Dear Students,

Welcome to the Linux Utilities Tutorial with Sample examples. Here we cover only few most used Linux commands, rest all left to explore.

Regards,

Savitri S,

Asst Professor, Dept of CSE,

PES University

Mail-id: savitris@pes.edu

Linux Utilities:

man command:

man - an interface to the on-line reference manuals

man is the system's manual pager. Each page argument given to man is normally the name of a program, utility or function. And command to check information about manual page is man man, then q to quit the manual page.

clear command:

clear - clear the terminal screen

cat command:

man cat or info cat - explore the command

Syntax:

cat [OPTION]... [FILE]...

cat command is to concatenate the files and print on the standard output. It accepts any number of files as its arguments and outputs the contents of each file, completely, one after another, on the screen (the standard output device).

Assume that we have three files, file1, file2 and file3 with the following contents of these files, respectively as:

file1: These are the contents of file1

file2: These are the contents of file2

file3: These are the contents of file3

The output of the command:

cat file1 file2 file3

These are the contents of file1

These are the contents of file2

These are the contents of file3

Assume now we want to cat two files were one of the file is not existing, as shown below:

cat file1 file 4;

The output of the command:

These are the contents of file1

cat: file4: No such file or directory

Suppose the file whose is present in some other directory then command to cat is shown below, assume file5 is present in the directory called Samplefiles:

cat Samplefiles/file5

The output of the command:

These are the contents of file5

By default, Linux outputs the contents of files, mentioned as arguments to the command cat, to the standard output (which is screen, by default). But it also provides another facility using which we can "redirect" the output of a command to some other file instead of standard output.This is called "output redirection" and to do this we can do as follows:

cat file1 file2 > file6

The result of above command would be the creation of a new file, file5, with the following contents:

file6: These are the contents of file1

These are the contents of file2

The contents of file6 are the concatenation of the contents of file1 and file2. In Linux, standard output is just another file. Hence, by using the facility of output redirection ">" we could replace the file "standard output" with our own file, file5. Similarly standard input (which is keyboard, by default) is also another file.

we are redirecting the text on standard input (keyboard) to file7 as shown below:

cat > file7 ; We can enter any text. So enter some text and move to a newline.To stop press [Ctrl+d] on the keyboard.This is one of the ways to create a file in linux, if file7 is not existing, if already existed before we execute this command then the contents of file7 would be overwritten.Here, because no input file is mentioned as its argument cat assumes that the file is standard input.

touch command:

man touch or info touch

syntax:

touch [OPTION]... FILE...

touch command creates a blank file without any text in it.This would be the case if the file does not exist already.

touch file1;

If the file1 already exists then update the access and modification times of each file to the current time, but contents of the file will not change.By making use of different options we can change only the access time as, -a : change only the access time along with touch command.

We can pass arguments more than one file in a single command:

touch file2 file3

Suppose the file4 need to be created in some other directory then command to create is shown below, file4 to be created in the directory called Samplefiles, without moving to that directory:

touch Samplefiles/file4

ln command:

man ln or info ln
The general syntax for the ln command is:

ln [-s] target [name]

The target parameter indicates the location or file to which the link should point. The optional name parameter specifies the name that link should be given. If no name is specified, the basename of the target will be used. The ln utility creates a hard link by default; the -s option indicates that a symbolic link should be created instead.

ln command is used to create links between files. Links allow more than one location to refer to the same information.There are two types of links, both of which are created by ln.
By default, it makes hard links; with the -s option, it makes symbolic (or soft) links

Hard links refer to the specific location of physical data. a hard link is a reference, or pointer, to physical data on a storage volume. On most file systems, all named files are hard links. The name associated with the file is simply a label that refers the operating system to the actual data. As such, more than one name can be associated with the same data. Though called by different names, any changes made will affect the actual data, regardless of how the file is called at a later time. Hard links can only refer to data that exists on the same file system.

Symbolic links, which refer to a symbolic path indicating the abstract location of another file, a symbolic link (often shortened to symlink) is a file that serves as a reference to another file.

Unlike a hard link, a symbolic link does not point directly to data, but contains a symbolic path which is used to identify a hard link (or another symbolic link). Thus, when a symbolic link is removed, the file to which it pointed is not affected. In contrast, the removal of a hard link will result in the removal of the file, if it is the last hard link to that file. As a result, symbolic links can be used to refer to files on other mounted file systems. A symbolic link whose target does not exist is known as an orphan.

Unlike hard links, if the target of a symbolic link is removed, the data is lost and all links to it become orphans. Conversely, removing a symbolic link has no effect on its target.

Let us look into example to understand the hard link and soft link:
Consider the file a.txt, the link count is 1 as shown below:

ls -l a.txt

-rw-rw-r-- 1 ubuntu ubuntu 53 May 22 10:17 a.txt

ln a.txt b.txt ; Here b.txt is not an existing file, so it creates a new file and copy content of a.txt to b.txt. And the count of the link is increased, it means there are two lines to the file with two different names.

ls -l a.txt b.txt ;

-rw-rw-r-- 2 ubuntu ubuntu 53 May 22 10:17 a.txt
-rw-rw-r-- 2 ubuntu ubuntu 53 May 22 10:17 b.txt

Now after hardlink, count is increased and now the

ln b.txt c.txt ; b.txt file was created earlier through hard link is now used to create hard link for another file c.txt, as mentioned below now the reference count is increased and also here we can notice that the count for all the files which hardlinked has same reference count.

ls -l a.txt b.txt c.txt

-rw-rw-r-- 3 ubuntu ubuntu 53 May 22 10:17 a.txt
-rw-rw-r-- 3 ubuntu ubuntu 53 May 22 10:17 b.txt
-rw-rw-r-- 3 ubuntu ubuntu 53 May 22 10:17 c.txt

rm c.txt ; Suppose one of the files is removed, that c.txt is removed , now the reference count is reduced and the count of the other two files is the same.

ls -l a.txt b.txt c.txt

ls: cannot access 'c.txt': No such file or directory
-rw-rw-r-- 2 ubuntu ubuntu 53 May 22 10:17 a.txt
-rw-rw-r-- 2 ubuntu ubuntu 53 May 22 10:17 b.txt

ln a.txt d.txt; Here d.txt is not an existing file and if the file is not existing then a hard link cannot be created.

ln: failed to create hard link 'd.txt': File exists

man size:

list section sizes and total size.

Syntax:

size [-A|-B|--format=compatibility]
[--help]
[-d|-o|-x|--radix=number]
[--common]
[-t|--totals]
[--target=bfdname] [-V|--version]
[objfile...]

Consider an executable file named print_op and ass the object/executable file name as input to the tool as shown below:

size print_op

text	data	bss	dec	hex	filename
1182	552	8	1742	6ce	print_op

The first three entries are for text, data, and bss sections, with their corresponding sizes. Then the total in decimal and hexadecimal formats. The last entry is for the filename.

By default, the size of sections is displayed in decimal. However, if you want, you can have this information on octal as well as hexadecimal. For this, use the -o and -x command line options. And Input the size file is object file as shown below:

size Sample_print.o -o

Output:

text	data	bss	oct	hex	filename
0131	00	00	131	59	Sample_print.o

size Sample_print.o -x

Output :

text	data	bss	dec	hex	filename
0x59	0x0	0x0	89	59	Sample_print.o

If we are using size to find out section sizes for multiple files in one go, then if we want, we can also have the tool provide totals of all column values. We can enable this feature using the **-t** command line option.

size print_op -t

text	data	bss	dec	hex	filename
89	0	0	89	59	print_op
89	0	0	89	59	(TOTALS)

The last row in the output has been added by the **-t** command line option.

nm command:

man nm

nm - list symbols from object files.

The **nm** command displays information about symbols in the specified *File*, which can be an object file, an executable file, or an object-file library. If the file contains no symbol information, the **nm** command reports the fact, but does not interpret it as an error condition. The **nm** command reports numerical values in decimal notation by default.

To list the static and external symbols of the object file i.e, a.out as:

nm –e a.out

To display symbol sizes and values as hexadecimal and sort the symbols by value, as:

nm –xv a.out

man su:

su - change user ID or become superuser

Syntax:

su [options] [username]

The su command is used to become another user during a login session. Invoked without a username, su defaults to becoming the superuser.

sudo command:

man sudo -explore the command

sudo, sudoedit — execute a command as another user.

sudo allows a permitted user to execute a command as the superuser or another user as specified by the security policy. The security policy determines what privileges, if any, a user has to run sudo. The policy may require that users authenticate themselves with a password or another authentication mechanism. If authentication is required, sudo will exit if the user's password is not entered within a configurable time

limit. This limit is policy-specific; the default password prompt timeout for the sudoers security policy is unlimited.

apt-get command:

man apt-get

apt-get - APT package handling utility -- command-line interface

apt-get is a command from the APT suite of tools that is used to install, upgrade or remove software packages.

The *apt-get* command obtains information about available packages from the sources listed in **/etc/apt/sources.list** and then uses that information to upgrade or install packages.

To obtain updated information about packages available from the installation sources listed in **/etc/apt/sources.list**, use the *apt-get update* command.

To upgrade all installed packages, use the *apt-get upgrade* command. We can use the *-u* option to display the complete list of packages which will be upgraded.

To install a package by its name, use the *apt-get install PACKAGE_NAME* command.

To remove a package by its name, use the *apt-get remove PACKAGE_NAME* command.

To clear out information about retrieved files, use as sudo apt-get clean.

Few References Used along with man page for each of the commands:

https://en.wikipedia.org/

https://geek-university.com/