ls – list contents of directory

ls command is one of the most commonly used tools in Unix. You simply cannot underestimate the importance of being able to confirm exactly what files and directories are available to you, and **ls** does its job perfectly. Most of recent Linux and Unix distros support ls colorized output which is shown above.

ls – basic syntax

Like many commands in Unix, **ls** expects you to specify a file or directory name for it to inspect:

```
$ ls file1 dir2
```

This command above will list (check if it exists) file called file1, and will also show contents of the dir2 directory (if dir2 exists).

The most basic way to use **ls command** is to simply make it list files and directories in your current directory. You don't need to specify any parameters for it, just type the command itself.

In this example below, I'm in **~/proj** directory (~ sign simply means my home directory, so with homedir **/home/greys** the **~/proj** means **/home/greys/proj**) where some of my Linux-based projects are:

```
greys@xps:~/proj $ ls
ansible bash glebreys.com gleb.reys.net python unixtutorial
unlocker writing
```

All of these names – ansible, bash, glebreys.com, etc are the directories in that **/home/greys/proj** directory.

ls – most common usage

If you're interested in a particular file or directory, you should specify the filename as a parameter to ls.

This command will simply output the filename of the file you

have specified:

```
$ ls /etc/passwd
/etc/passwd
```

Similarly, it’s possible to confirm that a certain file is not found:

```
$ ls /etc/passwd5
ls: /etc/passwd5: No such file or directory
```

You may also specify a directory (the output in this example is abridged) to confirm the contents of it:

```
$ ls /etc
Muttrc
Net
X11
adduser.conf
adjtime
aliases
aliases.db
alternatives
apache2
apm
apt
at.deny
...
```

And finally, the most common way **ls** is used: the long version of the output, which is invoked using the -l parameter:

```
$ ls -l /etc
total 1064
-rw-r--r-- 1 root root   8063 Mar  8  2007 Muttrc
-rw-r--r-- 1 root root    611 Mar  5  2007 Net
drwxr-xr-x 5 root root   4096 Sep  7 04:44 X11
-rw-r--r-- 1 root root   2077 Aug  3  2006 adduser.conf
-rw-r--r-- 1 root root     44 Aug  3  2006 adjtime
-rw-r--r-- 1 root root     51 Mar 25  2007 aliases
-rw-r--r-- 1 root root  12288 Sep  7 05:01 aliases.db
drwxr-xr-x 2 root root   4096 Sep  7 05:03 alternatives
drwxr-xr-x 8 root root   4096 Sep 26 03:02 apache2
drwxr-xr-x 6 root root   4096 Aug  3  2006 apm
drwxr-xr-x 4 root root   4096 Sep  7 04:34 apt
-rw-r----- 1 root root    144 Aug  3  2006 at.deny
```

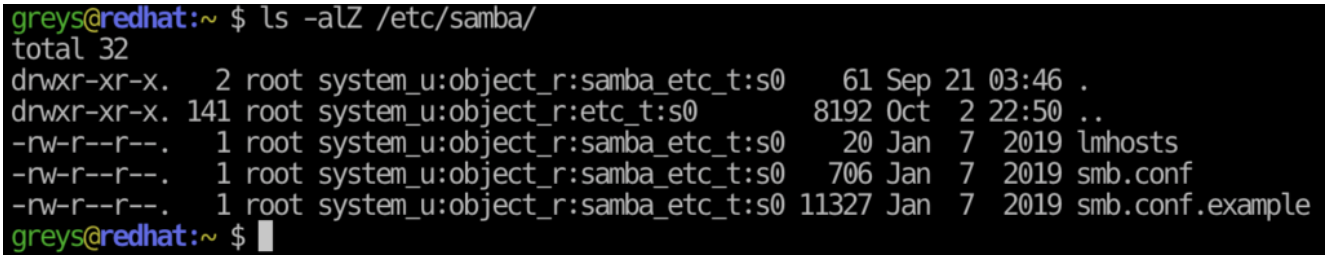
Show Symlinks with ls

Symbolic links can be inspected to show destination files using pretty standard **ls -l** combination:

Showing symbolic links with ls command

List SELinux Contexts with ls

ls command can be used with the **-Z** parameter to show file contexts in SELinux enabled systems:



List SELinux contexts with ls

Such output allows you to confirm the type of each directory entry, the access permissions, the number of links to this file, the ownership (user and unix group which own it), size in bytes, date of the last modification, and, finally, the name of the directory entry.

See also

- basic Unix commands
- lrwxrwxrwx
- Unix Commands
- chmod and chown
- SELinux