PATHOGEN MULTIOMICS AND BIOINFORMATICS

Lisbon 2023

COURSE DOCUMENTATION

July 3rd – 7th, 2023

U LISBOA UNIVERSIDADE DE LISBOA
FACULDADE DE FARMÁCIA Universidade de Lisboa
iMed. ULisboa Research Institute for Medicines

PATHOGEN MULTIOMICS AND BIOINFORMATICS 2023

:: Course Overview ::

Course Director:

- João Perdigão, PhD, Research Institute for Medicines (iMed.ULisboa), Faculty of Pharmacy of the University of Lisbon (PORTUGAL)

  E- mail: jperdigao@ff.ulisboa.pt

Teaching Staff:

- Taane Clark, PhD, London School of Hygiene and Tropical Medicine

  E-mail: Taane.Clark@lshtm.ac.uk
- Susana Campino, PhD, London School of Hygiene and Tropical Medicine

  E-mail: Susana.Campino@lshtm.ac.uk
- Jody Phelan. PhD, London School of Hygiene and Tropical Medicine

  E-mail: Jody.Phelan@lshtm.ac.uk
- Ifeaniy Ezeonwumelu, PhD, Gladstone Institute of Virology and Immunology, UC San Francisco

  E-mail: ifeanyi.ezeonwumelu@gladstone.ucsf.edu
- Matthew Higgins, MSc, London School of Hygiene and Tropical Medicine

  E-mail: Matthew.Higgins@lshtm.ac.uk
- Emilia Manko, MSc, London School of Hygiene and Tropical Medicine

  E-mail: Emilia.Manko@lshtm.ac.uk

Course Contents

The course is structured in seven distinct modules (Total Hours: 35) that covers basic Next Generation Sequencing analysis workflows:

Course Schedule

	3rd July	4th July	5th July	6th July	7th July
	Monday	Tuesday	Wednesday	Thursday	Friday
09:00	**Module 1** Mapping	**Module 3** Pathogen Toolbox	**Module 5** RNA-Seq and Transcriptomics	**Module 6** Metagenomics	
10:00					**Invited Guest Speaker Lectures**
11:00					
12:00					Lunch Break
13:00	Lunch Break	Lunch Break	Lunch Break	Lunch Break	
14:00	**Module 2** De novo Assembly	**Module4** Introduction to Phylogenetics (Part 1)	**Module 4** Introduction to Phylogenetics (Part 2)	**Module 7** GWAS	**Invited Guest Speaker Lectures**
15:00					
16:00					

In addition, the last day will cover a series of guest seminars.

Course documentation includes an introduction the topics covered in the course as well as exercises/tutorials. Moreover, further documentation concerning the installation of the computing system, overview and basic Linux commands is also provided.

:: Course Computing System Overview ::

Introduction

This course will make use of a Virtual Operating System based on CentOS 7. This is a free, enterprise-class, community-supported Linux distribution that is sourced from Red Hat Enterprise Linux (RHEL).

For this course we will use a Virtual CentOS image that will run on Oracle VirtualBox, a free virtualization platform that runs across multiple systems. Every computer made available for this course is already pre-installed with Virtual Box already configured with CentOS 7. You just need to find the shortcut on your desktop, double-click it, then select CentOS Virtual Machine (VM) from the VM list on the left and click Start. This will open a new window and CentOS with all course files and bioinformatic software necessary for the course will soon start.

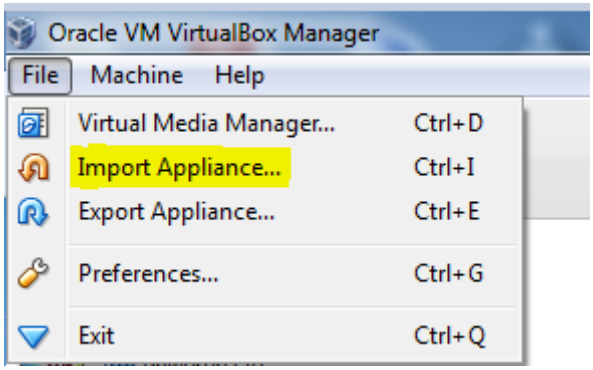
The image of this VM is also available (approx. size 40Gb) if you wish to take it and carry out some of the course analysis at home. Just ask the course instructors for the link to download this image. The following section explains how to import this image to VirtualBox on your computer.

Importing the OS Virtual disk into VirtualBox

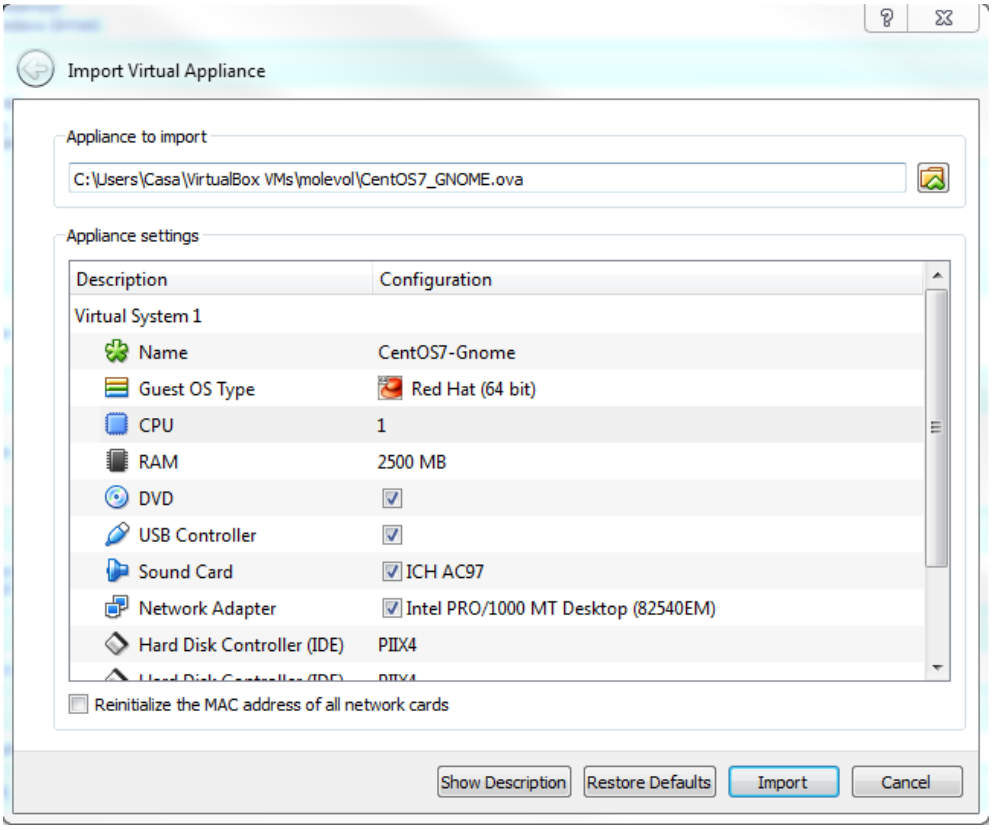
There are two main methods for importing the course VM to your VirtualBox installation: using an image disk file (vdi or vmdk file) or using an exported Open Virtualization Format Archive (OVF or OVA file). You will therefore need to check which type of file was made available to you and proceed accordingly.

Import from a OVA/OVF Archive:

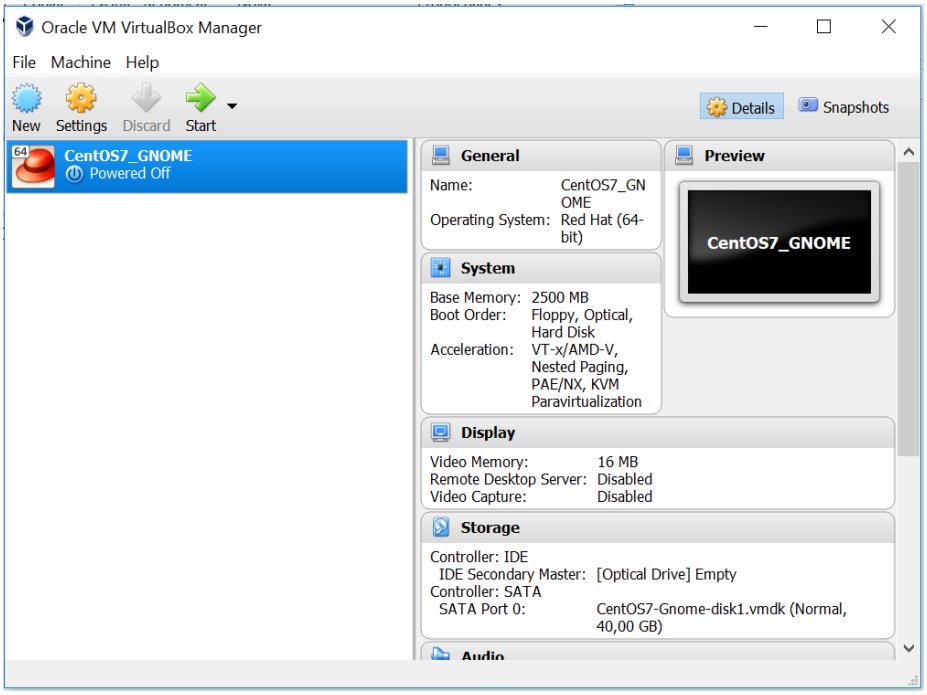
To import a VM in OVA/OVF format go to *File* > *Import Appliance*…



Then, select the OVA/OVF file from its location on your computer and click on *Import*.

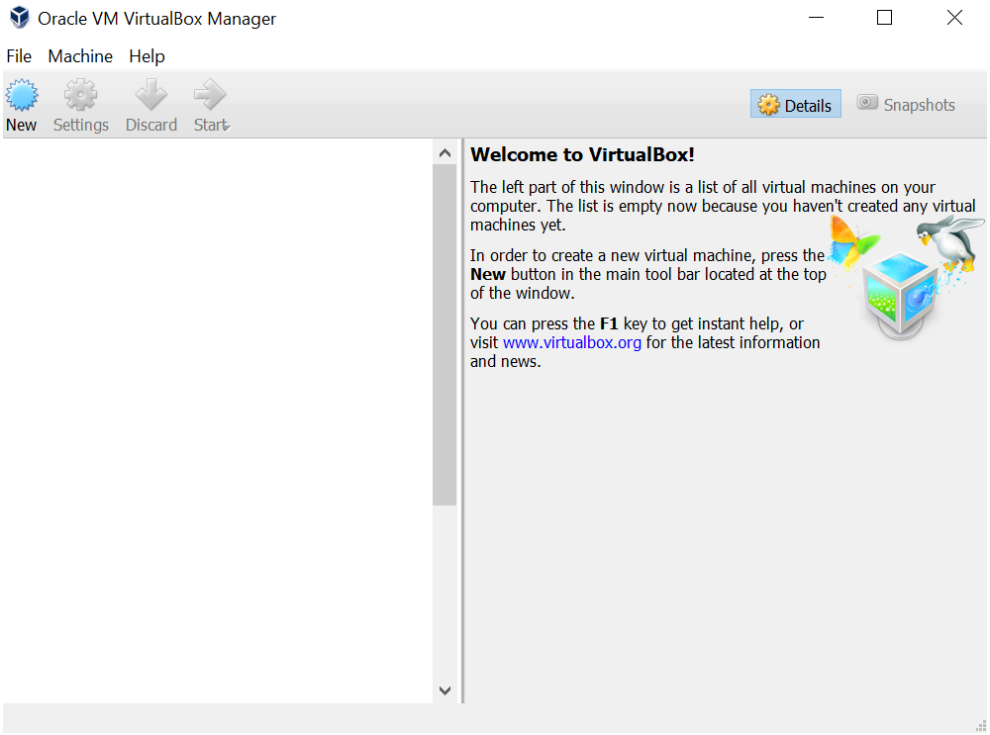


It may take some time to finalize the importation process.

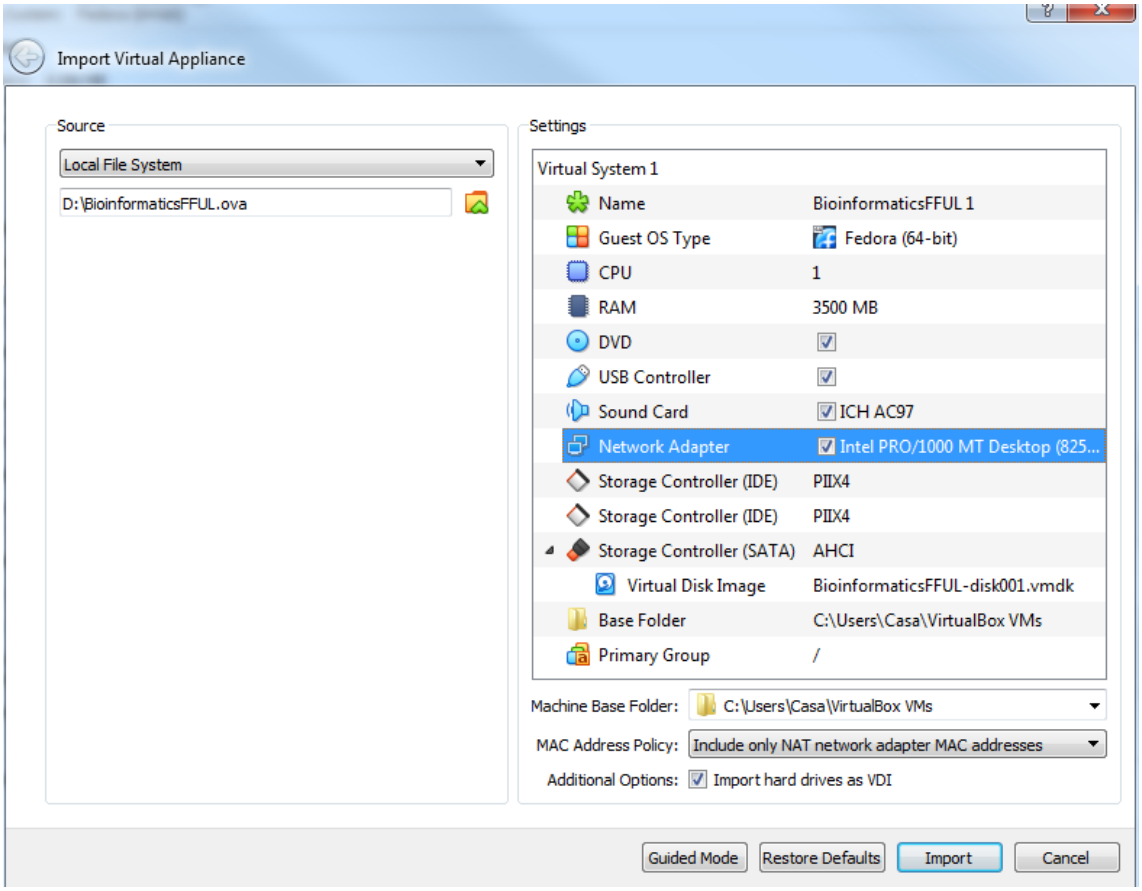


After the Importation process this VM will appear on VirtualBox VM list. To start it just select it and click *Start*.

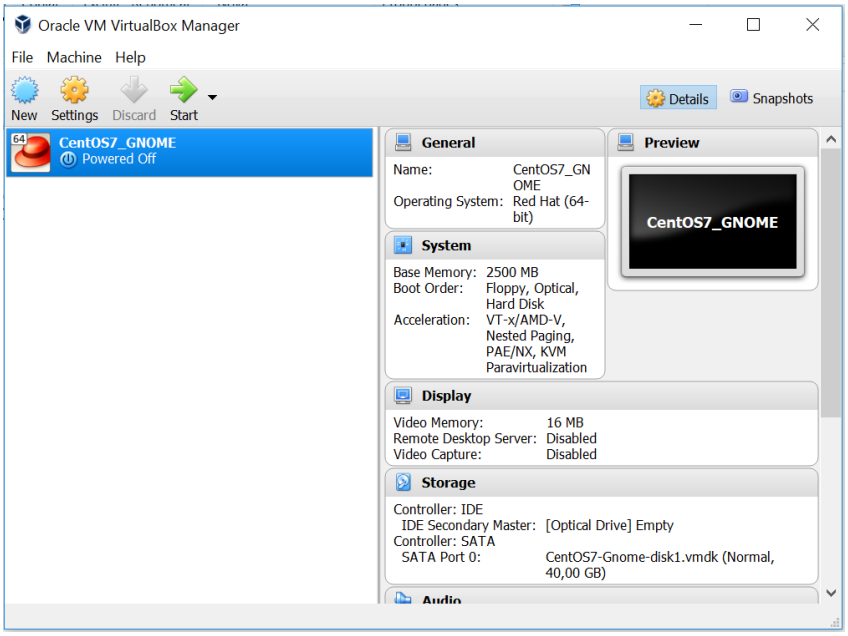
Import from a vdi or vmdk disk file:



Start VirtualBox and click *New*.



You can give it the name of your choice (e.g. BionformaticsFFUL), select Guest OS type as Fedora (64-bit). For memory size allocation it will depend on your system. For this course we recommend a minimum of 3000 Mb. After clicking on *Import* this VM will appear on VirtualBox VM list. To start it just select it and click *Start*.



About the Course VM

As mentioned above the course VM runs on a CentOS Stream Linux distribution. It comes with a GNOME Graphical User Interface (GUI), which is a free and open-source desktop environment that runs on Linux distributions.

This CentOS installation is configured to log in automatically with user centos. The user passwords are the following:

User: centos

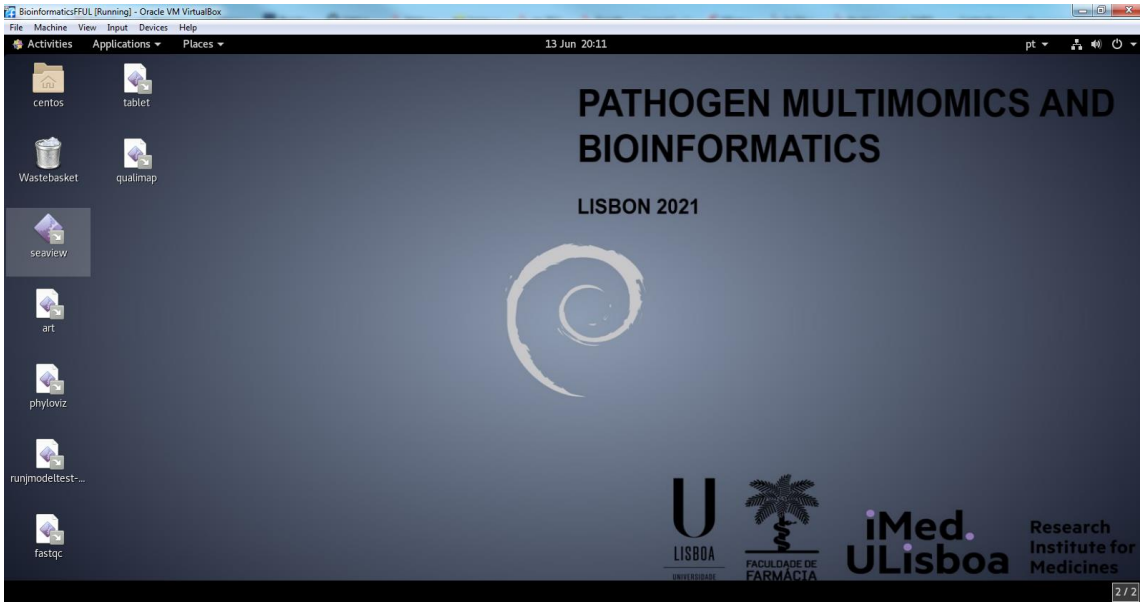
Password: centos

User: root

Password: centos

root is the super-user and it won't be necessary to use this account during the course. It may eventually become necessary if you deploy this VM onto your own computer and wish to undertake some system changes.

After logging in you will see the following desktop or a similar one:



Notice the desktop is already populated with some of the software that you will use over the course. Also, there are direct shortcuts to:

- your *home* directory folder (centos user *home* directory in this case) (Path: /home/centos/).
- open up a terminal/console window. Every time you open a new window from this shortcut it will open on your home directory.

You will find the course files within your home directory in the following folders/directories:

- **Module1** – Module 1 working directory with some files already present;
- **Module2** - Module 2 working directory with some files already present;
- **Module3** - Module 3 working directory with some files already present;
- **Module4** - Module 4 working directory with some files already present;
- **Module5** - Module 5 working directory with some files already present;
- **Module6**- Module 6 working directory with some files already present;
- **Module7** - Module 7 working directory with some files already present;
- **course_files** – Contains the FASTQ files to be used over the course;

Under the menu *Applications* in the desktop top bar you will find links to additional software and tools already installed (*e.g.*, Libre Office [an open-source alternative to Microsoft Office] or System Monitor).

:: Linux Command-line Basics ::

What is Linux?

During the 70's programmers from the Bell laboratories (AT&T) developed a new Operating System (UNIX) to overcome the existing limitations of the Operating Systems and Programming Languages of that time. Although initially used only for internal use at AT&T, universities and research centers became increasingly interested in this new OS with AT&T making available its source code. This enabled this new OS to expand and new extensions and tools to de developed by the scientific community.

However, AT&T started to restrict the licensing of UNIX and in order to protect the free software a new foundation called Free Software Foundation was created along with a new licence: General Public Licence (GPL). And from this new foundation a new UNIX-based OS was born: GNU (which means GNU is Not UNIX). In the 90's, Linus Torvalds inspired by the GNU project developed a new OS kernel: Linux, fully compatible with UNIX and GNU environments. Although quite rudimentary and incomplete, at its beginning Linux became increasingly popular, reaching one hundred new users in its first week. Over the last decades since its creation it has become a very robust and sophisticated OS and is currently at the same level of other commercial OSs, if not beyond.

Why on earth should I use Linux and the command-line?

First, it's free! And, being an open-source OS, each user can be a contributor to Linux development thereby increasing its reliability and stability since any error or bug can be readily corrected and incorporated in future releases.

Two of the most important Linux features are that this is a multitask and multiuser OS. Multiple users can be logged in simultaneously and can start multiple tasks/programs – called processes. Furthermore, each user is prevented from interfering with another user's work.

Presently, many Linux distributions already come bundled with graphical desktop environments (e.g. GNOME or KDE) but the command-line provides the user with a more flexible and enhanced control of the system. In the beginning it may be hard to know the commands but as the user gets familiar and learns how to use the command-line, it is possible to achieve a performance that is superior to the graphical interfaces.

Do I have to memorize all the commands?

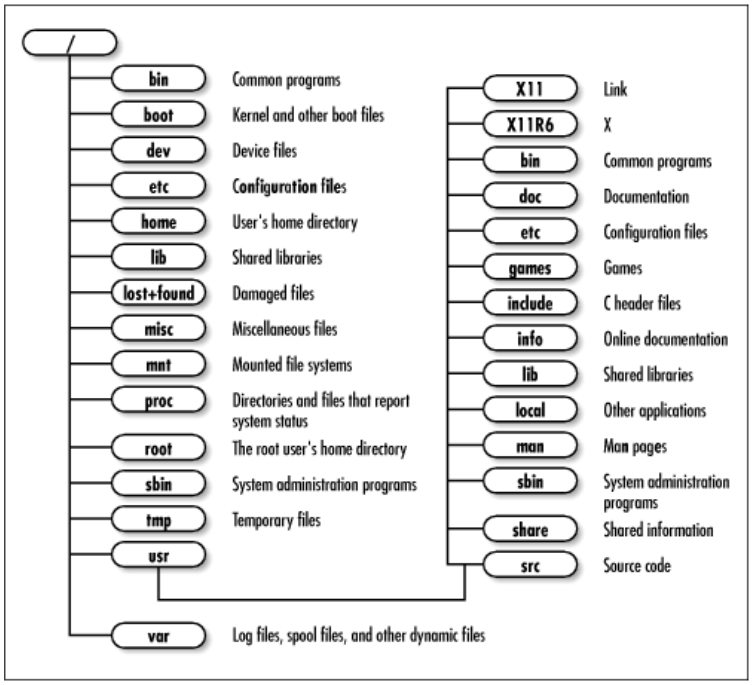
Actually no! Some of the commands are essential while others you will probably never use them. Linux comes with a plethora of basic commands and programs that you won't even know they are there. You'll end up by naturally assimilate this "new language" as you use these commands and if you don't recall any specific command, just look it up on the textbooks, online on webpages or the extensive community support forums or, in this document – that's what those are for!

Filesystem

As in windows or MacOS, files are organized in folders or directories where each directory can have multiple sub-directories and so on. The main directory (/) is called **root** and contains all directories and folders in the system. Each user only has access to its home directory unless if granted access to other directories or has administrative privileges. Each user home directory is located on /home/[user login] (in this course VM: /home/centos).

There is a special user: the **super-user** called **root**, because it has access to the root directory.

Take a look at an example of a Filesystem hierarchy:



The command-line

You can access to the command-line by opening a terminal window using the shortcut on your desktop. It is called command-line because the user must write commands in the form of text. Once you open the command-line interface you will see an initial message looking like this:

```
[centos@localhost ~]$
```

This is called the prompt and it tells you that the system is waiting for commands. It also tells you that you are logged in as user *centos* at (@) the *localhost* (the machine name you are using) and you are at your home directory (~). This is important because you can use the command-line to switch to another user and to connect to another machine.

Basic commands

You can start by obtaining a list of the files and sub-directories that exist in your current directory by typing:

```
$ ls
```

```
[centos@localhost ~]$ ls
commands.txt          Downloads   Music             Public       test.sh
course_documentation  files_copy  NC000962_3.fasta  R            Videos
course_files          Module1     NC000962_3.gbk    RAST_output
Desktop               Module2     other             Templates
Documents             Module3     Pictures          test.Rexec
[centos@localhost ~]$
```

Or if you want to obtain a more detailed list (long) list you can type:

```
$ ls -l
```

```
[centos@localhost ~]$ ls -l
total 17544
-rw-rw-r--. 1 centos centos     5429 Sep  1 18:51 commands.txt
drwxrwxr-x. 2 centos centos        6 Sep 11 10:22 course_documentation
drwx------. 2 centos centos     4096 Aug 19 18:45 course_files
drwxr-xr-x. 2 centos centos     4096 Sep  7 11:26 Desktop
drwxr-xr-x. 2 centos centos        6 Nov  1  2014 Documents
drwxr-xr-x. 5 centos centos     4096 Sep 11 10:26 Downloads
drwxrwxr-x. 2 centos centos     4096 Aug 19 18:29 files_copy
drwxrwxr-x. 3 centos centos       61 Sep  8 14:14 Module1
drwxrwxr-x. 3 centos centos     4096 Sep 11 10:23 Module2
drwxrwxr-x. 6 centos centos       66 Sep 14 16:39 Module3
drwxr-xr-x. 2 centos centos        6 Nov  1  2014 Music
-rw-r--r--. 1 centos centos  4474567 Jul  6 13:25 NC000962_3.fasta
-rw-r--r--. 1 centos centos 13443692 Jul 23  2013 NC000962_3.gbk
drwxrwxr-x. 5 centos centos       35 Sep 11 10:31 other
drwxr-xr-x. 2 centos centos     4096 Sep  7 11:36 Pictures
drwxr-xr-x. 2 centos centos        6 Nov  1  2014 Public
drwxrwxr-x. 3 centos centos       44 Aug 16 23:38 R
drwxrwxr-x. 2 centos centos        6 Sep  8 17:54 RAST_output
drwxr-xr-x. 2 centos centos        6 Nov  1  2014 Templates
-rwxrwxr-x. 1 centos centos       80 Aug 17 02:20 test.Rexec
-rwxrwxrwx. 1 centos centos       80 Aug 17 02:21 test.sh
drwxr-xr-x. 2 centos centos        6 Nov  1  2014 Videos
[centos@localhost ~]$
```

Important note: the commands are case-sensitive so Ls or LS will not work. Keep this always in mind!!

This more exhaustive list gives a number of details that we will not cover here but notice the modification date and time and size in bytes. Notice that NC000962_3.gbk file has a size of 13443692 bytes. It seems a lot but it is only 13.4 Mb!

Now, how do we find the full path of the directory where we are working? Just type:

```
$ pwd
```

It means print working directory outputs this:

```
[centos@localhost ~]$ pwd
/home/centos
```

What if I want to change directory:

```
$ cd Module1
```

cd means change directory and you can use the full path of a directory to change to a distant directory from where you are (e.g. cd /home/centos/Module1/vcfs). When you don't use the full path, cd assumes that the directory you are specifying exists in the present working directory. Now type pwd. What do you see?

```
$ pwd
```

```
[centos@localhost Module1]$ pwd
/home/centos/Module1
```

To go back type:

```
$ cd ..
```

On the command-line, two dots (..) means the directory above while one dot (.) means the current directory. That is why if you type `cd .` it will not get you anywhere but the present directory.

File manipulation

Let's say we wish to manipulate files and directories. To create a new directory type:

```
$ mkdir test_dir
```

To remove it:

```
$ rmdir test_dir
```

If the directory contains files rmdir will not work,you have to type:

```
$ rm -r test_dir
```

Which is more drastic. Try this and use the cd or ls command to go inside and see your directory.

And what about files?

You can copy (cp command) or move (mv command) files easily from the command-line:

```
$ cp NC000962_3.fasta Documents
```

This command copied NC000962_3.fasta file to your Documents directory. To avoid errors youl would write it in the following format:

```
$ cp ./NC000962_3.fasta ./Documents/
```

That way it is more intelligible that you are copying it to a folder. But here you are not specifying if you want the copied file to remain with the same name in the destination folder, the prompt assumes that it should stick with the same name. Otherwise:

```
$ cp ./NC000962_3.fasta ./Documents/test.fasta
```

The mv command works approximately in the same way, except that it deletes the original file.

To remove a file just type:

```
$ rm ./Documents/test.fasta
```

Can we visualize text files on the command-line? Sure, let's use the cat command (from your home directory, to go there from anywhere type cd ~) and type:

```
$ cat NC000962_3.fasta
```

Did you see it all? Probably not. By the way, cat takes multiple files and can concatenate those, hence the name cat. You can use CTRL+S and CTRL+Q to stop the scroll or, scroll up using SHIFT+PgUp or SHIFT+PgDn when the listing stops.

To see a file gradually with stoppings, type:

```
$ more NC000962_3.fasta
```

If you press ENTER or SPACE it will scroll down, to exit just press q.

Now let's stop for a moment and introduce three other characters:

- >, redirects the output to a file and erases a previously existing file if it has the same name;
- >>, redirects the output but appends it to the end of an existing file;
- |, named pipe character, does exactly that, pipes or channels the output of a command to another.

We can, for example, do:

```
$ ls -l > list.txt
```

This created a new file (list.txt). Let's read the content:

```
$ more list.txt
```

Is it familiar?

And if you do this:

```
$ ls -l >> list.txt
$ more list.txt
```

What now?

Wild-cards

To end this basic Linux tutorial. We'll introduce wild-cards and regular expressions. Wild-cards are special characters that enable you to call many files recursively. These are:

Character	Meaning
*	Any sequence of one or more characters
?	Any single character
[]	A sequence of one or more characters containing the characters within brackets

Let's use wild-cards to list files that only start with NC:

```
$ ls -l NC*
```

Or if we only want fast files:

```
$ ls -l *.fasta
```

You can use wild-cards on file operations as well:

```
$ cp ./NC* ./Documents/
```

What happened? Do you notice you have copied both files to the documents directories?

Let's delete those:

```
$ rm ./Documents/NC*
```

You just deleted two files with a single command. Be aware of the risks of using wild-cards as you can easily delete thousands of files with a single command. But, notice the power that comes with the command-line.

Hope this basic introduction is helpful for the following course Modules.