NCP2209 – Operating System

LINUX - LISTING FILES AND DIRECTORIES

Listing files and directories

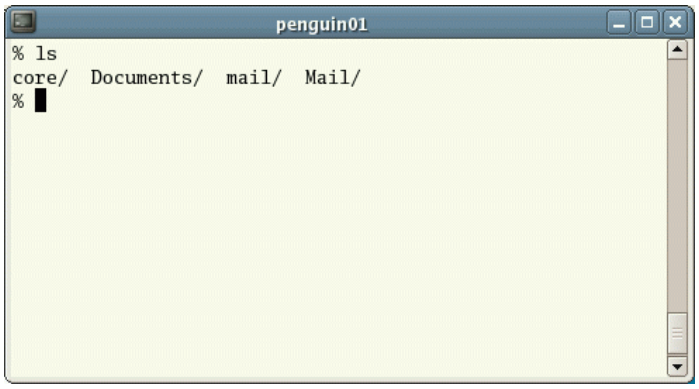
ls (list)

When you first login, your current working directory is your home directory.

To find out what is in your home directory, type

```
% ls
```

The **ls** command (lowercase L and lowercase S) lists the contents of your current working directory.

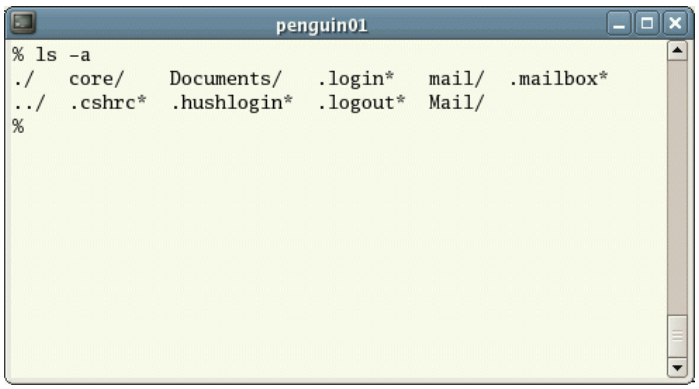


ls does not, in fact, cause all the files in your home directory to be listed, but only those ones whose name does not begin with a dot (.) Files beginning with a dot (.) are known as **hidden files** and usually contain important program configuration information.

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.



ERROL JOHN ANTONIO
erroljohn.antonio@ue.edu.ph

ls is an example of a command which can take options: **-a** is an example of an option. The options change the behavior of the command.

Now try to execute the command **ls -l.** l stands for long listing. Use the following ls command to view detailed information in the given directory contents.

What have you observed?

Below are the other **ls command options**:

ls command main options:

option	description
ls -a	list all files including hidden file starting with '.'
ls --color	colored list [=always/never/auto]
ls -d	list directories - with ' */'
ls -F	add one char of */=>@| to enteries
ls -i	list file's inode index number
ls -l	list with long format - show permissions
ls -la	list long format including hidden files
ls -lh	list long format with readable file size
ls -ls	list with long format with file size
ls -r	list in reverse order
ls -R	list recursively directory tree
ls -s	list file size
ls -S	sort by file size
ls -t	sort by time & date
ls -X	sort by extension name

Examples:

List directory *Documents/Books* with ***relative*** path:

```
$ ls Documents/Books
```

List directory */home/user/Documents/Books* with ***absolute*** path.

```
$ ls /home/user/Documents/Books
```

List **root** directory:

```
$ ls /
```

List **parent** directory:

```
$ ls ..
```

List **user's home directory** (e.g: /home/user):

```
$ ls ~
```

List with **long format**:

```
$ ls -l
```

Show **hidden files**:

```
$ ls -a
```

List with **long format** and **show hidden files**:

```
$ ls -la
```

Sort by **date/time**:

```
$ ls -t
```

Sort by **file size**:

```
$ ls -S
```

List all **subdirectories**:

```
$ ls *
```

Recursive directory tree list:

```
$ ls -R
```

List only text files with **wildcard**:

```
$ ls *.txt
```

ls redirection to output file:

```
$ ls > out.txt
```

List directories only:

```
$ ls -d */
```

List files and directories with **full path**:

```
$ ls -d $PWD/*
```

ERROL JOHN ANTONIO
erroljohn.antonio@ue.edu.ph

Making Directories

mkdir (make directory)

We will now make a subdirectory in your home directory to hold the files you will be creating and using in the course of this tutorial. To make a subdirectory called myFolder in your current working directory type

```
mkdir myFolder
```

To see the directory you have just created, type

```
ls
```

Try!

Create a directory using the command:

- **mkdir Academics**
- **mkdir Academics -v**

What have you observed?

- **mkdir Academics/Machine Problems**
- **mkdir -p Academics/Machine Problems**
- **mkdir -p Academics/'Machine Problems'**

What have you observed?

Changing to a Different Directory

pwd (parent working directory)

The command **pwd *directory*** displays the current directory.

cd (change directory)

The command **cd *directory*** means change the current working directory to '*directory*'. The current working directory may be thought of as the directory you are in, i.e. your current position in the file-system tree.

Changes into the given directory, or into the home directory when no parameter is provided.

To change to the directory you have just made, type

```
cd myFolder
```

Type **ls** to see the contents (which should be empty)

The directories . and ..

Still in the **myFolder** directory, type

```
ls -a
```

As you can see, in the **myFolder** directory (and in all other directories), there are two special directories called (**.**) and (**..**)

The current directory (.)

(**.**) means the current directory, so typing

```
cd .
```

NOTE: there is a space between cd and the dot

means stay where you are (the **myFolder** directory).

This may not seem very useful at first, but using (**.**) as the name of the current directory will save a lot of typing, as we shall see later in the tutorial.

The parent directory (..)

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

```
cd ..
```

will take you one directory up the hierarchy (back to your home directory). Try it now.

Note: typing **cd** with no argument always returns you to your home directory. This is very useful if you are lost in the file system.

Examples:

Change to **home directory** (determined by $HOME environment variable):

```
$ cd
```

Also change to **home directory**:

```
$ cd ~
```

Change to **root directory**:

```
$ cd /
```

Change to **parent directory**:

```
$ cd ..
```

Change to **subdirectory** *Documents*:

```
$ cd Documents
```

Change to **subdirectory** *Documents/Books*:

```
$ cd Documents/Books
```

Change to **directory with absolute path** */home/user/Desktop*:

```
$ cd /home/user/Desktop
```

Change to directory name with white space - *My Images*:

```
$ cd My\ Images
```

or

```
$ cd "My Images"
```

or

```
$ cd 'My Images'
```

Pathnames

pwd (print working directory)

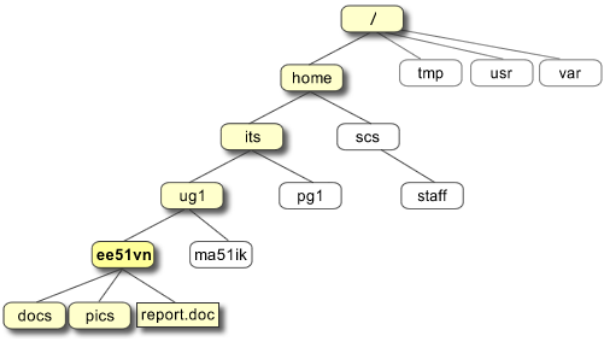
Pathnames enable you to work out where you are in relation to the whole file-system. For example, to find out the absolute pathname of your home-directory, type **cd** to get back to your home-directory and then type

```
pwd
```

The full pathname will look something like this -

```
/home/its/ug1/ee51vn
```

which means that **ee51vn** (your home directory) is in the sub-directory **ug1** (the group directory),which in turn is located in the **its** sub-directory, which is in the **home** sub-directory, which is in the top-level root directory called " / " .



Examples:

Change directory to /usr/src directory and print working directory:

```
$ cd /usr/src
$ pwd
/user/src
```

Change directory to home directory and print working directory:

```
$ cd ~
$ pwd
/home/user
```

Change directory to parent directory of the home directory and print working directory:

```
$ cd ~/..
$ pwd
/home
```

Change directory to root directory and print working directory:

```
$ cd /
$ pwd
/
```

Home Directories and Pathnames

Understanding pathnames

First type cd to get back to your home-directory, then type

```
ls myFolder
```

to list the contents of your myFolder directory.

Now type

```
ls backups
```

You will get a message like this -

```
backups: No such file or directory
```

The reason is, **backups** is not in your current working directory. To use a command on a file (or directory) not in the current working directory (the directory you are currently in), you must either **cd** to the correct directory, or specify its full pathname. To list the contents of your backups directory, you must type

```
% ls myFolder/backups
```

~ (your home directory)

Home directories can also be referred to by the tilde **~** character. It can be used to specify paths starting at your home directory. So typing

```
ls ~/myFolder
```

will list the contents of your myFolder directory, no matter where you currently are in the file system.

What do you think

```
ls ~
```

would list?

What do you think

```
% ls ~/..
```

would list?

References:

RapidTables. (n.d.). Linux. https://www.2daygeek.com/linux-unix-ls-command-display-directory-contents/

Post Lab. (n.d.). Linux Tutorial for Beginners. https://www.purdue.edu/postlab/resources/linux-tutorial/

ERROL JOHN ANTONIO
erroljohn.antonio@ue.edu.ph