#   © 2012 Cadence Design Systems, Inc.

# Licensed under the Apache License, Version 2.0 (the "License");

# you may not use this file except in compliance with the License.

# You may obtain a copy of the License at

#

# http://www.apache.org/licenses/LICENSE-2.0

#

# Unless required by applicable law or agreed to in writing, software

# distributed under the License is distributed on an "AS IS" BASIS,

# WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

# See the License for the specific language governing permissions and

# limitations under the License.

**Configuring the Intel Linux NVMe Test Suite Environment**

1. **Introduction**

This document provides additional information necessary to setup a Linux environment to run the Intel NVMe compliance suite against real hardware. The Intel provided configuration document for setting up a Linux QEMU environment can be found at <https://github.com/nvmecompliance/manage/blob/master/readme.startHere.odt>.

1. **References**

*NVM Express Revision 1.0b*

*NVM Express Revision 1.0c*

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

1. **Linux NVMe Driver & Test Software**

<https://github.com/nvmecompliance>

<https://github.com/nvmeqemu/>

<https://github.com/nvmecompliance/tnvme/wiki>

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

1. **Linux OS Requirements**

Ubuntu 10.10 Server 64 bit is the recommended OS for running the Linux driver and test suite. Earlier releases of Ubuntu may not support the features needed. Non-Ubuntu Linux releases should work, but again the kernel level support must be present. Target any distribution as long as it supports a kernel satisfying (2.6.35.22 <= target kernel < 3.0)

It is recommended to not use a VMWare environment to run the Linux OS. Ubuntu 10.10 should be a directly bootable OS. It can be set up on a dual boot system if running Windows as the main OS.

1. **Creating a bootable Ubuntu drive (host OS)**

This document assumes that Windows is the boot OS and Ubuntu is being made the 2nd boot option. If an acceptable Ubuntu bootable environment is already available then this section can be skipped.

* 1. Ubuntu 10.10 download site: <http://releases.ubuntu.com/10.10/>
* Download [64-bit PC (AMD64) desktop CD](http://releases.ubuntu.com/10.10/ubuntu-10.10-desktop-amd64.iso) to a location on your on your hard drive. This is an ISO file.
  1. Create a bootable CD of Ubuntu 10.10
* Requires CD burning software that will make a bootable ISO file. If your system has no such software, Express Burn from NCH Software has a free sample download of their software which can be used at <http://www.nch.com.au/burn/index.html>. Other packages are available as well.
* Start the CD burn software and follow directions to save the downloaded Ubuntu file to a CD.
  1. Installing Ubuntu
* Select the CD and start the ‘webui’ which will walk you through creating a bootable OS for Ubuntu.

1. **Downloading the Linux software to the host OS**

*1st Reference:* <https://github.com/nvmecompliance/manage/blob/master/readme.startHere.odt> under *section 3.0*

* Reboot the system and select Ubuntu
* Create an nvme directory and subdirectories. (I placed mine at /home/<user name>)
  + /home/<user name>/nvme
  + /home/<user name>/nvme/dnvme
  + /home/<user name>/nvme/tnvme
  + /home/<user name>/nvme/manage
  + /home/<user name>/nvme/qemu
* Downloading and installing
  + Select FireFox and navigate to <https://github.com/nvmecompliance>
    - Select ‘tnvme’, ‘dnvme’ and ‘manage’ one at a time and navigate to the ‘download’ selection and ‘save’ the files. They will be saved under the ‘Downloads’ directory on Ubuntu.
    - *Note: Intel documentation says to ‘clone’ the files but I did a download since I had no intention of copying any changes back to the repository.*
    - On Ubuntu select each file in the Downloads directory and unzip. Copy the contents of each unzipped directory to their respective /<directory>/nvme/dnvme; /<directory>/nvme/tnvme; /<directory>/nvme/manage
* Select FireFox and navigate to <https://github.com/nvmeqemu/>
  + - Select ‘qemu’ and navigate to the ‘download’ selection and ‘save’ the file. They will be saved under the ‘Downloads’ directory on Ubuntu.
    - *Note: Intel documentation says to ‘clone’ the files but I did a download since I had no intention of copying any changes back to the repository.*
    - On Ubuntu select the file in the Downloads directory and unzip. Copy the contents of the unzipped directory to /<directory>/nvme/qemu

1. **Install the Linux Libraries on the host OS**

*1st Reference:* <https://github.com/nvmecompliance/manage/blob/master/readme.startHere.odt> under *section 4.1.1.x*

7.1 Open a terminal window on the host OS and install the following libraries:

* sudo apt-get install libxml2
* sudo apt-get install libxml++2.6-dev libxml++2.6-doc
* sudo apt-get install g++
* sudo apt-get install libboost1.42-all-dev
* sudo apt-get install qemu-KVM
* sudo apt-get install openssh-server
* sudo apt-get install linux-headers-2.6.35-22-server
* sudo apt-get install zlib1g-dev
* sudo apt-get install libsdl-dev
* sudo apt-get install libaio-dev

7.2 At the Linux prompt type the following command to enable port 22 which will allow the virtual OS to network with the host OS:

* iptables –A INPUT –p tcp - -dport ssh –j ACCEPT
* iptables –L # verify that ssh is running
* iptables-save

*Note: these commands need to be issued whenever the host OS is booted to enable port 22.*

1. **Creating the QEMU Environment (virtual OS)**

*1st Reference:* <https://github.com/nvmecompliance/manage/blob/master/readme.startHere.odt> under *section 4, 5, 6, 7*

* Building QEMU
* cd /<directory>/nvme/qemu
* ./buildQemu.sh
  + - Download the Ubuntu 10.10 ISOfile from: <http://releases.ubuntu.com/10.10/>
* Download [64-bit PC (AMD64) desktop CD](http://releases.ubuntu.com/10.10/ubuntu-10.10-desktop-amd64.iso) to the Downloads directory.
  + Create a directory: ../nvme/manage/vkernel and copy the ISOfile to this directory.
  + Create a directory /<directory>/nvme/manage/vdisk and cd /<directory>/nvme/manage and

execute: ./vdiskCreate.sh 10

* + At the Linux prompt: cd /<directory>/nvme/manage and execute:

./runQemu.sh 10GB.img <Ubuntu ISOfile name>

* + Wait for the QEMU window to appear and Ubuntu to start the install process. Follow the instructions for installation.
  + After installation of Ubuntu 10.10 it should shutdown. From inside the host OS open a terminal window and execute the following to boot the virtual OS:
* cd /<directory>/nvme/manage
* ./runQemu.sh 10GB.img

1. **Setting up the virtual OS environment (virtual OS)**

*1st Reference:* [https://github.com/nvmecompliance/manage/blob/master/readme.startHere.odt under *section 10*](https://github.com/nvmecompliance/manage/blob/master/readme.startHere.odt%20under%20section%2010)

*2nd reference* [*https://help.ubuntu.com/community/SSHFS*](https://help.ubuntu.com/community/SSHFS)

* After the virtual OS has booted, open a terminal window and install:
* sudo apt-get install libfuse2
* sudo apt-get install runit
* Connecting to the host OS:
  + At the Linux prompt type:

sudo gpasswd –a $USER fuse # $USER is name of user when Ubuntu was installed

* You need to create a local mount point which will be used to mount to the host OS. It is less confusing to create the mount point on the same directory path as the directory path on the host OS where our target directory is at. The host target directory is:

$USER@<IP address>:/<directory>/nvme

which for my system was:

username@158.140.84.83:/home/username/nvme

So a corresponding local mount point would be: /home/username/nvme

*Note: you need the IP address of the host OS. So switch back to the host OS by doing a “ctrl-alt” and at the Linux prompt enter: ifconfig and find the IP address.*

*Note: I used ‘nvme’ since we are compiling and running from nvme and its subdirectories. So all my examples use ‘nvme’.*

* Once you have the local mount point and the host OS IP address, at the Linux prompt execute:
  + sshfs –o idmap=user $USER@<IP address>:/<directory>/nvme ~/nvm

For my system this command line would be:

*sshfs –o idmap=user* [*username@158.140.84.83:/home/username/nvme*](mailto:bpowers@158.140.84.83:/home/bpowers/nvme) *~/nvme*

* You should receive a request for a password for the host OS. Type the password and you should be logged into the host OS at the mount point.

**NOTE: The sshfs command in the Intel documentation section 10.1.3.1 did not work for me so I used the command from Ubuntu:** [*https://help.ubuntu.com/community/SSHFS*](https://help.ubuntu.com/community/SSHFS)

* At the Linux prompt cd to /<directory>/nvme and do an “ls –l”. You should see all the files under the nvme directory on the host OS system.

1. **Install the Linux Libraries on the virtual OS**

10.1 Open a terminal window on the virtual OS and install the following libraries:

* sudo apt-get install libxml2
* sudo apt-get install libxml++2.6-dev libxml++2.6-doc
* sudo apt-get install g++
* sudo apt-get install libboost1.42-all-dev

*NOTE: I installed these libraries on both the host and virtual OS, because they are the identical kernel revision, to allow compiling from either OS.*

1. **Building the Test Suite and Device Driver (virtual OS)**

*Reference*:

*<https://github.com/nvmecompliance/tnvme/wiki/>*

[*https://github.com/nvmecompliance/dnvme/wiki/Compiling*](https://github.com/nvmecompliance/dnvme/wiki/Compiling)

[*https://github.com/nvmecompliance/tnvme/wiki/Compiling*](https://github.com/nvmecompliance/tnvme/wiki/Compiling)

11.1 Building the kernel driver

At the Linux prompt execute:

* cd /<directory>/nvme/dnvme
  + NEVER do “sudo make clean” for this will cause damage to the installed libraries. Please refer to <https://github.com/nvmecompliance/dnvme/wiki/Compiling> for details.
* vi Makefile
  + *Locate the line with the KDIR variable and replace the variable with: 2.6.35-22-server and save.*
* make
* cp dnvme.ko ~/.
* insmod ~/dnvme.ko
* chmod 777 /dev/nvme0

11.2 Building the Test Suite

At the Linux prompt execute:

* cd /<directory>/nvme/tnvme
* make

1. **Running the Test Suite (virtual OS)**

*Reference:*

[*https://github.com/nvmecompliance/tnvme/wiki/Executing*](https://github.com/nvmecompliance/tnvme/wiki/Executing)

[*https://github.com/nvmecompliance/tnvme/wiki/Command-Line-Options*](https://github.com/nvmecompliance/tnvme/wiki/Command-Line-Options)

At the Linux prompt execute:

* cd /<directory>/nvme/manage
* ./runtnvme.sh –help
  + Look inside file *./Logs/current* for the output