Everything Is a File

To Linux, a file is just a stream of bits and bytes. Linux doesn't care what those bits and bytes form; instead, the programs running on Linux care. To Linux, a text document and a network connection are both files; it's your text editor that knows how to work with the text document, and your Internet applications that recognize the network connection.

- Linux (and Unix) filenames can be up to 255 characters in length. Unlike Windows and Mac OS machines, Linux boxes are case-sensitive when it comes to filenames.Case-sensitivity also means that commands and filenames **must be entered exactly to match their real command names or filenames**.
- **`/` is never an option because that particular character is used to separate directories and files**. You could use a dash, forming books-to-buy.txt, but I find that underscores work nicely as word separators while remaining more unobtrusive than dashes. **If you do use a dash, though, do not place it at the beginning of a filename**. For `\ [] {} * ? ' ''` must be escaped by placing a `\` in front of the characters, which tells the shell that it should ignore their special usage and treat them as simple characters. A simpler method that's a bit less onerous is to surround the filename with `""`, which function similarly to the `\`

File System

1. `ls -R ~/iso` The -R option traverses the iso directory recursively, showing you the contents of the main iso directory and every subdirectory as well. Each folder is introduced with its path relative to the directory in which you started followed by a colon, and then the items in that folder are listed.

   `ls -1 (or ls --format=single-column)`/`ls -m or --format=commas`

   `ls -r` reverse the default output of the command and options you're inputting

   `ls -X` sort alphabetically by the file extension

   `ls -t (or ls --sort=time)` sort a directory's contents by date and time

   `ls -S (or ls --sort=size)` sort by size

   `ls -a` displays both hidden and unhidden items

   `ls -h` transform file size to kilobytes(K), megabytes(M) and gigabytes(G), a rounding is taken place

   `ls -F`: `*` Executable `/`Directory `@`Symbolic link `|`FIFO `=`Socket

   `ls -l`: `-` Regular file `-` Executable `d` Directory `l` Symbolic link `s` Socket `b` Block device `c` Character device `p` Named pipe

In each case, `r` means "yes, read is allowed"; `w` means "yes, write is allowed" (with "write" meaning both changing and deleting); and `x` means "yes, execute is allowed." A `-` means "no, do not allow this action." If that `-` is located where an `r` would otherwise show itself, that means "no, read is not allowed." The same holds true for both `w` and `x`. (Normally `rwx` means the file's owner, the file's group, and all the other users on the system can read, write or execute this file)

In the case of a directory, `r` means that the user can list the contents of the directory with the `ls` command. A `w` indicates that users can add more files into the directory, rename files that are already there, or delete files that are no longer needed. That brings us to `x`, which corresponds to the capability to access a directory in order to run commands that access and use files in that directory, or to access subdirectories inside that directory.

`ls -1 | grep m`/

2. `cd ~` `~` is like an alias meaning your home directory

   `cd /`回到根目录

   `cd ..`回到上一级目录

   `cd -` takes you back to your previous directory

3. `echo 'Hello World'`

   `echo $USER` `echo $HOME` `echo ~` `echo 'Hello World' > foo.txt` `echo 'Hey How' >> foo.txt`

4. `touch`创建空文件

   `touch` can simultaneously update both the access and modification times for a file (or folder). You can be more specific, if you'd like. If you want to change just the access time, use the `-a` option (or `--time=access`); to alter the modification time only, use `-m` (or `--time=modify`).

   You can only use the `touch` command on a file and change the times if you have write permission for that file. Otherwise, `touch` fails.

   `touch -t` Instead, you can pick whatever date and time you'd like, as long as you use this option and pattern: `-t [[CC]YY]MMDDhhmm[.ss]`. It's very important that you include the zeroes if the number you want to use isn't normally two digits or your pattern won't work. (check the book for more details)

5. `cp file1 file2` copy `file1` named `file2` to the same directory

   `cp dir1/file1 dir2/file2` copy `file1` in `dir1` named `file2` to `dir2`

   `cp dir1/file1 .` copy `file1` in `dir1` to current directory

   `cp -v ~/pix/on_floor_0[1-3].jpg .` copy `on_floor_01`, `copy on_floor_02`, `copy on_floor_03` to the current directory and show the steps

   `cp -i ~/pix/on_floor_0[1-3].jpg .` If you tried once again to copy the same files but used the `-i` option this time, you'd get `cp: overwrite './on_floor_01.jpg'?` many times

   `cp -R dir1 dirl2` the directory, as well as its contents, are copied.

   `cp -a` copy files as perfect backups in another directory. `-a` ensures that `cp` doesn't follow symbolic links (which could grossly balloon your copy), preserves key file attributes such as owner and timestamp, and recursively follows subdirectories.

6. `mv file1 file3`将当前目录下文件file1更名为file3

   `mv file2 dir2`将文件file2移动到目录dir2下

7. `rm file3`删除文件file3 (rmdir 命令只能删除空的文件夹)

   `rm -r dir1`删除目录dir1(`-r`连同子目录删除。`-f`强制删除， 即忽略不存在的文件)

`mkdir -pv` tells you what `mkdir` is doing every step of the way, so you don't need to actually check to make sure `mkdir` has done its job.

`cat`显示文件内容 (`space`显示下一行`enter`显示下一页)

`cat file1 file2`依顺序显示file1,file2的内容

`cat file1 file2>file3`把file1,file2的内容结合起来，再“重定向（`>`）”到file3文件中， 如果file3是已经存在的文件，那么它本身的内容被覆盖，而变成file1+file2的内容

`cat file1>>file2`这将变成将file1的文件内容“附加”到file2的文件后面，而file2的内容依然存在，这种重定向符`>>`比`>`常用，可以多多利用

`more`分页显示文件内容

`head -20`查看文件的前几行

`tail -30`查看文件的后几行

`clear`这个命令是用来清除屏幕的，它不需要任何参数

`grep -n name *.[chS]`用于查找文件中符合字符串的哪行。

vim

`:wq`保存并退出`q!`不保存文件并退出vi

`vi -r filename`在上次正用vi编辑时发生系统崩溃，恢复filename

`)`光标移至句尾 `(`光标移至句首

`H`光标移至屏幕顶行 `M`光标移至屏幕中间行 `L`光标移至屏幕最后行

`0`（注意是数字零）光标移至当前行首 `$`光标移至当前行尾

`Ctrl+u`向文件首翻半屏`Ctrl+b`向文件首翻一屏

`Ctrl+d`向文件尾翻半屏 `Ctrl+f`向文件尾翻一屏

`yy`/ `p`

（复制当前行）拷贝一行到剪贴板或取出剪贴板中内容的命令（`:n`切换文件）

`/string`/`?string`

从光标所在处向后或向前查找相应的字符串的命令。`N`上一个，`n`下一个

`x`删除字符 `dd`删除整行到行首 `d$`删除整行到行尾

attention

1. unless you specify `-i` (negate case lock), all files, folders and directories named with an upper case will not be shown
2. the `rm -rf/` command means remove (`rm`) - recursive (`r`) - force (`f`) home (`/`) and it will delete every folder, file and directory within your Linux OS. It is equivalent of wiping your entire hard drive clean.
3. `/` means root.