

Linux Administration

Files

Xavier Belanger

**This work is licensed under
a Creative Commons Attribution-ShareAlike 4.0 International License.**

<http://creativecommons.org/licenses/by-sa/4.0/>

You are free to:

- **Share** — copy and redistribute the material in any medium or format
- **Adapt** — remix, transform, and build upon the material for any purpose, even commercially.

The licensor cannot revoke these freedoms as long as you follow the license terms.

Under the following terms:

- **Attribution** — You must give appropriate credit, provide a link to the license, and indicate if changes were made. You may do so in any reasonable manner, but not in any way that suggests the licensor endorses you or your use.
- **ShareAlike** — If you remix, transform, or build upon the material, you must distribute your contributions under the same license as the original.
- **No additional restrictions** — You may not apply legal terms or technological measures that legally restrict others from doing anything the license permits.

Basic notions

- Files are organized in a very hierarchical way.
- The top the file system is / (root).
- Almost everything managed by the system is a file (processes, devices, network connections, ...).

Filesystem Hierarchy Standard (FHS)

/bin, /sbin	binaries
/boot	boot configuration
/dev	devices
/etc	configuration files
/home	end-users files
/lib	libraries
/proc	processes
/root	root's home directory
/tmp	system temporary directory
/usr	applications
/var	“variable” files

Storage status

- *mount* will display how many files systems (local and remote) are available.
- *df* (disk free) will give you what space is available for each file system.
- *du* (disk used) list the size of the files in the current directory and all sub-directories.
- You can use *du* to summarize the size of a specified directory: *du -s <directory>*
- For both *df* and *du*, use the *-h* and *--si* option to get a more human readable output

Navigating the file system

- The *cd* (change directory) command will let you move to another directory.
 - Go to the parent directory: `cd ..`
 - Go to a relative directory: `cd ../../tmp`
 - Go to a specific directory: `cd /usr/local/bin`
 - Go to your personal directory: `cd ~` or `cd`
 - Go to the previous working directory: `cd -`
- The *pwd* (print working directory) command displays where you currently are.

Listing files

- `ls` is the go-to command to list files, plenty of options are available, depending how much details you need (formatting, additional information, sorting).
- `tree` is a somewhat “graphical” tool that will display files with their directory structure.

Hidden files

- Files and directories with a name starting with a dot are not displayed by default.
- Dot files and directories are mostly used to store user configuration and preferences.
- To display dot files, use the `ls` command with the `-a` option.
- Except for their name, dot files are manipulated like regular files.

Creating directories and files

- Use the command `mkdir` to create a new directory:
mkdir <directory>
- You can create a directory and subdirectories all at once with the `-p` option:
mkdir -p <directory>/<sub-directory>
- You can create an empty file with the `touch` command:
touch <file>

Copying and moving files

- To copy a file, use the *cp* command:
- *cp* *<source file>* *<destination file>*
- You can copy a full directory with the *-a* argument:
cp -a <source directory> *<destination directory>*
- *mv* is the command to use to move file:
mv <source file> *<destination>*
- The *mv* command is also used to rename files.

Deleting files

- Deleting files is performed by using the `rm` command:
rm <file>
- Beware: by default there is no confirmation and files are deleted “forever” (no trash/recycle bin). Use the `-i` option to get a confirmation prompt.
- Deleting directories is done with the *rmdir* command (the directory must be emptied first).
- Deleting all files and directories recursively without confirmation can be done with *rm -Rf*.
You must have a very good reason to use this.

Filename elements

- When provided a full path filename the *dirname* and *basename* commands let you split the directory name from the file name (respectively).
 - *dirname* <full path name>
 - *basename* <full path name>