


## Запуск Windows поверх Ubuntu Linux 7.04

 [QEmu](#) позволяет запускать несколько ОС поверх другой, например, Windows поверх Linux. QEmu использует qemu, драйвер ускорения, включенный в поставку Ubuntu 7.04 Feisty Fawn для того, чтобы выполнять Windows с приемлемой скоростью.

Ubuntu 7.04 Feisty Fawn также включает в себя новую версию rdesktop, которая способна запускать отдельные графические приложения из VM на вашем обычном рабочем столе. Подробнее – в разделе «[Монолитная виртуализация](#)».



Примечание: Большинство компьютеров, поставляемых с начала 2006 года поддерживают технологии аппаратной виртуализации VT или Pacifica, значительно ускоряющие виртуализацию. Для того, чтобы проверить это, выполните:

```
grep -E '^flags.*(vmx|svm)' /proc/cpuinfo
```

Если Вы получаете какой-то результат, Вам имеет смысл обратиться к разделу «[KVM \(Kernel Virtual Machine\)](#)», и использовать KVM вместо QEmu и KQEmu по причине более высокого быстродействия.

Чтобы установить *Ubuntu* в QEmu, прочтите [Installation/QemuEmulator](#).

## Установка Windows в QEmu на Feisty Fawn 7.04

---

Эта статья описывает Windows XP, но вы можете аналогично установить любые ранние версии Windows, возможно, они будут более приемлемы по причине более высокого быстродействия.

- Установите требуемые пакеты (см. [InstallingSoftware](#)):

```
sudo apt-get install qemu kqemu-source module-assistant
```

- Соберите модуль kqemu из пакета Ubuntu:

```
sudo module-assistant prepare
sudo module-assistant build kqemu
sudo module-assistant install kqemu
```

- Настройте udev на создание узла /dev/kqemu во время загрузки:

- Создайте файл /etc/udev/rules.d/60-kqemu.:

```
KERNEL=="kqemu", NAME="%k", MODE="0660", GROUP="kqemu"
```

- Измените содержание файла /etc/modprobe.d/kqemu:

```
options kqemu major=0
```

- Создайте группу "kqemu" и добавьте в нее пользователей, которым будет разрешено пользоваться QEMU с ускорением:

```
sudo addgroup --system kqemu
```

```
sudo adduser myUser kqemu
```

Убедитесь что в итоге вы имеет права на запуск QEmu (например выйдите, и повторно войдите в систему).

- Загрузите kqemu

```
sudo modprobe kqemu
```

- Если вы хотите активизировать модуль kqemu при каждой загрузке, добавьте "kqemu" в /etc/modules.
- Создайте виртуальный диск для установки Windows

```
qemu-img create -f qcow windows.img 2G
```

будет создан одиночный файл максимальным объемом 2 Гб под названием windows.img в текущем каталоге. Для Windows XP приемлем любой объем не

менее 1.5 Гб. Виртуальный диск заполняется по мере использования, и не может превышать заданный при создании объем.

**Примечание:** Если вы хотите устанавливать Windows XP Service Pack 2, создавайте диск не меньше 3 Гб.

**Примечание:** Используя образы формата qcow, вы не сможете их монтировать стандартными утилитами. Для избежания этого опустите ключ '-f qcow', однако это заставит систему распределить полный объем диска сразу. Вы можете сконвертировать образ этого типа в qcow, следуя инструкциям ниже.

- Установим Windows XP. Вставим диск в привод и выполним:

```
qemu -localtime -cdrom /dev/cdrom -m 384 -boot d windows.img
```

Это запустит Qemu, заставив его использовать локальный таймер, windows.img в качестве жесткого диска, и /dev/cdrom в качестве CD, с 384 Мб оперативной памяти (хороший минимум для XP) и загрузиться с CD.

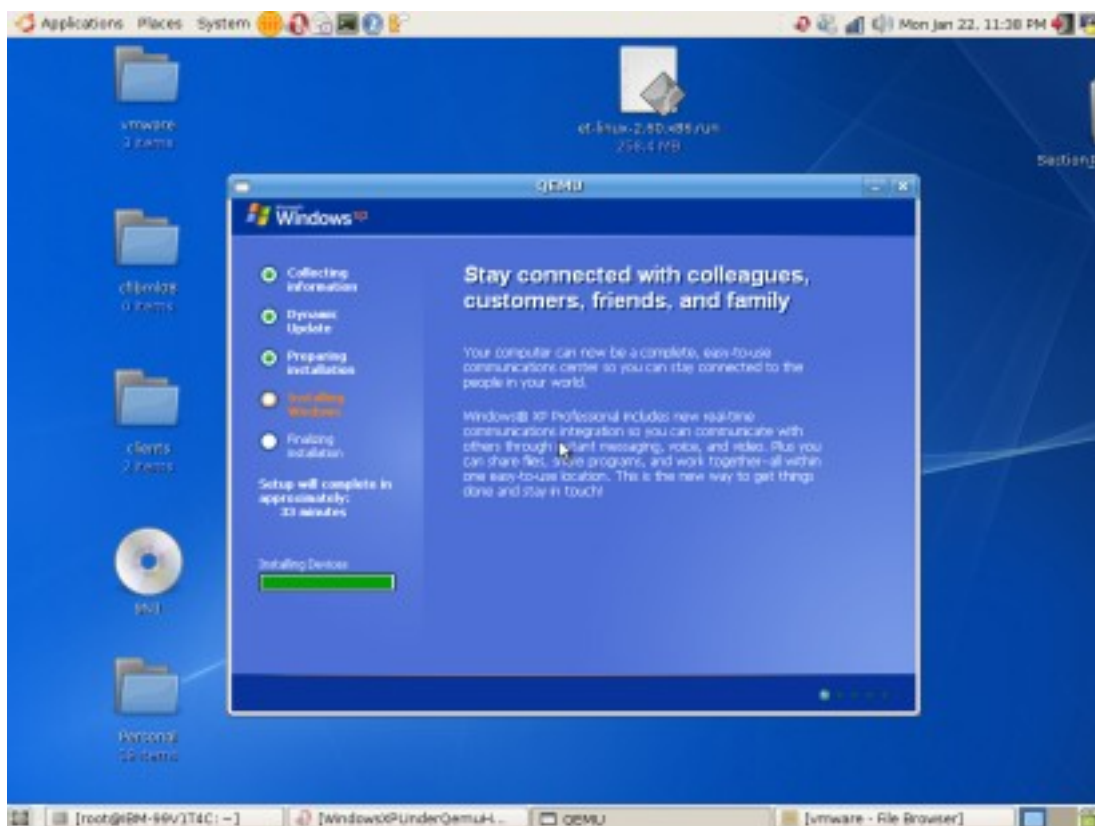
Можно использовать образ .iso:

```
qemu -localtime -cdrom cdiimagefile.iso -m 384 -boot d windows.img
```

**Примечание для пользователей 64-разрядных систем:** Используйте 'qemu-system-x86\_64' вместо 'qemu'. Иначе опция "-kernel-kqemu" (ускорение ядром) будет игнорирована.

**Примечание по установке Windows 2000:** Используйте опцию '-win2k-hack'. Иначе инсталлятор может выдать ошибку "Not enough disk space" (Нет свободного места на диске).

QEmu запустится, загрузится с CD, и запустит программу установки XP. На некоторых системах последняя фаза начальной установки может длиться очень долго (см. рисунок). В этом случае просто перезапустите QEmu.



После нескольких перезагрузок вы получите работающую Windows XP.

Щелкнув по окну QEmu, Вы передаете управление виртуальной машине. Для возврата обратно нажмите Ctrl + Alt. Кроме того, вы можете переходить в полноэкранный режим и обратно при помощи Ctrl + Alt + F

## Сеть

---

QEmu предоставляет два типа сети. В обоих режимах, внутри гостевой системы Windows создается виртуальный сетевой адаптер.

Документация по сетевому функционалу QEmu

- <http://fabrice.bellard.free.fr/qemu/qemu-doc.html>

### Сетевой стек режима пользователя

В этом режиме QEmu управляет сетевыми интерфейсами самостоятельно. QEmu предоставляет DHCP сервер, присваивающий динамические IP адреса гостевым системам. TCP- и UDP-порты перенаправляются с учетом параметров командной строки

```
QEMU VLAN      <-----> Firewall/DHCP server <-----> Host network
                |
                |          (10.0.2.2)
                |
                |-----> DNS server (10.0.2.3)
                |
                |-----> SMB server (10.0.2.4)
```

Плюсы

- Простая настройка

Минусы

- Доступны только TCP и UDP соединения, команда `ping` не будет работать.
- Требуется предварительная настройка входящих соединений

## TAP

В случае TAP-соединения, QEmu соединяет гостевые системы с основной, используя сетевой мост TAP. Linux создает виртуальный сетевой интерфейс (tap0), доступный для использования штатным образом. Это дает возможность перенаправлять сетевой трафик в гостевые ОС, используя штатные средства Linux.

Документация по TAP

- <http://www.ubuntuforums.org/showthread.php?t=179472>
- <http://compsoc.dur.ac.uk/~djw/qemu.html>
- <http://bbs.archlinux.org/viewtopic.php?id=27298>

За

- Полная эмуляция IP стека

Против

- Сложная настройка. Проблемы при попытке запускать QEmu без привилегий администратора.

## Дополнительные возможности

---

### Общий доступ к файлам

Если у вас настроена [Samba](#), QEmu позволит Вам обмениваться файлами между основной и гостевой системами. Пример подразумевает наличие разделяемого каталога с названием 'qemu\_share'.

Запустите QEmu следующим образом:

```
qemu -smb $HOME/qemu_share -m 384 -localtime windows.img
```

Внутри Windows отобразите папку как сетевой диск:

```
net use e: \\10.0.2.2\qemu_share
```

IP адрес вашего хоста всегда будет 10.0.2.2. если у вас запущены какие-то службы, вы можете их использовать из гостевой системы по IP-адресу

## Монтирование виртуального диска.

Убедитесь, что QEmu **НЕ ЗАПУЩЕН**.

Создайте папку '/media/qemu'. Выполните:

```
sudo mount -o loop,offset=32256 windows.img /media/qemu
```

## Контрольные точки VM

Вы можете пользоваться т.н. «оверлейными файлами» QEmu для сохранения состояния системы. Эти файлы меньше, чем виртуальный диск и содержат только изменения, которые произошли с момента создания оверлея. В случае краха системы вам достаточно удалить старый оверлей, и система будет возвращена к первоначальному состоянию.

```
qemu-img create -b windows.img -f qcow windows.ovl
```

теперь вы должны запускать Windows так:

```
qemu windows.ovl -enable-audio -user-net -localtime -smb qemu_share
```

QEmu также воспринимает команды из оболочки гостевой ОС. Для этого, находясь в гостевой ОС, нажмите CTRL+ALT+2. Это активизирует монитор QEmu. Используя монитор вы можете менять CD, посылать нажатия клавиш, приостанавливать выполнение, и т.п. За подробностями обратитесь к [официальной документации](#). Для выхода из монитора QEmu нажмите CTRL+ALT+1

## Преобразование монтируемого диска в формат qcow

```
qemu-img convert windows.img -O qcow windows.qcow
```

## Решение проблем

---

Если у Вас возникают проблемы с мышью в QEmu, перед запуском выполните

```
export SDL_VIDEO_X11_DGAMOUSE=0
```

---

# KVM (Kernel Virtual Machine)

---

Kernel Virtual Machine является частью Ubuntu 7.04.

## Запуск Windows XP или Ubuntu внутри KVM поверх Ubuntu 7.04 Feisty Fawn

### Установка KVM на Ubuntu 7.04 Feisty Fawn

---

- Установите пакеты '**kvm**' и '**qemu**'. (См. [InstallingSoftware](#)).
- Загрузите драйвер:
  - Для процессоров AMD

```
sudo modprobe kvm-amd
```

- Для процессоров Intel

```
sudo modprobe kvm-intel
```

Для Intel-компьютеров может потребоваться обновление BIOS и разрешение использования виртуализации в настройках BIOS перед продолжением. Иначе `modprobe kvm-intel` будет выдавать ошибку "Operation is not supported". Подробнее см. дискуссию:

<http://ubuntuforums.org/showthread.php?t=350691>

- Сделайте `/dev/kvm` доступным пользователю:

```
sudo chmod 777 /dev/kvm
```

или

```
sudo adduser $USER kvm  
# logout and back in
```



## Установка гостевой Windows XP

---

- Создайте виртуальный диск.

```
qemu-img create windows.img -f qcow 6G
```

- Запустите KVM и установите Windows

Вставьте установочный диск Windows и выполните:

```
kvm -no-acpi -m 384 -cdrom /dev/cdrom -boot d windows.img
```

- -m устанавливает объем оперативной памяти - 256М – реальный минимум.

Для избежания проблем с ACPI, когда установка Windows XP запросит "Press F6 for additional drivers", нажмите F5. Установочная программа позволит выбрать тип компьютера. Выберите «Standard PC» и продолжайте установку.

**ВАЖНО:** ключ '-boot d' заставляет KVM to пытаться загрузиться первым с CD-ROM. После установки загрузите Вашу VM так:

```
kvm -no-acpi -m 384 -cdrom /dev/cdrom windows.img
```

Вы можете создать ярлык панели, указав полный путь к образу (/home/user/windows.img)

Если вы получаете ошибку:

```
Could not configure '/dev/rtc' to have a 1024 Hz timer. This is not a fatal error, but for better emulation accuracy either use a 2.6 host Linux kernel or type 'echo 1024 > /proc/sys/dev/rtc/max-user-freq' as root.
```

Исправьте ее:

```
sudo -s  
echo 1024 > /proc/sys/dev/rtc/max-user-freq
```

Если Ваша машина грузится и работает слишком медленно, обратитесь к статье [http://kvm.qumranet.com/kvmwiki/Windows\\_ACPI\\_Workaround](http://kvm.qumranet.com/kvmwiki/Windows_ACPI_Workaround)

## Установка гостевой Ubuntu Edgy

---

- Создайте файл виртуального диска.

```
qemu-img create edgy.img -f qcow 6G
```

- Запустите KVM и установите Edgy

Вставьте установочный диск Edgy и выполните:

```
kvm -m 256 -cdrom /dev/cdrom -boot d edgy.img
```

На процессорах Intel стандартный установщик вызывает проблему реального режима ([real mode](#)). Используйте [netboot mini.iso](#) или альтернативный установщик.

Загрузите вашу VM

```
kvm -m 256 edgy.img
```

### ***Продвинутая настройка сети***

Если Вам это не нужно, Вам вполне подойдет стандартная документация QEmu, релевантная KVM.

## Определения

---

VLAN: сегмент виртуальной сети. Вы можете представлять себе его как виртуальный сетевой коммутатор. Вы подключаете виртуальные сетевые интерфейсы в VLANы, и связываете их, создавая стеки коммутаторов, максимально похожие по своему функционированию на установку реальной сети.

### Сетевой стек режима пользователя, режим по умолчанию

---

Если Вы не определяете сетевые параметры, KVM по умолчанию создает сетевой интерфейс, соединенный с частным VLAN. На этом VLAN также эмулируется одиночный узел, действующий как DHCP сервер и шлюз по умолчанию. Все соединения инициированные гостевой системой, маршрутизируются через внутренний стек KVM и представляются на хосте как сокеты, открытые процессом KVM с адреса 127.0.0.1.

Сети режима пользователя KVMобычно содержат два адреса: 10.0.2.2 (хост) and 10.0.2.15 (гость).

В нотации KVM это будет:

```
$ kvm -net nic -net user ...
```

Другими словами: добавить сетевую карту, и соединить ее с vlan 1. Также, добавить виртуальный хост в vlan 1. Виртуальный хост через NAT соединяется с аппаратным узлом.

Так как этот режим реализован с использованием протокола SLIRP, то UDP не поддерживается. Если вам это необходимо, Вам придется использовать другие методы.

## Связь между VLAN

---

### Множественные сетевые карты и VLAN

Что если вам понадобится VM с более чем одной сетевой картой? Вам потребуется создать несколько VLAN для подсоединения сетевых карт. Для создания множественных VLAN, присвойте каждому уникальный номер. Например, так создаются 2 сетевые карты и 2 VLAN:

```
$ kvm -net nic,vlan=0 -net socket,listen=:8010,vlan=0 -net nic,vlan=1 -net user,vlan=1 ...
```

### Соединение по Socket

Итак, мы можем объединять машины в VLANы, что нужно сделать чтобы объединить VLANы? Самый простой способ – задействовать сокет TCP. Один VLAN ожидает соединений:

```
$ kvm -net nic -net socket,listen=:8010 ...
```

Другие устанавливают соединения:

```
$ kvm -net nic -net socket,connect=127.0.0.1:8010 ...
```

Если вы опускаете адрес, то прослушиваются все присвоенные адреса. Чтобы прослушивать только локальные VLAN, укажите:

```
$ kvm -net nic -net socket,listen=127.0.0.1:8010 ...
```

Кроме того, это позволяет объединять VLANы, расположенные на разных аппаратных узлах.

## Групповые сокеты

Если вы хотите присоединять много VLAN-инициаторов, вы должны использовать групповые сокеты.

```
$ kvm -net nic -net socket,mcast=230.0.0.1:1234 ...  
$ kvm -net nic -net socket,mcast=230.0.0.1:1234 ...  
$ kvm -net nic -net socket,mcast=230.0.0.1:1234 ...
```

Это заставляет принимать не более чем 3 разных VLAN. Кадры, посланные любым VLAN, транслируется в остальные.

## VDE

Множественные VLAN могут быть также объединены одним VDE. VDE описаны ниже.

## Соединение VLANs с хостом

---

### Устройство TAP

Большинство (все?) virtual-to-physical соединений сделаны как TAP. TAP-устройства – обычные сетевые интерфейсы, наподобие eth0, eth1, lo, и т.п. Один конец TAP соединен с VLAN, другой – соединен и смаршрутизирован на интерфейс хоста.

```
$ kvm -net nic -net tap ...
```

Эта команда создает новый уникальный интерфейс tap (tap0, tap1, etc). Для автоматизации при запуске интерфейса предназначен скрипт /etc/qemu-ifup. По умолчанию /etc/qemu-ifup просто присваивает интерфейсу IP-адрес 172.20.0.1. Вы можете явно указать имя сети, используя опцию ifname=IF, и другой скрипт, используя script=SCRIPT:

```
$ kvm -net nic -net tap,ifname=qtap0,script=/var/vm/vm0.ifup
```

Такой подход решает проблему с сетью для одиночной VM, однако не подходит для массового хостинга, так как каждая гостевая система требует собственный TAP.

## Усложненная работа с сетью

---

Итак, как же мы можем объединить виртуальные машины в сеть, и предоставить доступ к внешней сети? Существует множество разных способов со своими pro et contra. Это одна из причин по которой сетевые инженеры делают большие деньги. Ниже изложены некоторые техники.

### **Виртуальные сетевые карты на VDE, VDE через Tap на хост и через NAT в сеть**

Это позволяет гостевым системам соединяться друг с другом, с хостом, внешними сетями. Кроме того, это позволяет соединяться хосту с любой гостевой ОС. Однако это не позволяет внешним сетям иметь входящий доступ к гостям (ничто не мешает вам настроить входящие соединения вручную через `-redir`, или `ssh -L`). В любом случае эта методика не затрагивает конфигурацию сети хоста. Вы должны запускать эти команды каждый раз при загрузке VM.

- **\$ sudo sh -c "echo 1 > /proc/sys/net/ipv4/ip\_forward"**

добавим "tun" в /etc/modules.conf. И запустим

- **\$ sudo modprobe tun.**
- **\$ sudo apt-get install vde dnsmasq**
- **\$ sudo /etc/init.d/dnsmasq stop**
- **\$ sudo vde\_switch -tap qtap0 -daemon**

теперь vde\_запущен на /tmp/vdectl (используйте -socket PATH для указания местоположения сокета).

- **\$ sudo ifconfig qtap0 10.111.111.254 broadcast 10.111.111.255 netmask 255.255.255.0 up**
- **\$ sudo iptables -t nat -A POSTROUTING -o `route -n | egrep '^0\.0\.0\.0 ' | awk '{print \$8}'` -j MASQUERADE**
- **\$ sudo dnsmasq --log-queries --user=nobody --dhcp-leasefile=/var/tmp/dnsmasq-leasefile --dhcp-range=10.111.111.129,10.111.111.199,255.255.255.0,10.111.111.255,8h -interface=qtap0 --domain=qemu.lan -d**
- **\$ sudo vdeq kvm -hda v2.qcow -boot c -net nic -net vde -m 192**

В госте, укажите **nameserver 10.111.111.254** в /etc/resolv.conf. Убедитесь, что DHCP выдает чистые IP адреса.

В госте выполните, **ping 10.111.111.254** и он должен работать. На хосте - **ping 10.111.111.140** (или другой адрес гостя).

Из гостя пингуйте любой внешний адрес. Если он пингуется – все настроено правильно.

## Прямой мост во внешнюю сеть

Эта техника подменяет умолчательный интерфейс хоста мостом. Когда вы соединяете гостевую систему с мостом, он отображается во внешнюю сеть так, как если бы реально существовал. Это более надежный, но и более трудный путь. Будьте осторожны! Эта техника делает все виртуальные сетевые карты видимыми. И убедитесь что каждая из них имеет корректный MAC адрес! Когда вы настраиваете сеть дома, это самый легкий способ.

- В командной строке KVM укажите следующий параметр. Убедитесь что MAC уникален в контексте вашей сети.

```
-net nic,macaddr=00:xx:xx:xx:xx:xx -net tap
```

- Исправьте `/etc/network/interfaces`

```
auto lo br0 eth0

iface lo inet loopback

iface br0 inet dhcp
    bridge_ports eth0

iface eth0 inet manual
```

- Для интерфейса `eth0`

Иногда после исправления `/etc/network/interfaces` может потребоваться перезагрузка.

# Монолитная виртуализация

## Монолитная виртуализация

Вместо того, чтобы иметь доступ к целому рабочему столу Windows, вы можете запускать виртуализованные приложения напрямую с Вашего рабочего стола, используя последнюю версию (version 1.5.0) пакета **rdesktop**, поставляемого с Ubuntu 7.04, и Windows XP Professional's Terminal Services.



- Запустите Windows XP pro .
  - Используя VMware, настройте host-only networking, и запомните адрес VM.
  - Для Qemu используйте команду ниже


```
qemu -m 384 -redir tcp:3389::3389 windows.img
```

Это перенаправляет любые соединения на порт 3389 локальной машины на VM на порт 3389, где запущена служба терминалов.

- Войдите администратором в VM. Выполните **Start** → **Control Panel** → **System (Пуск – Панель управления - Система)**. На вкладке **Remote**



(Удаленное управление) отметьте **Allow users to connect remotely to this computer (разрешать удаленные подключения)**. Кликните **ОК**.

- Внутри VM, загрузите  <http://www.cendio.se/files/thinlinc/seamlessrdp/seamlessrdp.zip>. Извлеките в **C:\seamlessrdp**. Выйдите из VM.
- Установите **rdesktop** на хост. (см. [InstallingSoftware.](#))
- Проверьте приложение.

```
rdesktop -A -s "c:\seamlessrdp\seamlessrdpshell.exe C:\Program Files\Internet Explorer\iexplore.exe" <IP of VM>:3389 -u administrator -p password
```

Теперь вы можете создать ярлык для указанной команды.

