



LinuxTutorial Documentation

Release latest

Sebastian Schmeier (<https://sschmeier.com>)

Feb 11, 2020

CONTENTS

1	An very quick introduction to Linux	3
1.1	Preface	3
1.2	Learning outcomes	3
1.3	Introduction	3
1.4	The Linux desktop environment	4
1.5	Some words regarding the Linux file-system	4
1.6	Open a terminal	6
1.7	Proxy settings	8
1.8	The Command-line interface (CLI)	10

This is a very basic introductory tutorial to Linux.

AN VERY QUICK INTRODUCTION TO LINUX

1.1 Preface

This tutorial is based on a Linux/Unix *command-line*. Using the *command-line* requires a Linux/Unix operating system. The easiest way to try out a Linux system without actually installing it on your computer is a [LiveCD](#)¹. A LiveCD is a CD/DVD that you prepare (e.g. burn a Linux distribution on it) and insert in your computer. You would restart you computer and can run Linux from the CD/DVD without any installation requirements. This is helpful for trying out a distribution of Linux, not for actual work.

Another route would be to use a virtual machine. A virtual computer that runs within your normal host system, e.g. Windows or MacOSX. The software to create a virtual machine is free, e.g. [VirtualBox](#)².

Common flavors of Linux ready for download are e.g. [Ubuntu](#)³.

An accompanying lecture for this tutorial is available at [figshare](#)⁴.

1.2 Learning outcomes

1. Be able to operate comfortably the command-line.
2. Be able to navigate the unix directory structure on the command-line.
3. Be able to start command-line programs and getting help/information about programs.
4. Be able to investigate text files with command-line commands.
5. Be able to investigate the content of text-files on the command-line.
6. Be able to explain the concept of a unix pipe.

1.3 Introduction

We will go through some introductory material explaining the shell syntax. This is bash syntax but most of it will work on other shells (tcsh, zsh) as well.

What is a shell? Here I shamelessly quote [Wikipedia](#)⁵:

“In computing, a shell is a user interface for access to an operating system’s services. In general, operating system shells use either a command-line interface (**CLI**) or graphical user interface (GUI), depending on a computer’s role and particular operation. . .”

“**CLI** shells allow some operations to be performed faster in some situations, especially when a proper GUI has not been or cannot be created. However, they require the user to memorize

¹ https://en.wikipedia.org/wiki/Live_CD

² <https://www.virtualbox.org/>

³ <https://help.ubuntu.com/community/LiveCD>

⁴ <http://dx.doi.org/10.6084/m9.figshare.1506799>

⁵ <https://goo.gl/g9x4tE>

all commands and their calling syntax, and also to learn the shell-specific scripting language, for example bash script.”

1.4 The Linux desktop environment

An example of an environment in Linux (here LXDE from Lubuntu) and is similar to other user interfaces found in Windows or MacOSX (Fig. 1.1).

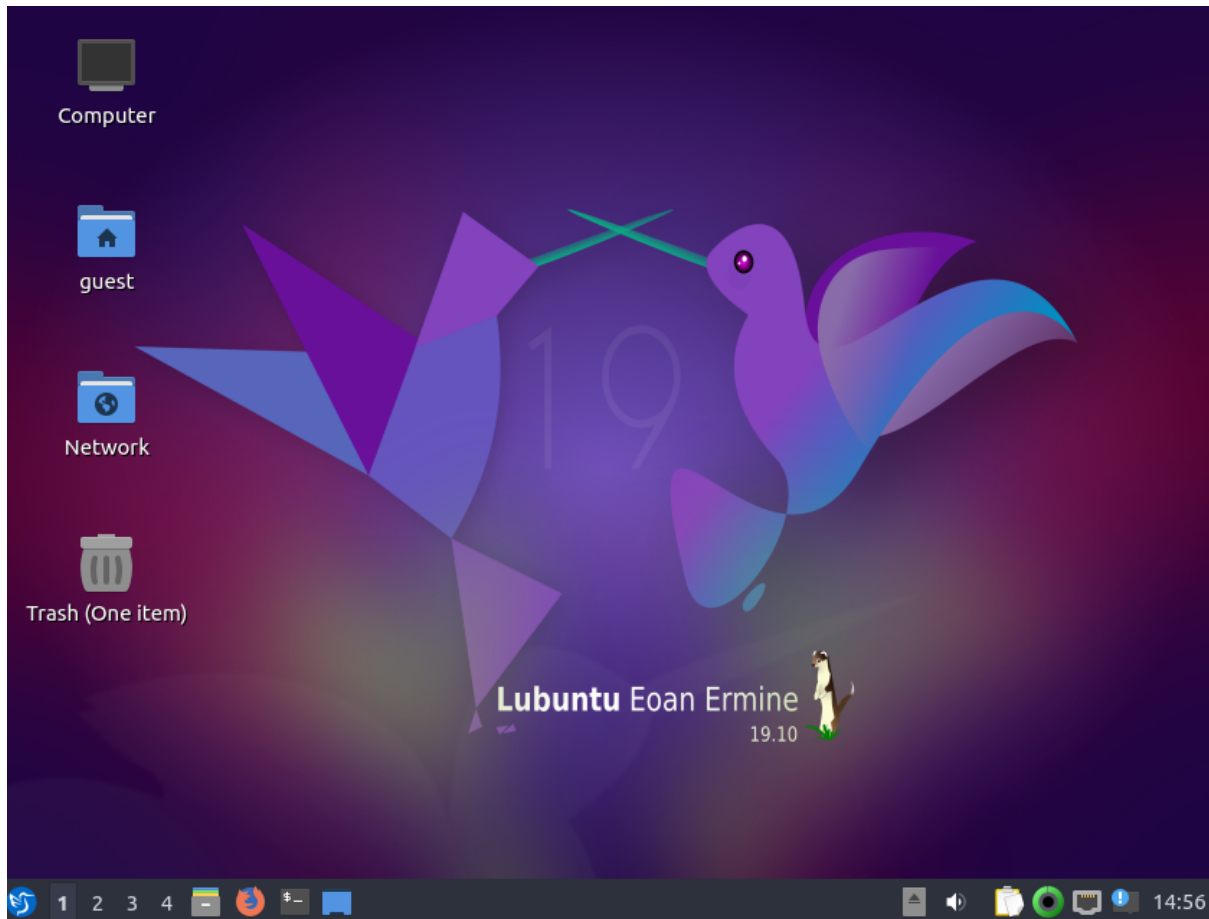


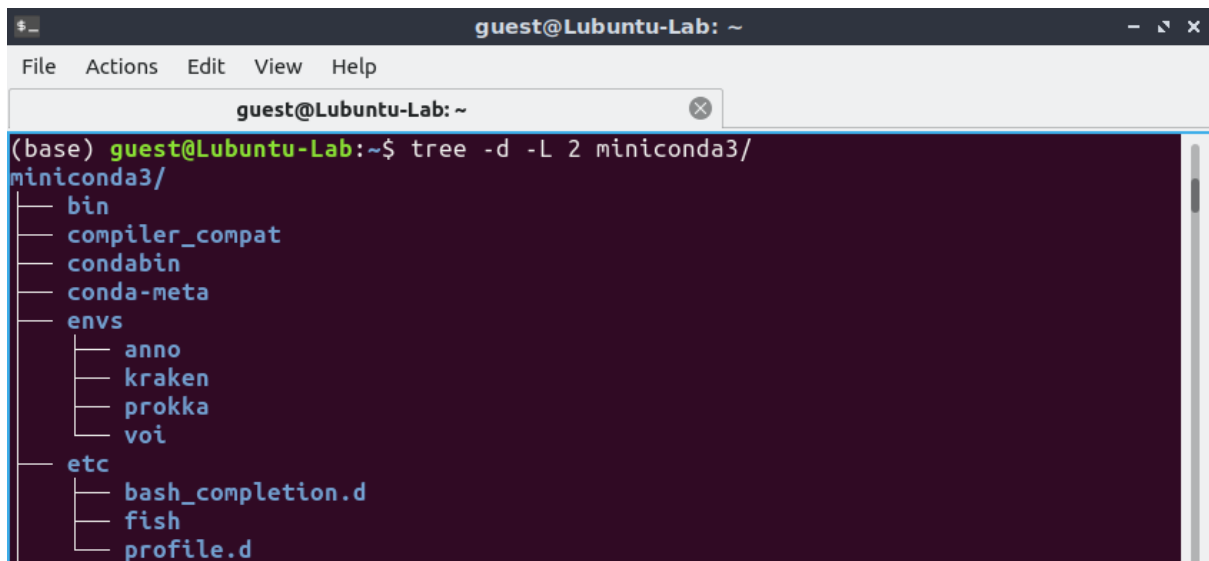
Fig. 1.1: The Linux desktop environment LXDE as seen on Lubuntu (a variety of Ubuntu).

1.5 Some words regarding the Linux file-system

The directory structure in a Linux system is not much different from any other system you worked with, e.g. Windows, MacOSX. It is essentially a tree structure (Fig. 1.2).

To navigate the file-system you can use a file-manager e.g. “Files” the default file manager in the Unity window manager used by BioLinux (Fig. 1.3).

However, on the command-line we navigate via commands and not via mouse clicks. Why is it necessary to use the command-line in the first place? Strictly speaking it is not, if you do not want to make use of programs on the command-line. However, the power of the Linux system becomes only obvious once we learn to make use of the command-line, thus navigating the directory structure via commands is one of the **most important skills** for you to learn.



```

guest@Lubuntu-Lab: ~
File Actions Edit View Help
guest@Lubuntu-Lab: ~
(base) guest@Lubuntu-Lab:~$ tree -d -L 2 miniconda3/
miniconda3/
├── bin
├── compiler_compat
├── condabin
├── conda-meta
├── envs
│   ├── anno
│   ├── kraken
│   ├── prokka
│   └── voi
└── etc
    ├── bash_completion.d
    ├── fish
    └── profile.d
  
```

Fig. 1.2: Quick look at the directory tree structure on the command-line.

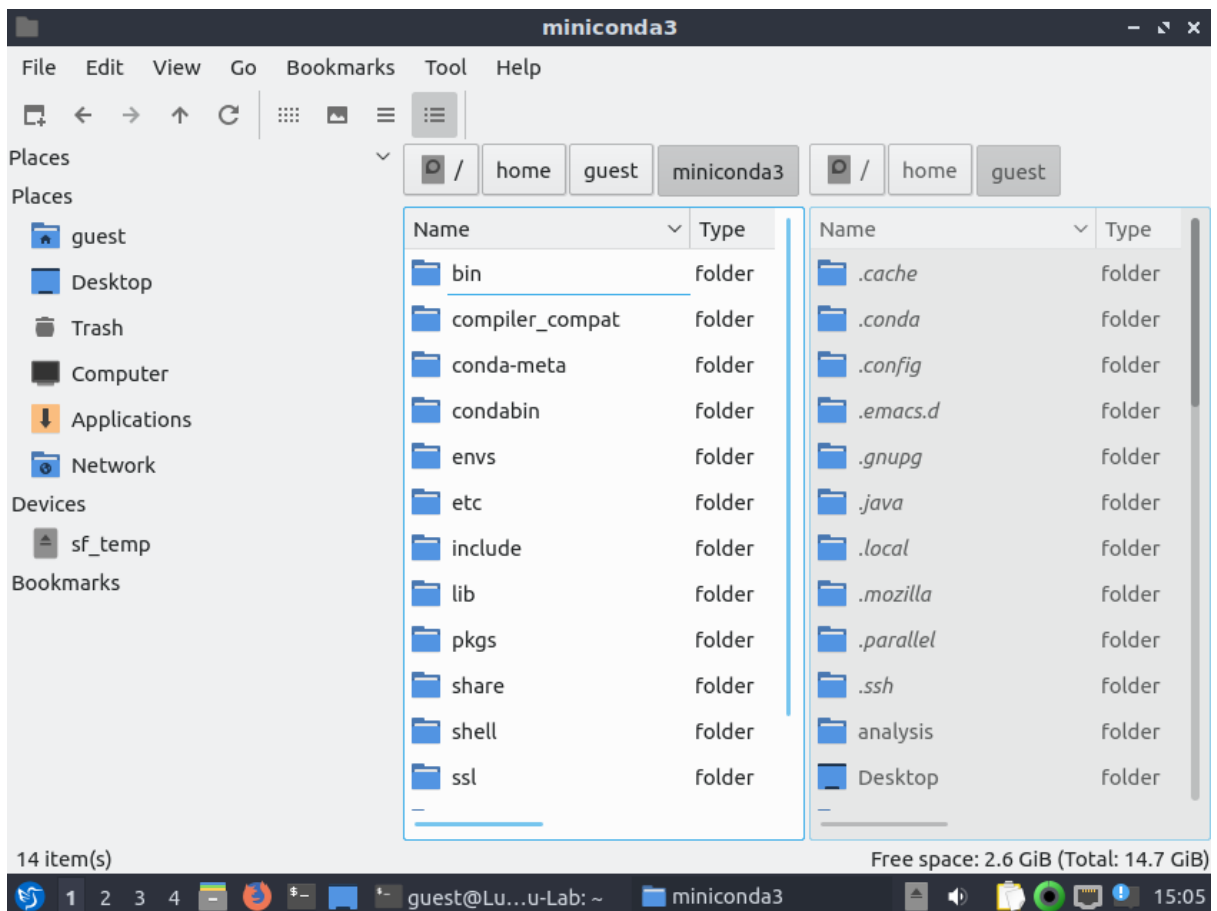


Fig. 1.3: Quick look at the directory tree structure in the “Files” GUI.

1.6 Open a terminal

Open a terminal window and you are ready to go. On your linux desktop find: **System Tools** -> **QTerminal** (for LXDE environment) or type “Terminal” in the search box (Fig. 1.4).

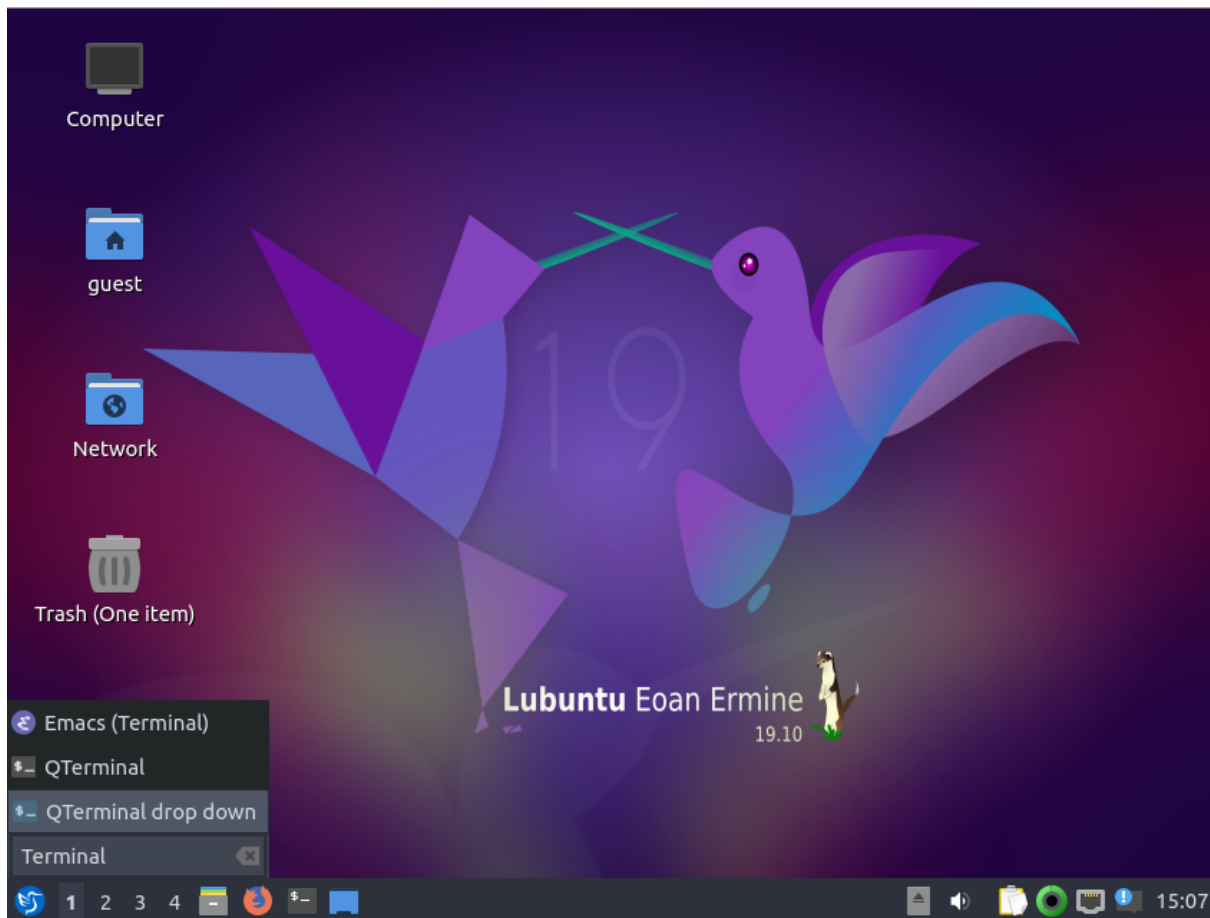


Fig. 1.4: LXDE search bar.

Fig. 1.5 shows an example of how a terminal window might look like (it is very easy to change its appearance). You will see this window to execute the commands to work with files and biological data. However, it is by no means restricted to “biological data”, once you know how to handle the command-line many tasks based on files will be easily achieved using various programs available here.

Note: The command-line prompt (e.g. \$ or >) indicates that the shell/terminal is waiting for commands from us. These will be sent to the computer to execute. As long as you do not see the prompt, the computer is busy processing your request.

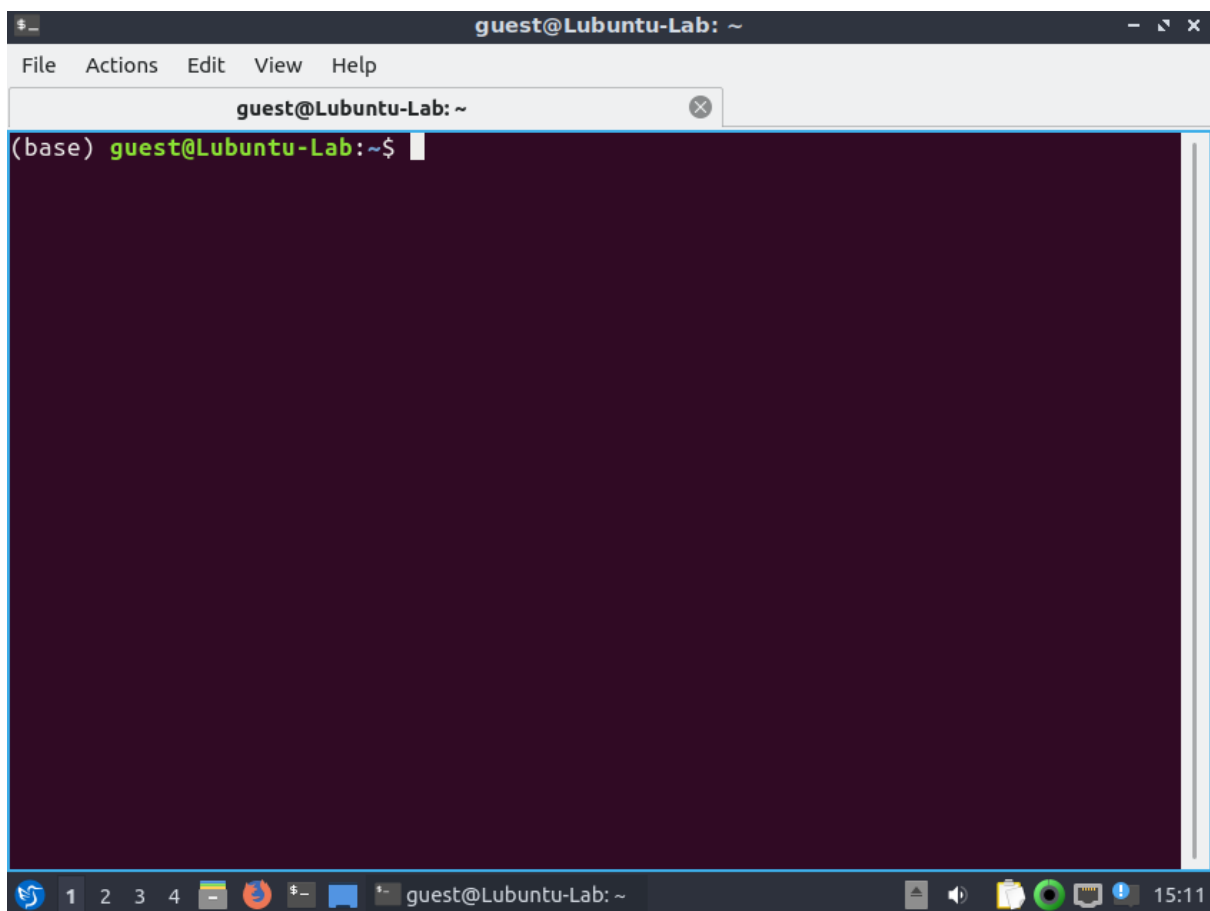


Fig. 1.5: An example of a terminal window in Unity.

1.7 Proxy settings

You might encounter problems connecting to the internet. So this is most likely the case if your university has restrictions in place to make the network more secure. One of these measures to make a network more secure is a proxy. However, we need the internet. Follow these steps to get connected.

1.7.1 Open network settings

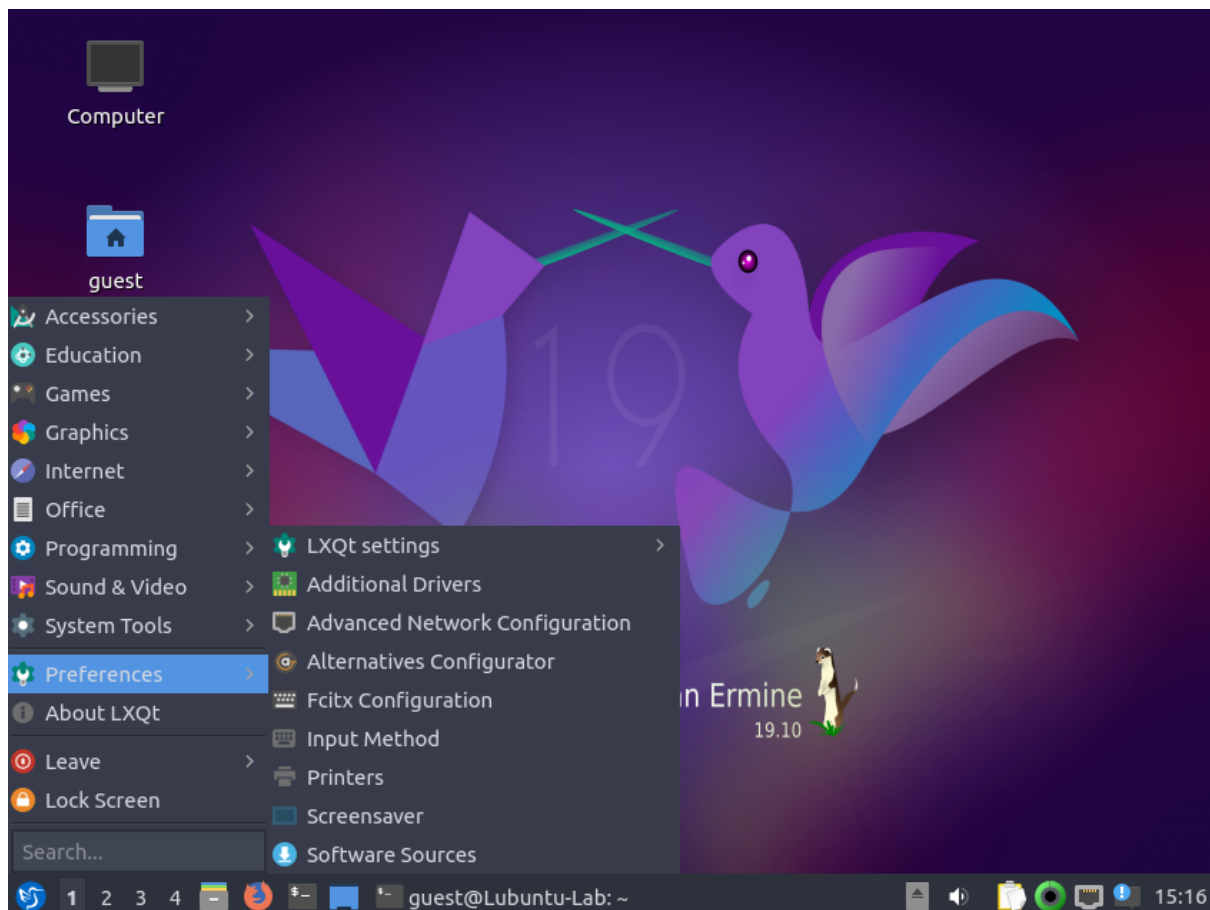


Fig. 1.6: Accessing network settings.

1.7.2 Change proxy settings to automatic

After changing the proxy settings to automatic you can open a web-browser and you should be asked for your network *username* and *password*. After you typed those and hit *Enter*, you should be connected.

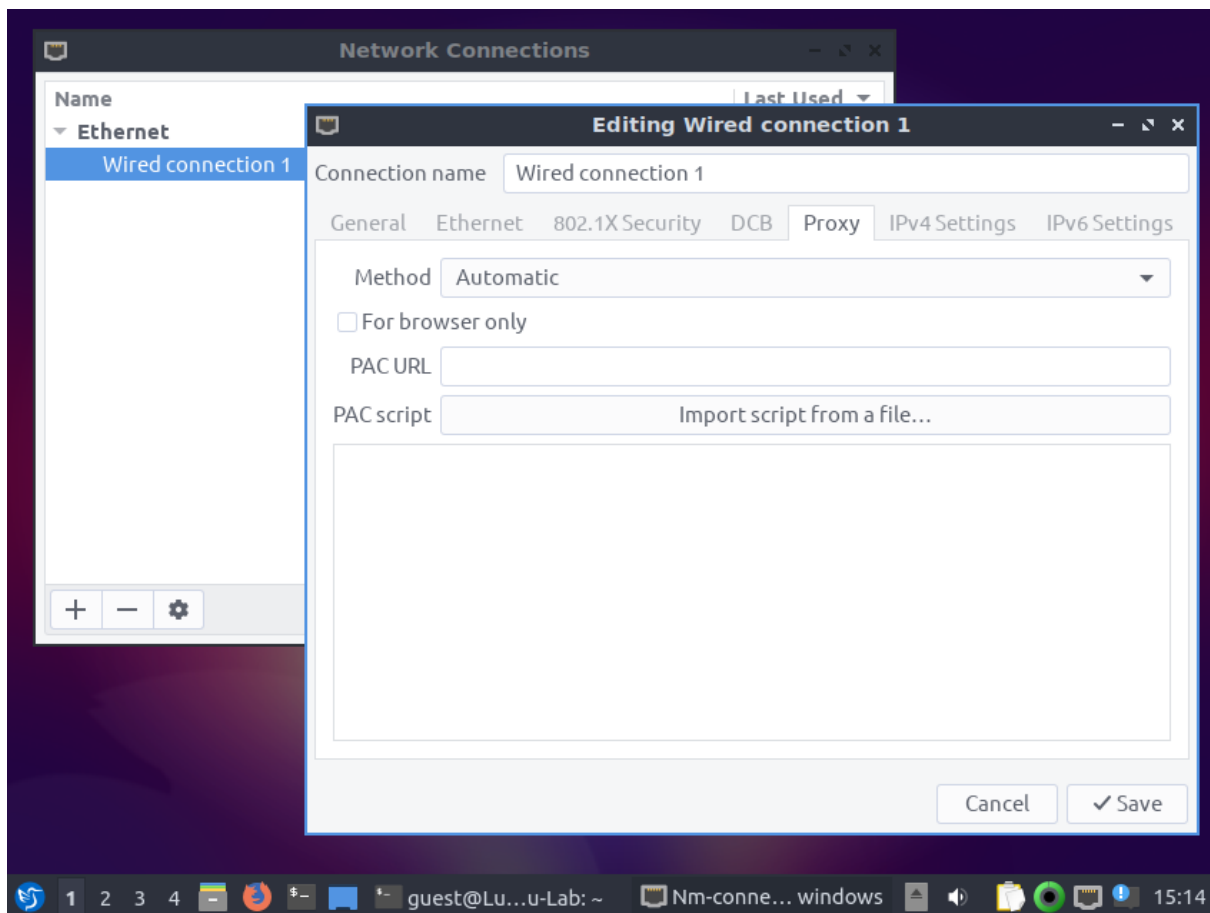


Fig. 1.7: Changing proxy settings.

1.8 The Command-line interface (CLI)

Pre-2016 you would find my own introductory course in this place. However, since 2016 we will be using the excellent material from the [Software Carpentry](http://software-carpentry.org)⁶ Foundation. I am a SWC affiliated volunteer instructor and we are teaching basic computer skills to scientists, with the goal of general computational up-skilling in the sciences. The material is created in a collaborative manner and tested over and over in many workshops.

Please follow the link to the material at the [Software Carpentry](http://swcarpentry.github.io/shell-novice)⁷ and have the webpage open. Otherwise you will work in the terminal window aka the shell.

⁶ <http://software-carpentry.org>

⁷ <http://swcarpentry.github.io/shell-novice>