6.033 Computer System Engineering
Spring 2009

M.I.T. DEPARTMENT OF EECS

6.033 - Computer System Engineering **UNIX Hands-On Assignment**

Hands-on 1: The UNIX File System

This hands-on assignment is due at the beginning of Recitation 4. Before attempting this hands-on, you should read Sections III and IV of *The UNIX Time-Sharing System*, which is also assigned for this recitation.

This hands-on assignment will guide you in exploring the user visible layers of the UNIX file system.

I. Warmup

Log into an Athena machine. (athena.dialup.mit.edu is ok) and get access to the Athena command prompt. If you logged into an Athena dialup machine, you should see the prompt in your ssh client. If you logged into an Athena workstation you may need to open a terminal window. The prompt should look something like this:

```
athena%
```

Use the `add` command to gain access to the 6.033 utilities you will use in this assignment.

```
athena% add 6.033
```

Some of the questions on this hands-on assume your UNIX shell is set to `tcsh`, the (current) default on Athena. Run the following command to ensure you're running the appropriate shell.

```
athena% tcsh -f
```

II. Looking Around

By default, the system will set your *current working directory* to your *home directory*, your personal name space where you can create your own files, directories and links. You can view the contents of your current working directory with the `ls` command.

```
athena% ls
```

To learn more about the contents of the directory, you can use `ls` with the `-l` option or any of the other options listed in the ls manual page; you can read the manual by typing `man ls`.

This course makes use of Athena, MIT's UNIX-based computing environment. OCW does not provide access to this environment.

```
athena% ls -l
athena% man ls (read the man page)
```

Use the `pwd` command to learn the *absolute path* of your current working directory. This will tell you where you are in the directory name space even if you move around in the directories.

```
athena% pwd
```

The output of `pwd` reveals where the Athena administrators store your home directory. For example, `/afs/athena.mit.edu/user` tells us that the your home directory is stored as a user in the Athena name space.

`stat` is a program that reports detailed information about files including its inode number, reference count (links), file type and other metadata. To use it, type `stat` followed by a file name at the command prompt. Run `stat` on your home directory:

```
athena% stat .
```

III. Creating Directories and Files

Now create a directory named `6.033-handson1` in your home directory for this assignment using the `mkdir` command. You can learn more about the `mkdir` command by typing this:

```
athena% man mkdir (read the man page)
```

Use `ls` to verify that the new directory exists. Now change your current working directory to your new `6.033-handson1` directory using the `cd` command and verify that your working directory has changed using `pwd`.

```
athena% cd 6.033-handson1
athena% pwd
```

View the contents of your new directory using `ls -a -l`. `ls` normally hides the directories "." and "..", but the `-a` option forces it to show them.

```
athena% ls -a -l
```

Question 1: Change to the '.' entry in your new directory. What happens to your working directory? Next, change to the '..' entry. What happens to your working directory?

Question 2: Describe a scenario where you might need to use the '.' directory.

Change your current directory back to your new `6.033-handson1` directory and stat the current directory; note the reference count. Now create a couple files in your new directory using the `touch` command and stat the directory again.

```
athena% stat .
```

```
athena% touch foo bar
athena% ls
athena% stat .
```

Question 3: What has changed in the stat output and why has it changed?

Now create a subdirectory `baz` in `6.033-handson1` using `mkdir` command and stat the directory once more.

```
athena% mkdir baz
athena% stat .
```

Question 4: What has changed in the stat output this time and why has it changed? Why does the reference count only change when you create a new directory?

IV. Creating Links

The `ln` command can create both hard links and soft (symbolic) links. Read the man page for more information.

```
athena% man ln
```

First stat your file `foo` and read the output information. Then create a hard-link named `foo-lnk` and stat both `foo` and `foo-lnk`.

```
athena% stat foo
athena% ln foo foo-lnk
athena% stat foo
athena% stat foo-lnk
```

Note that everything about `foo` and `foo-lnk` is identical except for their directory entries. If you modify `foo` you will see the modifications in `foo-lnk`.

```
athena% echo Hello >> foo
athena% cat foo-lnk
```

Now create a symbolic link to `foo` and note that the symbolic link differs from the original file in several ways. Creating the symbolic link does not increase the reference count of `foo` and the symbolic link does not share an inode with `foo`.

```
athena% stat foo
athena% ln -s foo foo-slnk
athena% stat foo
athena% stat foo-slnk
```

Question 5: One reason for supporting symbolic links is to allow linking from one disk to another disk, but you can also create a symbolic link to a file on the same disk. Name one advantage and one disadvantage of using symbolic links on the same disk.

Now cd to the 6.033 Athena locker with the command:

```
athena% cd /mit/6.033
```

Your home directory is accessible by the path /mit/YOUR_USERNAME (replace YOUR_USERNAME with your username). Try to change to your home directory with the command:

```
athena% cd ../YOUR_USERNAME
```

Question 6: What happened? Why?

Like your home directory, the 6.033 locker's absolute path is much longer than /mit/6.033 and /mit/6.033 is only a symbolic link. You can learn the absolute path name by typing pwd or by typing:

```
athena% ls -l /mit
```

Question 7: You can reach the 6.033 locker with the path /afs/athena.mit.edu/course/6/6.033. Why does Athena also provide the /mit/6.033 symbolic link?

Question 8: How would you change the file system to make this command (cd /mit/6.033; cd ../YOUR_USERNAME) actually change to your home directory?

V. The Search Path

The UNIX shell has a configuration variable named PATH that tells the shell where to look in the file system for programs we type on the command line. You can see your PATH variable with this command:

```
athena% echo $PATH
```

You can configure your shell to search the current working directory by adding '.' to the PATH using these commands:

```
athena% setenv OLDPATH $PATH
athena% setenv PATH .:$PATH
```

Now, cd to /mit/6.033 and run our demo program using the following commands. Your shell will find the "demo" program because it is in your working directory.

```
athena% cd /mit/6.033
athena% demo
```

Oh no, something terrible just happened! Just kidding, the demo program did not actually do anything. Verify that nothing happened with ls.

```
athena% ls -l
```

Question 9: What happened to `ls`? Why isn't it listing files like it did before? (Hint: set your path back to its original state: `setenv PATH $OLDPATH`)

Usually, it is a bad idea of have '.' in your PATH, because it is easy to run the wrong programs by accident. Instead, you can use '.' explicitly to run programs in your working directory like this:

```
athena% ./demo
```

Question 10: How long did this assignment take you to complete up to this point?

VI. Extra Credit

Question 11: Appendix 2.A describes how (in UNIX) a file's inode maps to the file's data blocks using direct pointers and indirect blocks. An alternative strategy, as used in FAT, stores a file's data as a linked list of FAT entries, each corresponding to a block (*cluster*). Name one advantage and one disadvantage of the linked list strategy.

Question 12: Some shells like `bash` try to make '..' always work properly, namely, `cd /mit/6.033; cd ../$USER` will place you in your home directory. Does `bash` always get this behavior correct?