**Chapter 2**

**File Systems and Logging in**

**1. File System**

The Unix file system includes directories containing files and directories, each directory of which can contain yet more files and directories. Our own home directory, however, probably doesn’t contain any directories (except . and .. ofcourse), which prevents us from exploiting what we call the virtual file cabinet of the file system.

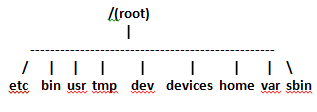
Managing filesystems is one of the important tasks of Systems Administrator. Unix filesystems when originally designed to favor its users. Unix filesystems look like an inverted tree, so that a root is at top and then branches underneath it. Major tasks for Systems administrators are:

**Tasks**

* Making files available to users.
* Managing and monitoring the system's disk resources.
* Protecting against file corruption, hardware failures, user errors through backup.
* Security of these filesystems, what users need and have access to which files.
* Adding more disks, tape drives, etc when needed.

When Unix operating systems is intalled, some directories depending upon the Unix being installed are creaed under / (or root) , such as /usr /bin /etc /tmp /home /var. As we can attach many disks to Unix operating systems, it is recommended that when working in professional environment / and /usr /home are on separate disks.

Same time the files are organized into a tree structure with a root named by the character '/'. The first few levels of the tree look like this:



We will have a look on each and every file or directory in this hierarchy:

|  |  |
| --- | --- |
| etc | Contains all system configuration files and the files which maintain information about users and groups. |
| bin | Contains all binary executable files (command that can be used by normal user also) |
| usr | Default directory provided by Unix OS to create users home directories and contains manual pages |
| tmp | System or users create temporary files which will be removed when the server reboots. |
| dev | Contains all device files i.e. logical file names to physical devices. |
| devices | Contains all device files i.e. physical names of physical devices |
| home | Default directory allocated for the home directories of normal users when the administrator don’t specify any other directory. |
| var | Contains all system log files and message files. |
| sbin | Contains all system administrator executable files (command which generally normal users don’t have |

**2. How do we login to Unix system?**

We need to have an account created by unix administrator for that particular system. For example if We want to login to unix systems named **will-1** , open up a new session to **will-1** and at prompt enter the user name and password as supplied by Administrator.

**How to connect**

When We connect to a UNIX computer and when We log in locally using a text-only terminal, We will see the prompt:

**Login:**

At this prompt, type in the username and press the enter/return/ key. Remember that UNIX is case sensitive (i.e. Will, WILL and will are all different logins). We should then be prompted for the password:

**Login : will-1**

**password :**

Type the password in at the prompt and press the enter/return/ key. Note that the password will not be displayed on the screen as we type it in.

If We mistype the username or password we will get an appropriate message from the computer and We will be presented with the **login:** prompt again. Otherwise We should be presented with a shell prompt which looks something like this:

**$**

To log out of a text-based UNIX shell, type "exit" at the shell prompt (or if that doesn't work try "logout "; if that doesn't work press **ctrl-d**).

**3. Changing the password**

The UNIX command to create/change the password is ***Passwd***:

**$ passwd**

The system will prompt us for our old password, then for our new password. To eliminate any possible typing errors we have made in our new password, it will ask We to reconfirm our new password.

Remember the following points when choosing our password:

* Avoid characters which might not appear on all keyboards, e.g. '£'.
* Make it at least 7 or 8 characters long and try to use a mix of letters, numbers and punctuation.

Note: One of the things we should do when we log in for the first time is to change our password.

**4. File Types**

There are four types of files in the Unix file system.

**4.1. Ordinary Files**

An ordinary file may contain text, a program, or other data. It can be either an ASCII file, with each of its bytes being in the numerical range 0 to 127, i.e. in the 7-bit range, or a binary file, whose bytes can be of all possible values 0 to 255, in the 8-bit range.

**4.2. Directory Files**

Suppose that in the directory x we have a, b and c, and that b is a directory, containing files u and v. Then b can be viewed not only as a directory, containing further files, but also as a file itself. The file b consists of information about the directory b; i.e. the file b has information stating that the directory b has files u and v, how large they are, when they were last modified, etc.

**4.3. Device Files**

In Unix, physical devices (printers, terminals etc.) are represented as ``files.'' This seems odd at first, but it really makes sense: This way, the same read() and write() functions used to read and write real files can also be used to read from and write to these devices.

**4.4. Link Files**

Suppose we have a file X, and type

**$ ln X Y**

If we then run **ls** , it will appear that a new file, Y, has been created, as a copy of X, as if we had typed

**$ cp X Y**

However, the difference is the **cp** does create a new file, while **ln** merely gives an alternate name to an old file. If we make Y using **ln**, then Y is merely a new name for the same physical file X.

**5. Creating Directories and Files**

**5.1. Create a File**

**$ cat > file Enter text and end with ctrl-D**

**$ vi file Edit file using the vi editor**

**5.2. Display File Contents**

**$ cat file display contents of file**

**5.3. Make a Directory**

**$ mkdir directory-name**

**5.4. Making a sub Directory**

To create a directory (dir1) which is having child sub directory(dir2) also in a single command.

**$ mkdir –p dir1/dir2**

**5.5 Denoting Paths**

We can mention the SOURCE path of DESTINATION path in any format as specified below

|  |  |
| --- | --- |
| **cd /** | go to the root directory |
| **cd** | go to your login (home) directory |
| **cd ~username** | go to username's login (home) directory not allowed in the Bourne shell |
| **cd ~username/directory** | go to username's indicated directory |
| **cd ..** | go up one directory level from here |
| **cd ../..** | go up two directory levels from here |
| **cd /full/path/name/from/root** | change directory to absolute path named note the leading slash |
| **cd path/from/current/directory** | change directory to path relative to here. note there is no leading slash |