

Linux Overview

Logging On

Requires a user name and a password.

Only the root user (i.e. the administrator) can create a new user.

Linux Overview - users

There are two different types of users:

Ordinary user: limited in what he / she can do. Usually has permission to read (view files), write (create & save files) and execute (run executables) in his /her home directory. Users usually have permission to run certain commands but would not have access to parts of the system.

Root user (also known as the superuser): is an administrator and can do everything. The root user has control over what others users can do and which parts of the system they have access to.

Linux Overview – users (cont.)

Every Linux user has the following attributes:

username: used to log into the system

password: used to authenticate the user

user id: unique integer generated by the system

group id: a unique integer generated by the system

home directory: where the user can store his / her files

default shell: the default terminal shell

Linux Overview - groups

Linux users can belong to a group (or several groups)

It's possible to enable a group of users to have certain privileges while denying others. For example, some users may have access to specific software while others do not.

When a new Ubuntu user is created a group (with the same name as the user) is also created.

In some Linux distributions, all new users are added to a common group (e.g. in Suse Linux).

Linux Configuration files

Linux stores information about users in the following files:

`passwd` – has an entry for each user. It contains information about each user such as user id, home directory, etc.

`shadow` – contains the encryption of users' passwords (can only be read by root)

`group` – lists groups and group members.

All these files are in the `/etc` directory

Linux passwd file

Example:

gerard:x:1000:1000:Gerard Mac Sweeney,,,:/home/gerard:/bin/bash

joe:x:1001:1001:joe,,,:/home/joe:/bin/bash

tony:x:1002:1002:tony,,,:/home/tony:/bin/bash

kate:x:1003:1005::/home/kate:/bin/bash

The numbers represent the user & group ids (uid + gid)

The x indicates that the encrypted password is contained in the shadow file.

Linux shadow file

Example

```
joe:$6$6d/77tiY.l4V.jfC$BfDVhWndx8PIBe0lWoKSjGEUT/Gf6qlp6tgkV1hm6ulfg2l7K7TL  
xlNZ8c84nlwQhL6H8WPIKWwQ.AzdUbsa4/:17423:0:99999:7:::
```

```
tony:$6$AS36G$VPhHTImCN.S8LnP3z.ZWeegJUrfrcCzQ7IFbfDGmYgxe1/5JnjNN7KB  
WRAzS/9HYw/h697DC6XvdT14dctO1.:17423:0:99999:7:::
```

```
kate!:17423:0:99999:7:::
```

The string (after the user name) is the encryption of the user's password.

The user **kate** has yet to be given a password

Linux group file

Example

```
groupZ:x:1003:tony,joe
```

This group has a user id 1003.

The group has two members.


Interfacing with Linux

Once logged on, user can interface with Linux using the GUI (graphical user interface) or entering commands directly.

One enters commands directly using the command line interface i.e. the terminal program.

Users need to know the correct syntax for each command.

Linux Terminal Program



The screenshot shows a Linux desktop with a green background. On the left, there are icons for 'root's Home', 'wastebasket', and 'CNAME Terminal'. A terminal window titled 'Terminal' is open in the center. The terminal shows the following commands and output:

```
linux-lsw5:~ # cd /media
linux-lsw5:/media # ls -l
total 4
-rwxr-xr-x 1 root root 240 Dec  6 11:35 ex2
linux-lsw5:/media #
```

The terminal window has a menu bar with 'File', 'Edit', 'View', 'Search', 'Terminal', and 'Help'. The desktop taskbar at the bottom shows 'Computer' and 'Terminal' icons, along with system status icons and the date 'Tue 30 Jan, 11:33'.

Linux Terminal Program

When you launch the terminal you will probably see the bash shell (other shells are available but bash is often the default).

Running commands at the terminal uses fewer resources than a GUI and provides more functionality

It's possible to write your own short operating systems programs known as scripts.

Linux Commands

ls lists files in current directory

pwd current directory

cat view file contents

cal displays calendar

date displays date

su switches user

cp copy a file

mv move a file & change file name

Linux Documentation

There is a wealth of documentation available. Obviously one can search the Internet using any search engine.

Basic commands have built in documentation. Every command has a manual (man) page which should provide anything you might require about a specific command.

For example, if you require information about the ls command (lists files in current directory), enter the following:

```
man ls
```

Linux Documentation (cont.)

The `man` pages list all command options relating to the command e.g.

`ls` can be used with in a variety of options such as:

`ls -l` provides a long listing, giving more detailed information about each file.

`ls -i` lists files with their i-node numbers.

Linux Documentation (cont.)

man pages are not always user-friendly, but should (with some patience) provide what you need.

When using the command, use space to scroll through and q to quit.

An alternative to man is info e.g.

info ls

Linux Directories

Linux directories (or folders) are containers that may contain sub-directories and files.

A directory is a type of file and has the same attributes as an ordinary file such as ownership, permissions, etc.

As well as directories, Linux has other types of file such as symbolic links, device, etc.

Linux Directory Structure

Like windows, Linux has a hierarchical directory structure.

At the top of the tree is: /

There are a number of top-level directories under /. These may vary from distribution to distribution. However, you would expect to see the following:

Linux Directory Tree

/bin: contains binaries, the commands that you use to perform tasks such as ls and cat.

/etc: contains configuration files such as passwd, shadow and group.

/home: contains the home directory for each user.

/root: root's home directory

Linux Directory Tree (cont.)

`/dev`: device files – used to make devices (such as storage) available.

`/sbin`: system binaries – contains commands used by root, such as `groupadd` and `usermod`.

Others: `/lib`, `/usr`, `/tmp` and many others.

Linux Directory Tree (cont.)

To view top level directories:

Use the GUI to navigate to the top of the tree.

Using the terminal program:

```
cd /  
ls
```

Alternatively

```
tree -L 1
```

Linux Directory Tree (cont.)

To view all directories in a tree formation:

```
tree -d | less
```

(space to scroll, q to exit)

Install tree

```
apt-get install tree (if necessary)
```

```
man tree (for documentation)
```

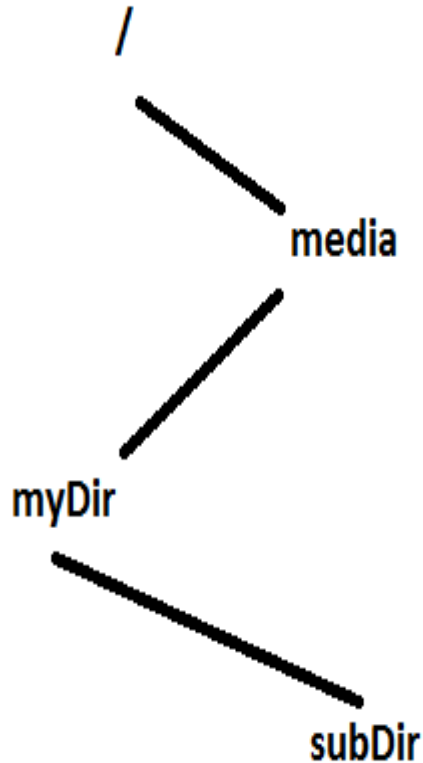
Absolute PATH

The absolute (or full) path is given using the / symbol (which donates the top of the tree).

A directory might have a path such as /media/myDir/subDir

This means that subDir is a subdirectory of myDir which in turn is a subdirectory of media which is a subdirectory of /.

Absolute PATH (cont.)



Every file and directory has an absolute path.

Home Directory

When a user logs on, he / she generally has a home directory.

Here the user can store his / her own files including sub-directories and their contents.

It is usually the default login directory i.e. where a user goes after logging in.

It is possible for a user to have no home directory e.g. a guest user in an Internet café or a public library.

Home Directory (cont.)

Users 'own' their home directories and files they create here.

Users can add, remove and edit home directory contents.

Users can allow (or prevent) others permission to their home directory and home directory contents.

Home Directory (cont.)

Usually, home directories are subdirectories of /home.

A user joe's home directory would be /home/joe.

It is possible that home directories could be located elsewhere on the system.

Files

A file is a collection of data items stored on disk. What ever you store in a computer - it must be a file. Files are associated with devices like hard disk, etc. Every file has a location i.e. an absolute path.

The command **ls -l** (long listing) displays the different properties of files in the current directory. There are different types of files such as:

Regular (ordinary): this is the most common type of file. It's identified by the **-** at the beginning of the permission string.

Example:

```
-rw-r--r-- 1 joe joe 16 Jan 31 13:56 myFile
```

Types of File

Directory: in a typical system about 10% of files are directories. It's identified by the **d** at the beginning of the permission string.

Example:

```
drwxr-xr-x 2 joe groupX 4096 Jan 31 13:57 newDir
```

Symbolic (soft) link: this is a short cut to an other file. It's identified by the **l** at the beginning of the permission string.

```
lrwxrwxrwx 1 joe joe 11 Jan 31 13:57 passLink -> /etc/passwd
```

Other types also exist such as device files.

File Attributes

Every file has a number of properties (attributes). Some of these can be seen when entering `ls -l`

```
-rw-r--r-- 1 joe groupX 16 Jan 31 13:56 myFile
```

The first item is the permission string (`-rw-r--r--`). This is followed by the number of hard links (1), owner (joe), the group (groupX), the size in bytes (16), date (Jan 31), time (13:56) and file name (myFile).

More information can be displayed e.g. `ls -li` displays the information node (i-node) number; `ls -ls` displays the number of blocks required to store the file. Try **man ls**.

There are many commands used with files such as **cat**, **rm**, **touch**, **cp**, etc.

File Permissions

Consider the following.

```
-rw-r--r-- 1 bill staff    28 Oct 2 11:43 temp
```

- The **-rw-r--r--** is known as the permission string. The first character indicates the type of file. The other nine characters represent permissions.
- The permissions are read **r**, write **w** and execute **x**. The first three are **user** permissions; the next three are for the **group** the final three are for **others**.
- The above file **temp** can be read and written to by the owner, read by members of the group **staff** and read by all users. The above permissions are the usual default permissions when a file is created. Permissions can be changed by the owner or by the root user.

Ordinary File Permissions

Example

```
-rwxr-xr-- 1 joe groupX 34 Feb 1 12:50 myFile
```

This is an ordinary file

The owner of the file is joe

joe can read (view file contents), write (make & save changes) and myFile (if it's a program or a script) the file myFile

Members of groupX can read and execute myFile

Other system users can read myFile

Directory Permissions

Directories also have permissions but have slightly different meanings.

Example

```
drwxr-x-- joe groupY 1024 2 Feb 12:50 myDir
```

joe owns this directory and can change any of its nine permissions

joe can see directory contents and add or remove files to / from it.

Members of groupY can see directory contents (& copy them) but can not add or remove contents.

Other users have no permission.

Editors

Although strictly not part of the operating system, all Linux systems come with at least one editor.

There are many types of editors available both GUI (`gedit`) or command line (`nano`)

Editors are used to create, edit and save files.

Usually, users only have permission to save files in their home directories.

The `vi` editor used to be the standard UNIX editor and is still included in all distributions.

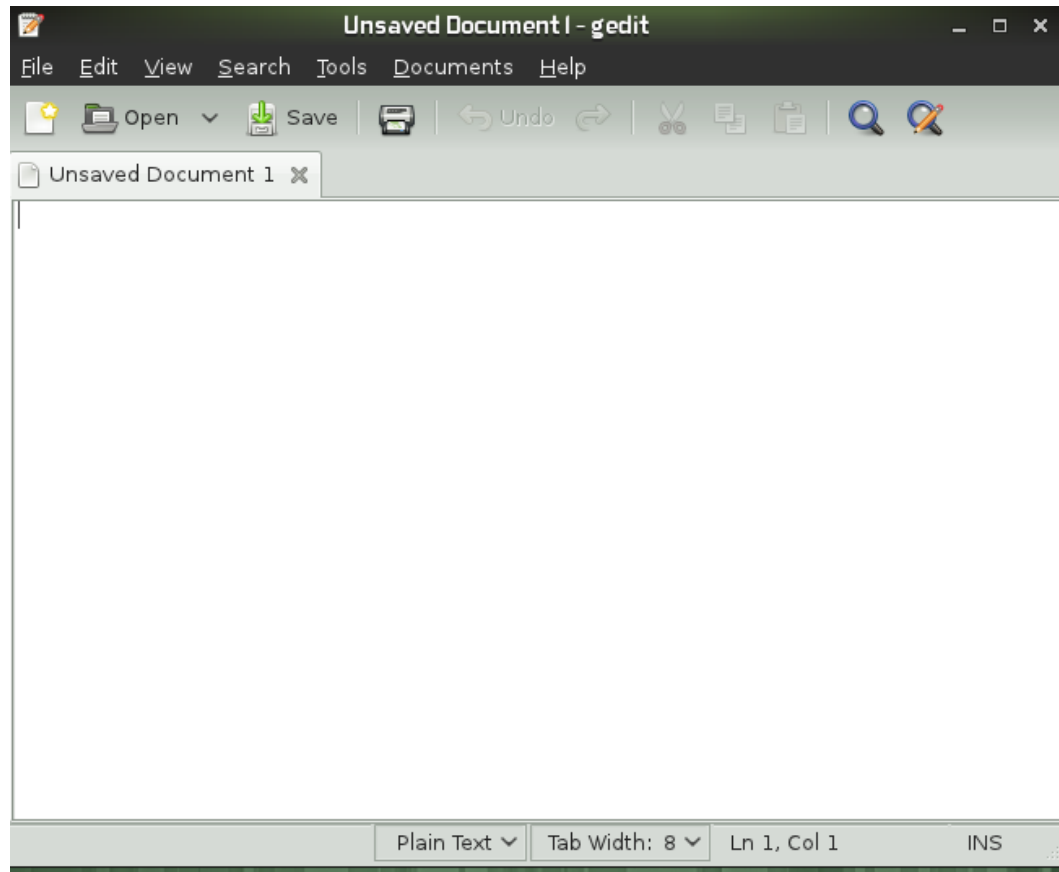
Editors (cont.)

gedit runs in a GUI environment.

gedit is similar to Windows notepad

Uses drop down menus for different functions.

gedit Editor



Editors (cont.)

Nano has a command line interface.

No GUI (e.g. no use of mouse) available.

Text can be entered and commands require use of the **CTRL** key e.g. to exit **CTRL**+ x

nano Editor

```
GNU nano 2.2.6      New Buffer

^G Get Help ^O WriteOut ^R Read Fil ^Y Prev Pag ^K Cut Text ^C Cur Pos
^X Exit     ^J Justify  ^W Where Is ^V Next Pag ^U UnCut Te ^T To Spell
```