List of Stuff

GPS

/Where am I?/ - *pwd* /Path to local directory./ - *ls* - *tab* /*Note here* This command completes the current command. It is an autocompletion function. **Tab the world.**/ - *cat* /Displays a file as a stream of text./ - *man* /Invokes the manual for a command; ex.: **man cp** tells us all flags and adverbs for copy.** - /-h - */-help/* - history* /Displays the history of what that has been done during this session. (Better, it remembers a determined amount of lines.)/ - *cd* /Change Directory./ - *echo* /Reproduces the text on screen; OR, if followed by the operator >, it reproduces it within a file (rewriting the file); OR, if followed by the operator >>, it appends it within a new file (adding the contents)./ - *clear* /Clears the current screen./

Files

- *mkdir* /Creates a directory./
- *touch* /Creates a new file./
- cp /Deadly face/ /It copies the contents of a file inside another file. And it *replaces* them!/i
- mw /Deadly face/ /It moves.
- rm /Deadly face/ /It removes.

Operators

- > /Replaces the contents of a file./ /Deadly face./
- >> /Appends to the contents of a file./
- / /This breaks down a long screen output into smaller portions visible one at a time./
    - *less* is a paginator for this specific function; it lets you scroll along a text.
- *!* /We are not doing this now./

Search

- *find*
- *grep* /Used after the pipe *I*, it returns the result of a search./
- rwk *?*

Paths/Shortcuts

- / root
- ~ home
- - previous

Flags

/Flags are truly universal/ - *-i* /This asks for confirmation before executing the actual command./ - *-r* /Super dangerous. It means recursively; when applied to **remove**, for instance, it deletes each file within a directory, then the directory itself./

> Dennis' Pep Talk Linux philosophy: 1) use simple commands; 2) produce plaintext. Combine 1) and 2) and you will get a very powerful set of tools. Think of /Streaming data/ and /projecting data/, therefore controlling the cector that is communication - and text. Linux is a *textual system*.

> Syntax All that I have on my command list are **Verbs**. To this **Verbs** I can append **Adverbs** and finally *Objects**. Let us now imagine that we are the** Parser**. The** Parser **considers** SPACE **and** SYNTAX**. /Important note from DT: always use files that have** No spaces in it**. And try** not to use capitals**.

> Good policy Add a backslash when you are talking about a directory: *trash/*.

> An essential secret. We are just using shortcuts here, especially when moving files within a same directory or from a directory to the upper or lower. It should always be remembered that in truth the command is referring to longer, elaborate paths that start from the root folder every time. If we use current directory and upper directory, we are using **relative paths** (?); if we define the entirety of a directory position - the entire path - we are using its **absoultue path**.

> Food for thought. Start thinking about **textual streams**.

Week 6

D.T.: please ask questions.

D.T.: file are files: they do not depend from anything else; when, instead, I am installing some sort of elaborate software - Firefox, OpenOffice or so on - an elaborate group of files, interrelated and dependent from both themselves and other files in the hard drive, is simultaneously copied in different areas of the hard disk. Therefore, installing is an elaborate and flexible manner of copying.

D.T.: people who use Macs (posers) had to install additional software in order to attend today's class; programs usually rely on current versions of files installed by the operating system. How does Linux work? Linux has a vast repository of all the software that is available for any version of its operating system; the repository is also careful not to incur in any software conflict.

D.T.: programming languages similarly possess repositories. Think of Ruby and Python: they have repositories that work in the same way, libraries from which we can obtain software without bothering about conflicts, and dependencies.

Friends!

- **head** *It shows us the first few lines of the file.*
- **tail** *It shows us the final few lines of the file.*
- **cat**

Network

- **wget** *Used to download stuff from the interwebs.*

D.T.: let us think about files and extensions. We know what file names are, and we know what extensions are. For instance, when we have a **.pdf** file, the operating system and the user know that the file, regardless of its specific name, is only readable by Adobe. Somewhere in our operating system, the **.pdf** extension is associated with the specific software Acrobat Reader - so that, whenever we automatically open the file, Acrobat Reader is automatically used. If we change the extension, the operating system will not know what software to use when opening the file. *But* the file will anyway be readable - because the first few lines of the file contain the essential information that make it readable for its corresponding software. If we change the extension, the file is still readable. If we change the first few lines, regardless the extension, we **won't** be able to open it. > *So, what have we learnt?* The extension is the sort of **attunement** that my operating system might need in order to correctly open a file.

Search

- **locate** *Use it to find files.* **Locate** uses a database of files that is located somewhere in my hard disk.loc
- **find**
- **grep**
- **awk**

D.T.: someone who's able to use these four tools is insanely cool. > D.T.: patterns can become extremely complicated; but starting, for instance, with a star ** * **, can be helpful and easy. Try removing all files that end with** .pdf** - the command would be: `rm -ri *.pdf`.