

This presentation was written by Alan Pope.

It was designed to accompany a talk given at the Hampshire Linux User Group December 2005 meeting.

Some notes accompany each slide but of course some of it may not make total sense if you were not at the meeting.

If you have any questions about this presentation you will find details of where to get more information at the end.

A faint, light-colored illustration of Tux the penguin is centered in the background of the slide. Tux is standing and facing forward, with his characteristic black and white plumage and a small red bow tie.

# Contents

- What is QEMU
- Why I would want it
- The competition & How they stack up
- What it can and can't do
- What's in the pipeline
- Where to get QEMU/kqemu/qvm86
- Other tools/utilities
- Installing guests
- Tasks for the reader
- More advanced things
- Links to on-line resources
- A demo (or some screenshots if I screw up my laptop)

The aim of this presentation is to provide an overview of qemu.

What it is, what it does and how it can be useful. It will also cover briefly some of the alternatives, comparing where appropriate the functionality and features of each

Within the notes are sample commands that can be used to install, compile and run qemu.

The presentation, demos and scripts have been created and tested on Ubuntu 5.10 (Breezy), but of course should work on many other Linux distros. The current (10<sup>th</sup> December 2005) CVS was used in demos.

# What is QEMU

*"QEMU is a generic and open source processor emulator which achieves a good emulation speed by using dynamic translation."*

- It's a virtual machine
  - No, not like java
  - Yes, like a software computer within a hardware computer
- Similar in operation to Bochs / DOSEMU / VMWare / Virtual PC
  - There are differences – we'll come to those!

Emulators, virtual machines and simulators have been around for many years. People use emulators to re-live their past, playing old Spectrum, Commodore or Arcade games. Computers of today are excellent at emulating machines of yester-year. A Sinclair Spectrum with just one relatively low speed CPU can easily be emulated by a PC or even a mobile phone of today.

Researchers and designers use emulators for simulating theoretical devices such as new CPU designs where the device being emulated doesn't physically exist yet. This often results in emulation which is slower than running the code on the (as yet non-existent) device itself. This is to be expected as future devices are often more powerful and have hardware features better than current hardware.

Qemu kind of falls between those two groups. It emulates recent-past to current to immediate future architectures. The humble IBM Compatible PC with its 8086 or 80286 PC can easily be emulated by a Pentium of today. Emulating a fast Sparc or AMD Athlon processor on a midrange PC might be somewhat more akin to swimming in treacle. You can do it, but progress will be slow.

## Why I would want it

- Run a legacy OS (e.g. Win95,98 etc)
- Run an OS for a different CPU/Architecture
- Gain familiarity with an installer
- Preparation for exams/certifications
- To document a process
- Test application portability, or installation requirements
- Test applications on OSs you don't want to dedicate a whole machine to
- Test websites in a browser that isn't supported on your host OS
- To test a kernel without rebooting the host
- Because it's fun/cool/interesting
- Because I can
- What other reasons can *you* think of?

As you can see there are a myriad of reasons why you might want to emulate another computer system.

Developers may want to test applications in multiple environments, but may not have the resources or space for many physical computers. Emulation allows them to have many virtual machines (VMs) on one box.

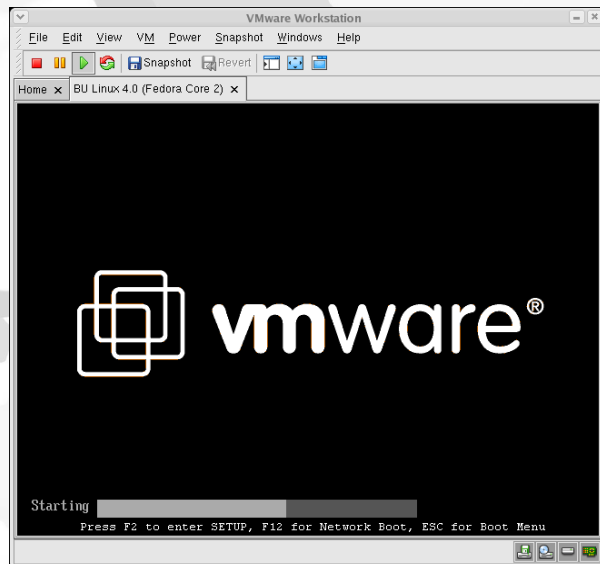
Users may ~~want~~ need to run Windows to enable them to use their online banking service or other broken/non-standards-compliant websites.

Linux users may want to test out home made kernels to see if they function, before committing to rebooting.

Of course, it's also quite a fun thing to do.

## The competition - VMWare

- Non-FLOSS licence
- Free runtime 'player'
- Very robust/reliable
- Industry standard



Very popular in corporate environments for emulating computers running Windows based server products, VMWare is a proprietary x86 emulator.

# The competition – Virtual PC

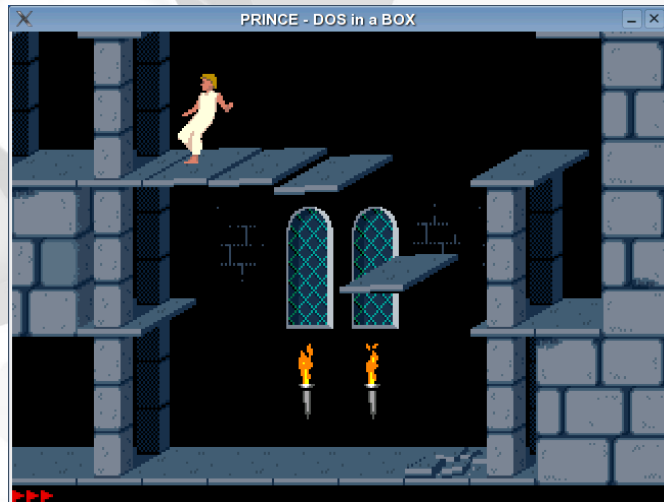
- Microsoft Product
  - Feel free to skip the rest of this page
- Non-FLOSS licence
- Aquired from Connectix
- Questionable support



Initially developed by Connectix, and now owned by Microsoft, Virtual PC really hit it big as a PC emulator on the Mac. Enabling Mac owners to run Windows based software was a key selling feature.

## The competition – DOSEMU

- GPL
- Useful for emulating DOS machines
- That means retro gaming!



DOSEMU is at \*nix users wanting to run DOS based applications on a newer OS. DOSEMU has quite a following amongst people who like to play their favorite 'old-school' DOS based games.

One great benefit of DOSEMU is that it can easily access files in your home directory. This makes the task of getting files "into" the emulated machine much easier.

QEMU can also do this.

## The competition – Others

- Plex86
- Bochs
- Denali
- Linux-VServer
- Mac on Linux
- PearPC
- SVISTA 2004
- TRANGO
- Integrity Virtual Machine
- OpenVZ
- Virtuozzo
- Xen (of course)

Wikipedia has a great comparison of these tools available  
[http://en.wikipedia.org/wiki/Comparison\\_of\\_virtual\\_machines](http://en.wikipedia.org/wiki/Comparison_of_virtual_machines)

## How they stack up

- <Spreadsheet here>

# What can QEMU do - I

- Major features
  - It's GPL (except qemu)
  - Can emulate quite a range of CPUs /architectures
  - Available for wide range of platforms
  - Emulates up to 4 IDE disks or CDROMs
    - Can boot from a disk or CD/DVD image (ISO)
  - Configurable networking options
    - Option 1: Built in firewall/NAT
    - Option 2: Bridged networking
  - Suspendable operation
  - Read-only images (and cow/qcow)

When booting from a CD to install an OS it's often required to change media during the installation. As QEMU supports mounting ISO images of media there's nothing to physically eject.

To switch media in QEMU you must switch to the monitor and issue a sequence of commands to inform the emulator that you wish to eject one media and mount another.

## What can QEMU do - II

- A non-GPL kernel accelerator/virtualiser kqemu
- A GPL project has been started to duplicate this called qvm86
- Has utils for managing disk images
- Supports USB
- Can utilise images of a fixed size, or ones that grow on demand
- Has 'monitor' for debugging and interacting with the emulator directly

The author of QEMU has produced a proprietary (non-open) kernel module called 'kqemu' which can be loaded prior to booting up QEMU. It provides a level of virtualization to improve performance of the virtual machine.

The developer has requested that in order to open up the source for kqemu, his project should be sponsored, or he should in some way be reimbursed.

By way of response another developer has created an open version called 'qvm86' which provides the same functionality.

## What can QEMU do - III



- Emulated hardware

```
00:00.0 Host bridge: Intel Corp. 440FX - 82441FX PMC [Natoma] (rev 02)
00:01.0 ISA bridge: Intel Corp. 82371SB PIIX3 ISA [Natoma/Triton II]
00:01.1 IDE interface: Intel Corp. 82371SB PIIX3 IDE [Natoma/Triton II]
00:02.0 VGA compatible controller: Cirrus Logic GD 5446
00:03.0 Ethernet controller: Realtek Semiconductor Co., Ltd. RTL-8029(AS)
```

Note that all other emulators tend to simulate the PIIX4 as opposed to the PIIX3

## What can't it do

- Doesn't have integrated front end like Vmware
- Not good if you already have a Windows partition
- Hardware 3D

There is a 3<sup>rd</sup> party front end for QEMU available which performs many of the tasks you might use the command line for. It also makes it easier to manage disk images.

It is possible to boot to a windows install on an existing partition, however it's not advised to do so as windows is notorious for having a hissy fit when the hardware changes. You may already be booting the windows partition with your own native hardware. The emulated machine will probably differ considerably – unless of course you have a realtek net card, piix chipset and cirrus logic video card on your PC!

## What's in the pipeline

- Support for host devices
  - USB, Serial & Parallel ports
- VLAN network options
- Emulation of 64-Bit Windows
- MAC OS X emulation
- MIPS & MIPSel User Linux Emulation

Mentioned on IRC on 4<sup>th</sup> December 2005..

```
<filip2307|SaD> btw, QEMU is close to support Win64 .. with the patch I sent  
this morning and ACPI implementation it installs and runs err,  
www.volny.cz/xnavara/qemu/win64/
```

## Where to get QEMU

- Packages in major distros (at time of writing)
  - Debian Sarge
  - Ubuntu Breezy
  - SUSE
  - Mandrake
  - Fedora
- Compile from source
  - Some pre-requisites – see the wiki
  - Doesn't currently compile on GCC4

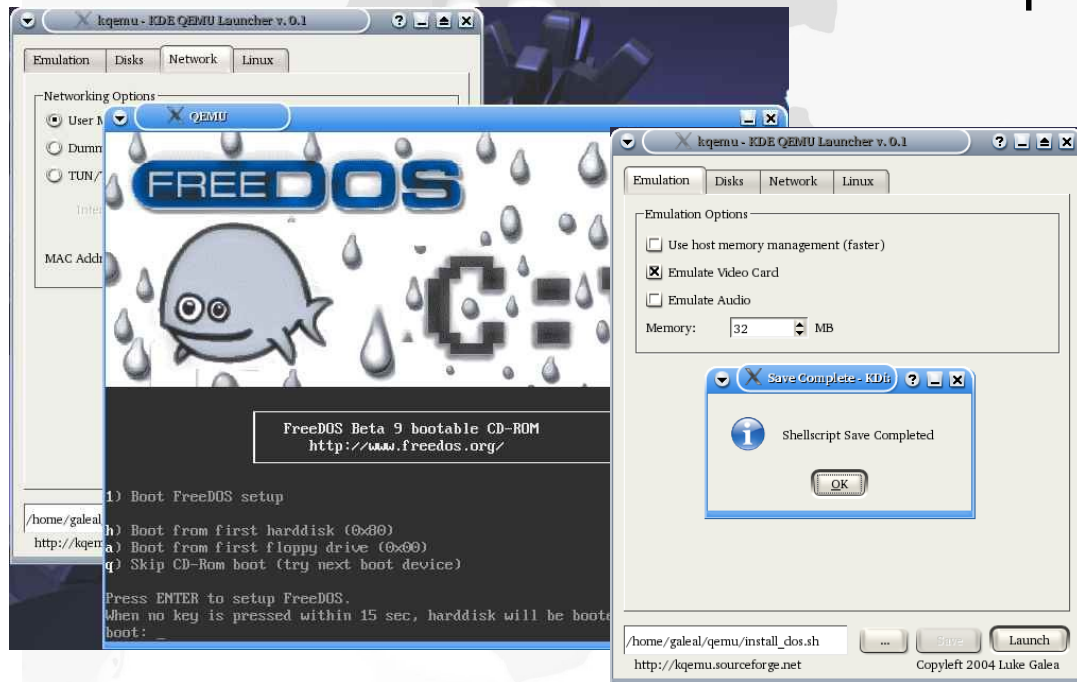
- Debian
  - <http://packages.debian.org/qemu>
- Ubuntu
  - <http://packages.ubuntu.com/qemu>
- SUSE
- Mandrake/Mandriva/Fedora
  - <http://rpmfind.net/linux/rpm2html/search.php?query=qemu>
- Source access from CVS
  - <http://savannah.nongnu.org/cvs/?group=qemu>
  - <http://savannah.nongnu.org/cvs/?group=qvm86>
  - KQEMU is not (currently) OSS, so no CVS access.

## Other tools and utilities

- KRDesktop (formerly kqemu)
  - <http://kqemu.sourceforge.net/>
- Qemu-launcher
  - <http://emeitner.f2o.org/projects/qemu-launcher>
- qGUI
  - <http://perso.wanadoo.es/comike>

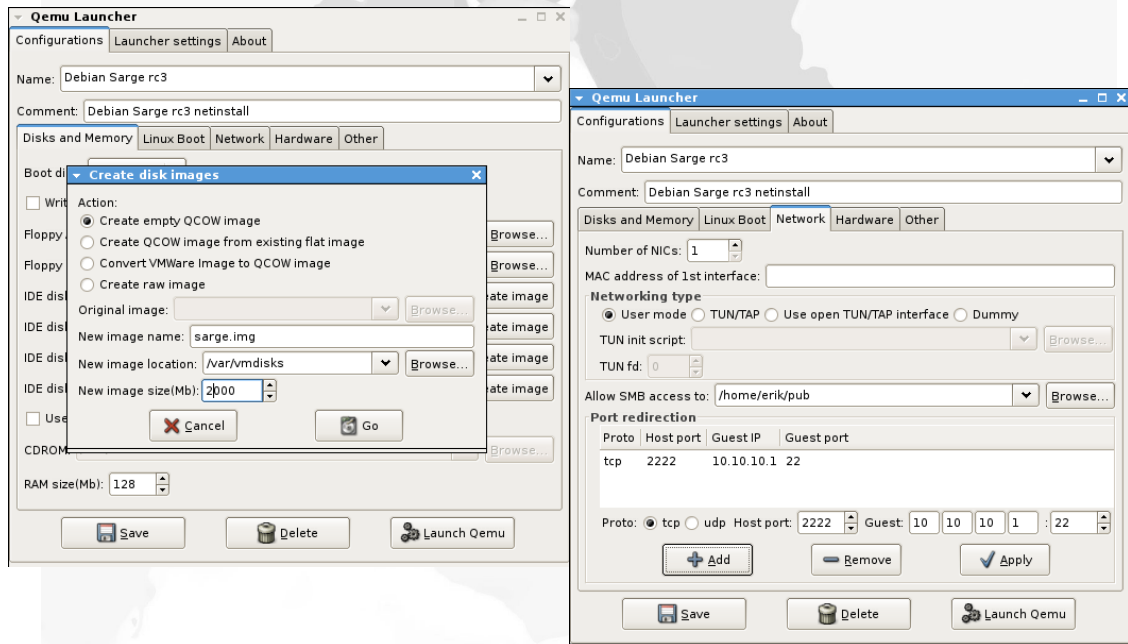
I believe the name of kqemu (the frontend) was changed as it clashed with the closed-source kqemu kernel module.

## Other tools and utilities - KRDesktop



- KRDesktop (formerly confusingly KQEMU) - <http://kqemu.sourceforge.net/>

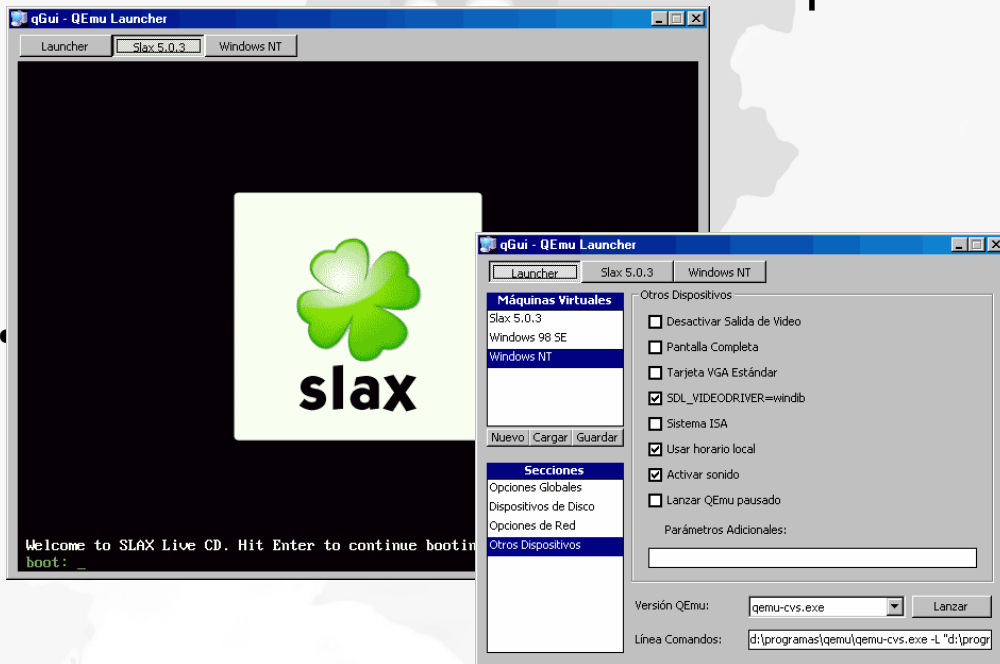
# Other tools and utilities – Qemu Launcher



- Qemu-launcher

- <http://emeitner.f2o.org/projects/qemu-launcher>

# Other tools and utilities - qGUI



- qGUI
  - <http://perso.wanadoo.es/comike>

# Creating Disk Images

- Each guest needs at least one disk image
- `qemu-img` is the tool supplied with QEMU to do this

```
$ qemu-img create myimagefile.img 1G
Formating 'myimagefile.img', fmt=raw, size=1048576 kB
```

Before actually starting `qemu` you really need a disk image to install your guest operating system into. `Qemu` comes with a utility called '`qemu-img`' which can create disk images.

```
$ qemu-img
qemu-img version 0.7.2, Copyright (c) 2004-2005 Fabrice Bellard
usage: qemu-img command [command options]
QEMU disk image utility
```

Command syntax:

```
create [-e] [-b base_image] [-f fmt] filename [size]
commit [-f fmt] filename
convert [-c] [-e] [-f fmt] filename [-O output_fmt] output_filename
info [-f fmt] filename
```

Command parameters:

'filename' is a disk image filename  
'base\_image' is the read-only disk image which is used as base for a copy on write image; the copy on write image only stores the modified data  
'fmt' is the disk image format. It is guessed automatically in most cases  
'size' is the disk image size in kilobytes. Optional suffixes 'M' (megabyte) and 'G' (gigabyte) are supported  
'output\_filename' is the destination disk image filename  
'output\_fmt' is the destination format  
'-c' indicates that target image must be compressed (qcow format only)  
'-e' indicates that the target image must be encrypted (qcow format only)

Supported format: vfat vpc bochs dmg cloop vmdk qcow cow raw

## Booting and Installing guests

- QEMU can be started from the command line

```
qemu -hda myimagefile.img (name of image)
      -boot d (boot from cd)
      -cdrom myiso.iso (cd iso image)
      -enable-audio (well duh)
      -usb (enable usb support)
      -m 128 (qty of RAM for vm)
```

Before actually starting qemu you really need a disk image to install your guest operating system into. Qemu comes with a utility called 'qemu-img' which can create disk images.

Booting qemu is as simple as the command above.

- Windows

- 95/98/NT/2k/XP all work in Qemu
- Win9x doesn't like 'cow' disks
- qvm86 only supports 2k/XP though

- Fedora

- Doesn't like small 'cow' disks



**DEMO!**

Boot Windows XP

Boot another OS (if time)

## Tasks for the reader

- Install it!
- Create a disk image
- Slap in a CD of some OS, or download an ISO
- Start the emulator!
- Go through your normal installation routine
- ???
- Profit!

## More advanced things



- Compiling from latest stable source
  - qemu can't be compiled under GCC4
  - kqemu can be compiled under GCC4
- Compiling from cvs
  - Not as painful as you might imagine
- Bridged networking

The HantsLUG wiki has some pretty comprehensive documentation on compiling Qemu, but be sure to update it if you find any oddities on your own platform/distro!

On Ubuntu 5.10 (Breezy) it can be as easy as getting and unpacking the kernel source, setting a few environment variables, and issuing a slightly tweaked “configure” (to cope with qemus inability to be compiled by GCC4) and “make” commands.



## Tips

- IRC channel #qemu on freenode is helpful and yields good accurate responses to questions
- Compiling from source isn't as painful as it looks!
- Compiling from CVS is just as easy

Compiling from source requires the kernel headers/source handy.  
If you have the source but not compiled it you can simply unpack it and issue the following in the directory containing the uncompressed kernel source.

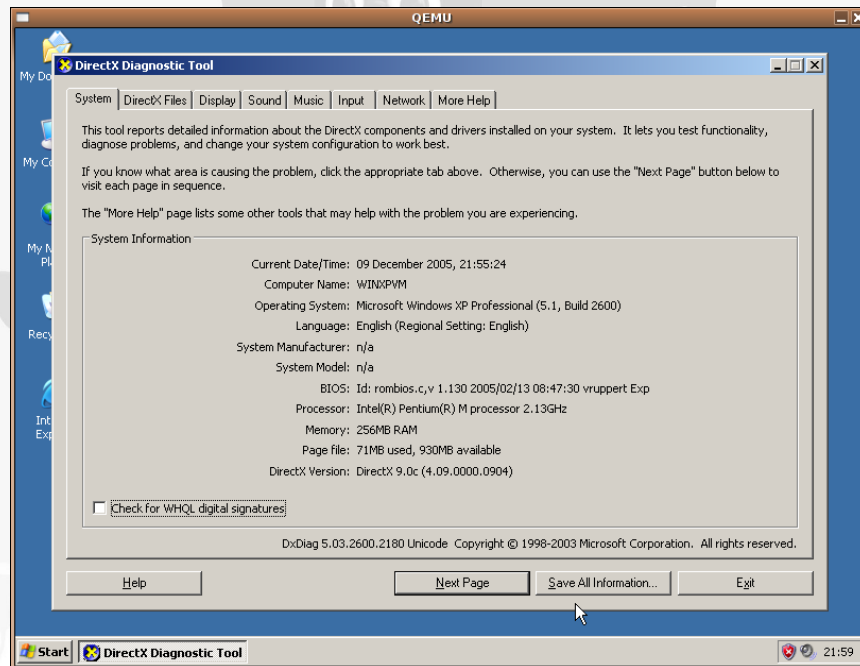
```
make oldconfig  
make include/linux/version.h  
make include/asm
```

Then move on to compile qemu itself.

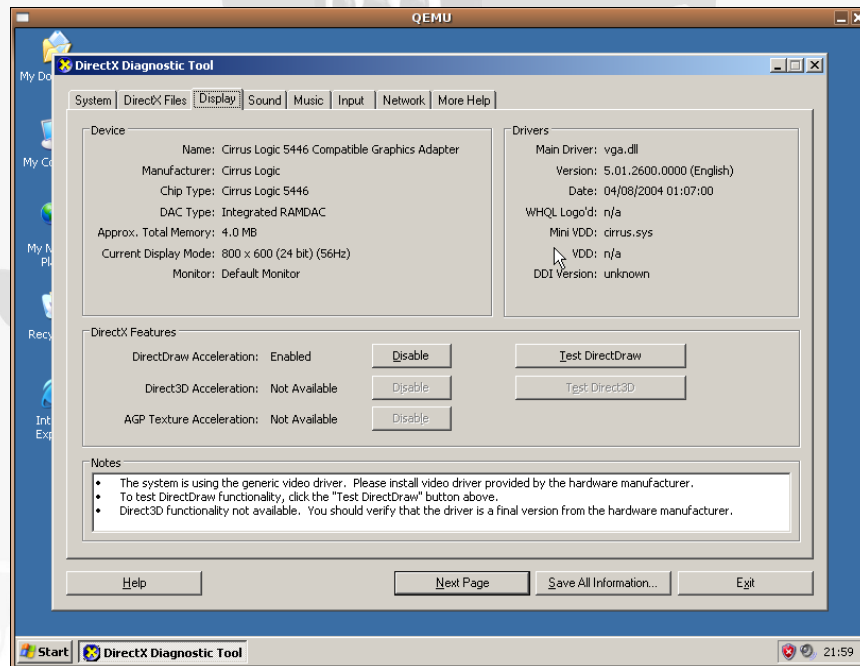
# Links to on-line resources

- Latest version of this presentation
  - <http://hants.lug.org.uk/cgi-bin/wiki.pl?TechTalks>
- Application Homepage
  - <http://fabrice.bellard.free.fr/qemu/>
- Savannah (CVS) page
  - <http://savannah.nongnu.org/projects/qemu/>
- Wikipedia Entry
  - <http://en.wikipedia.org/wiki/QEMU>
- Fedora Package
  - <http://fedoraneers.org/tchung/qemu/>
- Frontend
  - <http://www.davereyn.co.uk/qemu.htm>
- Discussion Forum
  - <http://m2.dad-answers.com/qemu-forum/index.php>
- Unofficial #qemu Wiki
  - <http://lilly.csoft.net/~jeffryj/cgi-bin/moin.cgi/FrontPage>
- Hants LUG pages
  - <http://hants.lug.org.uk/cgi-bin/wiki.pl?LinuxHints/QemuEmulation>
- Wikipedia comparison of virtual machines
  - [http://en.wikipedia.org/wiki/Comparison\\_of\\_virtual\\_machines](http://en.wikipedia.org/wiki/Comparison_of_virtual_machines)
- Free Operating Systems Zoo
  - <http://free.oszoo.org/>

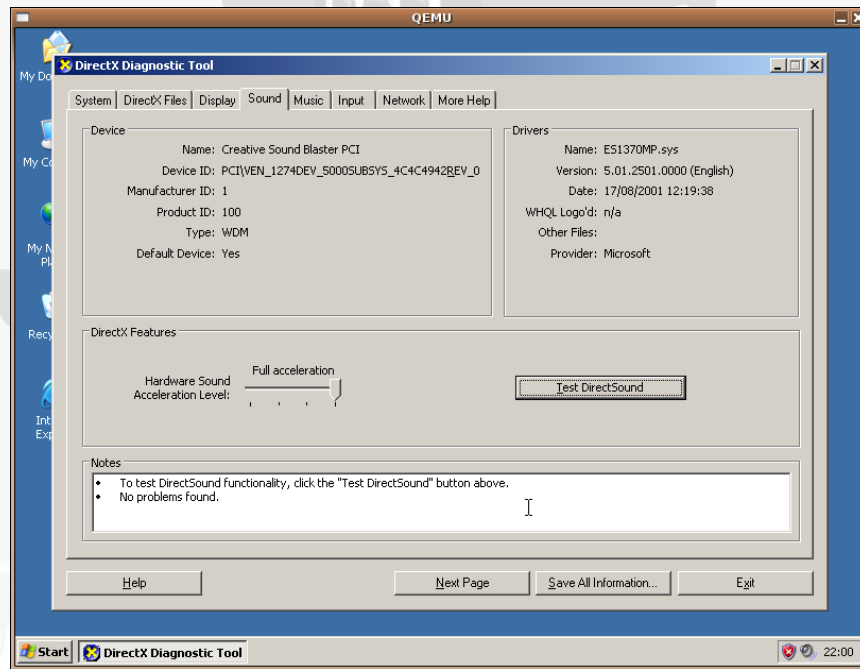
# Some screenies.. XP under QEMU



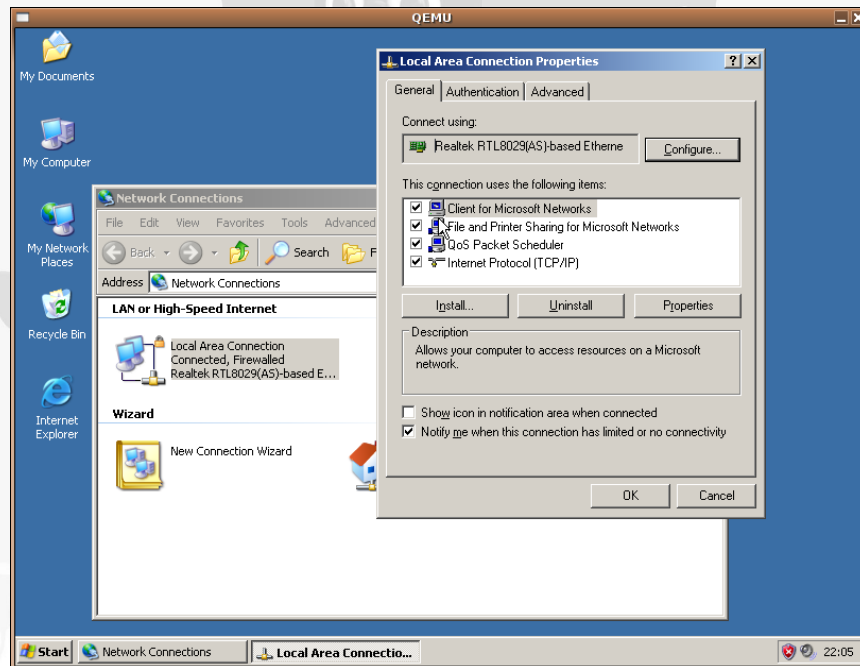
# Emulated Video Hardware



# Emulated Audio Hardware



# Emulated Network Device



Questions?

Thank you.

