

MINIX3 в QEMU

Циллорик О.И.

< olej@front.ru >

Редакция 2.18

от 15.12.2009

Оглавление

Установка QEMU.....	2
Акселератор.....	5
Установка MINIX3.....	7
Загрузка с CD.....	7
Управление консолями в QEMU.....	8
Установка MINIX3 на диск.....	9
MINIX3 реальный и виртуальный.....	11
Оживляем сетку.....	12
Удалённый доступ.....	16
QEMU & X.....	17
Установка X.....	17
Удалённый доступ к X.....	22
X в терминале QEMU.....	24
Дополнительные источники информации.....	27

Заглавие этого текста выглядит, возможно, как-то не самым удачным образом, ... как «заяц в белом вине», но разговор пойдёт об установке операционной системы MINIX3 под виртуальной машиной QEMU, и не только об установке, но и настройке, и использовании некоторых не совсем очевидных особенностей. Все показанные действия будут выполняться в операционной системе Linux, для определённости — CentOS 5.2 (Fedora core 6), всё тоже самое будет сохраняться для дистрибутивов, использующих инсталлятор yum и пакетную систему rpm (Fedora, RedHat); в других дистрибутивах Linux всё будет практически тоже, с учётом отличий инсталляции пакетов применяемым в дистрибутиве системой инсталляций.

Но сначала краткие предварительные замечания относительно типографской разметки текста: что и как будет выглядеть. Некоторые действия (установка программ и др.) требуют выполнения команд с правами root. Далее по тексту, будут по возможности тщательно указываться необходимость прав root при выполнении команды — использованием значка приглашения в записи команды: # - только root полномочия, \$ - не требует root привилегий.

Примечание: Выполнение отдельной команды с правами root будет иногда показываться в записи:

```
$ su -c 'команда'
```

что то же самое (почти то же самое — отличие в переменных окружения, в путях поиска файлов), что и:

```
# команда
```

Все приводимые в тексте консольные команды проверялись дополнительно непосредственно перед написанием текста, записи выполнения консольных команд приводятся прямым (без редактирования) копированием с терминала. Такой текст (скопированный с терминала) будет показываться моноширинным шрифтом. Кроме того, в большинстве случаев (особо это касается ввода многострочных команд, где отдельные строки завершаются «\») пользовательский ввод в записи команды будет показан жирным шрифтом, а ответный вывод от системы — обычным. Короткие цитаты из различных источников информации будут показываться курсивом.

Вывод команд приведен, возможно, несколько с избыточными подробностями, но это сделано сознательно, для возможностей сравнения ожидаемого и получаемого результатов при воспроизведении, особенно при изменениях версий используемых команд.

Установка QEMU

Начнём с предположения, что виртуальная машина QEMU ещё не установлена в нашем Linux. Можно устанавливать её из исходных кодов, но я покажу инсталляцию бинарных пакетов из RPM-формата, а инсталляцию из исходных кодов мы увидим позже, при доустановке акселератора к QEMU (установку QEMU можно произвести так же и из того же источника). Ищем доступный репозиторий и нужные пакеты в нём:

```
# yum list qem*
```

```
...
```

```
Available Packages
```

qemu.i386	2:0.10.5-1.e15.2	epel
qemu-common.i386	2:0.10.5-1.e15.2	epel
qemu-img.i386	2:0.10.5-1.e15.2	epel
qemu-system-arm.i386	2:0.10.5-1.e15.2	epel
qemu-system-cris.i386	2:0.10.5-1.e15.2	epel
qemu-system-m68k.i386	2:0.10.5-1.e15.2	epel
qemu-system-mips.i386	2:0.10.5-1.e15.2	epel
qemu-system-ppc.i386	2:0.10.5-1.e15.2	epel
qemu-system-sh4.i386	2:0.10.5-1.e15.2	epel
qemu-system-sparc.i386	2:0.10.5-1.e15.2	epel
qemu-system-x86.i386	2:0.10.5-1.e15.2	epel
qemu-user.i386	2:0.10.5-1.e15.2	epel

Примечание: здесь показан результат на моём компьютере. Он может оказаться заметно шире, чем у вас, потому как использовался поиск в одном из дополнительно подключенных репозиториях yum, а yum показывает из найденных результатов самую последнюю найденную версию. Посмотреть текущие подключенные к yum репозитории можно:

```
# yum repolist
```

```
...
```

repo id	repo name	status
addons	CentOS-5 - Addons	enabled
adobe-linux-i386	Adobe Systems Incorporated	enabled
base	CentOS-5 - Base	enabled
epel	Extra Packages for Enterprise Linux 5 -	enabled
extras	CentOS-5 - Extras	enabled
livna	rpm.livna.org for 5 - i386	enabled
planetccrma	Planet CCRMA 5 - i386	enabled
planetcore	Planet CCRMA Core 5 - i386	enabled
updates	CentOS-5 - Updates	enabled

В принципе, стандартно определённого при установке Linux репозитория base достаточно для наших целей: из всех показанных пакетов **необходимым** будет только `qemu.i386` (остальные будут **достаточными** для виртуализации не только OS, но и архитектуры процессора); отличия будут только в представленной версии, вот результат выполнения на другом компьютере:

```
# yum list qem*
```

```
...
```

```
Available Packages
```

qemu.i386	qemu.i386 0:0.9.0-4	base
-----------	---------------------	------

Но и этого достаточно для наших целей! Если же вам захочется добавить репозиторий epel к списку репозиториях yum, то делается это так:

```
# rpm -Uvh http://download.fedora.redhat.com/pub/epel/5/i386/epel-release-5-3.noarch.rpm
```

Загружается http://download.fedora.redhat.com/pub/epel/5/i386/epel-release-5-3.noarch.rpm

предупреждение: /var/tmp/rpm-xfer.1BRu2J: Заголовок V3 DSA signature: NOKEY, key ID 217521f6

Подготовка... ##### [100%]
1:epel-release ##### [100%]

Продолжаем установку QEMU, устанавливать будем **всё**, не жалея места:

yum install qem*

...

Dependencies Resolved

```
=====
```

Package	Arch	Version	Repository	Size
---------	------	---------	------------	------

```
=====
```

Installing:

qemu	i386	2:0.10.5-1.el5.2	epel	14 k
------	------	------------------	------	------

Installing for dependencies:

pulseaudio-libs	i386	0.9.10-1.el5.3	epel	166 k
qemu-common	i386	2:0.10.5-1.el5.2	epel	209 k
qemu-img	i386	2:0.10.5-1.el5.2	epel	103 k
qemu-system-arm	i386	2:0.10.5-1.el5.2	epel	803 k
qemu-system-cris	i386	2:0.10.5-1.el5.2	epel	438 k
qemu-system-m68k	i386	2:0.10.5-1.el5.2	epel	498 k
qemu-system-mips	i386	2:0.10.5-1.el5.2	epel	2.6 M
qemu-system-ppc	i386	2:0.10.5-1.el5.2	epel	2.4 M
qemu-system-sh4	i386	2:0.10.5-1.el5.2	epel	911 k
qemu-system-sparc	i386	2:0.10.5-1.el5.2	epel	747 k
qemu-system-x86	i386	2:0.10.5-1.el5.2	epel	1.4 M
qemu-user	i386	2:0.10.5-1.el5.2	epel	4.7 M

Transaction Summary

```
=====
```

Install	13 Package(s)
Update	0 Package(s)
Remove	0 Package(s)

Total download size: 15 M

Is this ok [y/N]: y

Downloading Packages:

(1/13): qemu-common-0.10.	100%	=====	209 kB	00:07
(2/13): qemu-system-x86-0	100%	=====	1.4 MB	00:49
(3/13): qemu-system-m68k-	100%	=====	498 kB	00:16
(4/13): qemu-system-cris-	100%	=====	438 kB	00:14
(5/13): qemu-img-0.10.5-1	100%	=====	103 kB	00:03
(6/13): qemu-system-ppc-0	100%	=====	2.4 MB	01:23

```
(7/13): qemu-system-mips- 100% |=====| 2.6 MB 01:28
(8/13): qemu-system-sh4-0 100% |=====| 911 kB 00:30
(9/13): qemu-0.10.5-1.el5 100% |=====| 14 kB 00:00
(10/13): pulseaudio-libs- 100% |=====| 166 kB 00:05
(11/13): qemu-system-spar 100% |=====| 747 kB 00:24
(12/13): qemu-user-0.10.5 100% |=====| 4.7 MB 02:49
(13/13): qemu-system-arm- 100% |=====| 803 kB 00:26
```

Running rpm_check_debug

Running Transaction Test

Finished Transaction Test

Transaction Test Succeeded

Running Transaction

```
Installing: qemu-common ##### [ 1/13]
Installing: pulseaudio-libs ##### [ 2/13]
Installing: qemu-system-arm ##### [ 3/13]
Installing: qemu-system-sparc ##### [ 4/13]
Installing: qemu-system-sh4 ##### [ 5/13]
Installing: qemu-system-mips ##### [ 6/13]
Installing: qemu-system-ppc ##### [ 7/13]
Installing: qemu-system-cris ##### [ 8/13]
Installing: qemu-system-m68k ##### [ 9/13]
Installing: qemu-system-x86 ##### [10/13]
Installing: qemu-user ##### [11/13]
Installing: qemu-img ##### [12/13]
Installing: qemu ##### [13/13]
```

Unable to look at what's on dbus

Installed: qemu.i386 2:0.10.5-1.el5.2

Dependency Installed: pulseaudio-libs.i386 0:0.9.10-1.el5.3 qemu-common.i386 2:0.10.5-1.el5.2 qemu-img.i386 2:0.10.5-1.el5.2 qemu-system-arm.i386 2:0.10.5-1.el5.2 qemu-system-cris.i386 2:0.10.5-1.el5.2 qemu-system-m68k.i386 2:0.10.5-1.el5.2 qemu-system-mips.i386 2:0.10.5-1.el5.2 qemu-system-ppc.i386 2:0.10.5-1.el5.2 qemu-system-sh4.i386 2:0.10.5-1.el5.2 qemu-system-sparc.i386 2:0.10.5-1.el5.2 qemu-system-x86.i386 2:0.10.5-1.el5.2 qemu-user.i386 2:0.10.5-1.el5.2

Complete!

Всё! Проверяем доступность установленного QEMU:

```
$ which qemu
```

```
/usr/bin/qemu
```

```
$ qemu
```

```
QEMU PC emulator version 0.10.5, Copyright (c) 2003-2008 Fabrice Bellard
```

```
usage: qemu [options] [disk_image]
```

```
...
```

- здесь вы получите обстоятельнейшую справку о деталях командной строки QEMU. Но установить — этого мало, хорошо бы ещё и проверить работоспособность и оценить качество QEMU прежде двигаться дальше. Отправляемся на сайт проета QEMU: <http://www.qemu.org/download.html> и на этой странице находим достаточно много образов различных OS собранных для загрузки в качестве образца, загрузим из них:

FreeDOS floppy disk image

<http://odin.fdos.org/odin2005/odin1440.img>

Выполняем команду (некоторые опции командной строки будут обсуждены позже, или см. внимательно ту справку, которую выводит `qemu` при запуске без параметров):

```
$ qemu -m 8M -fda odin1440.img -boot a
Could not open '/dev/kqemu' - QEMU acceleration layer not activated: No such file or directory
```

Может быть дополнительно ещё и такое сообщение:

```
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.
```

Для устранения такого сообщения, как в нём и указано, сделаем:

```
# su -c 'echo 1024 > /proc/sys/dev/rtc/max-user-freq'
```

С 1-м сообщением мы станем разбираться чуть далее по тексту.

Но, тем не менее, запуск FreeDOS произошёл, как это показано на рисунке:

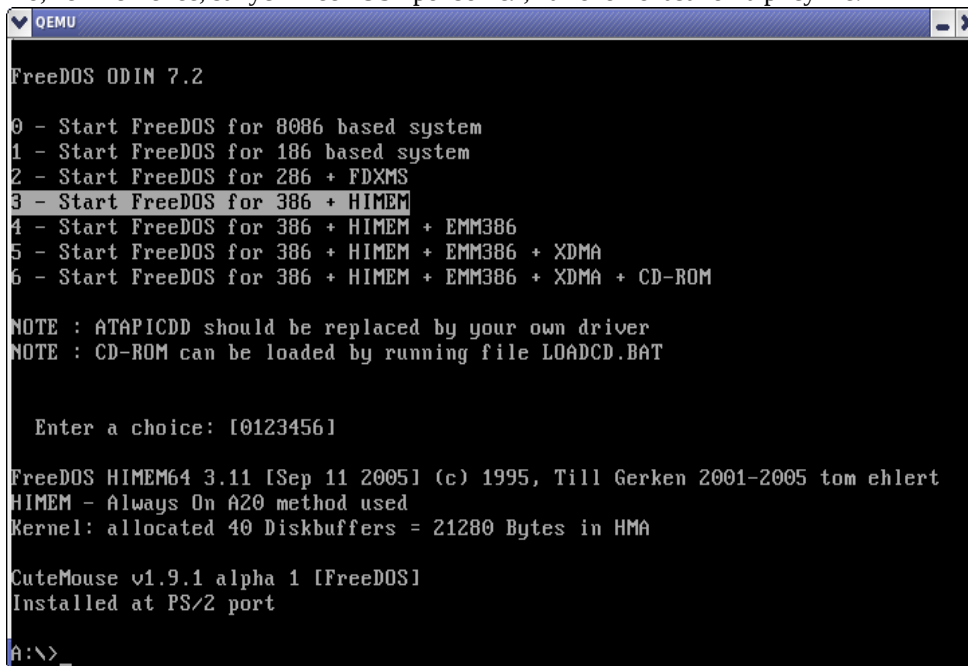


Рис.1.1

Старт виртуальной FreeDOS системы.

Примечание (в качестве напоминания): как делать снимки окон в Linux? Без этого любые описания или документация — голословные, и содержат, как правило, ошибки. Обычный способ: нажать PrintScreen и дождаться появления окна программы. Но часто (при большом числе установленных средств, которые переопределяют и перехватывают клавиши) это может не работать. Тогда делаем прямой вызов программы из командной строки, например (в KDE):

```
$ ksnapshot &
```

```
[1] 10115
```

Акселератор

Теперь возвращаемся к первому из показанных выше сообщений QEMU. Это он «жалуется» на отсутствие акселератора `kqemu`. Вот что пишут относительно этого на <http://www.xakep.ru/magazine/xa/118/094/1.asp> (интересная статья о QEMU, с которой стоит ознакомиться):

По сравнению с другими известными виртуальными машинами, QEMU работает достаточно шустро. Но есть еще модуль QEMU Accelerator Module (KQEMU), позволяющий выполнять часть кода напрямую на реальном процессоре, минуя виртуальный. Это неплохо ускоряет работу гостевой системы. Без этого модуля запуск виртуальной ОС замедляется примерно в пять раз.

Пакет акселератора вы с помощью `yum` не найдёте, и мы поставим его из исходных кодов. Загрузим <http://www.qemu.org/kqemu-1.4.0pre1.tar.gz> с того же сайта проекта QEMU, которым мы уже пользовались. Далее (в удобном нам произвольном каталоге, результат `make` я показываю расширенно — здесь видно, что создаётся модуль ядра Linux):

```

$ tar -zxvf kqemu-1.4.0pre1.tar.gz
$ cd kqemu-1.4.0pre1
$ ./configure
Source path      /usr/src/kqemu-1.4.0pre1
C compiler       gcc
Host C compiler  gcc
make             make
host CPU         i386
kernel sources   /lib/modules/2.6.18-92.el5/build
kbuild type      2.6
$ make
make -C common all
...
gcc -D__KERNEL__ -nostdinc -iwithprefix include -I. -I.. -D__ASSEMBLY__ -c -o
i386/nexus_asm.o i386/nexus_asm.S
gcc -D__KERNEL__ -nostdinc -iwithprefix include -I. -I.. -D__ASSEMBLY__ -c -o
i386/monitor_asm.o i386/monitor_asm.S
...
gcc -D__KERNEL__ -nostdinc -iwithprefix include -I. -I.. -D__ASSEMBLY__ -c -o
i386/kernel_asm.o i386/kernel_asm.S
ld -r -o ../kqemu-mod-i386.o kernel.o i386/kernel_asm.o
make[1]: Leaving directory `/usr/src/kqemu-1.4.0pre1/common'
make -C /lib/modules/2.6.18-92.el5/build M=`pwd` modules
make[1]: Entering directory `/usr/src/kernels/2.6.18-92.el5-i686'
  CC [M]  /usr/src/kqemu-1.4.0pre1/kqemu-linux.o
cp /usr/src/kqemu-1.4.0pre1/kqemu-mod-i386.o /usr/src/kqemu-1.4.0pre1/kqemu-mod.o
  LD [M]  /usr/src/kqemu-1.4.0pre1/kqemu.o
Building modules, stage 2.
MODPOST
WARNING:      could      not      find      /usr/src/kqemu-1.4.0pre1/.kqemu-mod.o.cmd
for /usr/src/kqemu-1.4.0pre1/kqemu-mod.o
  CC      /usr/src/kqemu-1.4.0pre1/kqemu.mod.o
  LD [M]  /usr/src/kqemu-1.4.0pre1/kqemu.ko
make[1]: Leaving directory `/usr/src/kernels/2.6.18-92.el5-i686'
$ su -c'make install'
./install.sh

```

Этого достаточно для инсталляции модуля, но недостаточно для загрузки:

```

# lsmod | head -n3
Module                Size  Used by
mga                   62145  3
drm                   65493  4 mga
# modprobe kqemu
# lsmod | head -n3
Module                Size  Used by
kqemu                 131108  0

```

```
mga                62145  3
# chmod a+rw /dev/kqemu
# ls -l /dev/kqemu
crw-rw-rw- 1 root root 10, 62 Ноя  7 15:57 /dev/kqemu
```

- загрузили модуль (плюс проделали ещё некоторые манипуляции с его правами, которые стационарно легко уладить с помощью udev). Теперь загрузка виртуальной машины должна происходить без единого предупреждения:

```
$ qemu -m 8M -fda odin1440.img -boot a
...
```

Всё! Теперь вы готовы виртуально выполнять любую OS на любой процессорной платформе.

Примечание: при последнем показанном запуске можно получить сообщение:

```
Version mismatch between kqemu module and qemu (00010300 00010400) - disabling kqemu use
```

И это очевидно не то, чего мы добивались! Это связано с версиями qemu и kqemu, как сказано на сайте проекта QEMU они должны соответствовать друг другу:

<http://www.qemu.org/kqemu-1.3.0pre11.tar.gz> Use with QEMU <= 0.9.1. Full source code available under the GPL license.

<http://www.qemu.org/kqemu-1.4.0pre1.tar.gz> Use with the QEMU development version in the Subversion repository

А как было показано ранее, в начале этого раздела, с помощью yum мы может установить как версию qemu 0.9.x, так и 0.10.x.

Установка MINIX3

Загрузка с CD

Теперь у нас всё готово для запуска MINIX3 под QEMU. Скачиваем последний по версии образ загрузочного CD, на время написания этого текст это был http://www.minix3.org/download/minix_R3.1.5-r5612.iso.bz2 . Если образ сжат bzip2 (некоторые версии сжаты, другие — нет), то разархивируем его, возможно, предварительно проверив для контроля целостность архива:

```
$ bzip2 -t minix_R3.1.5-r5612.iso.bz2
$ bzip2 -d minix_R3.1.5-r5612.iso.bz2
$ ls -l
-rw-rw-r-- 1 olej olej 640679936 Ноя  6 10:25 minix_R3.1.5-r5612.iso
```

Вот теперь перед нами ISO-образ последнего релиза OS MINIX3. Накатаем (любой программой для того предназначенной, например K3b в Linux) этот образ на CD. Это LiveCD, поэтому мы можем грузить OS непосредственно с него, или устанавливая на носитель. Сначала проделаем только загрузку:

```
$ qemu -m 100M -cdrom minix_R3.1.5-r5612.iso -boot d
```

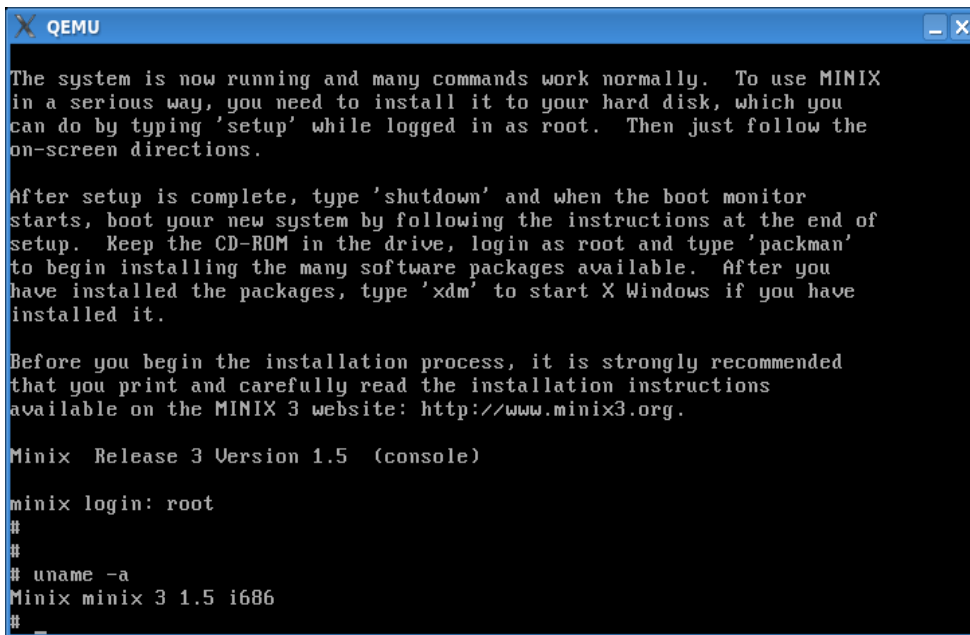
Точно того же результата (внимание!) мы достигаем и используя записанный ранее LiveCD, командой:

```
$ qemu -m 100M -cdrom /dev/cdrom -boot d
```

или

```
$ qemu -m 100M -cdrom /dev/hda -boot d
```

Примечание: это у меня CD-ROM /dev/hda (именно так!) у вас он, скорее всего, будет под другой литерой.



```
QEMU
The system is now running and many commands work normally. To use MINIX
in a serious way, you need to install it to your hard disk, which you
can do by typing 'setup' while logged in as root. Then just follow the
on-screen directions.

After setup is complete, type 'shutdown' and when the boot monitor
starts, boot your new system by following the instructions at the end of
setup. Keep the CD-ROM in the drive, login as root and type 'packman'
to begin installing the many software packages available. After you
have installed the packages, type 'xdm' to start X Windows if you have
installed it.

Before you begin the installation process, it is strongly recommended
that you print and carefully read the installation instructions
available on the MINIX 3 website: http://www.minix3.org.

Minix Release 3 Version 1.5 (console)

minix login: root
#
#
# uname -a
Minix minix 3 1.5 i686
#
```

Рис.2.1

Управление консолями в QEMU

Только что мы установили 1-й экземпляр VM MINIX3. Здесь следует сразу же коротко остановиться на том, как управлять (горячие клавиши) консолями в QEMU. Это замечание будет относиться ко всем последующим установкам, обсуждаемым дальше, и к любым вашим установкам.

Прежде всего нам нужно уметь переключаться между консолями установленного MINIX3, как вы помните, их по умолчанию 4, но это число может быть легко увеличено. Для переключения консолей MINIX3 нужно:

1. чтобы ввод был захвачен окном QEMU – для этого щёлкните левой кнопкой мышки на поле окна терминала QEMU...
2. используйте комбинации: [правый Alt]+[Fn], где n – 1, 2, 3, 4 – номер консоли MINIX3...
3. или используйте [правый Alt]+[>-] и [правый Alt]+[<-] (стрелки) для перемещения к следующей и предыдущей консоли, соответственно.

Далее, гостевая система, работающая в окне QEMU (например, рис.1.1), может захватывать клавиатуру, а главное, мышь для своей работы, о чём и как было сказано выше. При этом у пользователя может возникнуть ложное впечатление, что система (Linux) в которой выполняется qemu — зависла (курсор мыши исчез), но это не так: органы управления просто принадлежат в данный момент (захвачены) виртуальной операционной системой. Чтобы их освободить нажмите [Ctrl]+[Alt].

Клавиша [правый Alt] в терминальной системе MINIX3 переключает раскладку клавиатуры. Если при переключении консолей вы нажали эту клавишу, но затем передумали нажимать [Fn], то консоль у вас окажется переключенной в альтернативную раскладку: если вы установили русский шрифт, то вам повезло, и вы видите русские литеры, если нет – видите «краказябры»; в том и другом случае вы не можете дальше вводить команды (и даже просто забить введенные символы не можете). Для того, чтобы вернуть раскладку – снова нажмите «одинокий» [правый Alt].

Помимо отображения консолей гостевой OS, в нашем случае MINIX3, QEMU имеет ещё 3 собственных консоли, которые вызываются:

[Ctrl]+[Alt]+[2] – консоль управления QEMU (рис.2.2): позволяет выполнять множество команд-операций, например, подключать виртуальные диски динамически, и многое другое; одна из команд help (вывод её показан на рис.2.2) выводит всю справочную информацию по командам консоли;

[Ctrl]+[Alt]+[3] – консоль терминала на последовательном интерфейсе serial0;

[Ctrl]+[Alt]+[4] – консоль терминала на последовательном интерфейсе parallel0;

Вернуться из дополнительных консолей к отображению консолей гостевой OS можно по [Ctrl]+[Alt]+[1].

Детальное описание управляющих консолей исчерпывающе описано в документации QEMU (источники указаны в конце текста), и выходит далеко за рамки нашего рассмотрения.


```

X QEMU
log item[,...] -- activate logging of the specified items to '/tmp/qemu.log'
savevm taglid -- save a VM snapshot. If no tag or id are provided, a new snapshot is crea
loadvm taglid -- restore a VM snapshot from its tag or id
delvm taglid -- delete a VM snapshot from its tag or id
stop -- stop emulation
c/cont -- resume emulation
gdbserver [port] -- start gdbserver session (default port=1234)
x /fmt addr -- virtual memory dump starting at 'addr'
xp /fmt addr -- physical memory dump starting at 'addr'
plprint /fmt expr -- print expression value (use $reg for CPU register access)
i /fmt addr -- I/O port read
sendkey keys -- send keys to the VM (e.g. 'sendkey ctrl-alt-f1')
system_reset -- reset the system
system_powerdown -- send system power down event
sum addr size -- compute the checksum of a memory region
usb_add device -- add USB device (e.g. 'host:bus.addr' or 'host:vendor_id:product_id')
usb_del device -- remove USB device 'bus.addr'
cpu index -- set the default CPU
mouse_move dx dy [dzt] -- send mouse move events
mouse_button state -- change mouse button state (1=L, 2=M, 4=R)
mouse_set index -- set which mouse device receives events
wavcapture path [frequency bits channels] -- capture audio to a wave file (default freque
stopcapture capture index -- stop capture
menseave addr size file -- save to disk virtual memory dump starting at 'addr' of size 'si
(qemu) help

```

Рис.2.2

Установка MINIX3 на диск

Для установки на виртуальный диск нужно прежде создать файл, который будет имитировать этот виртуальный диск. Поскольку это будет рабочий диск MINIX3 на достаточно долгое время, то его размер должен быть весьма значительным (от нескольких десятков Mb до Gb), поэтому прежде есть смысл поинтересоваться наличием свободного места на дисковом пространстве Linux:

```
$ df
```

Файловая система	1K-блоков	Исп	Доступно	Исп%	смонтирована на
/dev/hdf6	7640636	6842276	403968	95%	/
tmpfs	143956	0	143956	0%	/dev/shm
/dev/hde1	3775880	2491336	1284544	66%	/mnt/win_c
/dev/hdf4	7151004	3426472	3724532	48%	/mnt/win_d
/dev/hde5	6221016	1116	6219900	1%	/mnt/win_e

Я буду создавать виртуальный диск минимального размера, на который будет устанавливаться MINIX3, размером 90Mb (это близко к минимальному размеру 79Mb, который будет указывать этот релиз системы, остаток ~10Mb отдадим под /home), это годится для экспериментов, но вряд ли достаточно для рабочей системы:

```
$ qemu-img create minix3-disk 90M
```

```
Formatting 'minix3-disk', fmt=raw, size=92160 kB
```

```
$ ls -l minix3-*
```

```
-rw-r--r-- 1 olej olej 94371840 Ноя 7 18:54 minix3-disk
```

Хотя то же самое можно сделать и при помощи стандартной утилиты dd, для примерно того же размера, например, так:

```
$ dd of=minix3-disk1 bs=1024 seek=90000 count=0
```

```
...
```

```
$ ls -l minix3-*
```

```
-rw-r--r-- 1 olej olej 94371840 Ноя 8 20:01 minix3-disk
```

```
-rw-rw-r-- 1 olej olej 92160000 Ноя 11 10:36 minix3-disk1
```

Примечание: есть отличие этих двух способов; утилита dd позволяет создать только raw-образ, который представляет собой файл, заполненный нулями. Утилита qemu-img поддерживает несколько различных форматов образа, указать на которые можно при помощи параметра `format`. По умолчанию создаются qcow-файлы (qemu Copy On Write). Этот формат поддерживает шифрование (AES, 128 бит) и компрессию, но возможны еще:

raw-, cow- (User Mode Linux), vmdk- (VMWare) или cloop- (сжатый loop, обычно используемый на LiveCD). Многие предпочитают использовать raw. Этот формат не поддерживает сжатие, но если образ находится на разделе с файловой системой, поддерживающей дыры (holes), например ext2/3, то сжатие будет обеспечено самим драйвером файловой системы. И у этого способа есть еще один несомненный плюс – можно монтировать в дерево ФС и работать как с обычным дисковым разделом. Утилита `qemu-img` поддерживает параметр `convert`, позволяющий преобразовывать образы из одного формата в другой.

Двигаемся дальше: загружаем CD (в точности так, как это делалось ранее) и приступаем к установке на заготовленный виртуальный диск:

```
$ qemu -m 100M -hda minix3-disk -cdrom /dev/cdrom -boot d -localtime
```

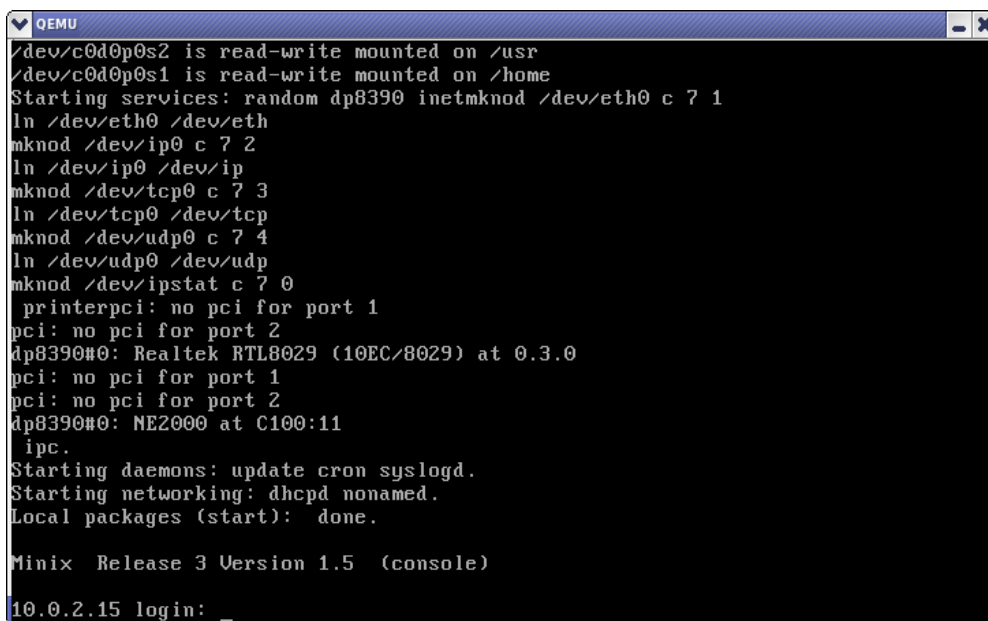
Дальше, вооружившись текстом по инсталляции: <http://www.minix3.ru/docs/setup-russian.pdf>, пошагово следуем этой инструкции установки на раздел реального диска, никаких принципиальных различий не встречаем: зарегистрировавшись в MINIX3 как `root` (без пароля)... поехали:

```
# setup
```

В этом месте можете спокойно пойти отдохнуть, выпить чашечку кофе — копирование файлов (копируются порядка 7700 файлов) в виртуальную файловую систему будет продолжаться ... (если вы ведёте установку с записанного CD) минут 20-40. После завершения установки производим загрузку только что установленной системы с виртуального диска (рис.2.3):

```
$ qemu -m 100M -hda minix3-disk -cdrom /dev/cdrom -boot c -localtime
```

```
...
```



```
QEMU
/dev/c0d0p0s2 is read-write mounted on /usr
/dev/c0d0p0s1 is read-write mounted on /home
Starting services: random dp8390 inetmknod /dev/eth0 c 7 1
ln /dev/eth0 /dev/eth
mknod /dev/ip0 c 7 2
ln /dev/ip0 /dev/ip
mknod /dev/tcp0 c 7 3
ln /dev/tcp0 /dev/tcp
mknod /dev/udp0 c 7 4
ln /dev/udp0 /dev/udp
mknod /dev/ipstat c 7 0
printerpci: no pci for port 1
pci: no pci for port 2
dp8390#0: Realtek RTL8029 (10EC/8029) at 0.3.0
pci: no pci for port 1
pci: no pci for port 2
dp8390#0: NE2000 at C100:11
ipc.
Starting daemons: update cron syslogd.
Starting networking: dhcpd nonamed.
Local packages (start): done.
Minix Release 3 Version 1.5 (console)
10.0.2.15 login:
```

Рис.2.3

И далее, наверное, вам захочется в MINIX3 установить интересующие вас пакеты; набираем в консоли:

```
# packman
```

```
...
```

Примечание: обратите внимание, что инсталлятору `packman` выбранные вами пакеты нужно указать не по именам, а по порядковым номерам, в которых он вам их (пакеты) предварительно перечисляет на экран.

Для наших дальнейших целей (для воспроизведения показанных далее иллюстраций) понадобятся, как минимум, пакеты:

```
openssh-4.3p2
```

```
openssl-0.9.8a
```

```
ncurses-5.5
```

```
X11R6.8.2
```

Инсталляция пакетов тоже может занять довольно продолжительное время.

MINIX3 реальный и виртуальный

На другом компьютере (это специально подчеркнуто, чтобы вы не соотносили имена разделов дисков с предыдущим изложением), на котором был уже ранее установлен «реальный» MINIX3 в раздел диска /dev/hda2 :

```
# fdisk /dev/hda
```

Устр-во	Загр	Начало	Конец	Блоки	Id	Система
/dev/hda1		1	511	4104576	b	W95 FAT32
/dev/hda2		512	956	3574462+	81	Minix / старый Linux
/dev/hda3	*	957	1723	6160927+	4f	QNX4.x 3-я часть
/dev/hda4		1724	2491	6168960	f	W95 расшир. (LBA)
/dev/hda5		1724	1787	514048+	82	Linux своп / Solaris
/dev/hda6		1788	2491	5654848+	83	Linux

- и запустим QEMU вот так:

```
$ qemu -m 50M -hda /dev/hda -boot c -localtime
```

```
qemu: could not open disk image /dev/hda2
```

Что, в общем, понятно (права доступа):

```
# ls -l /dev/hda
```

```
brw-r----- 1 root disk 3, 0 Ноя  8 08:54 /dev/hda
```

Но, поскольку очень уж не хочется выполнять qemu под правами root, то сделаем это так:

```
# chmod a+rw /dev/hda
```

И ещё раз выполняем:

```
$ qemu -m 50M -hda /dev/hda -boot c -localtime
```

...

В результате, мы проходим путь загрузки в qemu от загрузчика GRUB (рис.2.4) до загрузки (выбрав в меню «Minix 3») реально установленного (/dev/hda2) экземпляра MINIX3; на рис.2.5 показано выполнение установленного в составе «реального» MINIX3 MidnightComander, выполняющийся в окне qemu.

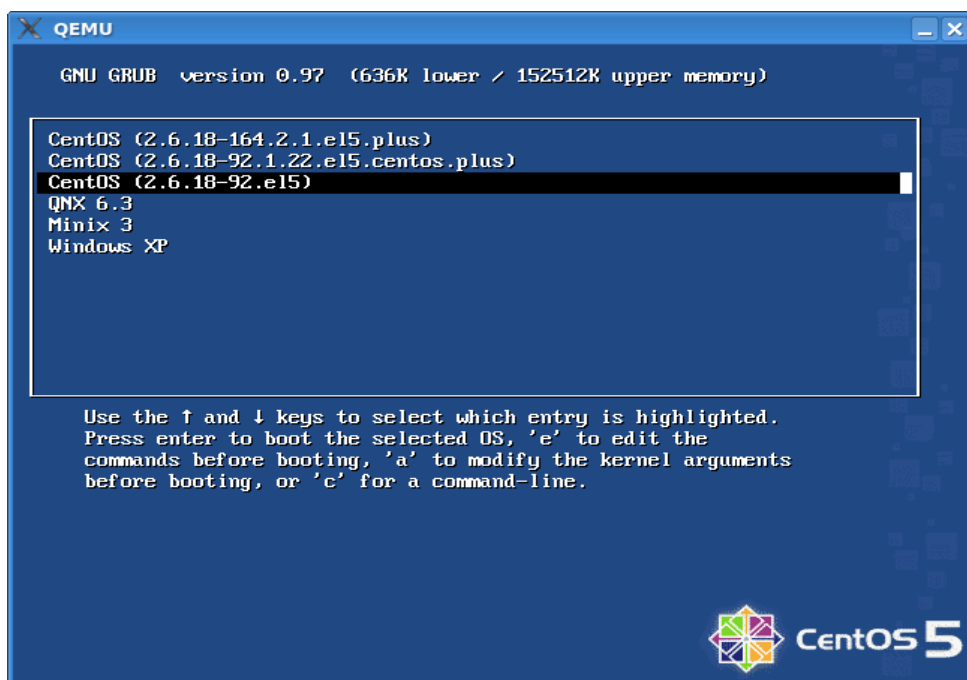


Рис.2.4

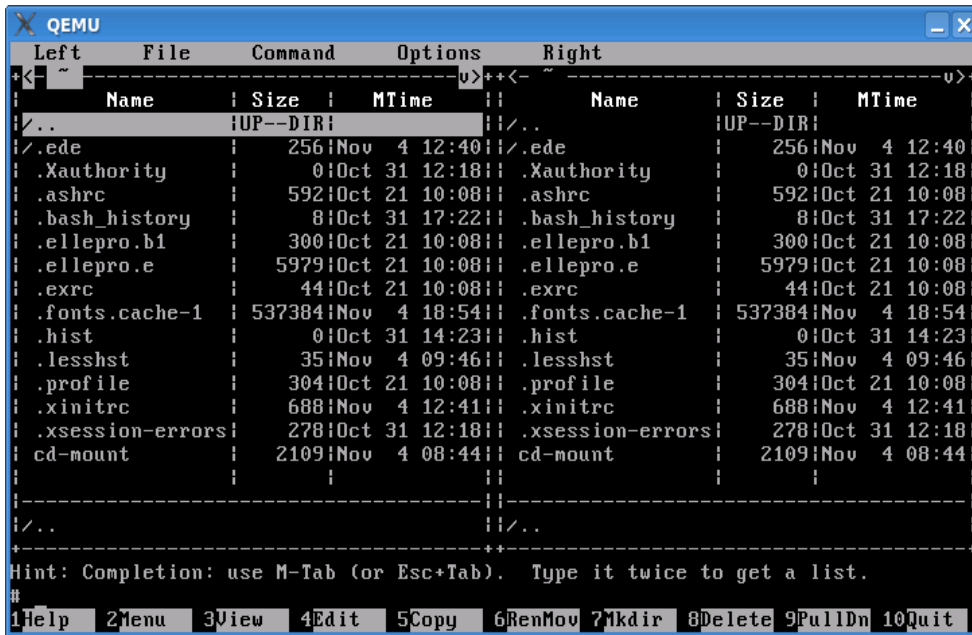


Рис.2.5

Зачем нужно такое «изращение»? Таким путём мы имеем возможность запускать **один и тот же экземпляр** ОС:

- либо как реальную систему, для устранения любых эффектов виртуализации, адекватной проверки характеристик и возможностей системы без всяких привносимых особенностей.
- другим случаем целесообразности загрузки системы как реальной является выполнение больших объёмов работ, особенно дисковых, реальная система быстрее виртуальной на порядок, и может даже не один.
- либо как виртуальную, но с дополнительными возможностями обмена данными с другими файловыми системами, например, MINIX3 очень капризен к устанавливаемым сетевым картам, но в таком варианте можно установить «виртуальное» сетевое соединение с базовой системой и по нему организовать обмен данными.

Примечание: используя показанную «дуальность» MINIX3 в разделе диска, я хочу обратить внимание на ещё одну возникающую возможность: быструю и комфортную установку (а особенно переустановку) MINIX3 в раздел:

1. загрузка с LiveCD и установка собственно системы в раздел /dev/hda2 :

```
# qemu -cdrom /dev/cdrom -hda /dev/hda -boot d
...
# setup
...
# shutdown now
```

2. перегрузка в той же конфигурации и установка пакетов:

```
# qemu -cdrom /dev/cdrom -hda /dev/hda -boot d
...
# packman
```

Этот же шаг установки (доустановки) пакетов можно выполнять с тем же успехом при «реальной» загрузке MINIX3 из раздела HDD; если предполагается устанавливать большой объём (или все) пакетов, что часто бывает при начальной установке, то рационально перегрузиться в реальную систему MINIX3: установка всех пакетов с LiveCD в реальной системе может произойти за 10-15 мин. (на быстрых процессорах и быстрее), в виртуально загруженной системе этот же процесс может растянуться и на 2-3 часа.

Оживляем сетку

В документации QEMU сказано:

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

Сеть к виртуальной ОС также предполагается как виртуальная, т.е. мы должны установить в этой виртуальной подсети соединение MINIX3 к базовому Linux, а оттуда уже, по необходимости, обеспечит роутинг в реальную LAN, или в наружу в Интернет.

Предостережение: убедитесь, что создаваемая (как описано далее) виртуальная подсеть не совпадает или перекрывается (по совокупности IP:маска) с существующими подсетями на Linux хосте – в противном случае вы потеряете множество времени, толкуя весьма странные эффекты в сетевой подсистеме.

В моём случае в Linux уже установлена достаточно развитая сетевая подсистема:

```
# ifconfig -a
eth0      Link encap:Ethernet  HWaddr 00:60:52:07:4F:4B
          inet addr:192.168.1.7  Bcast:192.168.1.7  Mask:255.255.255.248
          inet6 addr: fe80::260:52ff:fe07:4f4b/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:2337 errors:0 dropped:0 overruns:0 frame:0
          TX packets:2631 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:820615 (801.3 KiB)  TX bytes:355058 (346.7 KiB)
          Interrupt:10 Base address:0x4000

eth1      Link encap:Ethernet  HWaddr 00:4F:49:00:B0:F9
          BROADCAST MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 b)  TX bytes:0 (0.0 b)
          Interrupt:10 Base address:0xec00

...
```

Здесь видно 2 сетевых интерфейса: (eth0) подключенный к LAN 192.168.1.7:255.255.255.248, и другой (eth1) в данный момент не активный (во 2-й строке нет UP). Для того, чтобы быть уверенным в неискажаемости результатов, виртуальную подсеть к MINIX3 будем строить в совсем другой подсети IP адресов: 192.168.2.0:255.255.255.0. Для этого...

Со стороны Linux:

1. создаём в /etc 2 скрипта (старта и останова) для тунельного интерфейса, они имеют права исполнения и установленный SUID бит:

```
# ls -l /etc/qem*
-rwsr-sr-x 1 root root 78 Ноя 8 16:32 /etc/qemu-ifdown
-rwsr-sr-x 1 root root 116 Ноя 8 16:32 /etc/qemu-ifup
# cat /etc/qemu-ifup
#!/bin/sh
echo ----- tap up -----
sudo /sbin/ifconfig $1 192.168.2.6
# cat /etc/qemu-ifdown
#!/bin/sh
```

```
echo ----- tap down -----
```

```
#!/sbin/ifconfig $1 down
```

Скрипт остановки здесь реально ничего не делает ... но может быть, при необходимости, наполнен содержанием.

2. запускаем виртуальную машину так:

```
$ su -c'qemu -m 100M -hda minix3-disk -boot c -localtime -net nic,vlan=0 -net tap,vlan=0'
```

```
...
```

- здесь запуск происходит от имени root, что связано с необходимостью инициализации сетевого интерфейса (это крайне нежелательно, но как от этого избавиться, и возможно ли — это вы разберётесь при желании сами):

```
# ifconfig -a
```

```
...
```

```
tap0      Link encap:Ethernet  HWaddr 00:FF:F7:C0:85:56
          inet addr:192.168.2.6  Bcast:192.168.2.255  Mask:255.255.255.0
          inet6 addr: fe80::2ff:f7ff:fec0:8556/64  Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:82 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:500
          RX bytes:0 (0.0 b)  TX bytes:17858 (17.4 KiB)
```

Шлюз по умолчанию в LAN установлен через интерфейс eth0 (192.168.1.1):

```
# route
```

```
Kernel IP routing table
```

Destination	Gateway	Genmask	Flags	Metric	Ref	Use	Iface
192.168.1.0	*	255.255.255.248	U	0	0	0	eth0
192.168.2.0	*	255.255.255.0	U	0	0	0	tap0
169.254.0.0	*	255.255.0.0	U	0	0	0	eth0
default	192.168.1.1	0.0.0.0	UG	0	0	0	eth0

Со стороны MINIX3:

1. В /etc/inet.conf пропишем:

```
eth0 dp8390 0 { default; };
```

- qemu устанавливает до 4-х сетевых интерфейсов типа NE2000, а сетевая плата NE200 в MINIX3 находится в драйвере dp8390 (/usr/src/drivers/dp8390).

2. Сетевому интерфейсу (умалчиваемому) присвоим IP адрес (после экспериментов это следует поместить куда-то в стартовый скрипт, например, /etc/profile):

```
# ifconfig -I /dev/ip
```

```
ifconfig: /dev/ip: Host address not set
```

```
# ifconfig -I /dev/ip -h 192.168.2.4
```

```
# ifconfig -av
```

```
/dev/ip0: address 192.168.2.4 mtu 1500
```

После этого мы можем пинговать достижимость хоста Linux, как показано на рис.3.1.

Или, напротив, достижимость MINIX3 со стороны хоста Linux:

```
QEMU
supports multiple virtual terminals. Just use ALT+F1, F2, F3 and F4 to
navigate among them.

To get rid of this message, edit /etc/motd.

#
# ifconfig -a
# ls /dev/ip*
/dev/ip /dev/ip0 /dev/ipstat
# cat /dev/ipstat
cat: /dev/ipstat: I/O error
# ifconfig -h
option requires an argument -- h
Usage: ifconfig [-I ip-device] [-h ipaddr] [-n netmask] [-m mtu] [-ival]
# ifconfig -I /dev/ip0
ifconfig: /dev/ip0: Host address not set
# ifconfig -I /dev/eth0
ifconfig: Unable to get IP configuration: Illegal ioctl for device
# ifconfig -I /dev/ip0 -h 192.168.2.4
#
# ping 192.168.2.4
192.168.2.4 is alive
# ping 192.168.2.6
192.168.2.6 is alive
#
```

Рис.3.1

```
# ping 192.168.2.4
PING 192.168.2.4 (192.168.2.4) 56(84) bytes of data.
64 bytes from 192.168.2.4: icmp_seq=1 ttl=96 time=406 ms
64 bytes from 192.168.2.4: icmp_seq=2 ttl=96 time=81.6 ms
64 bytes from 192.168.2.4: icmp_seq=3 ttl=96 time=114 ms
--- 192.168.2.4 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2000ms
rtt min/avg/max/mdev = 81.691/201.049/406.757/146.080 ms
```

```
QEMU
# Usage: rarpd [-d[level]] network-name ...
# ping 192.168.1.1
write: Destination not reachable
[1] Done (1) rarpd
# ping 192.168.1.1
write: Destination not reachable
# irdpd
^C
# irdpd &
# ping 192.168.1.1
write: Destination not reachable
# add_route -g 192.168.2.4 -d 0.0.0.0 -m 1 -n 0.0.0.0 -I -v
add_route: unable to open('-v'): No such file or directory
# add_route -g 192.168.2.4 -d 0.0.0.0 -m 1 -n 0.0.0.0 -v
adding output route to 0.0.0.0 with netmask 0.0.0.0 using gateway 192.168.2.4
# ping 192.168.1.1
192.168.1.1 is alive
# pr_routes -a
ent # if dest gateway dist pref mtu flags
0 ip0 0.0.0.0/0 192.168.2.4 1 0 0 static
# ping qnx.org.ru
ping: unknown host (qnx.org.ru)
# ping 72.249.144.181
72.249.144.181 is alive
#
```

Рис.3.2

Но это ещё не даёт нам возможность доступа к шлюзу (192.168.1.1) в LAN и далее. Причина этого – отсутствие роутинга. Всё, что необходимо для установления роутинга (add_route и др.), показано снимком экрана MINIX3 на рис.3.2.

Примечание: показанный для образца ping qnx.org.ru в Интернет — недостижим, поскольку в нашем MINIX3 не работает (не настроена?) служба разрешения имён (DNS и др.); но если мы определим IP адрес тестируемого хоста вручную:

```
$ nslookup qnx.org.ru
Server: 195.5.51.182
```

```
Address: 195.5.51.182#53
Non-authoritative answer:
Name: qnx.org.ru
Address: 72.249.144.181
```

- то непосредственно по IP хост в Интернет достижим, что и показывает последняя команда на рис.3.2 (после установления роутинга).

Удалённый доступ

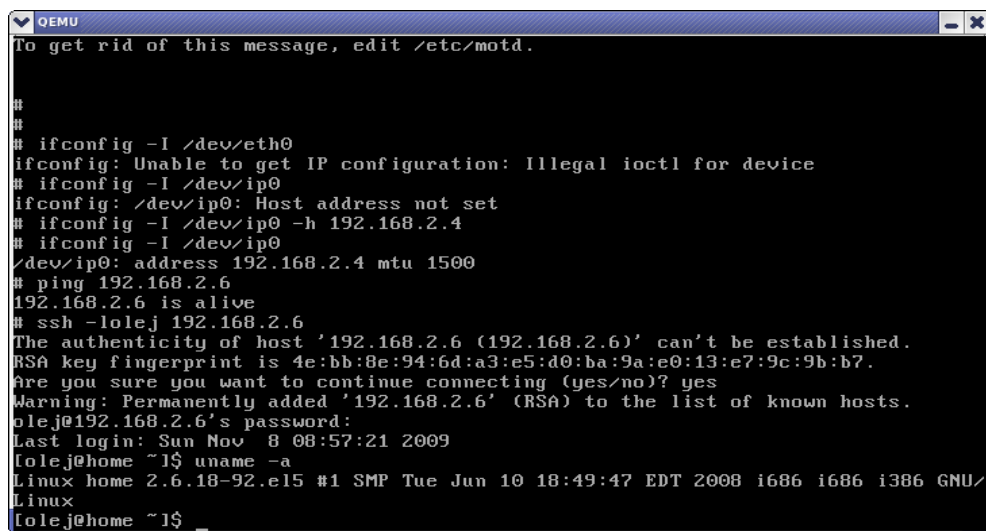
Работать с консоли MINIX3 занятие ... не комфортное и утомительное. Но, если у нас есть теперь сеть из «комфортного» Linux в «не комфортный» виртуальный MINIX3, то мы можем просто всю работу в ОС MINIX3 производить из терминалов (сколько угодно много по числу) Linux.

1. Доступ к текстовой консоли по SSH.

```
$ ssh -luser 192.168.2.4
Password: ...
```

Примечание: к этому месту как раз вовремя или назначить (изменить) пароль для root (при установке MINIX3 root остался с пустым паролем) командой `passwd`; либо добавить нового пользователя MINIX3 командой `adduser` — `ssh` не допускает обращений пользователей без паролей (хотя, возможно, этого и можно добиться настройками `sshd`?).

2. Симметричным образом можно и со стороны MINIX3 получить SSH-доступ к Linux:



```
QEMU
To get rid of this message, edit /etc/motd.

#
#
# ifconfig -I /dev/eth0
ifconfig: Unable to get IP configuration: Illegal ioctl for device
# ifconfig -I /dev/ip0
ifconfig: /dev/ip0: Host address not set
# ifconfig -I /dev/ip0 -h 192.168.2.4
# ifconfig -I /dev/ip0
/dev/ip0: address 192.168.2.4 mtu 1500
# ping 192.168.2.6
192.168.2.6 is alive
# ssh -lolej 192.168.2.6
The authenticity of host '192.168.2.6 (192.168.2.6)' can't be established.
RSA key fingerprint is 4e:bb:8e:94:6d:a3:e5:d0:ba:9a:e0:13:e7:9c:9b:b7.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '192.168.2.6' (RSA) to the list of known hosts.
olej@192.168.2.6's password:
Last login: Sun Nov 8 08:57:21 2009
[olej@home ~]# uname -a
Linux home 2.6.18-92.e15 #1 SMP Tue Jun 10 18:49:47 EDT 2008 i686 i686 i386 GNU/
Linux
[olej@home ~]#
```

Рис.3.3

3. Доступ из одной из панелей mc в Linux по SSH к хосту MINIX3.

Это достигается, например, следующей последовательностью действий-меню: F9 -> «Левая панель» -> «Shell соединение», а на предложение ввести имя компьютера вводим: `user@192.168.2.4` (это именно в точности соответствует SSH-команде, показанной выше — здесь другой синтаксис).

Теперь мы имеем то, что показано на рис.3.4, где в левой панели — хост MINIX3 (192.168.2.4), в правой — Linux (192.168.2.6), и мы привычными «менюшными» операциями `mc` по F3 / F4 / F5 / F6 / F7 / F8 - оперируем с файлами на любом из хостов и гоняем их туда-сюда.

Примечание: операция копирования с шифрованием потока по SSH — достаточно ресурсоёмкая операция (если ещё и учесть, что на одном конце она выполняется в виртуальной машине), показанная на рис.3.4 операция копирования дистрибутивного пакета `mc` на MINIX3 занимала у меня 1-2 минуты.

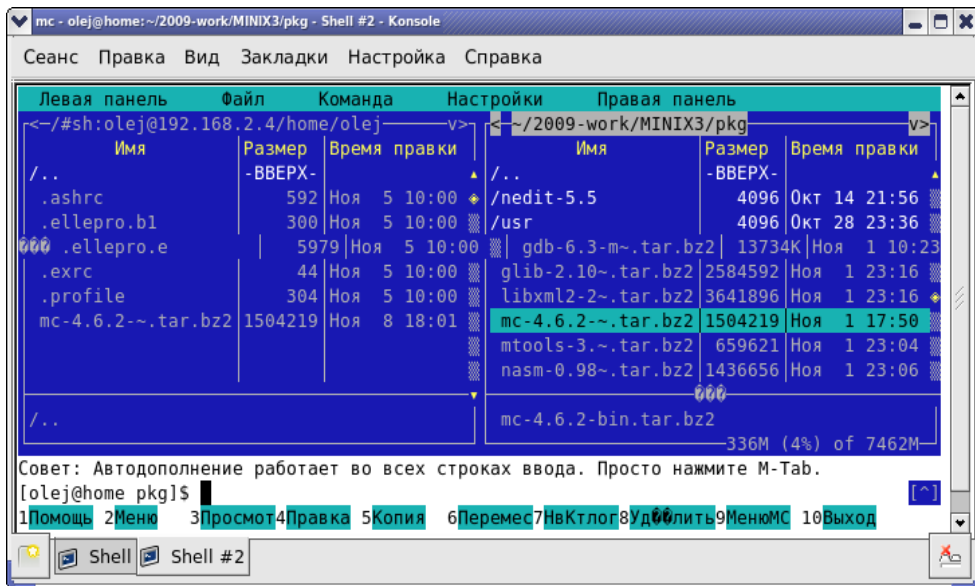


Рис.3.4

QEMU & X

Установка, настройка и использование графической X системы в MINIX3 – тема, выходящая далеко за рамки рассмотрения MINIX3 под QEMU, по которой (особенно настройке Xorg) имеется великое множество ресурсов в Интернет. Здесь я только бегло упомяну некоторые особенности X, которые особенно касаются работы под QEMU.

Прежде всего то, что часто упускается из виду: X система – сугубо сетевая система, X протокол – сетевой протокол; без корректной установки сети в MINIX3, хотя бы в отношении единственно локального хоста – X система работать не станет!

Установка X

Установка X не имеет каких-то выдающихся особенностей, отличающих её установку от других пакетов с дистрибутивного LiveCD, за исключением, пожалуй, размера: бинарный пакет X11R6 содержит более 10 тыс. файлов – это может, при некоторых обстоятельствах, исчерпать таблицу i-узлов дискового раздела. Кроме того, мне хотелось бы установить минимальную систему с X, включая минимальные сетевые средства (SSL/SSH), установить эти пакеты в бинарном виде (исходные тексты не устанавливать), с тем, чтобы использовать этот файл (виртуальный диск) в дальнейших сетевых экспериментах. В связи с этим, я далее показываю **полную** установку системы и пакетов «с нуля»; кроме того, это будет показано на другом (гораздо более быстром в связи с объёмами работы) компьютере, с отличающимися IP подсетей (не соотносите с предыдущим описанием!). Итак...

Создаём виртуальный диск размером ~300Mb (далее станет видно, что меньше – мало, а больше – это переизбыток, таким образом это минимальный размер, в который можно установить MINIX3 + X, и то с некоторыми ухищрениями, о которых дальше...):

```
$ time dd if=/dev/zero of=minix3-300 bs=1k count=300k
307200+0 записей считано
307200+0 записей написано
   скопировано 314572800 байт (315 MB), 2,25056 секунд, 140 MB/s
real    0m2.351s
user    0m0.168s
sys     0m1.948s
```

Обратите внимание, что диск создаётся командой dd в другом синтаксисе, чем ранее ... для разнообразия, но это по-другому работает.

Примечание: времена в этом разделе, где они будут показаны, относятся к DualCore компьютеру 2x1.6Ghz.

Загрузка с LiveCD и установка MINIX3 на виртуальный HDD:

```
$ qemu -hda minix3-300 -cdrom minix_R3.1.5-r5612.iso -boot d -localtime -kernel-kqemu
```

```
QEMU
Unless you are an expert, you are advised to use the automated
step-by-step help in setting up.

Press ENTER for automatic mode, or type 'expert':

--- Substep 3.1: Select a disk to install MINIX 3 -----
Probing for disks. This may take a short while..... Probing done.
The following disk was found on your system:

Disk [0]: /dev/c0d0, 299 MB
Free space (299 MB)

Enter the disk number to use: [0] 0

--- Substep 3.2: Select a disk region -----
Please select the region that you want to use for the MINIX 3 setup.
If you select an in-use region it will be overwritten by MINIX. The
following region were found on the selected disk:

[0] Free space (299 MB)

Enter the region number to use or type 'delete': [0] 0_
```

Рис.4.1

После загрузки:

```
# setup
```

При определении HDD я даже не стал входить в режим «expert» – всё происходит автоматически.

Вот здесь – важнейший трюк (который я выше упомянул как «ухищрение»): вместо умолчания размера блока 4Кб, я определяю 1Кб. Если этого не сделать, вы не сможете установить X в HDD объёмом менее ~1.2Gb!

```
QEMU
This is the point of no return. You have selected to install MINIX 3
into region 0 of disk /dev/c0d0. Please confirm that you want
to use this selection to install MINIX 3.

Are you sure you want to continue? Please enter 'yes' or 'no': yes

--- Step 4: Reinstall choice -----
No old /home found. Doing full install.

--- Step 5: Select the size of /home -----
MINIX will take up 79 MB, without /home.
How big do you want your /home to be in MB (0-219) ? [43] 4
4 MB Ok? [Y] y

--- Step 6: Select a block size -----
The default file system block size is 4 KB.
Block size in kilobytes? [4] 1_
```

Рис.4.2

Конфигурируем сеть. Если мы «угадаем» параметры здесь, на этапе установки системы, то нам не придётся настраивать сеть вручную, как описывалось выше. Виртуальную сеть будем строить на IP=192.168.3.4 со стороны гостевого MINIX3, и IP=192.168.3.6 со стороны базового Linux.

```

X QEMU
by MINIX are marked with *. Please choose:

0. No Ethernet card (no networking)
1. Intel Pro/100
2. 3Com 501 or 3Com 509 based card
3. Realtek 8139 based card (also emulated by KVM)
4. * Realtek 8029 based card (also emulated by Qemu)
5. NE2000, 3com 503 or WD based card (also emulated by Bochs)
6. AMD LANCE (also emulated by VMWare and VirtualBox)
7. Different Ethernet card (no networking)

Ethernet card? [4]

Configure network using DHCP or manually?

1. Automatically using DHCP
2. Manually

Configure method? [1] 2
Hostname [minix1]: qemu-minix
IP address [10.0.0.1]: 192.168.3.4
Netmask [255.255.255.0]:
Gateway: 192.168.3.6
Primary DNS Server [1]: 192.168.9.2
Secondary DNS Server [1]: 192.168.9.254

```

Рис.4.3

Всё, установка с LiveCD закончилась, можно переходить к установке пакетов, для этого перегружаемся с HDD:

```

$ qemu -hda minix3-300 -cdrom minix_R3.1.5-r5612.iso -boot c -localtime -kernel-kqemu
...

```

После регистрации как root:

```
# packman
```

- и устанавливаем пакеты X11R6 (#113) и SSL/SSH (#71/72, на рисунках не показаны).

```

X QEMU
11 bcrypt-1.1      Cross platform file encryption utility (134 kB)
12 binutils-2.16.1 A collection of GNU binary tools (15024 kB)
13 bison-2.1       Parser generator (requires gnu m4 in m4 package) (2144 kB)
14 catdoc-0.94.2   view various file types such as ms word in text (400 kB)
15 CSSC-1.0.1     SCCS Version Control Software Clone (6224 kB)
16 cvs-1.11.21    Concurrent versioning system (640 kB)
17 diffutils-2.8.1 GNU diff and friends (272 kB)
18 dungeon-2.7.1   Text adventure dungeon exploration game (336 kB)
19 ede-1.1         Equinox Desktop Environment 1.1 (needs EFLTK) (38704 kB)
20 efltk-2.0.6     EFLTK - Extended Fast Light Toolkit 2.0.6 (7072 kB)
21 emacs-21.4     The EMACS editor (25504 kB)
22 exim-4.66      Exim - MTA (overwrites aliases, sendmail, mailq) (5808 kB)
23 fb             file browser (36 kB)
Format examples: '3', '3,6', '3-9', '3-9,11-15', 'all'
Package(s) to install (RETURN or q to exit)? 113
Get source(s) too? (y/N) N

Installing from /mnt/install/packages/X11R6.8.2.tar.bz2 ..
Setting size of /usr/X11R6/bin/X to 8000000
/usr/X11R6/bin/X: Stack+malloc area changed from 15728640 to 8000000 bytes.
Installed ok.

Showing you a list of packages using more. Press q when
you want to leave the list.
Press RETURN to continue..

```

Рис.4.4

Посмотрим состояние файловой системы (особо /usr) после установки всех нужных нам пакетов (рис.4.5).

```

#
# df
Filesystem      Size (kB)    Free      Used    % Files%  Mounted on
/dev/c0d0p0s0   16384        11740     4644    29%    27%      /
/dev/c0d0p0s2   286424      29116     257308  90%    45%     /usr
/dev/c0d0p0s1    4096        3941      155     4%     1%     /home
/dev/c0d2p2      614400      92244     522156  85%    24%     /mnt
# time fsck /dev/c0d0p0s2

Checking zone map
Checking inode map
Checking inode list

blocksize = 1024      zonesize = 1024

 15856 Regular files
   734 Directories
    0 Block special files
    0 Character special files
 22886 Free inodes
    0 Named pipes
  1452 Symbolic links
 29116 Free zones
 3:06.86 real      4.93 user      3:01.88 sys
#

```

Рис.4.5

Занято 90% дискового пространства /usr и 45% таблицы i-узлов. Обратите внимание на время, которое 2x1.6Ghz компьютер «разгребал» сильно заполненную файловую систему MINIX3.

После этого:

```

# shutdown
> off

```

К этому времени у нас ещё нет сети: она предварительно настроена со стороны MINIX3, но нет тунеля со стороны Linux. Содержимое файла /etc/qemu-ifup:

```

$ cat qemu-ifup
#!/bin/sh
echo ----- tap up -----
sudo /sbin/ifconfig $1 192.168.3.6

```

Запускаем систему с поддержкой сети (обратите внимание на #):

```

# qemu -hda minix3-300 -cdrom minix_R3.1.5-r5612.iso -boot c -localtime \
-kernel-kqemu -net nic,vlan=0 -net tap,vlan=0
----- tap up -----
...

```

```

# QEMU
ipc.
Starting daemons: update cron syslogd.
nonamedLocal packages (start): sshd Generating SSH1 RSA host key: Ok
Generating SSH2 RSA host key: Ok
Generating SSH2 DSA host key: Ok
done.

Minix Release 3 Version 1.5 (console)
qemu-minix login: root

To install X Windows, run 'packman' with the install CD still in the
drive. To start X Windows after you have installed it, login as root
and type: 'xdm'. For more information about configuring X Windows, see
www.minix3.org.

If you do not have sufficient memory to run X Windows, standard MINIX 3
supports multiple virtual terminals. Just use ALT+F1, F2, F3 and F4 to
navigate among them.

To get rid of this message, edit /etc/motd.

# ping 192.168.3.6
192.168.3.6 is alive
#

```

Рис.4.6

Первая проба: ping. Хорошо видно запуск демонов и генерацию ключей SSH при старте.

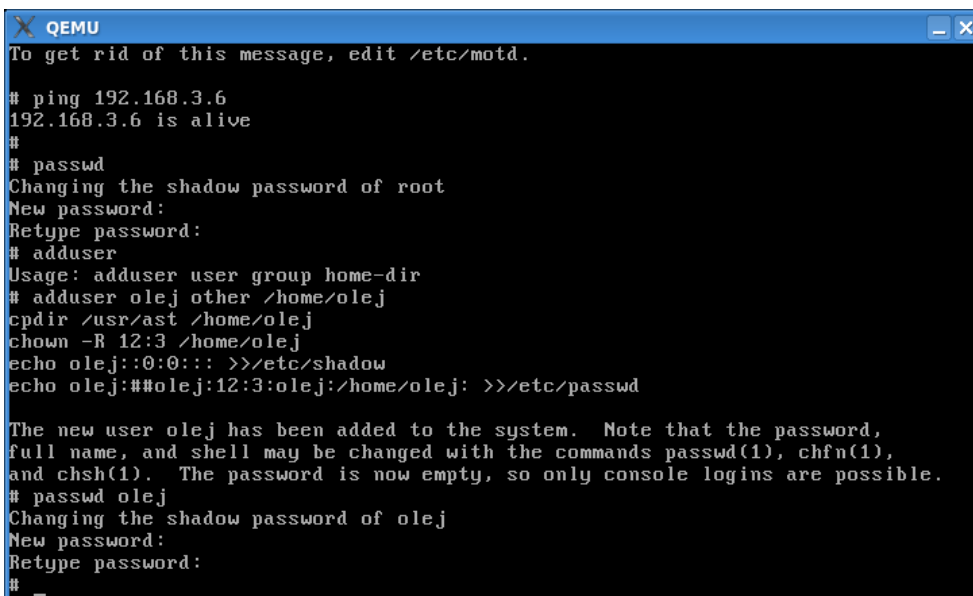
Проверки сети с встречной стороны:

```
# ifconfig tap0
```

```
tap0      Link encap:Ethernet  HWaddr F2:6E:E6:E8:BB:89
          inet addr:192.168.3.6  Bcast:192.168.3.255  Mask:255.255.255.0
          inet6 addr: fe80::f06e:e6ff:fee8:bb89/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:7 errors:0 dropped:0 overruns:0 frame:0
          TX packets:87 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:500
          RX bytes:420 (420.0 b)  TX bytes:14932 (14.5 KiB)
```

```
$ ping 192.168.3.4
```

```
PING 192.168.3.4 (192.168.3.4) 56(84) bytes of data.
64 bytes from 192.168.3.4: icmp_seq=1 ttl=96 time=15.2 ms
64 bytes from 192.168.3.4: icmp_seq=2 ttl=96 time=6.66 ms
64 bytes from 192.168.3.4: icmp_seq=3 ttl=96 time=6.60 ms
--- 192.168.3.4 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2000ms
rtt min/avg/max/mdev = 6.604/9.518/15.281/4.075 ms
```



```
QEMU
To get rid of this message, edit /etc/motd.

# ping 192.168.3.6
192.168.3.6 is alive
#
# passwd
Changing the shadow password of root
New password:
Retype password:
# adduser
Usage: adduser user group home-dir
# adduser olej other /home/olej
cpdir /usr/ast /home/olej
chown -R 12:3 /home/olej
echo olej::0:0:: >>/etc/shadow
echo olej:##olej:12:3:olej:/home/olej: >>/etc/passwd

The new user olej has been added to the system. Note that the password,
full name, and shell may be changed with the commands passwd(1), chfn(1),
and chsh(1). The password is now empty, so only console logins are possible.
# passwd olej
Changing the shadow password of olej
New password:
Retype password:
#
```

Рис.4.7

Дальше нельзя жить с root без пароля, и без дополнительных пользователей (SSH не позволит).

```
$ ssh -l root 192.168.3.4
```

Подобное мы уже видели, но здесь отчётливей: из Linux в MINIX3 как root (рис.4.8).

Замечательной особенностью показанного выше экрана есть то, что это консоль MINIX3, куда (и откуда) теперь можно произвольно копировать тексты команд и результаты их выполнения. Вот образцы выполнения нескольких команд консоли MINIX3, которые ранее просто никаким иным способом (кроме графического скриншота) невозможно было бы вкопировать в этот текст:

```
# uname -a
```

```
Minix qemu-minix 3 1.5 i686
```

```
# df
```

Filesystem	Size (kB)	Free	Used	% Files%	Mounted on
/dev/c0d0p0s0	16384	11204	5180	32% 28%	/

```

/dev/c0d0p0s2      286424      27933      258491  91%   45%   /usr
/dev/c0d0p0s1      4096        3928        168    5%    2%    /home

```

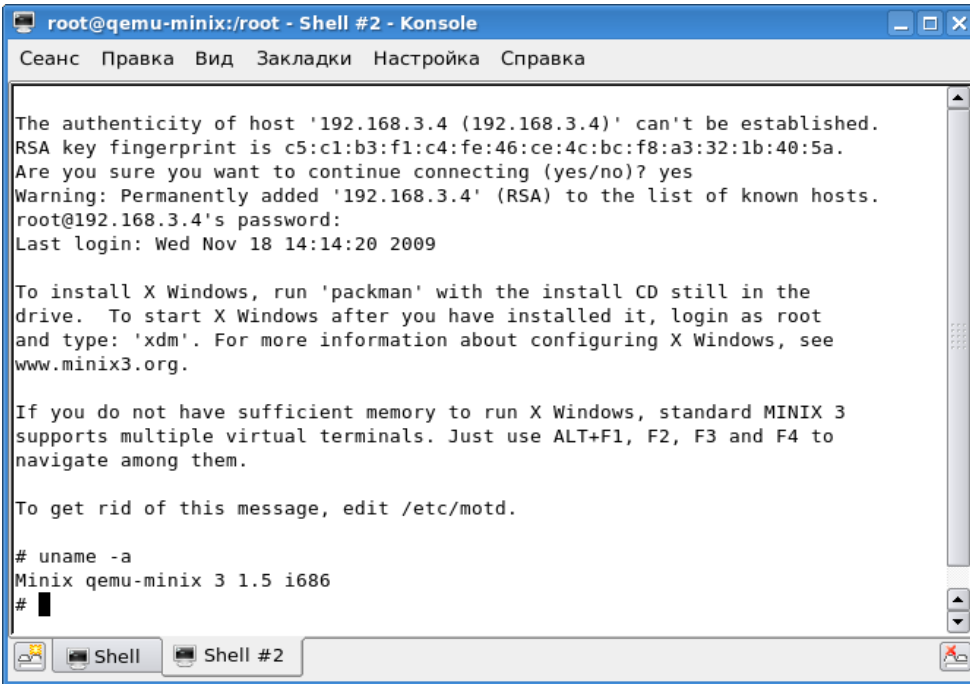


Рис.4.8

И ещё в пользу такого решения нужно отметить, что создать таких «терминалов MINIX3» на экране своей рабочей станции можно сколь угодно много, а переключаться между ними — с большой лёгкостью, чего отнюдь не скажешь о способе переключения между ограниченным числом консолей MINIX3.

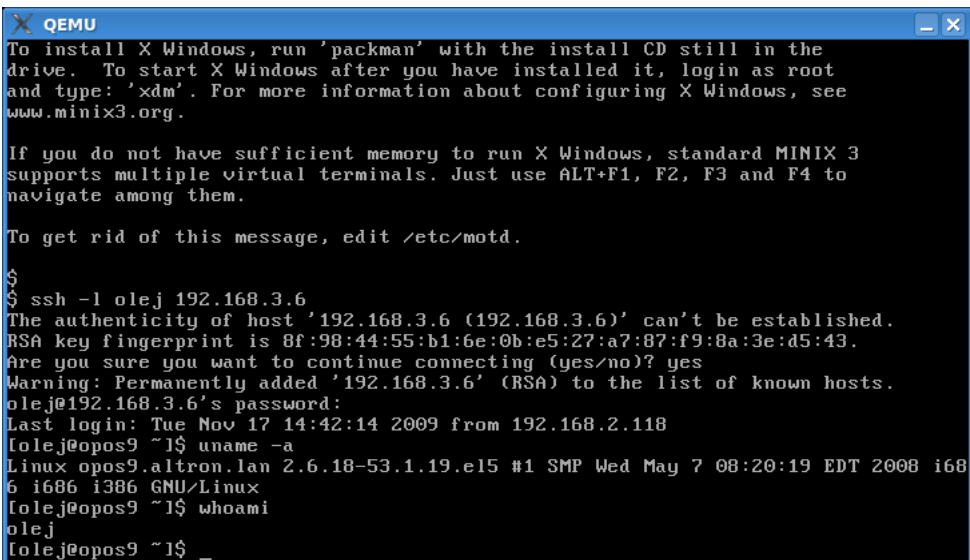


Рис.4.9

Из MINIX3 в Linux как user...

Удалённый доступ к X

А теперь мы проделаем удалённый доступ к X в MINIX3 из графического окружения базового Linux. При этом X-приложения MINIX3 выполняются, осуществляя графический ввод-вывод на X-сервер работающий в Linux.

Первый способ основан на «прямом» использовании сетевого протокола X. Для этого нужно предварительно обеспечить ряд условий со стороны X сервера:

1. Разрешить на Linux X-сервере доступ от удалённого хоста (или вообще от всех хостов), дальше показана проверка того, что это состоялось:

```
$ xhost +192.168.3.4
```

192.168.3.4 being added to access control list

```
$ xhost
```

access control enabled, only authorized clients can connect

```
INET:192.168.3.4
```

```
SI:localuser:olej
```

2. Проверит, что X-сервер запущен с разрешённым доступом TCP, например так:

```
$ ps ahx | grep Xorg
```

```
329 pts/10 S+ 0:00 grep Xorg
```

```
23476 tty7 Ss+ 84:49 /usr/bin/Xorg :0 -br -audit 0 -auth /var/gdm/:0.Xauth vt7
```

Если TCP доступ запрещён (а так часто и бывает после инсталляции Linux), то последняя строка (запуска Xorg) будет выглядеть подобно:

```
... /usr/bin/Xorg :0 -br -audit 0 -auth /var/gdm/:0.Xauth -nolisten tcp vt7
```

Тогда это нужно изменить. Например, можно воспользоваться менеджером:

```
# gdmsetup
```

- установить разрешение TCP доступа, и перезапустить X систему:

```
# gdm-restart
```

(не стоит здесь пугаться, что рабочая сессия здесь закроется, и будет запущена новая, начиная с login).

3. Войти в удалённую систему ... тем же SSH, к примеру:

```
$ ssh -l root 192.168.3.4
```

```
root@192.168.3.4's password:
```

```
Last login: Wed Nov 18 15:17:15 2009
```

4. Запустить требуемое GUI приложение в этой удалённой системе (в точности то же самое можно сделать и с консоли MINIX3 в QEMU):

```
# xclock -display 192.168.2.108:0.0
```

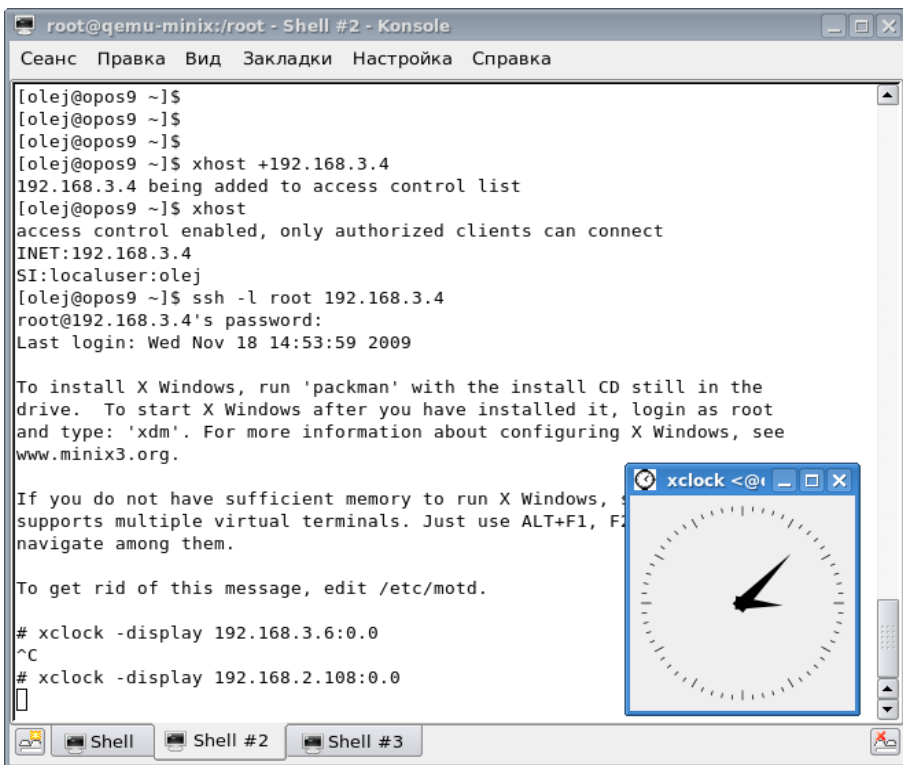


Рис.4.10

Запуск X приложения, используя «нативный» сетевой X протокол.

Второй способ по принципам работы – совершенно другой, использует возможности SSH-протокола. Но он позволяет получить этот же результат. В этом случае работа идёт медленнее, т.к. весь трафик шифруется SSH, но зато соединение – защищено.

Особое внимание обратите на то, что **оба** описанных способа удалённого доступа к X – ни коим образом не используют особенности QEMU, и в **равной степени применимы** при работе как в виртуально запущенном MINIX3, так и реально.

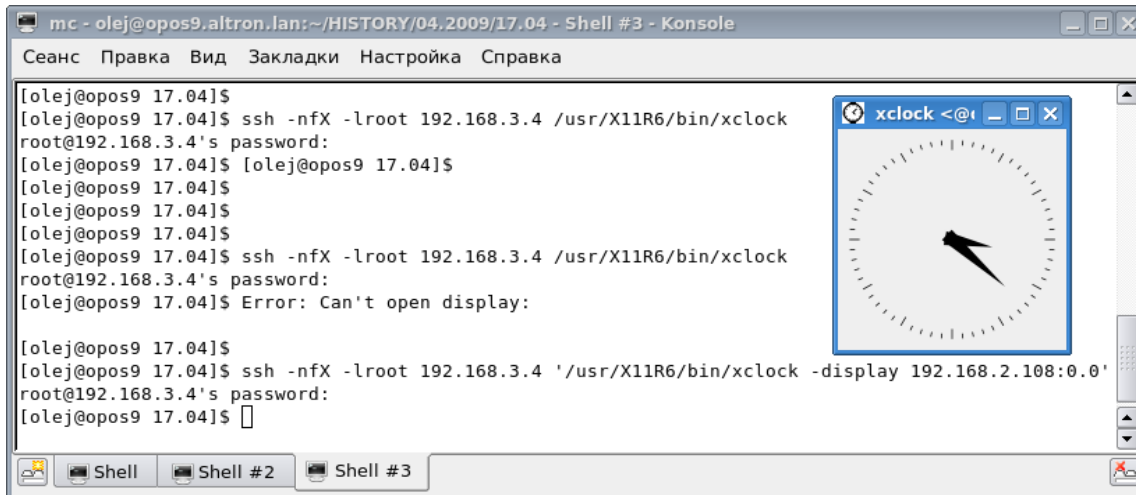


Рис.4.11 Запуск через SSH.

X в терминале QEMU

Есть другой, диаметрально противоположный показанному выше способ выполнения GUI приложений MINIX3 — непосредственно в терминале QEMU, и X-сервере, выполняющемся на хосте MINIX3. В принципе, при корректно установленной сетевой подсистеме (обязательное условие для работы X-системы, даже на автономном компьютере, поскольку X — это сетевой протокол) — добиться запуска менеджера оконной системы с отображением в терминал QEMU удастся простой командой (с терминала MINIX3):

```
# xinit
```

Но при этом часто получается такое разрешение по умолчанию X-сервера (2560x1600, например), и такой, соответствующий ему, размер терминала QEMU (который изменяется при переходе в графический режим), который намного превосходит и экран X-сервера базовой Linux системы ... что мы лучше начнём с основ настройки X-сервера MINIX3, и только потом вернёмся к его запуску. Все действия мы станем производить не с консоли MINIX3, а с терминала SSH, как объяснялось ранее:

```
# qemu -hda minix3-300 -boot c -localtime -kernel-kqemu -net nic,vlan=0 -net tap,vlan=0
```

```
----- tap up -----
```

```
...
```

```
$ ssh -l root 192.168.3.4
```

```
The authenticity of host '192.168.3.4 (192.168.3.4)' can't be established.
```

```
RSA key fingerprint is c5:c1:b3:f1:c4:fe:46:ce:4c:bc:f8:a3:32:1b:40:5a.
```

```
Are you sure you want to continue connecting (yes/no)? Yes
```

```
Warning: Permanently added '192.168.3.4' (RSA) to the list of known hosts.
```

```
root@192.168.3.4's password: Last login: Thu Nov 19 11:55:51 2009
```

```
# pwd
```

```
/root
```

```
# Xorg -configure
```

```
CSocketCreateListener: sa_family = 2
```

```
_XSERVTransMakeAllCOTSServerListeners: failed to create listener for local
```



```
X Window System Version 6.8.2
Release Date: 9 February 2005
X Protocol Version 11, Revision 0, Release 6.8.2
Build Operating System: Minix3 Vrije Universiteit
Current Operating System: Minix qemu-minix 3 1.5 i686
```

...

List of video drivers:

```
mga
glint
nv
tga
s3
s3virge
```

...

Your xorg.conf file is /root/xorg.conf.new

To test the server, run 'X -config /root/xorg.conf.new'

- перечисляется большой список видеоадаптеров, которого показано только начало.

Примечание: время выполнения этой команды настолько значительно (1-2-3 мин. - никакой реакции), что вам может показаться, что VM висит... Но это не так. Дождитесь её завершения. Это плата за виртуализацию! Вообще, всё, что показано в этом разделе, здесь и далее, относительно GUI в терминале QEMU, у меня на процессоре ~600Mhz выполнялось мучительно долго; но на другом компьютере 2x1.6Ghz — всё показанное достаточно приемлемо для работы.

Предупреждение: после выполнения этой конфигурационной команды, все (4) текстовые консоли MINIX3 в терминале QEMU — стали чёрными, «погасли» ... были переведены в какой-то несоответствующий видеорежим. Этого не должно быть, это что-то из артефактов реализации (терминальной подсистемы?). Но после перезапуска QEMU всё восстановится.

После выполнения предыдущей команды мы имеем в /root новый конфигурационный файл:

```
# ls /root
```

```
.ashrc .ellepro.b1 .ellepro.e .exrc .fonts.cache-1 .profile .xinitrc xorg.conf.new
```

Проверяем тестовый старт X при этих новых параметрах:

```
# X -config /root/xorg.conf.new
```

...

```
X Window System Version 6.8.2
Release Date: 9 February 2005
X Protocol Version 11, Revision 0, Release 6.8.2
Build Operating System: Minix3 Vrije Universiteit
Current Operating System: Minix qemu-minix 3 1.5 i686
```

...

```
(==) Log file: "/var/log/XLogFile.0.log", Time: Thu Nov 19 12:46:59 2009
```

```
(++) Using config file: "/root/xorg.conf.new"
```

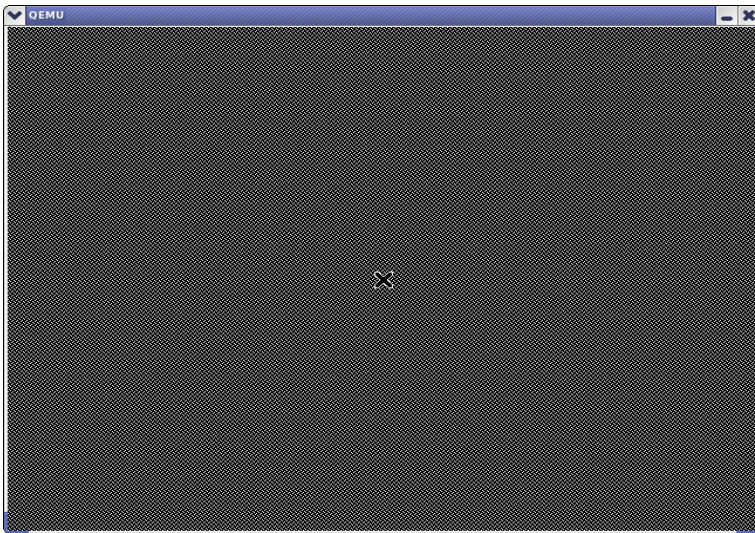


Рис.4.12

При этом мы должны получить на терминале QEMU тестовую картинку X (рис.4.12). Вы можете даже подвигать крестик на картинке... только происходит это мучительно долго (я поначалу даже решил, что отшибло мышку в VM), см. предупреждение выше. Всё, поскольку у нас не остаётся работоспособного терминала QEMU, нам остаётся только перезагрузить QEMU, опять же из терминала SSH:

```
# shutdown now
Broadcast message from root@qemu-minix (tty0)
Thu Nov 19 12:59:43 2009...
The system will shutdown NOW
Sending SIGTERM to all processes ...
Read from remote host 192.168.3.4: Connection reset by peer
Connection to 192.168.3.4 closed.
```

После новой загрузки QEMU, опять же в терминале SSH (при желании то же можете выполнить и в консоли MINIX3):

```
# cp xorg.conf.new /etc/X11/xorg.conf
# ls /etc/X11/
app-defaults  fs  lbxproxy  proxymngr  rstart  twm  xdm  xinit  xkb  xorg.conf
xserver      xsm
# xinit
...
X Window System Version 6.8.2
...
(==) Log file: "/var/log/XLogFile.0.log", Time: Thu Nov 19 14:41:34 2009
(==) Using config file: "/etc/X11/xorg.conf"
...
```

После этого запуска (продолжительная пауза) вы должны получить в окне QEMU нечто подобное показанному на рисунке (здесь работает оконный менеджер twm, если вы перепределили оконный менеджер в файле \$HOME/.xinitrc, то детали отображения изменятся):

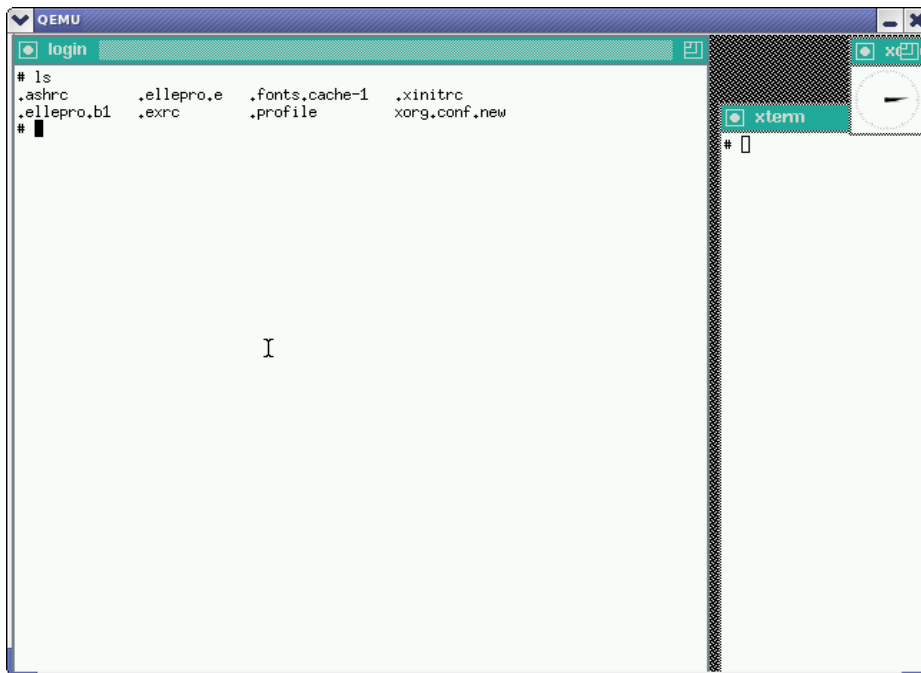


Рис. 4.13

```
# ps -ax
  PID TTY  TIME CMD
...
  107  ?   1:31 sshd
...
  120  p0   4:13 xinit
  121  p0   9:01 X
  123  p0  25:12 xterm
  124  p0   1:17 twm
  125  p0   6:05 xclock
  126  p0  24:34 xterm
  127  p0  24:22 xterm
...
  132  ?   0:15 sshd
...
```

Дополнительные источники информации

1. Документация пользователя эмулятора процессора QEMU

Перевод: Павел Марьянов , март 2006

<http://jack.kiev.ua/docs/qemu-doc-ru.html>

2. Виртуальный полигон: Эмулируем аппаратное обеспечение различных платформ с помощью QEMU

Владимир Ляшко

<http://www.xakep.ru/magazine/xa/118/094/1.asp>

3. Установка MINIX3

Перевод: Роман Игнатов, Павел Макаров

<http://www.minix3.ru/docs/setup-russian.pdf>

4. Описания не документированных команд MINIX3 `add_route`, `pr_routes` и др.

<http://www.os-forum.com/minix/net/general-comment-display.php?commentid=171>

5. Сайт проекта QEMU

<http://www.qemu.org/>

и, в частности, полный комплект документации

<http://www.qemu.org/user-doc.html>