Hands-On UNIX I Exercise:

Exercise 1: Users.

Root can create a new user account on your machine. To do that we shall use the `pw` command.

So first pick a username that you want to use. For example, use your first name, last name, a combination of both, or whatever you prefer. For purposes of this example we use *username*. So, to add a new user to your system type:

```
# pw useradd username -m
 -s /usr/local/bin/bash
```

This creates the user *username*. The "-m" says add a new home directory of /usr/home/*username* and copies files from

/usr/share/skel to the new user's home directory. The default shell "bash" will be used.

Next you need to set the password fro the new userid, otherwise you won't be able to use the account properly. Some quick guidelines to picking a secure password:

· Password should be at least 6 characters in length

· Dont' use any words in a dictionary in any language – forwards or backwards.

· The password should contain a mix of letters, numebrs, and upper and lower-case letters.

For example, your root password is sufficiently secure. Change your new user's password using the command:

```
# passwd username
```

Enter your password of choice and confirm it. Please note that the terminal will NOT echo anything back to you as you set it.

Test your new user account by logging in on another terminal with your username and password.

Next, add a username for the instructors to be able to check the configuration of your computer using:

```
# pw useradd inst -G wheel -m
 -s /usr/local/bin/bash
```

Set the inst password to instpass – we shall log in and change it to something secure using:

```
# passwd inst
```

Exercise 2: Manipulating files and directories

Be careful in this exercise. Running as root (the Unix administrative account) means that you can easily damage your system, so we ask that you log out of your root account and log in as your own user account instead.

To logout, type

```
# exit
```

If you are unsure of how to proceed ask your instructor or assistants for help before continuing.

The first command that we are going to use is *man*, this is short for "man"ual. Read about each command to see the range of options that exist.

Many of the basic commands we'll be practicing are built in as part of your shell environment (that is you won't find a binary/program file for *cd*). To read about commands like *cp, cd, ls, mv, rm* in more detail you can just type:

```
$ man builtin
```

And, for a command like *ls* you can type:

```
$ man ls
```

And, even for a built-in command you can just type "man commandName", or something like:

```
$ man cd
```

and this will open the "builtin" man page for you.

If you have problems exiting from "man" press the "q" key. Also, you can use the keyboard arrows to move around in the descriptions.

As we move around directories an extremely useful command is *pwd*, which return the working directory name you are in. So, if you get lost just type:

```
$ pwd
```

We'll do this from time to time as we use directory commands.

Simplified Map of Unix Directory Tree

```
/                   ("root directory")
/etc                (contains configuration files)
/etc/rc.d           (contains system startup
scripts)
/root               (user root's home directory)
/tmp                (place to store TeMPorary
files)
/usr                (contains the majority of user
utilities, applications, home directories)
/usr/home           (home directories for users on
the system*)
/usr/local/etc      (contains third-party
configuration files and startup scripts*)
/var                (multi-purpose log, temporary,
transient, and spool files location)

*Different from Linux
```

For details on the (almost) complete directory tree under Unix/Lunx type "man hier"

Command Glossary

```
cd                  Change Directory
ls                  LiSt files
mkdir               MaKe DIRectory
mv                  MoVe files
pwd                 Print Working Directory
rm                  ReMove files
rmdir               ReMove DIRectory
touch               Update date on file/Create new
empty file if none exists
```

Note: There are some special files on UNIX, which are '.' and '..':

- '.' - "this" directory (the one you are "in" as reported by the pwd command
- '..' - the parent directory, i.e.: the one above

Now we are ready to practice a bit with the commands:

```
$ cd /
$ pwd
$ ls
$ ls -la
$ cd /tmp
$ cd ..
$ pwd
$ cd tmp
```

What's going on here? If you don't understand, ask.

```
$ cd (take you back to your home
directory)
$ pwd
$ touch text.txt
$ cp text.txt new.txt
$ mv text.txt new.txt
```

What's happening now? If prompted to overwrite, respond "y". Note that "username" is the name of the user account you created in the first exercise.

Now watch what happens if you try to copy a file on itself.

```
$ cp new.txt /home/username/.
$ cd ../../home/username
$ cd (to return to our home directory)
$ cp new.txt new.txt.bak
```

The tab key makes life much easier. Now type:

```
$ cd
$ mkdir tmp
$ mv new.* tmp/.
$ ls
```

Finally, we are going to remove the directory that contains the two archives.

```
$ cd tmp
$ rm *
$ cd ..
$ rmdir tmp
```

You can force this using a command like this:

```
$ rm -rf tmp
```

The use of "rm -rf" is **very dangerous!**, and, naturally, very useful. For example, if you are "root" and you type "rm -rf /*" this would be the end of your server. This commands says "remove, forcibly and recursively, everything" - Or, if you start in the root directory (/), remove all files and directories *without asking* on the entire server. If you want to use "rm -rf *" always take a deep breath and check where you are first (really, do this!):

```
$ pwd
```

First this says in what directory you are. If you are mistaken, then you have the opportunity to not remove files that you might really need.

Exercise 3: Getting Help

Now that you have FreeBSD up and running you probably want to have a way to figure out how to use it when there are no instructors around, or other FreeBSD-knowledgeable people. First and foremost, make it a habit to read the man pages (MANual pages) for the commands that you use. You might be surprised at some of the things these commands can do! In any case, as you have seen, when in doubt about what a command does, or how it works simply type:

```
$ man command
```

Optionally, you might find some additional information for some commands typing:

```
$ info command
```

And, to get information about both these commands type:

```
$ man man

$ info info
```

After this there is a large amount of documenation available to you in serveral ways. For instance, if you look in:

/usr/share/doc

you will find multiple FreeBSD articles in various languages available to you. In addition, the FreeBSD Handbook is available here
under /usr/share/doc/handblook. If you wanted to start reading the FreeBSD Handbook from your local hard drive you could use a web browser on the
file /usr/share/doc/handbook/index.html. In URL format in your web browser this looks like:
Note: we are likely not using a graphical interface at this time.

```
file:///usr/share/doc/handbook/index.html
```

There is, also, a FreeBSD FAQ file you can look at here:

```
file:///usr/share/doc/faq/index.html
```

After this, have a look at some of the available articles by doing the following in a terminal window.

```
$ cd /usr/share/doc/en/articles

$ ls
```

Finally, there are several papers available as well. Try:

```
$ ls /usr/share/doc/papers
```

If you have a network connection, then you can go to `http://www.freebsd.org/docs.html` for even more information. Become accustomed to the idea of using man to get specific information about commands, and then using these additional resources to get an overview of entire sub-systems of the FreeBSD operating system. If you want to read the FreeBSD Handbook online it is available here `http://www.freebsd.org/doc/en_US.ISO8859-1/books/handbook/index.html`. If you want to understand FreeBSD concepts in more depth, then the FreeBSD Handbook is *really* where you should begin.

Exercise 4: Editing files using vi.

We are going to set up our network by editing a file (surprise). You must be root for this exercise.

The vi editor uses "modes"

This is a critical point. The vi editor has two modes. These are:

- Command mode
- Input mode

To go back and forth between these modes when you are in vi you can press:

- ESCape key (command mode)
- Letter "i" for Input mode, letter "o" for input mode with newline below cursor

Remember this as it is confusing. The easiest thing to do when you get confused in vi is to press the ESCape key a

couple of times and start over.

Now let's do the following:

```
$ cd /home/username
$ touch temp.txt
$ vi temp.txt
```

Now you are in vi. Press the "i" key to switch to input mode.

Type something like, "VI is great! I think I'll be using vi from now on instead of Microsoft Word."

Press ENTER to add lines. Type some more stuff, whatever you like.

Here is a short list of vi commands:

Open: vi fn, vi -r fn, vi + fn, vi +n fn, vi +/pat fn
Close: :w, :w!, :wq, :wq!, :q, :q!
Movement: h,j,k,l, w, W, b, B, :n
Editing: i, o, x, D, dd, yy, p, u
Searching: /pattern, ?pattern, n, N

OK, let's save the file that you are in. To do this do:

Press the ESCape key to get in to command mode

Press ":" to get ready to issue a file command

Type "w" and press ENTER to save your file.

Press ":" to get back to the prompt to issue a file command

Press "q" to quite the file

Instead of the multiple steps you could have type ":wq" to write and quite at the same time. If you need to quit a file without saving it *after* you've made changes, then you press **:q!**. For many people this is the most important command to remember in vi :-).

Now make sure you are root and type:

vi /etc/rc.conf

add the following lines at the bottom (if the file exists) using "o" to open a new line under the cursor and put vi in "input" mode. Replace NN in the lines below with the (number of your pc + 10) e.g pc1.e0.ws.afnog.org should have 196.200.218.10/24 ; pc15.e0.ws.afnog.org shoud have 196.200.218.25/24.

ifconfig_rl0-"196.200.218.NN/24"

defaultrouter="196.200.218.254"

sshd_enable="YES"

Save and quit the file with :wq

Exercise 5: Editing files with ee:

EE is another common editor that is smaller than vi (i.e less powerful) but small (and old) enough that lots of UNIXes will come with it installed by default. We shall now (as root) use ee to create the file

/etc/resolv.conf

ee /etc/resolv.conf

The screen you see has two sections. One is some online help for using ee, the other is the actual file. ee does **not have modes; when you type it goes straight into the file. so type the following three lines:**

search   e0.ws.afnog.org

nameserver 196.200.218.250

nameserver 196.200.223.1

(i've put extra spaces in the above lines so you can see where they are; just use a single tab)

save the file by typing CTRL-C and typing exit at

the prompt that follows. (to quit without saving in case you made a mistake type quit instead of exit).