**Lecture 1 (week 1.2)**

Thursday 8th 2016

Your computer account

You need a Physics & Astronomy department computer account if you don’t already have one. This will be taken care of during the first day of class.

Choose a user name related to your real name, e.g. as a student I would choose one of

**LudoVanWaerbeke**

**LudoWaerbeke**

**LVanWaerbeke**

You have to setup a password. It has to be at least 12 characters, including at least one capital letter, one digit and one symbol.

NEVER EVER press the computer power switch button, since computers remain always on.

You login/logout pretty much like on any other computer.

The computer screen locks if a session is inactive for a few minutes, you will be asked to enter your password again to get back in.

Your session

Once you are logged in, if you are used to the operating systems Windows or OS X, the environment will appear rather dry and unusual at first. This is because the PHYS210 computers are running a different operating system called linux. Linux is an open source version of UNIX. This is a command line based environment. Pretty much, anything you do with a click and drag under Windows or OS X, you do with a written command (text you type) under linux.

In fact, under any operating system (Windows, OS X, Linux) you can use both the click and drag or the command line concepts. It is just that command lines are less practical for Windows and OS X and click and drag is less practical for Linux. (note: OS X operating system is based on UNIX, as a result it is equally good at click and drag and command lines).

With practice you will learn the power of using command lines (in short it gives you a nearly limitless ways to communicate with the computer and makes it do anything you want).

A program is basically a block of commands the computer executes sequentially.

I will show you how you can use your Windows or OS X computer at home to work remotely.

The working space

The program (aka application) which allows to communicate with the computer using command line is called Terminal. You can find it under the “Applications” menu. A Terminal opens a dialog window with the computer. It is sometimes also called the console. The console (terminal) displays a prompt. On our system, the default prompt is

**[username@host ~]$**

**username** is yours and **host** is the computer name.

Folders are called directory. Computers have a folder structure, which you navigate through with “explorer” under Windows and “finder” under OS X. You will learn below how to navigate through a linux directory structure. The full specification of a directory structure in command-line linux is:

**/dir1/dir2/dir3/dir4/…/mydir**

where the various directories/subdirectories are separated by a slash /.

Basic commands

Open a Terminal and try these:

**pwd** command gives your location in the directory structure

**ls** command that lists the file names in working directory

**cd** command allows you to change directory

Commands help

Many commands can run with optional arguments. The list of command options can be seen when you type

**man command**

Your home directory is **/home2/username**

Try these in various directory (using **cd**):

**ls -l**

**ls -la**

**ls -ltr**

Other useful basic commands

**less filename** displays the content of a text file one screen at a time (**more** does something similar)

**head -n filename** displays the first n lines of a text file (default n=10)

**tail -n filename** displays the last n lines of a text file (default n=10)

**wc filename** displays the number of lines, words and character in **filename**

**mkdir dirName** creates the directory **dirName**

**rmdir dirName** deletes entire directory **dirName**. If **dirName** is not empty **rmdir** returns an error message

**rm filename1 filename2 …** deletes files **filename1 filename2 …**

**rm -r dirName** deletes entire directory **dirName** even if directory is not empty, even its sub-directories, if any.

**cp inName outName** creates a copy of file **inName** and name it **outName**

**cp -r inDirName outDirName** does the same as **cp** if **inName** is a directory

**mv inName outName** moves **inName** to **outName** (same as **cp -r** followed by the deletion of **inName**)

Wildcards

Most commands take a list of files or directories. To minimize writing, it is useful to use the concept of wildcards (using the asterisk character \*)

**rm \*** deletes all files and directories in the current directory

**rm -r \*** will delete everything in the directory where it is executed, including sub-directories and their content.

TAB-completion

Once you have typed enough characters of a file of directory name that there is only one possible match, hit the **Tab** key, and the prompt will type the rest of it.

If you haven’t typed enough character, or ther eis no match, you will get a “beep” instead.

You can continue the TAB completion down the directory/subdirectory chain.

Text file editing

**nano** is a basic editor which allows you to create and edit text files

Pattern search

It is sometimes useful to search for patterns in a big ascii file.

**grep pattern filename** returns the lines containing the pattern **pattern** in file **filename**.

Redirecting output

Command lines generate output information on the screen. You can redirect this information in a file **filename** by adding “**> filename**” at the end of the command line.

You can pass on the output of a command as input for another command in the same line. This is done by putting “ | “ between the two commands, this is called “pipe”.

For instance the command:

**ls -l dirname | less** will show the detailed file list in directory **dirname** one screen at a time with the **less** command.

Permissions

In Linux, each file has permissions attributes which defines who can do what with files: read, write and execute. They are identified with the letters “r”, “w”, “x”.

You are the owner of files that are created by you. Files created by other users are “owned” by them. Most system files is “owned” by root, the system manager.

By default, the owner can read or write (but not execute) a file, and no one else can do anything.

Only the owner can change the permissions of a file.

UNIX files have 3 separate permission modes, for the owner, for everyone else and for an intermediate size group. In Linux jargon, owner is called user or u, group is called group or g, and everyone else is called others or o.

[ludougrad@tau ~]$ ls -ltr

total 16

drwxr-xr-x. 2 ludougrad ludougrad 6 Aug 3 14:55 Desktop

drwxr-xr-x. 2 ludougrad ludougrad 6 Aug 3 14:55 Downloads

drwxr-xr-x. 2 ludougrad ludougrad 6 Aug 3 14:55 Templates

drwxr-xr-x. 2 ludougrad ludougrad 6 Aug 3 14:55 Public

drwxr-xr-x. 2 ludougrad ludougrad 6 Aug 3 14:55 Music

drwxr-xr-x. 2 ludougrad ludougrad 6 Aug 3 14:55 Pictures

drwxr-xr-x. 2 ludougrad ludougrad 6 Aug 3 14:55 Videos

drwxr-xr-x. 3 ludougrad ludougrad 17 Aug 10 14:33 Documents

-rw-rw-r--. 1 ludougrad ludougrad 1897 Aug 10 14:56 message

-rw-rw-r--. 1 ludougrad ludougrad 9781 Aug 17 11:34 lecture\_1.1\_myversion.docx

[ludougrad@tau ~]$

The first set of “rwx” is for the user permission mode, the second set of “rwx” is for the group the user belongs to and the third set of “rwx” if for others. A “-“ means that specific permission is not granted. When the first letter is a “d” this indicates a directory, not a file. When the first letter is a “-“ sign, this indicates a regular file.

Examples of permissions changes:

**chmod o+rwx** **filename** means others should have read, write, and execute/access permissions to file **filename**.

**chmod u-w** **filename** means remove user (your own) write-access to **filename**, so you can’t accidentally alter or delete it.