Guide for the Physics Computing Lab (PCL) *

Richard Sonnenfeld †

September 2011

Revised:
August 2007, original version
Richard Sonnenfeld February 2008
Jacob Trueblood February 2009
RS and JT September 2011
RS, August 2012

Contents

*Thanks to Dave Raymond for most of the good ideas for setting up the lab.

†Thanks to Victor Alvidrez for lots of hacking and for quick and dirty documentation.

1 Lab Overview

1.1 Physics Computing Lab Goals

Though students have their own computers, and the TCC also provides computing support, the physics department has found it useful to have a lab in Workman Center. We have observed that students use it to finish off a class report and to communicate with instructors or friends during the day without leaving Workman. We believe it fosters student collaboration, and it is used in the teaching of computer intensive physics and astronomy courses. The lab is also a location in which software unique to physics can be installed.

The computing lab has historically suffered from the well-known *Tragedy of the Commons*, in which a public resource falls into disrepair through benign neglect. In 2007, the physics department computing committee (DCC) decided to redesign the lab. Our goals were as follows:

1. Provide a modern and pleasant computing experience.
2. Make the lab as robust as possible to minimize maintainance and downtime.
3. Expose students to the principles and practice of modern scientific computing.

To achieve these goals, we did the following:

1. Replaced the hardware to achieve higher performance.
2. Installed common hardware to simplify maintainance.
3. Installed automated backups on all student files.
4. Configured all machines identically so there was no need to camp out for a favorite machine.
5. Made an all Linux laboratory.

We have placed a restriction on our laboratory. Students do not have the power to install their own software. They must e-mail the physics DCC, or ask their professor to do so. We regret this inconvenience, but we hope it is made up for in the additional stability and availability of the computers themselves.

1.1.1 Why Linux?

We assume students enter college with a strong familiarity with Windows or Macintosh operating systems. Both of these systems (intentionally) separate the user from the hardware and the software. A user is expected to purchase the applications needed to accomplish a goal. While this can be an efficient way to get routine work done, this approach to computing is counter to the needs of a researcher.

Often researchers need to create or customize scientific software to do their job. They cannot purchase it. Further, even when a scientist *can* purchase more routine software or hardware, their limited research dollars are often better spent elsewhere. Finally, an ethical researcher should not pirate for-profit software; but this is always a temptation when money gets tight in the Windows world.

While excellent programming environments are available for Windows and Macintosh systems, Linux was designed from the ground up to be fully customized. All Linux systems come with excellent programming tools, and an experienced user can configure the system itself to run on anything from tiny embedded instruments to supercomputers. The overall ethos of Linux is *do it yourself*, and we find this to be much more consistent with a researcher's orientation. We do not pretend that Linux expertise comes without a time investment, but we believe that this expertise will serve a student throughout a multi-decade career – far beyond the next release of Windows or MacOS.

Linux arose from UNIX and serious main-frame scientific computing. It is designed for reproducability and stability.

1.1.2 How can I learn Linux?

All the commands in Linux are documented with `man` (manual) and `info`. As a newbie, it is hard to understand these *man pages*, but you get it once you have some experience. A big problem is to even know what commands exist. The command `apropos` helps you find commands that are related to other commands. Even so, you need to read some tutorials or books before this is of much help to you.

A witty engineer who works at Tech pointed out that the only activity about which there is more information on the web than there is about using Linux is one that is typically clothing optional.

`http://www.ee.surrey.ac.uk/Teaching/Unix/`
A good beginner's tutorial, which also recommends some books. I started with "Linux Administration Handbook", because for your office computer, you are the system administrator. Many of these tasks are now point and click (like Windows), but with Linux it is possible to know what is going on under the pretty interface, and you can still configure every aspect of a modern Linux distribution by editing text files at the command line. It is good to know what the point and click commands are doing for you. (They basically edit the text files for you).

`http://www.cl.cam.ac.uk/teaching/2004/UnixTools/slides-4up.pdf`
A nice introduction to Linux from Cambridge U. It is a bit heavy on the programming side.

`http://www.cl.cam.ac.uk/teaching/2004/UnixTools/`
This is the Cambridge course home page. Several other links here are also interesting.

`http://www.physics.nmt.edu/f̃sonnenf/research/LinuxCourse/linuxnotes6p.rs`
Richard Sonnenfeld's notes to himself as he went through Red-Hat, Debian and Gentoo Linux over past several years. Many of the commands apply across all distributions of Linux. These notes are organized roughly alphabetically by command name, and always include usage examples. They can at least suggest commands you might want to look up.

`www.tldp.org`
The Linux Documentation Project is the definitive documentation for Linux that can carry you all the way to full expert level. It is heavy going at first. Start with the "HOW-TO's", which are more in-depth than the other documentation and more tutorial in nature.

`www.linuxquestions.org`
One of the best Linux forums. Any question you have will likely be answered, though you should try to search to see if it was *already* answered. Cryptic error messages are well-understood here. Simply search on one and you will almost certainly find a post with a fix or an explanation. Start reading the "Linux-Newbie" forum to improve your background.

2 Details for Users

2.1 Logging in and out

If you really have never used Linux, then you are not used to everything being case sensitive. Thus the user `pluto` and the user `Pluto` are considered to be two separate people. Remember this anytime a command or file-name does not seem to work anymore.

Unlike *Windows* in which the windowing system and the operating system are one and the same, for Linux you have many choices of "window manager". The default window manager that comes with Debian Linux is called "gnome", so you will see a "gnome desktop manager" `(gdm)` unless you change it.

The log-in screen is self-explanatory. Type your userid e.g. `pluto`, then your password. Note the buttons at the bottom of the login Window, `Language, Session, Actions`. `Actions` includes `Shut Down`. The `Session` button lets you change your window manager. (The other window managers aren't working yet. This manual will change when they are).

Assuming you can log in, you'll want to logout. From the gnome window system, look under the `Desktop` menu. You will see `logout pluto` second from the bottom.

From the command line for a remote session, you logout via `logout` or `exit`.

2.2 Getting an account and changing your password

To get an account send an e-mail to `dcc@kestrel.nmt.edu` stating your preferred userid and an initial password.

The PCL uses a client-server architecture. This means that most of the services come from a "main" computer or *server* (on the floor in the back corner of the lab), while the individual computers (*clients*) use the server's resources. The PCL server is named *babelfish*.

Normally one changes the password in Linux by simply typing `passwd` and responding to the prompts. However, in the Physics Computing Lab, all password changes must be done on the server.

You'll notice that *babelfish* has neither a monitor nor a keyboard. This gives a good reason to learn how to log in to one computer from another. Log in to *any* computer in the PCL using your username and initial password. Assume your username is *pluto*. Open a terminal window and then:

```
pluto@tesla:~$ ssh pluto@babelfish

The authenticity of host 'babelfish (192.168.0.1)' can't be established.
RSA key fingerprint is 2b:ea:67:19:56:2c:d2:28:8b:9c:3e:e2:ca:14:e7:3b.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'babelfish,192.168.0.1' (RSA) to the list of known hosts.

pluto@babelfish's password: XXXXX

**************************************
* Welcome to the NMT Physics PC lab.  *
**************************************
No mail.
Last login: Sun Aug 12 18:51:43 2007 from hawk.nmt.edu

pluto@babelfish:~$ passwd
Changing password for pluto
(current) UNIX password: XXXXX
Enter new UNIX password: YYYYY
Retype new UNIX password: YYYYY

passwd: password updated successfully

pluto@babelfish:~$ exit

logout
Connection to babelfish closed.
pluto@tesla:~$
```

The fancy business with `RSA key fingerprint ...` only happens the first time you log on to *babelfish* from a given client machine.

After you've changed your password on babelfish, it won't change on any of the clients immediately. The password changes are copied to the clients hourly on the hour. Your new password will show up then. You can keep working until the hour with your old password.

2.3 NFS-Mounted home directories

The home directories of individual users are all located on *babelfish*. Thus, when a user logs in to any of the client machines in the physics computing lab, their files are always present. This makes all machines in the physics computing lab equivalent. It does not matter where a student works one day, they can continue the next at a different machine with no loss or need to copy files. An added advantage of this approach is that program configurations and options like browser bookmarks remain the same at all machines.

The goal of this approach is to provide user convenience. The Tech Computer Center (TCC) also mounts all its user accounts separately from individual machines. The TCC's technical approach seems to involve considerably more system overhead than ours. In the Physics computing lab, users should experience no detectable log-on delay.

2.4 Printing

Here is a sample session in a terminal windoww to check out if your printer works.

```
pluto@tesla:~$ echo "this is a test" >testing.txt
pluto@tesla:~$ lpr testing.txt
pluto@tesla:~$ lpr: Error - no default destination
pluto@tesla:~$ lpr -Porion testing.txt
pluto@tesla:~$ echo "Hooray -- success, and you can print any filetype this way"
Hooray -- success, and you can print any filetype this way
```

The PCL uses CUPS (Common Unix Printing System). CUPS is configured with a web browser. Point your browser at `localhost:631`. On the web-page that comes up, select the "Printers" tab. At least one printer, named *orion*, should appear. Try printing a test page. If it works, you are ready to go. For more control of printers, launch `gnome-cups-manager` from the command line. You can use this to make *carlson* your default printer.

2.5 Remote Access

Though the individual clients of the PCL are not directly accessible from off-campus, the server *babelfish* is. Because all user files are on the server, they are accessible from anywhere via `ssh, scp,` or other tools. Files from babelfish may be easily downloaded to (or uploaded from) computers running Windows by using the freeware program `WinSCP`.

Should a user or administrator need remote access to an individual *nmtphysics.net* client machine, it can be reached by first connecting to *babelfish*.

2.6 Xwindows Magic

Linux is descended from UNIX, and UNIX has been solving the problems of networked scientific computing for fifty years. Not only can you `ssh` to a remote machine and get a command line, but you can run graphical applications on a remote machine. Thanks to the `X` system developed many years ago at MIT.

Here is an example.

```
pluto@wilkening:~$ ssh pluto@babelfish
pluto@babelfish's password:
**************************************
* Welcome to the NMT Physics PC lab.  *
**************************************
pluto@babelfish:~$ xeyes
Error: Can't open display:
pluto@babelfish:~$ exit
logout
Connection to babelfish closed.

pluto@wilkening:~$ ssh -X pluto@babelfish
pluto@babelfish's password:
**************************************
* Welcome to the NMT Physics PC lab.  *
**************************************
/usr/bin/X11/xauth:  creating new authority file /home/pluto/.Xauthority
pluto@babelfish:~$ xeyes
pluto@babelfish:~$
```

xeyes is fun, but not useful. However, you can run serious scientific applications remotely. This will become useful if your prof. requests some new software and there is not sufficient time to install it on all the machines of the PCL. If it is installed simply on babelfish, you can run it by this mechanism from any remote machine (and several students can do it at once).

This is also useful if you want to run software from home on *babelfish* that you have not installed. This will work, provided you have Linux (or a Mac with XWindows) at home.

2.7 X-terminals

Some applications (for example, NRAO's IRAF) require that you shell into the computer with an *X-terminal*. If you open a terminal in *gnome* (under Accessories), the default command is *gnome-terminal*. This is as good as an X-terminal for many purposes, but if you need the real McCoy, type `xterm &` from inside a gnome terminal.

2.8 Backup

The Linux automatic scheduler (`cron`) is configured to backup the `babelfish:/home` directory daily at 5 am to an external hard-drive. It is thus unlikely that you will lose your work to hard-drive failure. You should find other methods to protect yourself against you deleting your own files by accident.

3 Details for new student office computers

The new student office computers are configured very similarly to the PCL client computers, but there are differences, so at the risk of repetition, I give installation instructions specifically for office computers.

The primary difference between a PCL computer and an office computer is that the office computer keeps its /home directories inside, instead of on a server.

Because of this, backup in your responsbility on an office computer, though you should feel free to use /`rsync` and back-up to *babelfish*.

3.1 Basic Installation on Office Computers

1. Insert the netinst CD. These instructions are specifically for a *Debian Squeeze netinstall* with a relase date of 10 December 2009. First screen should say `Press F1 for help. Hit Enter to boot.` Hit Enter.

2. There are a lot of screens and menus during the install that are self-explanatory. The choices are generally obvious. In any case where they are not obvious, accept the default settings, unless these instructions specifically say otherwise.

3. Your first examples of obvious choices are to select *English* from the first menu and *US* from the second.

4. The first non-default choice is when you get the screen `Configuring the network with DHCP`. Hit *Cancel*.

5. On the next screen, select `Configure the network manually`

6. The computer may again ask `Autoconfigure?` Just say no.

7. You are requested to type the IP address. Type `192.168.0.xx` Here `xx` is the number you have been assigned by the departmental computing committee.

8. You will be asked the netmask, `255.255.255.0`, the Gateway 192.168.0.1, and the hostname. The machine should correctly guess your hostname, because you have already told the DCC what you would like it to be, and we have added it to the /etc/hosts file of babelfish.

9. You will next be asked about partitioning. Select `Guided, use entire disk`. Also select `Set up LVM`. Finally select `separate /home /usr /var /temp`. You will next be asked about partitioning. Select `Guided, use entire disk`. Also select `Set up LVM`. Finally select `separate /home /usr /var /temp`.

10. You will be asked several times if you really mean to repartition the hard-drive. You do.

11. You will be asked to pick a root password. Pick something you will remember that is also secure. The root user has total control over your computer.

12. You will be asked to pick a username. This is the usual way you will log in to this computer, so pick a username that is easy to remember and type.

13. `Use a network mirror?` Yes.

14. `Mirror country?`. You could pick `US`, but for a faster install scroll up to the top of the menu and select `Enter information manually`. For mirror, type `gryphon.nmt.edu` and for mirror directory, accept `/debian/`.

15. For system type, select `Desktop`, (the default).

16. The final screen is `Configuring xserver-org`. Accept the defaults.

17. Say YES when asked whether to install GRUB.

18. You should now have a working Debian system. Reboot once, then continue to customize it as indicated in the next sections

3.2 After the basic installation on Office computers

1. All programs in Linux may be configured by editing simple text files. Some examples of these configuration files are `cupsd.conf, sources.list, fstab, .bashrc`. You need a text editor. `vi, vim` and `emacs` are the classical Linux editors, but they are all "lifestyles" (particularly emacs!). If you want to just get on with it, use `gedit` which will remind you of Windows notepad. If you want a text-based editor that still has a smaller learning curve than `vi or emacs`, try `nano or pico`. If you are editing system files, you need to be `root`.

2. Now that you have an editor, Edit `/etc/apt/sources.list`. Delete the lines that start with "deb cdrom:". You can also add two lines that list the US software repository in addition to gryphon. Your `sources.list` should appear as follows:

```
deb http://gryphon.nmt.edu/debian/ squeeze main
deb-src http://gryphon.nmt.edu/debian/ squeeze main

deb http://ftp.us.debian.org/debian/ squeeze main
deb-src http://ftp.us.debian.org/debian/ squeeze main

deb http://security.debian.org/ squeeze/updates main contrib
deb-src http://security.debian.org/ squeeze/updates main contrib
```

3. Now that you have updated `sources.list` run `apt-get update`, to ensure that the latest software is being installed in the following steps.

4. You can use `apt-get` to install additional software We recommend you install the following:

   `apt-get install ssh` – for remote access to other computers
   `apt-get install openssh-server` – for remote access to your computer
   `apt-get install gftp` – Gnome remote file-copy utility – needed for the next step
   `apt-get install kernel-image-2.6-k7` – this upgrades your kernel to take
   advantage of the AMD Athlon
   or Turion processor (if you have one).

   Do not use 64-bit kernels. They are still flaky.

5. Your `/etc/network/interfaces` is probably correct already, but just to be sure, it should be edited to look like the following:

```
# /etc/network/interfaces
#
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
iface eth0 inet static
address 192.168.0.<type your ip here>
netmask 255.255.255.0
network 192.168.0.0
broadcast 192.168.0.255
```

```
gateway 192.168.0.1
dns-nameservers 192.168.0.1
dns-search nmtphysics.net
```

6. Run `/etc/init.d/networking restart`. If this fails to bring up the network interfaces (as indicated by `/sbin/ifconfig`) run `/etc/init.d/ifdown eth0; /etc/init.d/ifup eth0`. Even though this should do the same thing, it seems to help.

7. If the screen resolution is correct, you are finished. You will know it is not correct if the text on the logon screen appears too large and the logon application crashes.

8. Otherwise, run `dpkg-reconfigure xserver-xorg`. Accept all the default suggestions. You will only customize the screen resolution (the last choice). For the Optiquest wxga, set it to 1680x1050. It is important to DESELECT the 1680x1200 if it comes up.

9. Restart computer again

10. With the nVidia motherboards, I had a strange situation in which the mouse pointer disappeared. You could still tell it was selecting things, but you could not see the pointer! This is evidently a known (if obscure) problem with the nVidia drivers. I fixed this by editing `/etc/X11/xorg.conf` and adding the line

    ```
    Option          "HWCursor"              "off"
    ```

    so that the Device section of the file looks like this:

    ```
    Section "Device"
       Identifier          "nVidia Corporation C51G [GeForce 6100]"
       Driver              "nv"
       BusID               "PCI:0:5:0"
       Option              "HWCursor"              "off"
    EndSection
    ```

11. On boot, you will see that the text is off the left-hand side of the screen. Fix this by editing `/boot/grub/menu.lst` and appending to the end of the lines that start with *kernel* the following: `vga=792`

12. Restart computer again, and if the display is still not quite right, push the "2" button once on the monitor.

3.3 Performing backups

Even if you are not using a computer that is physically in the Physics Computing Laboratory, you can use *babelfish* to perform backups. The utility `rsync` can maintain an image of your entire home directory on *babelfish*. Here is an example script to do this. It assume that your userid is "george" on both your office computer and babelfish.

```
#/home/george/rsyn.scr
#The below copies everything from my home directory in my office
#to my home directory on babelfish.
#The below practices -- It shows what it would do but does not do it.
rsync -avun --progress --delete  /home/george/ babelfish.nmt.edu:/home/george/backup/
#Remove the -n and it does it for real
rsync -avu --progress --delete  /home/george/ babelfish.nmt.edu:/home/george/backup/
```

4 Installation Details for PCL Administrators

The Physics Computing Lab system is very close to a generic Debian 6.0 (squeeze) Linux desktop configuration. It is installed specifically from *debian-60r0-i386-netinst.iso*.

4.1 Basic installation on clients

1. Insert the netinst CD. First screen should say `Press F1 for help. Hit Enter to boot.` Hit Enter.
2. There are a lot of screens and menus during the install that are self-explanatory. The choices are generally obvious. In any case where they are not obvious, accept the default settings, unless these instructions specifically say otherwise.
3. The first non-default choice is when you get the screen `Configuring the network with DHCP`. Hit *Cancel*.
4. On the next screen, select `Configure the network manually`
5. The computer may again ask `Autoconfigure?` Just say no.
6. You are requested to type the IP address. Type `192.168.0.xx` At the time the PCL was setup, 20 IP addresses were defined. Use the current `/etc/hosts` file to assign IP addresses to new users.
7. You will be asked the netmask, `255.255.255.0`, the Gateway 192.168.0.1, and the hostname. The machine correctly guesses hostnames, assuming they are already in the `hosts` file.
8. For partitioning, select `Guided, use entire disk`. Also select `Set up LVM`. Finally select `separate /home /usr /var /temp`.
9. The `username` step is non-obvious. Do NOT pick a username of an actual user. Use something strange, like `cthullhu`. This user can later be deleted, or just ignored. No one will use it.
10. `Use a network mirror?` Yes.
11. `Mirror country?`. You could pick `US`, but for a faster install scroll up to the top of the menu and select `Enter information manually`. For mirror, type `gryphon.nmt.edu` and for mirror directory, accept `/debian/`.
12. For system type, select `Desktop`, (the default).
13. The final screen is `Configuring xserver-org`. Here, accept the defaults, but verify that 1680x1050 is among them if you have the Optiquest wide-screen monitor
14. Say YES to GRUB.
15. You should now have a working Debian system. Reboot once, then continue to customize it as indicated in the next sections

4.2 After the basic installation on clients

1. Begin by doing everything in the "after basic ... on Office computers"
2. First add two lines to `tesla.etc.hosts`. These identify *babelfish*, which will be used for nfs-mounting. The lines are:

```
#gateway server
192.168.0.1   babelfish.nmtphysics.net babelfish gateway
```

3. To nfs-mount the client's home directory, you merely need to change `/etc/fstab`. Add these two lines.

```
babelfish:/usr/local/   /usr/local      nfs     ro,auto,nosuid,nodev
babelfish:/home         /home           nfs     rw,auto,nosuid,nodev
```

4. Reboot with the nfs-mounted home directory, but log-in as root. From the root command line, shell into babelfish and run `make-trusted-ssh` followed by `munchpwd`. Exit from *babelfish* and run `munchpwd` on the new client.

5. Now reboot and you should be able to log-in and see your directory on babelfish as your home directory.

6. If you get some message about being unable to mount your home directory then either it does not exist on babelfish, was not exported, was not correctly mounted, or there is something wrong with the password files. One big problem is caused if you create a user-account for yourself on a client machine. Do not do this. It DOES mess up the password files. You can rescue yourself by editing passwd and removing any user who does not have a group ID of 100.

7. Run the script `/usr/local/squeeze-customization/apt-get-desktop.sh` to add packages that need adding. Keep this script up-to-date by adding in newly requested software.

8. Amazingly, printing on the clients just seems to work. You do not need to do anything to set it up. See my instructions to students about the web-interface for CUPS.

9. Passwords are supposed to be automatically pushed by *babelfish*, which runs munchpwd hourly. `make-trusted-ssh` allows this to run without a login. The new script `munchpwdall` on babelfish TELLS the clients to run their local munchpwd's. This means the only cronjob for password synching is on babelfish, not on the clients.

10. Every detail of further installation is too tedious to put in this procedure, but this is mostly correct. In the folder `babelfish:/usr/local/squeeze-customization` are listed every script used and every configuration file modified. Files that start with "babelfish" are for the servers. Files that start with "tesla" are for the clients. Use those files.

5 System Administration

5.1 Burning install CDs

From *babelfish* one can burn an install CD as follows.
`wodim -v dev=/dev/cdrom debian-60r0-i386-netinst.iso`

5.2 Passwords and User Administration

New users are added by the currently serving DCC system administrator. Great care was taken to simplify user management. Users need only be added to babelfish with the `adduser` command. Babelfish runs an hourly cron job which copies `/etc/passwd` to all the client machines. A custom python script running on the clients and server merges the password files to automatically add new users to each of the client machines. To make this work, `/etc/adduser.conf` had to be customized so that all users are members of the same `user` group (`GID=100`).

The process works with a call to a custom script called "munchpwd". Munchpwdall runs on the hour on the server which causes munchpwd to also run on the clients. It actually sorts the `/etc/passwd` and `/etc/shadow` files. It leaves intact all the system information and only extracts user information. It does this by looking for a group ID of 100, and assumes that everything with `GID=100` is a user. The user information only is then copied from the server to the clients. This approach is important, because many

processes also have ID's and are stored in passwd files. The `passwd` files of different machines may be slightly different, so simple copying of identical files from the server to the clients could break the systems.

In order to enable the automatic copying from server to client, the clients must all be trusted. This is the role of `make-trusted-ssh`.

babelfish:/etc/deluser.conf was modified so that if you remove a user, all files are also removed.

If something goes wrong and you have users you do not seem to be able to delete, edit /etc/passwd and /etc/shadow – but do it with care!

5.3 Custom scripts

Many of the customization scripts and customized configration files are stored in `/etc`, `/usr/local/bin`, and `/usr/local/etc` on *babelfish*. The directory `/usr/local` is also exported to the clients. The complete and current set of custom scripts and configuration files is saved in `/usr/local/squeeze-customization` on *babelfish*.

5.4 Linux Distribution

The Physics Computing Lab system is very close to a generic Debian 6.0 (squeeze) Linux desktop configuration. It is installed specifically from *debian-60r0-i386-netinst.iso*. The installation options selected are generally the defaults, with primarily the customizations indicated below. The full customization procedure is completely documented in the files saved in `/usr/local/squeeze-customization` on *babelfish*, which are also referred to in the **Installation** sections of this guide.

5.5 Network Setup

The network is a star topology with a single server, *babelfish.nmt.edu*, servicing several clients. Only the server has an external IP address. The server uses ip-masquerading to forward packets from client machines, and acts as a firewall. Client machines are assigned fixed IP addresses as indicated in `/etc/hosts`.

```
# /etc/hosts

#localhost
127.0.0.1 localhost
#give the gateway a name
129.138.42.49 babelfish.nmt.edu

#gateway server
192.168.0.1   babelfish.nmtphysics.net babelfish gateway

# nmtphysics.net users
192.168.0.10  tesla.nmtphysics.net tesla
192.168.0.11  dyson.nmtphysics.net dyson
192.168.0.12  tarter.nmtphysics.net tarter
192.168.0.13  bednorz.nmtphysics.net bednorz
192.168.0.14  randall.nmtphysics.net randall
192.168.0.15  penzias.nmtphysics.net penzias
192.168.0.16  colgate.nmtphysics.net colgate
192.168.0.17  shoemaker.nmtphysics.net shoemaker
192.168.0.18  brook.nmtphysics.net brook
192.168.0.19  oppenheimer.nmtphysics.net oppenheimer
```

```
192.168.0.20  wilkening.nmtphysics.net wilkening

...
```

5.6 Setting up a printer on the server

If the printer is connected via USB to the server, do `lsusb` to check whether the printer is detected.

Since CUPS is installed, open a web-browser with `localhost:631`

1. Go to "Printers" tab
2. Choose "Add printer"
3. For "carlson", we selected as the driver "HP LaserJet 2200 CUPS+Gutenprint v5.0.0"
4. Go to "Administration" tab
5. Under "Basic Server Settings" assure the following are selected
   `x Show printers shared by other systems`
   `x Share published printers connected to this system`

5.7 Administering print jobs

To see what jobs are currently spooled on the printer, do `lpq -P carlson`. If you find stalled jobs, or a run-away job, or if one user has several identical jobs stacked up (because of out-of-paper or some other thing that paused the printer), you can clear all jobs via `cancel -a carlson`. The `cancel` command can also be used to clear specific problem jobs.

6 Installing Matlab

`Matlab` usually installs itself to `/usr/local`. Since in our lab, `/usr/local` is a shared read-only drive on babelfish, I thought it wiser to install Matlab individually on each client on the local harddrive. However, the directory `babelfish:/usr/local/matlab/etc` does contain a `license.dat` file which you can copy to the clients for use in new installations.

Up until Matlab was installed, the bulk of the hard-drive on each client was not being used, because the home directories on the clients are covered by the nfs mount. To use the directories without creating confusion, it was important to mount them as something other than `home`. `localhome` seems a sensible choice. `/etc/fstab` on each client should be modified as follows. (The example below is for the client "dyson").

```
/dev/mapper/dyson-home /localhome      ext3    defaults    0    2
```

Obviously, the directory `localhome` needs to be created. So `mkdir /localhome; mkdir /localhome/matlab`. Next the license.dat file needs to be copied in:

```
cp /usr/local/matlab/etc/license.dat /localhome/matlab
```

`vim /localhome/matlab/etc/license.dat` to assure it looks like this:

```
# MATLAB license passcode file.
# LicenseNo: 123321   HostID: ID=123321
SERVER this_host ID=123321 27000
#The numbers 123321 should be replaced by the actual License Number
#DAEMON MLM /usr/local/matlab/etc/lm_matlab -- This is the default line
# from the Mathworks -- It needed to be changed as follows
# Change made by R.Sonnenfeld 1/2008
DAEMON MLM /localhome/matlab/etc/lm_matlab
INCREMENT TMW_Archive MLM 18 01-jan-0000 0 8DE7102237010D312935 \
        VENDOR_STRING=1 HOSTID=DEMO SN=123321
INCREMENT MATLAB MLM 18 01-jan-0000 1 6D473082BF214B654385 \
        VENDOR_STRING=classroom DUP_GROUP=UH SN=123321
```

Matlab can be installed into `/localhome/matlab` thus:

```
mount -t iso9660 /dev/cdrom /media/cdrom
/media/cdrom/install -t
```

7 Running Matlab

The license manager needs to be started once/boot. The first time I started it, I started it as `admin`. Thereafter, I discovered you had to be logged in as `admin`, but then you could start it as `/localhome/matlab/etc/lmstart`. You can also check the status as `/localhome/matlab/etc/lmstat`.

You can run Matlab with the command `/localhome/matlab/bin/matlab`. Since that is hard to remember, I made two *identical* scripts called `matlab` and `matlab.scr` and placed them in /usr/local/bin. The scripts essentially just contain the line above.

8 Full Installation Details for the Server

Every detail of server installation is too tedious to put in this procedure, but here is how to get all the details. In the folder `babelfish:/usr/local/squeeze-customization` are listed every script used and every configuration file modified. Files that start with "babelfish" are for the server.

Full details, albeit tersely written, can be found in the directory `/usr/local/squeeze-customization/server/docs` in the file `/server-reinstall.txt`. Anything not covered is covered in the various "babelfish" named configuration files.

What follows are selected details of the installation. As time permits, I will likely add to this.

Highlights of the installation are:

```
Do a vanilla Debian 'squeeze' install

At the package installation (tasksel) prompt select:
 [x] print server
 [x] file server
 [x] base system
 [x] ssh server
 [x] desktop system (optional but is helpful for configuring cups)

Run the script apt-get-server.sh
```

Configure all the many things that need configuring, nfs, cron, munchpwd, printer, quotas, firewall and backup.

8.1 Configuring nfs

Once nfs is installed (`apt-get install nfs-common nfs-kernel-server`), the only file that needs to be modified is `/etc/exports`. Here are the two key lines:

```
/home tesla(rw,no_subtree_check) dyson(rw,no_subtree_check) #... it continues
/usr/local tesla(ro,no_subtree_check) dyson(ro,no_subtree_check)
```

`/home` is exported `rw` to be used by all users. The `/usr/local` directory is exported `ro` for scripts to be shared on. `no_subtree_check` was added to make the mount go faster – I think. I've forgotten precisely the issue but it seemed to help at the time!

8.2 Configuring cron

The following lines were added to `/etc/crontab @babelfish`.

```
#cron schedule for updating password files: on the hour
1 * * * * root /usr/local/etc/munchpwd_all
# Note: munchpwd_all supercedes munchpwd on babelfish
# It triggers munchpwd on the clients and means they do not need it
# in their crontabs anymore

#cron schedule for backup: 5am
0 5 * * * root /usr/local/bin/rsync_backup
```

The first line prepares any new passwords added in the past hour to be pulled from the clients. The clients also run `munchpwd`, but after the hour, and suck up the new users.

The next line checks whether any user has gone over quota. This is rather time-consuming, so it runs at 4 am.

At 5 am, the automatic sync. to the large external USB drive takes place.

8.3 Configuring Additional Security

8.3.1 Limiting local login

The following procedure limits local login on babelfish, this will prevent users from being able to login in to a terminal as well as gdm. Append the following line to */etc/security/access.conf*:

```
-: ALL EXCEPT admin richard raymond root:LOCAL
```

Uncomment the following line from */etc/pam.d/login*:

```
account required pam_access.so
```

Append the following line to */etc/pam.d/gdm* account:

```
required pam_access.so
```

8.3.2 Timed Logout

A timed logout is used to automatically logout root from a terminal after a given amount of inactivity. Append the following line to */root/.bashrc*:

```
TMOUT=3600
```

Here, 3600 is the number of seconds to wait before logout. Note this method only applies the autologout to the root user. Also note that auto-logout fails if a document is open in *vi*. Autologout may be disabled (for a single session) by entering `export TMOUT=0` in a terminal. It can be re-enabled by entering `export TMOUT=3600`.

8.4 Rescuing the system

We have needed to rescue the system a couple of times and reinstall the system drive. Here is a quickie procedure covering everything you need to know if you are reasonably good with Debian linux already.

```
1) Do a  generic squeeze install from netinst.  Tell it repository is gryphon.nmt.edu.
2) Important ip addresss 129.138.42.49 for babelfish, gateway 129.138.42.254, DNS 129.138.250.10, 12
3) Important files to retrieve from the USB key to be kept plugged into babelfish.
   /etc/exports
   /etc/hosts
   /etc/crontab
   /etc/init.d/firewall.sh
   /etc/network/interfaces
   /etc/adduser.conf
   /etc/deluser.conf
   /etc/rc.local -- to autostart matlab.
   All of /usr/local/
   All of /localhome  for matlab
4) /etc/init.d/networking restart
   restart nfs
5) /etc/init.d/firewall.sh -- Needed for clients to see internet
6) Steal from client a current /etc/passwd, and /etc/shadow.  Then
   modify munchpwd to suck the 100 users out of /etc/passwd and /etc/shadow
   and then cat them to /etc/passwd and /etc/shadow on babelfish.
7) Need rsa key on babelfish ... ssh-keygen -t rsa (all defaults thereafter)
8) Before this step, all clients need to be running. THEN From server make-trusted-ssh
9) Printer setup.  Use CUPS printer setup.
```