UNIX for Beginners

Presentation · January 2019

DOI: 10.13140/RG.2.2.22349.90089

CITATIONS
0

READS
705

2 authors:

Dhananjay Dasiram Jade
International Centre for Genetic Engineering and Biotechnology
7 PUBLICATIONS **25** CITATIONS
SEE PROFILE

Dinesh Gupta
International Centre for Genetic Engineering and Biotechnology
131 PUBLICATIONS **1,841** CITATIONS
SEE PROFILE

UNIX for Beginners

A beginners guide to the **Unix** and **Linux** operating system.

Dhananajay D. J., Dinesh Gupta

What is UNIX?

UNIX® License PlateUNIX is an operating system which was first developed in the 1960s, and has been under constant development ever since. By operating system, we mean the suite of programs which make the computer work. It is a stable, multi-user, multi-tasking system for servers, desktops and laptops.

UNIX systems also have a graphical user interface (GUI) similar to Microsoft Windows which provides an easy to use environment. However, knowledge of UNIX is required for operations which aren't covered by a graphical program, or for when there is no windows interface available, for example, in a telnet session.

- **Types of UNIX**

The Linux PenguinThere are many different versions of UNIX, although they share common similarities. The most popular varieties of UNIX are Sun Solaris, GNU/Linux, and MacOS X.

❖ The UNIX operating system

The UNIX operating system is made up of three parts; **the kernel, the shell and the programs**.

The kernel

The kernel of UNIX is the hub of the operating system: it allocates time and memory to programs and handles the filestore and communications in response to system calls.

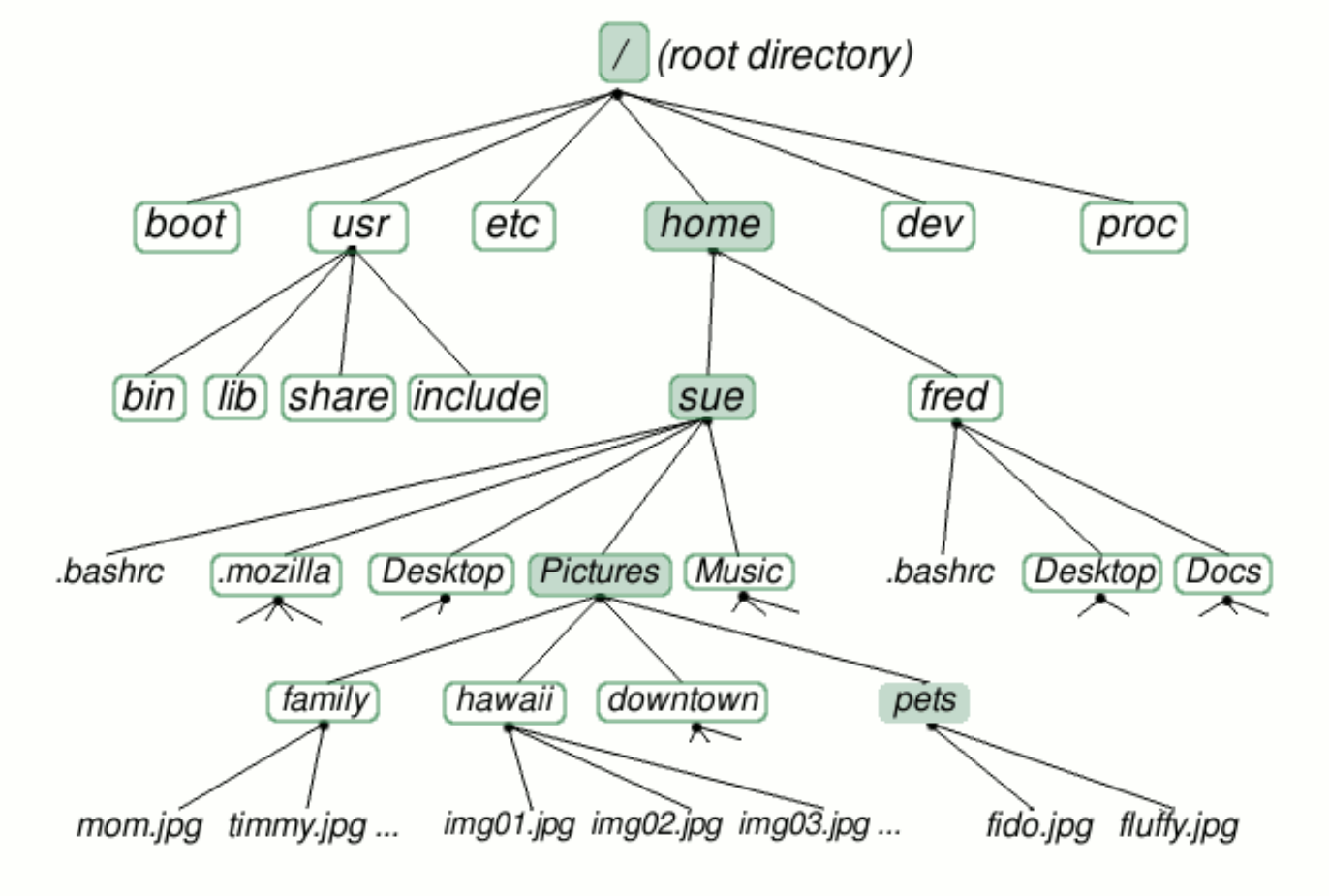
As an illustration of the way that the shell and the kernel work together.

The shell

The shell acts as an interface between the user and the kernel. When a user logs in, the login program checks the username and password, and then starts another program called the shell. The shell is a command line interpreter (CLI). It interprets the commands the user types in and arranges for them to be carried out. The commands are themselves programs: when they terminate, the shell gives the user another prompt (% on our systems).

The Directory Structure

All the files are grouped together in the directory structure. The file-system is arranged in a hierarchical structure, like an inverted tree. The top of the hierarchy is traditionally called **root** (written as a slash /)



UNIX is a command line interface. So, you write the commands you want executed.

Getting unix

There are many different ways to get unix.

- *Mac*: Mac is unix. Use the applications **X11** or **Terminal**.
- *Windows*: Download and install **MobaXTerm (**https://mobaxterm.mobatek.net/download-home-edition.html **)**.
- *Linux*: Use the applications **Terminal** default installed.

❖ Listing files and directories

ls (list)

To find out what is in your home directory.

To list all files in your home directory including those whose names begin with a dot, type

ls -a

As you can see, **ls -a** lists files that are normally hidden.

❖ Making Directories

mkdir (make directory)

To make a directory in you folder.

mkdir <Folder Name>

❖ Changing to a different directory

cd (change directory)

The command cd directory means change the current working directory to 'directory'.

The current directory (.)

In UNIX, (.) means the current directory, so typing

cd .

The parent directory (..)

(..) means the parent of the current directory, so typing

cd ..

❖ Pathnames

pwd (print working directory)

Pathnames enable you to work out where you are in relation to the whole file-system.

pwd

❖ Copying Files

cp (copy)

cp *file1 file2* is the command which makes a copy of **file1** in the current working directory and calls it **file2**

❖ Moving files

mv (move)

mv *file1 file2* moves (or renames) **file1** to **file2**

To move a file from one place to another, use the mv command.

❖ Removing files and directories

rm (remove), rmdir (remove directory)

To delete (remove) a file, use the **rm** command.

❖ Displaying the contents of a file on the screen

clear (clear screen)

Before you start the next section, you may like to clear the terminal window of the previous commands so the output of the following commands can be clearly understood.

Clear

cat (concatenate)

The command cat can be used to display the contents of a file on the screen. Type:

cat file.txt

less

The command less writes the contents of a file onto the screen a page at a time. Type

less file.txt

head

The **head** command writes the first ten lines of a file to the screen.

head file.txt

First 5 line

head -5 file.txt

tail

The **tail** command writes the last ten lines of a file to the screen.

tail file.txt

❖ Searching the contents of a file

Simple searching using **less**

Using less, you can search though a text file for a keyword (pattern). to search through **file.txt** for the word **'Biology'**, type

less file.txt

then, still in less, type a forward slash [/] followed by the word to search **/Biology**

grep

grep is one of many standard UNIX utilities. It searches files for specified words or patterns. then type

grep Biology file.txt

The **grep** command is case sensitive; it distinguishes between Biology and biology.

To ignore upper/lower case distinctions, use the -i option, i.e. type

grep -i biology file.txt

wc (word count)

A handy little utility is the **wc** command, short for word count. To do a word count on **file.txt**, type

wc -w file.txt

To find out how many lines the file has, type

% wc -l file.txt

❖ File system security (access rights)

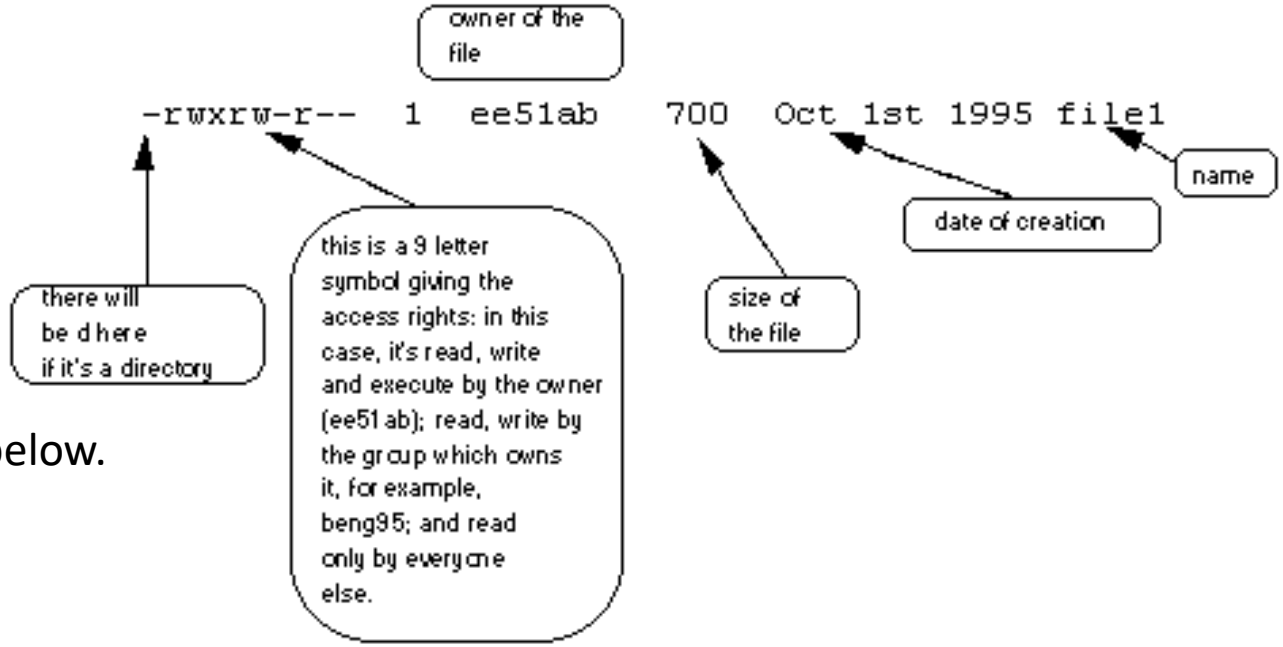
In your unix directory, type

ls -l (l for long listing!)

You will see that you now get lots of details

about the contents of your directory, similar to the example below.

File and directory access rights



- In the left-hand column is a 10 symbol string consisting of the symbols **d, r, w, x, -**, and, occasionally, **s or S**. If **d** is present, it will be at the left hand end of the string, and indicates a directory: otherwise - will be the starting symbol of the string.
- The 9 remaining symbols indicate the permissions, or access rights, and are taken as three groups of 3.
- The **left group of 3** gives the file permissions for the **user** that owns the file (or directory) (ee51ab in the above example);
- The **middle group** gives the permissions for the **group of people** to whom the file (or directory) belongs
- The **rightmost** group gives the permissions for **all others**.

❖ **Access rights on files.**

r (or -), indicates read permission (or otherwise), that is, the presence or absence of permission to read and copy the file

w (or -), indicates write permission (or otherwise), that is, the permission (or otherwise) to change a file

x (or -), indicates execution permission (or otherwise), that is, the permission to execute a file, where appropriate

❖ **Access rights on directories.**

r allows users to list files in the directory;

w means that users may delete files from the directory or move files into it;

x means the right to access files in the directory. This implies that you may read files in the directory provided you have read permission on the individual files.

❖ Changing access rights

chmod (changing a file mode)

Only the owner of a file can use chmod to change the permissions of a file. The options of chmod are as follows

For example, to remove read write and execute permissions on the file **dhananjay** for the group and others, type

chmod go-rwx dhananjay

This will leave the other permissions unaffected. To give read and write permissions on the file **dhananjay** to all,

chmod a+rw dhananjay

- **Merging files.**

  paste <file1> <file2>

  Concentrates corresponding lines of the given input files file1, file2, and so on. It treats each file as a column or columns of a table and pastes them together horizontally (parallel merging).

- **Sorting files.**

  sort <filename>

  Sorts the lines in the file alphabetically. Some options will make the sorting be numerically and/or on certain columns in the file. Extremely useful.

- **Downloading files from the internet.**

  wget <URL>

  An URL is the link you see in the browser. wget will download the page/data for you in current directory.

- **Printing text to the screen.**

  echo <text>

- **Getting time and date information.**

  date

- **Stopping runaway programs.**

  kill <PID>

  Killing the program is quite often the way to handle a badly behaving program. Then use the sure kill; kill -9 <PID>

- **Logging in to remote computers.**

  ssh -X <username>@<hostname>

  This will start a shell on the remote computer. You need to have an account on the remote machine first.

- **Transferring files to and from other computers.**

  ftp <hostname>

  After issuing the command, you will be asked for your username and password on the remote computer. You use the keywords "put" and "get" for file transfer, and "help" for getting help.

- **Editing text files.**

  nedit <filename>

  This is a very user friendly text editor. It is easy and intuitive to use and it has menus and all. It is used for making Perl programs, too. There are other editors like gedit or emacs. You can use the one you are used to. However, avoid Notepad or word processors like MS Word.

❖ **Text based editors**

vim <file name>

emacs <file name>

nano <file name>

❖ **Grahpical editors (using X)**

nedit <file name >

gedit <file name >

Reference

http://www.ee.surrey.ac.uk/Teaching/Unix/index.html