**COMP 3000 – Operating Systems**

**Getting Started with MINIX 3**

**Author:** Michel Barbeau

**Version:** January 7, 2018

To get an overview of MINIX 3, you may watch the Youtube video with the author of the system: [MINIX 3: A Reliable and Secure Operating System](https://www.youtube.com/watch?v=jiGjp7JHiYs).

There is an official [web site on MINIX 3](http://www.minix3.org/) that contains a lot of useful information (user is “root” and password is “student”).

Note that the official release (CD images) posted in the [Download section](http://wiki.minix3.org/doku.php?id=www:download:start) of the MINIX 3 web site **WILL NOT WORK FOR THIS COURSE**.

You need an image built using the latest [development snapshot](http://download.minix3.org/iso/snapshot/), i.e., snapshot: [minix\_R3.4.0rc6-d5e4fc0.iso.bz2](http://download.minix3.org/iso/snapshot/minix_R3.4.0rc6-d5e4fc0.iso.bz2).

I made available a [MINIX 3 image suitable for the course](http://download.scs.carleton.ca/virtualimages/COMP3000B-W18.ova). This image has been built in the MAC OS High Sierra/Virtual Box environment.

If for any reason this version does not fit your needs, you are welcome to build own image. It is a relatively straightforward process. There are detailed [installation instructions](http://wiki.minix3.org/doku.php?id=usersguide:doinginstallation).

In addition to the VirtualBox environment, I tested virtual machine installation on Linux Ubuntu 16.04/KVM. It works well.

On the other hand, I also tested MINIX on a BeagleBone Black. The HDMI interface is not supported. Also, it looks like the current master branch is broken for the BeagleBone.

**OpenSSH**

With a new image in a virtual machine environment, one of the first things to do is to install OpenSSH on MINIX 3. You will be able to open multiple MINIX 3 windows in the host environment, e.g. MAC OS.

There are [Setting Up SSH](http://wiki.minix3.org/doku.php?id=usersguide:settingupssh) instructions. On my system, I had to do the following:

1. Install:

pkgin update

pkgin install openssh

cp /usr/pkg/etc/rc.d/sshd /etc/rc.d/

printf 'sshd=YES\n' >> /etc/rc.conf

/etc/rc.d/sshd start

2. Enable SSH login as root, edit /usr/pkg/etc/ssh/sshd\_config and change the line that reads:

#PermitRootLogin prohibit-password

to:

PermitRootLogin yes

3. In a terminal of the host system (e.g., MAC OS), enter:

VBoxManage modifyvm "MINIX R3.4.0rc6" --natpf1 "guestssh,tcp,,2222,,22"

ssh -l root -p 2222 localhost

4. It may be convenient to disable host checking, on the host system (e.g., MAC OS), in file /etc/ssh/ssh\_config, append the lines:

Host \*

StrictHostKeyChecking no

UserKnownHostsFile=/dev/null

It is also possible to browse the MINIX 3 file system from an Eclipse Remote Explorer in the host environment, see installation instructions [here](http://wiki.minix3.org/doku.php?id=developersguide:eclipsetutorial). In the MAC OS environment, I had mixed success with it.

**Organization of the MINIX 3 Source Code**

Source code must be installed in repository /usr/src (already done in the [MINIX 3 image suitable for the course](http://download.scs.carleton.ca/virtualimages/COMP3000B-W18.ova)).

1. Update and install packages:

pkgin update

pkgin\_sets

2. Clone source code (check available disk space with df –h):

cd /usr

git clone git://git.minix3.org/minix src

3. In case and update is needed:

cd /usr/src

git pull

Master copies of header files are in directory /usr/src/include.

Makefiles, one per directory, control compilation.

**Compiling and Running MINIX 3**

The source code of the system utilizes makefiles. They are defined one above the other. To create a new version of MINIX, you can compile locally or globally.

1. Compiling a local makefile is useful to quickly check if your code changes compile correctly without going through the entire build process of the top level makefile.

2. The top level makefile is in directory /usr/src/. To build and deploy your MINIX OS changes, you must use the top level makefile:

cd /usr/src

make build

The very first build can take hours. Warnings are OK during the build process, but errors break the build process.

3. Once it is finished, you need to reboot MINIX:

reboot

4. After the first build, an incremental build can be much faster:

cd /usr/src

make build MKUPDATE=yes

5. You may compile and install the kernel only (will not work when modifications are involved at several different places):

cd /usr/src/releasetools

make hdboot

6. Results of past builds can be cleaned-up with:

make clean

**Last word**

Successful OS kernel development needs more thinking than writing. A good working methodology is important. Try and test small changes one-by-one. Before moving to the next step, make sure that you understand what you are doing and that changes you made work. Before doing any change, take snapshots of your virtual machine. In case it does not work, you can easily return to a working state.