
         3.2.2    2.13.90.0.18  2.3.2    2.4.20      Red Hat 9

PowerPC   3.3 [4]  2.13.90.0.18  2.3.1    2.4.20briq
         3.2

Alpha     3.3 [1]  2.14.90.0.4   2.2.5    2.2.20 [2]  Debian 3.0

Sparc32   2.95.4   2.12.90.0.1   2.2.5    2.4.18      Debian 3.0

ARM       2.95.4   2.12.90.0.1   2.2.5    2.4.9 [3]   Debian 3.0

[1] На Alpha для QEMU нужен атрибут 'видимости' gcc, доступный только
    для gcc версии >= 3.3.
[2] Нужен Linux >= 2.4.20 для поддержки исключения точности
    (не проверено).
[3] 2.4.9-ac10-rmk2-np1-cerf2

[4] gcc 2.95.x создаёт неверный код при использовании слишком большого числа
    переменных регистров. На PowerPC вам необходимо использовать gcc 3.x.
```

9.2 Windows

•Установите последние версии MSYS и MinGW с сайта http://www.mingw.org/. Вы можете найти подробные инструкции по установке в разделе загрузки и в FAQ'e.
•Загрузите devel-библиотеку MinGW для SDL 1.2.x (`SDL-devel-1.2.x-mingw32.tar.gz') с сайта http://www.libsdl.org. Распакуйте её во временный каталог и распакуйте архив `i386-mingw32msvc.tar.gz' в каталог MinGW. Отредактируйте скрипт `sdl-config', чтобы при запуске в нём использовался правильный каталог SDL.
•Распакуйте архив с последней версией QEMU.
•Запустите командный процессор MSYS (файл `msys.bat').
•Перейдите в каталог QEMU. Запустите `./configure' и `make'. Если у вас возникли проблемы с SDL, проверьте, можно ли запустить `sdl-config' из командной строки MSYS.
•вы можете установить QEMU в `Program Files/Qemu', выполнив команду `make install'. Не забудьте скопировать `SDL.dll' в `Program Files/Qemu'.

9.3 Кросс-компиляция в Linux для Windows

•Установите утилиты кросс-компиляции MinGW, доступные на сайте http://www.mingw.org/.

•Установите версию SDL для Win32 (http://www.libsdl.org), распаковав `i386-mingw32msvc.tar.gz'. Настройте переменную окружения PATH, чтобы `i386-mingw32msvc-sdl-config' мог быть запущен скриптом настройки QEMU.
•Настройте QEMU для кросс-компиляции под Windows:

```
./configure --enable-mingw32
```

Если необходимо, вы можете изменить кросс-префикс согласно префиксу, выбранному для утилит MinGW, с помощью опции -cross-prefix. Вы также можете использовать -prefix для указания пути установки Win32.

•Вы можете установить QEMU в каталог установки, выполнив `make install'. Не забудьте скопировать `SDL.dll' в каталог установки.

Замечание: в настоящий момент похоже, что Wine не может запустить QEMU для Win32.

9.4 Mac OS X

Патчи Mac OS X не полностью интегрированы в QEMU, поэтому всю необходимую информацию следует искать в архиве почтовой рассылки QEMU.

About This Document

This document was generated by Acid Jack on March, 19 2006 using texi2html 1.76.

The buttons in the navigation panels have the following meaning:

Button	Name	Go to	From 1.2.3 go to
[<]	Back	previous section in reading order	1.2.2
[>]	Forward	next section in reading order	1.2.4
[<<]	FastBack	beginning of this chapter or previous chapter	1
[Up]	Up	up section	1.2
[>>]	FastForward	next chapter	2
[Top]	Top	cover (top) of document	
[Contents]	Contents	table of contents	
[Index]	Index	index	
[?]	About	about (help)	

where the Example assumes that the current position is at Subsubsection One-Two-Three of a document of the following structure:

- 1. Section One
    - 1.1 Subsection One-One
        - ...
    - 1.2 Subsection One-Two
        - 1.2.1 Subsubsection One-Two-One
        - 1.2.2 Subsubsection One-Two-Two
        - 1.2.3 Subsubsection One-Two-Three <== Current Position
        - 1.2.4 Subsubsection One-Two-Four
    - 1.3 Subsection One-Three
        - ...
    - 1.4 Subsection One-Four