Setting up an NDT Server on Dell PowerEdge1750

Evan Mahone
St. Mary's College of Maryland
Office of Information & Technology

October 30, 2013

1 CentOS Install Preparation

- Download x86_64 version 6.4 iso from here[1]
- Burn the two parts of the ISO to two dvds as IMAGES
- Boot from the first DVD (by picking the optical drive from the boot menu accessed with F11 during startup)
- Wipe the server hard drive completely (using DBAN or any other program)
- Ensure a monitor, keyboard, and mouse are connected to server
- Ensure that you have a working internet connection wired via an ethernet cord

[1] http://mirrors.bluehost.com/centos/6.4/isos/x86_64/

2 Installing CentOS v6.4

For the following I chose packages which were a mix of minimal desktop and basic server described here[2]. I also included packages that I knew I needed later for configuring NDT. I also changed the default desktop to Xfce instead of gnome as I found it performed better on the server.

- If you haven't already, perform the check on the installation media - *Continue*
- Choose to install with video driver - *Continue*
- Language: English (English) - *Continue*
- Keyboard: U.S. English - *Continue*
- Type of devices: basic storage devices - *Continue*
- Choose to perform a fresh installation - *Continue*
- Hostname: ndt3.smcm.edu - *Continue*
- Nearest city in time zone: America/New York - *Continue*
- Choose a root password (mine is foobar) - *Continue*
- Choose to make a custom partition layout, using the three partitions described below
  1. **boot**: standard partition, mountpoint: /boot, filesystem: ext4, fixed size: 1024 MB
  2. **swap**: standard partition, filesystem: swap, fixed size: 8192 MB. Based on this[3]
  3. **root**: standart partition, mountpoint: /, filesystem: ext4, fill to maximum allowable size
- - *Continue*
- install bootloader on /dev/sda - *Continue*

2.1 Package Configuration

- from the list of installation types, choose minimal install
- choose to add an additional software repository
  - **repository name**: epel-release-6-8.noarch
  - **repository type**: HTTP/FTP
  - **repository url**: http://dl.fedoraproject.org/pub/epel/6/x86_64/
- Choose customize now - *Continue*
- On the packages screen, check only the following:
  - APPLICATIONS
    - Internet Browser
  - BASE SYSTEM
    - Base
    - Compatability Libraries
    - Debugging Tools
    - Hardware Monitoring Tools

[2] http://unix.stackexchange.com/questions/20379/centos-6-default-installation-options
[3] https://access.redhat.com/site/documentation/en-US/Red_Hat_Enterprise_Linux/4/html/System_Administration_Guide/Swap_Space.html

    - Performance Tools
  - DESKTOP
    - Desktop Debugging
    - Fonts
    - General Purpose Desktop (check only the following under optional packages)
      - gedit
    - Graphical Administration Tools
    - Input Methods
    - Legacy X Window System Compatability
    - X Window System
    - Xfce
  - DEVELOPMENT
    - Development Tools
  - SERVERS
    - Server Platform
- - *Continue*
- Now let it go through with installation - changing discs if needed
- Allow the server to reboot after the installation completes
- When the server reboots wait for the welcome screen to load - *Continue*
- Agree to license - *Continue*
- Create user (below is what I made)
  - **username**: seldon
  - **full name**: seldon
  - **password**: Asimov03
- - *Continue*
- Choose the synchronize date and time - *Continue*
- Choose to enable kdump
- Allow the server to reboot once again.
- Let it load. A login screen should pop up and you should be able to log in.
- Make sure to pick to use the default configuration once you log in for the first time.

3 Configuring CentOS

- Note that for all times that I user *seldon* you can replace it with the user you registered.
- Before doing anything else, install the software updates by going to the applications menu : administrative : software update and entering the root password (foobar)

3.1 Giving Root Priveleges

Before doing the following, make sure you know basically how to use vim. Refer here for how to use vim / visudo.[4]

Open a terminal and type the following (password is foobar):

```
$ su
# sudo /usr/bin/visudo
```

Move to the bottom of the document, and add the following lines:

```
#giving root priveleges to seldon
seldon ALL=(ALL) ALL
```

Save and exit, and you should now be able to use sudo. Whenever you use sudo you will have to type in the users password, not the root password (foobar).

3.2 Enabling EPEL repository in yum

```
$ sudo rpm -Uvh http://mirrors.kernel.org/fedora-epel/6/i386/epel-release-6-8.noarch.rpm
```

Check to ensure EPEL was enabled by issuing the command:

```
$ yum repolist
```

EPEL should be listed in there somewhere.

3.3 Enabling nux-desktop Repository in Yum

```
$ sudo rpm -Uvh http://li.nux.ro/download/nux/dextop/el6/x86_64/nux-dextop-release-0-1.el6.nux.noarch.rpm
```

Check to ensure nux-desktop was enabled by issuing the command:

```
$ yum repolist
```

nux-desktop should be listed somewhere in there

3.4 Installing Chromium (optional)

Open a terminal and type the following:

```
$ cd /etc/yum.repos.d
$ sudo wget http://people.centos.org/hughesjr/chromium/6/chromium-el6.repo
$ sudo yum install chromium -y
```

You can use chromium with the chromium command or from the applications menu.

---

[4] http://maeks84.wordpress.com/2008/05/29/how-to-use-visudo/

3.5 Installing Shutter (optional)

Shutter is a tool for screen capture.

```
$ sudo yum install shutter -y
```

You can use shutter via the command shutter.

3.6 Setting Up RPM Build Environment

These instructions are based off of the ones found here[5]

```
$ mkdir -p ~/rpmbuild/{BUILD,RPMS,SOURCES,SPECS,SRPMS}
$ echo '%_topdir %(echo $HOME)/rpmbuild' > ~/.rpmmacros
```

---

[5] http://wiki.centos.org/HowTos/SetupRpmBuildEnvironment

4 Installing Package Dependencies

4.1 ncurses-devel

ncurses-devel is needed for menuconfig

```
$ sudo yum install ncurses-devel.x86_64 -y
```

4.2 Python

python-devel-2.6.6-37.el6.x86_64 is needed for configuring the userland libraries.

```
$ sudo yum install python-devel -y
```

4.3 gtk2-devel

gtk2-devel is needed for the gutil utility from the userland libraries

```
$ sudo yum install gtk2-devel -y
```

4.4 gcc Compatability Libraries

Gcc compatability libraries are needed for the web100 libraries.

```
$ sudo yum install compat-gcc-34-g77
```

4.5 Sun JDK 1.5 Dependencies

The following 32-bit packages must be installed in order to get Sun JDK 1.5 working:

```
$ sudo yum install libXext.i686 libXt.i686 libXi.i686 libXp.i686 libXtst.i686 -y
```

The following 32-bit package needs to be installed to get JDK 1.5-alsa support:

```
$ sudo yum install alsa-lib.i686 -y
```

The following package needs to be installed to get JDK 1.5-fonts support:

```
$ sudo yum install chfontpath
```

As well, we need to link our mkfntdir binary to the path that JDK-1.5-fonts searches:

```
$ sudo mkdir X11R6 X11R6/bin
$ sudo ln -s /usr/bin/mkfontdir /usr/X11R6/bin/mkfontdir
```

4.6 Sun JDK 1.5

The following section is based off of instructions here [6].

- The sun JDK 1.5 is needed in order to build ndt-3.6.4.
- First, download the Sun JDK 1.5 update 15 rpm and put it in the home directory:

```
$ cd ~
$ wget http://mirrors.dotsrc.org/jpackage/1.7/generic/SRPMS.non-free/java-1.5.0-sun-1.5.0.15-1jpp.nosrc.rpm
```

- Next download the Sun JDK 1.5 update 15 self extracting binary from here: here. [7].
- It should be named *jdk-1_5_0_15-linux-i586.bin*
- Move the downloaded file to the */home/seldon/rpmbuild/SOURCES* directory
- Make it executable

```
$ chmod +x ~/rpmbuild/SOURCES/jdk-1_5_0_15-linux-i586.bin
```

In order to build the package, you must set your computer to emulate a different architecture, that is:

```
$ setarch i586
```

now build the rpm file:

```
$ rpmbuild --rebuild java-1.5.0-sun-1.5.0.15-1jpp.nosrc.rpm
```

if the installation goes over well you can *unset* the i586 architecture:

```
$ exit
```

now install all of the rpms as root:

```
$ su
# cd /home/seldon/rpmbuild/RPMS/i586
# rpm -Uvh java-1.5.0-sun-1.5.0.15-1jpp.i586.rpm
# rpm -Uvh java-1.5.0-sun-alsa-1.5.0.15-1jpp.i586.rpm
# rpm -Uvh --nodeps java-1.5.0-sun-fonts-1.5.0.15-1jpp.i586.rpm
# rpm -Uvh java-1.5.0-sun-plugin-1.5.0.15-1jpp.i586.rpm
# rpm -Uvh java-1.5.0-sun-devel-1.5.0.15-1jpp.i586.rpm
# exit
```

[6] http://wiki.centos.org/HowTos/JavaOnCentOS
[7] http://www.oracle.com/technetwork/java/archive-139210.html

5 Patching, Compiling, Installing, and Booting Your Custom Linux Kernel

The following instructions were adapted from the NDT cookbook, found here.[8]

5.1 Getting and Unpacking Resources

Getting and unpacking the linux kernel:

```
$ cd /usr/src
$ sudo mkdir tars
$ cd tars
$ sudo wget https://www.kernel.org/pub/linux/kernel/v2.6/linux-2.6.32.tar.gz
$ sudo tar -xvf ./linux-2.6.32.tar.gz
$ sudo mv ./linux-2.6.32 /usr/src/kernels/linux-2.6.32
```

Getting and unpacking the web100 patch

```
$ cd /usr/src/tars
$ sudo wget http://www.web100.org/download/kernel/2.5.27/web100-2.5.27-201001301335.tar.gz
$ sudo tar -xvf ./web100-2.5.27-201001301335.tar.gz
$ sudo mkdir /usr/src/kernel_patches
$ sudo mv ./web100 /usr/src/kernel_patches/web100
```

5.2 Patching Kernel

To apply the web100 patch to the linux kernel, issue the commands below:

```
$ cd /usr/src/kernels/linux-2.6.32
$ sudo patch -p1 < /usr/src/kernel_patches/web100/web100-2.6.32-2.5.27-201001301335.patch
```

5.3 Compiling and Configuring Kernel

To compile the kernel, issue the following commands:

```
$ cd /usr/src/kernels/linux-2.6.32
$ make clean
$ make mrproper
```

Copy the current kernel's boot config to our new kernel:

```
$ sudo cp /boot/config-2.6.32-358.el6.x86_64 ./.config
```

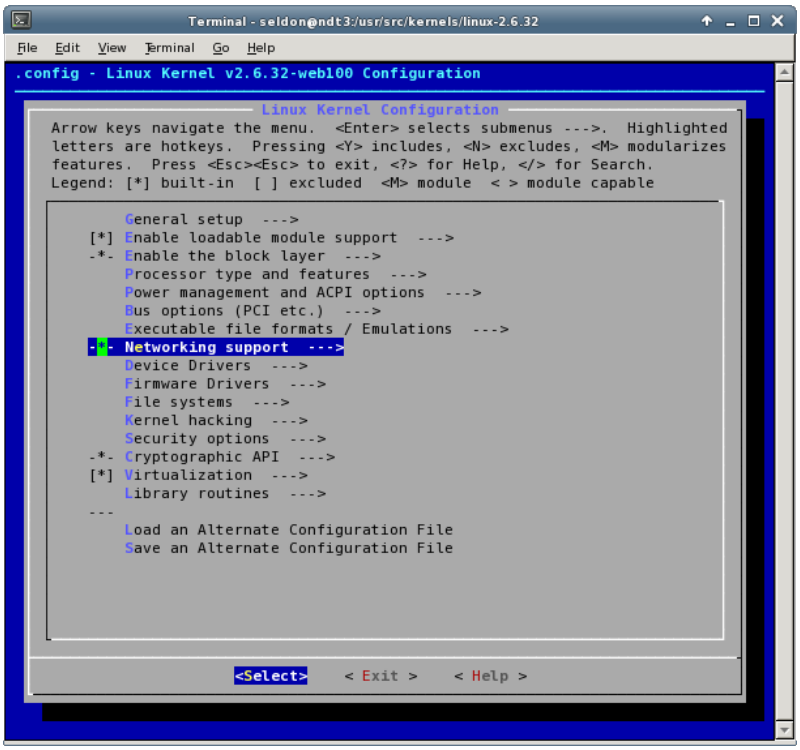
Configure the kernel:

```
$ sudo make menuconfig
```

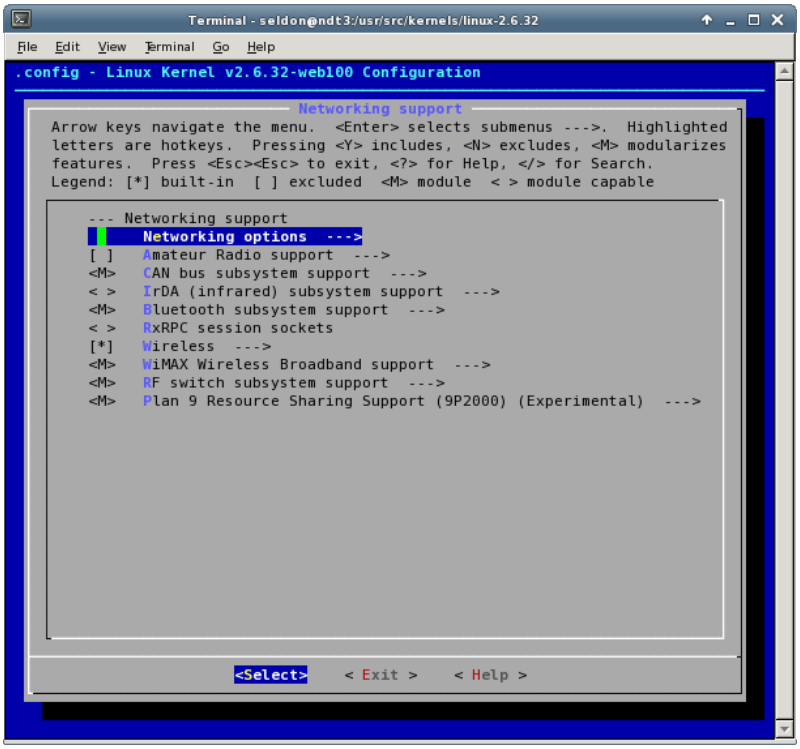
The configuration screen should load eventually. Follow the guide on the next few pages through the menuconfig GUI.

[8] www.internet2.edu/pubs/ndt-cookbook.pdf

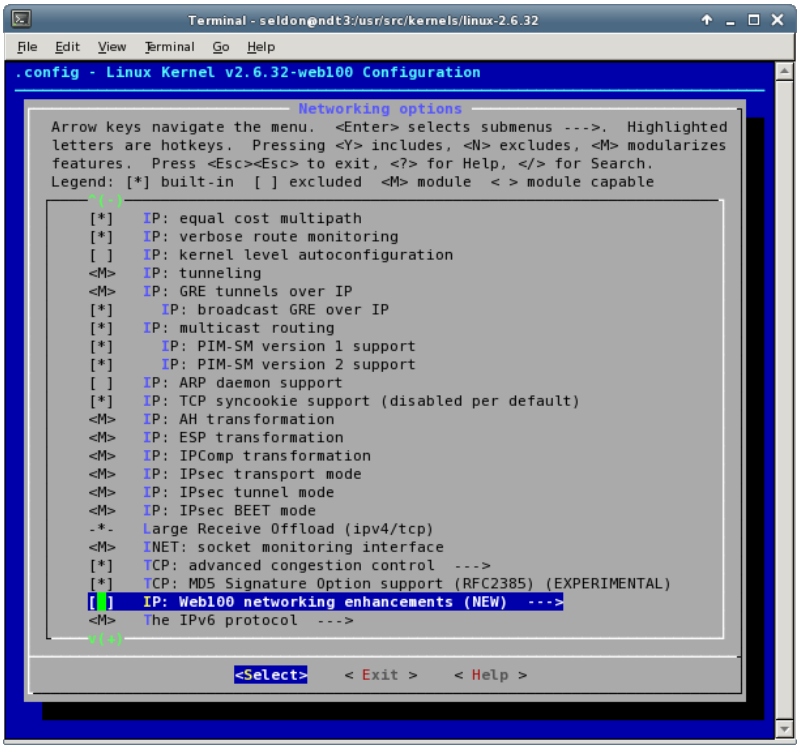
1. Navigate to *Networking Support* and hit enter.



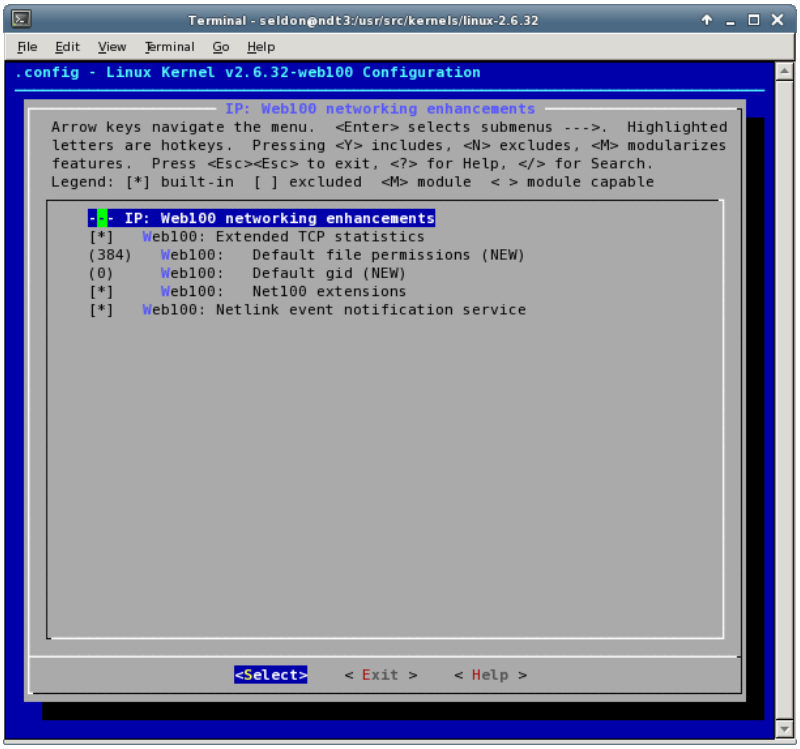
2. Navigate to *Networking Options* and hit enter.



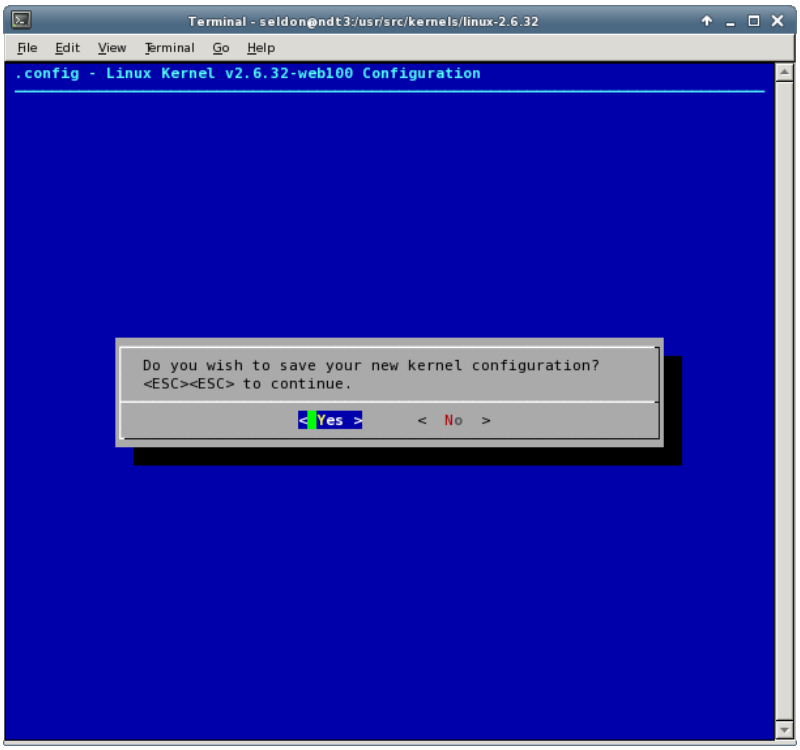
3. Navigate to *IP Web100 Networking Enhancements* and press y followed by enter.



4. Press y over *Web100 Extended TCP/IP Statistics.*
5. Ensure that the newly spawned *Web100: Default File Permissions* and *Default GID* match with the image below.
6. Press y over *Web100: Net100 extensions* and *Web100: Netlink Event Notification Service.*



7. Press escape twice.
8. Select yes and press enter when prompted to save your config,



5.4 Making RPM file

To begin making the RPM file, ensure first that you have made the RPMBUILD environment as described in this tutorial, and issue the following commands (this will take about 30 minutes):

```
$ cd /usr/src/kernels/linux-2.6.32
$ sudo make rpm
```

Now you must find where your RPM file was placed. If all went well, it should be in your RPMBUILD directory structure. However, it may end up in /root/rpmbuild. If this is the case, do the following:

```
$ su
# cp -ar /root/rpmbuild /home/seldon/
# chown -R seldon /home/seldon/rpmbuild
# exit
$ chmod 777 ~/rpmbuild
```

5.5 Installing Your RPM File

once your rpm is in your home directory:

```
$ cd /home/seldon/rpmbuild/RPMS/x86_64
$ sudo rpm -ivh kernel-2.6.32web100-1.x86_64.rpm
```

5.6 Preparing to Boot

Type the following into the terminal:

```
$ sudo mkinitrd /boot/initrd-2.6.32web100.img 2.6.32-web100
```

5.7 Configuring GRUB

To begin configuring grub, type the following into the terminal:

```
$ sudo gedit /etc/grub.conf
```

My grub.conf looked like this:

```
# grub.conf generated by anaconda
#
# Note that you do not have to rerun grub after making changes to this file
# NOTICE: You have a /boot partition. This means that
#         all kernel and initrd paths are relative to /boot/, eg.
#         root (hd0,0)
#         kernel /vmlinuz-version ro root=/dev/sda3
#         initrd /initrd-[generic-]version.img
#boot=/dev/sda
default=0
timeout=5
splashimage=(hd0,0)/grub/splash.xpm.gz
hiddenmenu
title CentOS (2.6.32-358.18.1.el6.x86_64)
      root (hd0,0)
      kernel /vmlinuz-2.6.32-358.18.1.el6.x86_64 ro root=UUID=e500f799-a5d4-485d-ba19-9a6537a94dc1 nomodeset
          rd_NO_LUKS KEYBOARDTYPE=pc KEYTABLE=us LANG=en_US.UTF-8 rd_NO_MD SYSFONT=latarcyrheb-sun16 crashkernel
          =128M rd_NO_LVM rd_NO_DM rhgb quiet
      initrd /initramfs-2.6.32-358.18.1.el6.x86_64.img
title CentOS (2.6.32-358.el6.x86_64)
      root (hd0,0)
      kernel /vmlinuz-2.6.32-358.el6.x86_64 ro root=UUID=e500f799-a5d4-485d-ba19-9a6537a94dc1 nomodeset rd_NO_LUKS
           KEYBOARDTYPE=pc KEYTABLE=us LANG=en_US.UTF-8 rd_NO_MD SYSFONT=latarcyrheb-sun16 crashkernel=128M
          rd_NO_LVM rd_NO_DM rhgb quiet
       initrd /initramfs-2.6.32-358.el6.x86_64.img
```

I changed it so that it looks like this:

```
# grub.conf generated by anaconda
#
# Note that you do not have to rerun grub after making changes to this file
# NOTICE: You have a /boot partition. This means that
#         all kernel and initrd paths are relative to /boot/, eg.
#         root (hd0,0)
#         kernel /vmlinuz-version ro root=/dev/sda3
#         initrd /initrd-[generic-]version.img
#boot=/dev/sda
default=0
timeout=5
splashimage=(hd0,0)/grub/splash.xpm.gz
#hiddenmenu
```

```
title CentOS-Web100 (2.6.32-358)
      root (hd0,0)
      kernel /vmlinuz-2.6.32-web100 root=UUID=e500f799-a5d4-485d-ba19-9a6537a94dc1 nomodeset rd_NO_LUKS
          KEYBOARDTYPE=pc KEYTABLE=us LANG=en_US.UTF-8 rd_NO_MD SYSFONT=latarcyrheb-sun16 crashkernel=128M
          rd_NO_LVM rd_NO_DM rhgb quiet
      initrd /initrd-2.6.32web100.img
title CentOS (2.6.32-358.18.1.el6.x86_64)
      root (hd0,0)
      kernel /vmlinuz-2.6.32-358.18.1.el6.x86_64 ro root=UUID=e500f799-a5d4-485d-ba19-9a6537a94dc1 nomodeset
          rd_NO_LUKS KEYBOARDTYPE=pc KEYTABLE=us LANG=en_US.UTF-8 rd_NO_MD SYSFONT=latarcyrheb-sun16 crashkernel
          =128M rd_NO_LVM rd_NO_DM rhgb quiet
      initrd /initramfs-2.6.32-358.18.1.el6.x86_64.img
title CentOS (2.6.32-358.el6.x86_64)
      root (hd0,0)
      kernel /vmlinuz-2.6.32-358.el6.x86_64 ro root=UUID=e500f799-a5d4-485d-ba19-9a6537a94dc1 nomodeset rd_NO_LUKS
           KEYBOARDTYPE=pc KEYTABLE=us LANG=en_US.UTF-8 rd_NO_MD SYSFONT=latarcyrheb-sun16 crashkernel=128M
          rd_NO_LVM rd_NO_DM rhgb quiet
       initrd /initramfs-2.6.32-358.el6.x86_64.img
```

I did this in the following steps:

1. Uncomment the *hiddenmenu* line to force grub to load.
2. Made a new kernel entry before all of the other entries:
   - CentOS-Web100 (2.6.32-358)
   - For the kernel part of the entry, I specified the web100 kernel, but then the rest of the line I had to borrow from the entry below, which is the kernel that you should be launched in now.

Now save and quit gedit, and reboot your system:

```
$ sudo reboot
```

GRUB should load, and you should select *CentOS-Web100 (2.6.32-358)* and wait for it to load. Once it loads, login to seldon.

6 Configuring Libraries

6.1 Getting and Unpacking Libraries

Getting and unpacking userland libraries:

```
$ sudo mkdir /usr/src/etc
$ cd /usr/src/tars
$ sudo wget http://www.web100.org/download/userland/version1.8/web100_userland-1.8.tar.gz
$ sudo tar -xvf web100_userland-1.8.tar.gz
$ sudo mv ./web100_userland-1.8 /usr/src/etc/web100_userland-1.8
$ sudo mv ./libpcap-1.4.0 /usr/src/etc
```

Getting and unpacking libpcap libraries:

```
$ cd /usr/src/tars
$ sudo wget http://www.tcpdump.org/release/libpcap-1.4.0.tar.gz
$ sudo tar -xvf libpcap-1.4.0.tar.gz
$ sudo mv ./libpcap-1.4.0 /usr/src/etc
```

6.2 Making and Installing Userland Libraries

Configure, make and install userland:

```
$ cd /usr/src/etc/web100_userland-1.8
$ sudo ./configure
$ sudo make
$ sudo make install
```

Configure make and install libpcap:

```
$ cd /usr/src/etc/libpcap-1.4.0
$ sudo ./configure
$ sudo make
$ sudo make install
```

Additionally, you need to make links. Add the following line to /etc/ld.so.conf

```
/usr/local/lib
```

you can do this via:

```
$ sudo nano /etc/ld.so.conf
```

now execute the command:

```
$ sudo /sbin/ldconfig
```

---

7 The NDT Program

This section follows from the NDT cookbook found here.[9]

7.1 Builing the NDT Program

Make sure that you have installed the JDK version 1.5. To Configure the system to use JDK 1.5 do the following and choose jre 1.5:

```
$ su
$ /usr/sbin/alternatives --config java
$ exit
```

Now download NDT:

```
$ cd /usr/src/etc
$ sudo wget http://software.internet2.edu/sources/ndt/ndt-3.6.2b.tar.gz
$ sudo tar -xvf ndt-3.6.2b.tar.gz
```

Now configure and make it:

```
$ cd /usr/src/etc/ndt-3.6.2b
$ sudo ./configure
$ sudo make
$ sudo make install
```

7.2 Testing the NDT program

In the NDT source directory, run the shell script provided:

```
$ cd /usr/src/etc/ndt-3.6.2b
$ sudo ./conf/create-html.sh
```

Following is how i filled out the prompt (where a blank means that I just pressed enter):

```
Enter your site name [Internet2] : test_site_01
Enter your site's location [Ann Arbor - MI] : St. Mary's - MD
Server connection info, enter 1 for 100 Mbps, 2 for 1 Gbps [2] : 2
Enter email userid [rcarlson] : hseldon
Enter email domain name [internet2.edu] : smcm.edu
Enter default subject line [Trouble report from ndt3.smcm.edu] :
The base web page 'tcpbw100.html' has now been created. You
must move this file into the ndt_DATA directory [/usr/local/ndt]
created during the 'make' process.
Do you want to install this file now? [yes] :
Enter location [/usr/local/ndt] :
```

Start server processes with the provided sample script:

[9] www.internet2.edu/pubs/ndt-cookbook.pdf

```
$ sudo ./conf/start.ndt
```

The first time I ran the script, I got the following output:

```
[1] 3515
[2] 3520
```

copy the startup script to the init.d directory:

```
$ sudo cp ./conf/ndt /etc/init.d/ndt
```