QEMU as Host for QEMU#

Introduction

The QEMU Neutrino port is a work in progress. Currently, it is quite stable and runs most important features. This page outlines the progress to date, and provides instruction for installation and usage of QNX Neutrino.

Roadmap#

Completed Work#

- modified source to compile on Neutrino
  - thanks to Colin Burgess for the initial work on this
- edited makefiles to install qemu without the standard “install” utility
- fixed enter key problem in QEMU monitor
- figured out how to enable user mode networking.
- fixed non-posix read() and write() calls that caused filesystem chaos
  - Actually stable now, and qcow2 images work, with snapshots - YAY!
- confirmed that qconn works (see Neutrino as guest in QEMU)
- constructed build files for better hardware detection

Remaining Work, in order of usefulness#

- write a TAP interface or similar for QNX allowing full networking
  - currently, you cannot ping or telnet the outside world, but web browsing and all internal traffic works
- Speed up disk accesses. Using growable formats is fairly slow.
- add support for audio devices, USB and possibly other peripherals
- add support for nto-user mode, which allows applications to run without procnto in the emulator
- implement speed-up code for x86 target, similar to kQEMU

Installing QEMU on Neutrino

Download the current build, NeutrinoAsHostForQEMU/qemu-0.9.0-qnx.tar.gz and extract it at /

```
tar xzvf qemu-0.9.0-qnx.tar.gz
```

You'll also need SDL installed. Using the installer, choose "Simple DirectMedia Layer (SDL) (x86)" from the online repository. Don't confuse it with the other ones.

Building from source#

Do all of this on a self-hosted Neutrino machine with the most recent tools installed.#

Step 1: Checkout the source from our trunk of the project here:#

```
svn checkout --username me@myemail.com http://community.qnx.com/svn/repos/qemu/trunk
```

Step 2: Install SDL support. #

3rd party support has been written for SDL, which Qemu relies on. Get it from the third party repository CD and use the installer in Photon. It is very important that you use the correct version. Using SDL for Neutrino 6.2.1 on 6.3.x will cause conflicts, and Qemu will not work. This might be tricky depending on where you get SDL from. Once I know how to make it work reliably with SDL in our online repository, I will post it here.

Step 3: Configure, Make, Install #

Qemu was built with GCC 3.4, and will not compile with gcc 2.x.x or 4.x, but works great with version 3.3.5. We have to configure it to use the proper compiler. Type:

```
./configure --cc=ntox86-gcc-3.3.5 --host-cc=ntox86-gcc-3.3.5
```

The configure script should tell you if you did anything wrong. Type make, and qemu will compile. Do a "make install" as root to install it on your machine.

Running QEMU#

QEMU on a Neutrino host is no different than for any other hosts, with the exception of features that are not implemented yet. See the wiki page, Neutrino as guest in QEMU for a guide.

Gochas:#

To use the QEMU monitor, serial port, or parallel port outputs, we need to press <cnt><alt><#>. By default in Photon, these key combinations switch workspaces and are not passed to applications. We have to disable this by setting an environment variable before Photon starts:

```
export PHWMOPTS=”-S”
```

Either type this line from text mode before starting Photon, or add it to the end of a file like /etc/profile.

Another thing to be aware of is that the QNX installer CD requires at least 2Gb of disk space, but the QNX filesystem limits file sizes to less than 2Gb. To get around this, use either growable image formats (not raw) or make custom images for Neutrino using buildfiles, mkifs, and dinit. See the other wiki for help.