POLITECNICO
DI TORINO

Introduction to Linux

Operating Systems – Sarah Azimi

Before Linux

- In 80’s, Microsoft DOS was the dominated OS for PC
- Apple MAC was better, but expensive.
- UNIC was much better, but much much more expensive.
- People were looking for a UNIX based system, which is cheaper and can run on PC.
- DOS, MAC and UNIX were proprietary, i.e., the source code of their kernel is protected.
- No modification is possible without paying high license fee.

Beginning of Linux

- A famous professor Andrew Tanenbaum, developed Minix, a simplified version of UNIX that runs on PC.
- Minix is for class teaching only. No intension for commercial use.

Linus Torvalds

- In 1991, a second-year student of Computer Science at the university of Helsinki, developed the preliminary kernel of Linux, known as Linux version 0.0.1.
- Soon more than a hundred people joined the Linux camp. Then hundreds of thousands.
- It was licensed under GNU General Public License, this ensuring that the source codes will be free for all to copy, study and to change.

Linus Torvalds

Linux Today

- Linux has been used for many computing platforms.
  - PC, supercomputers,….
  - Not only character user interface but graphical user interface is available.

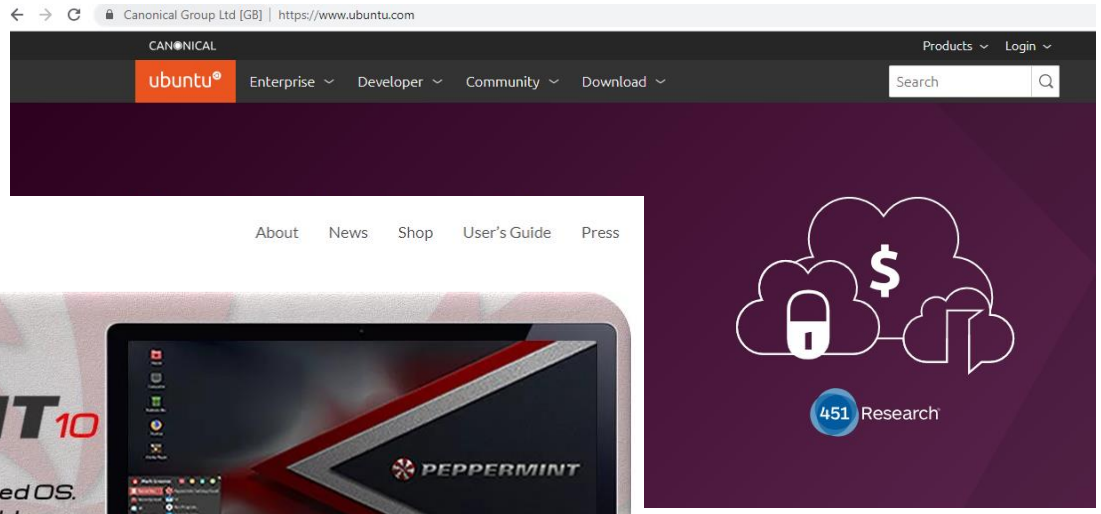
Linux Distributions

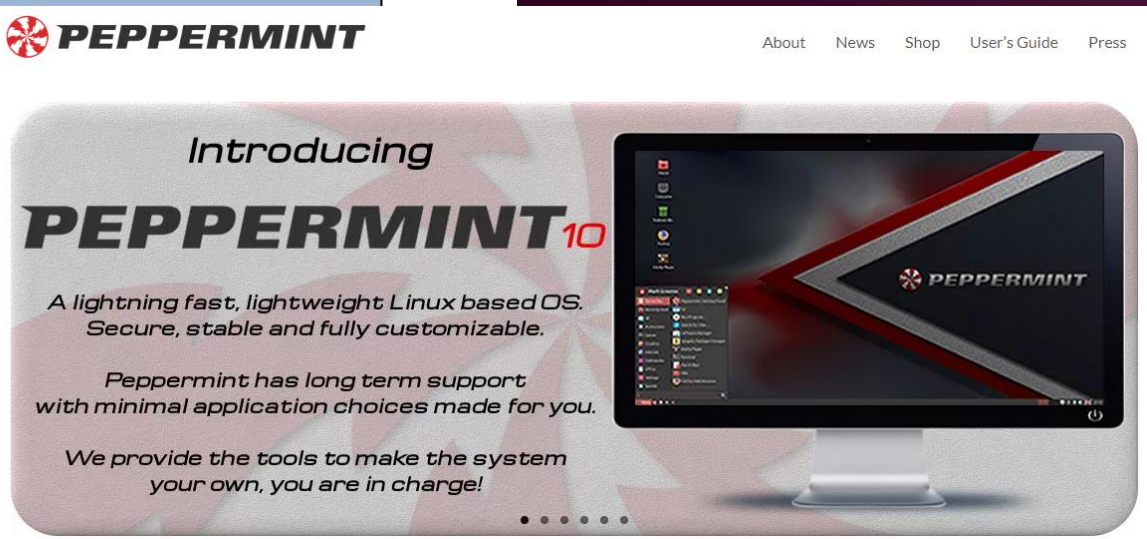
- Red Hat Linux
  - One of the original Linux distribution. The commercial, non-free version is Red Hat Enterprise Linux, which is aimed at big companies using Linux servers and desktops in a big way.
- Debian GNU/Linux
  - A free software distribution. Popular for use on servers. However, Debian is not what many would consider a distribution for beginners, as it is not designed with ease of use in mind.
- **Ubuntu**
  - One of the free and most incredibly easy to use free distributions of Linux.

Virtual Machines and Virtualization

- Enables a single PC or server to simultaneously run multiple operating systems or multiple sessions of a single OS.
- A machine can host numerous applications, including those that run on different operating systems, on a single platform.
- Host operating system can support a number of virtual machines (VM)
  - Each has the characteristics of a particular OS and, in some versions of virtualization, the characteristics of a particular hardware platform.

What to do now?





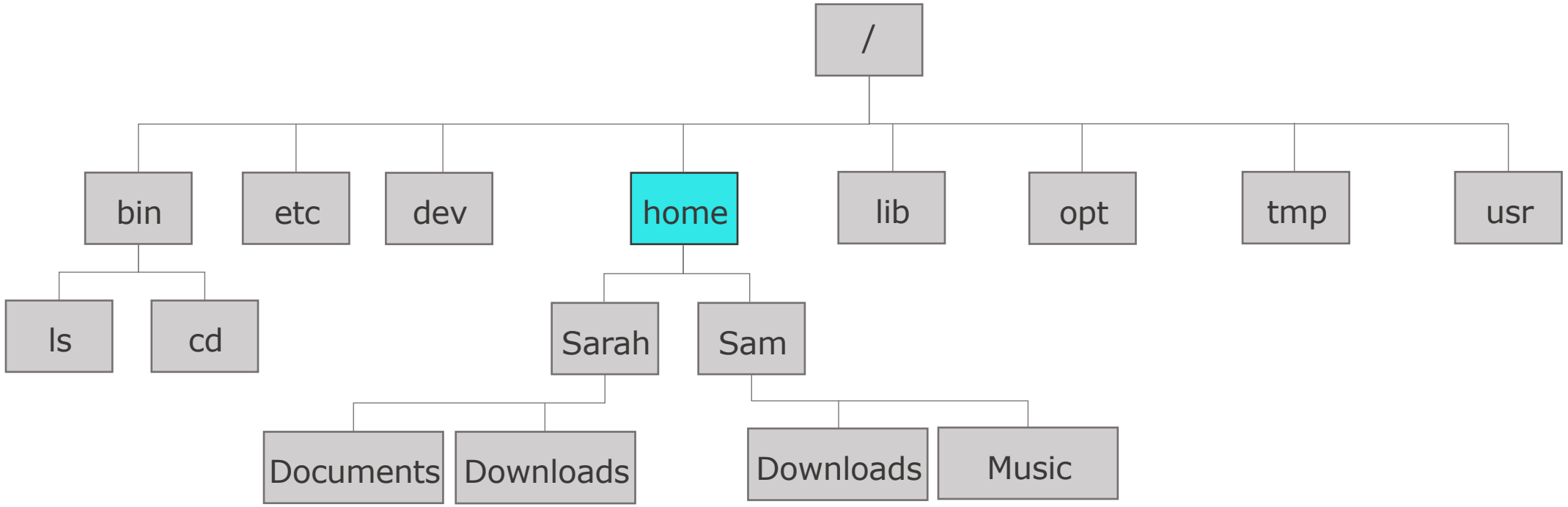
Peppermint_installation_guide.pdf

Online Manual

- All Linux commands have an online documentation
  - **man <command>**
  - **whatis <command>**

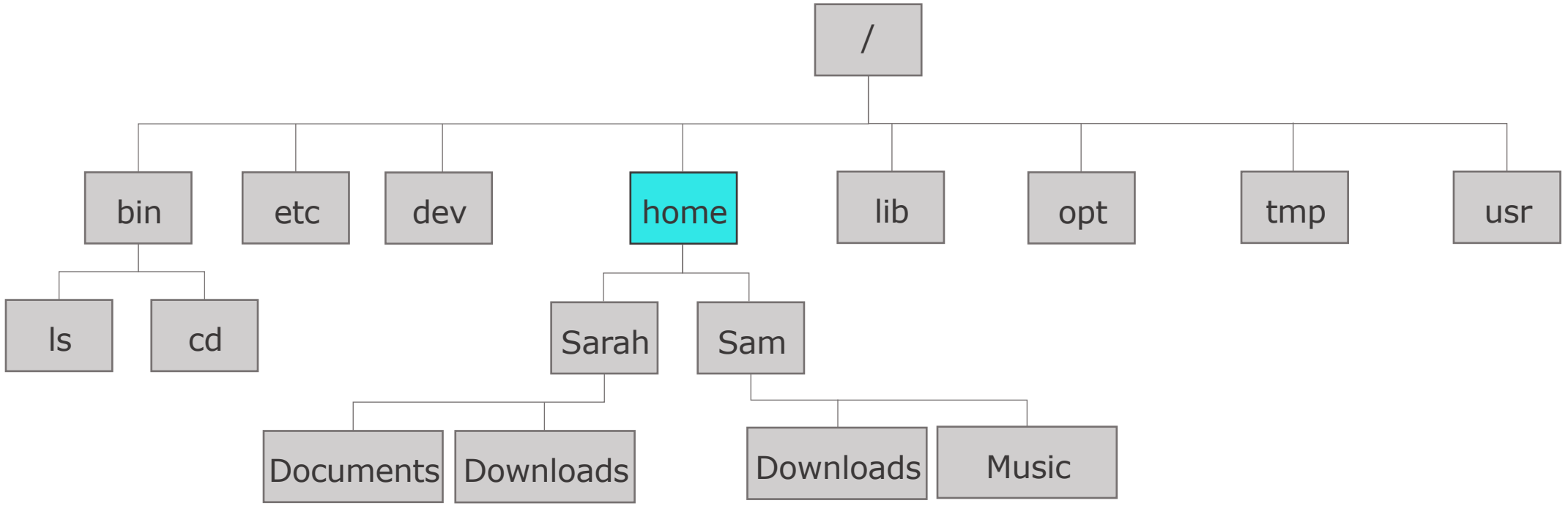
Directory Tree

- When you log on the Linux OS using your username, you are automatically located in your home directory.
- **Home** Every user gets her own folder in here, named for her logic account. So, the user who logs in with Sarah, has the directory /home/sarah, where all her personal files are kept.



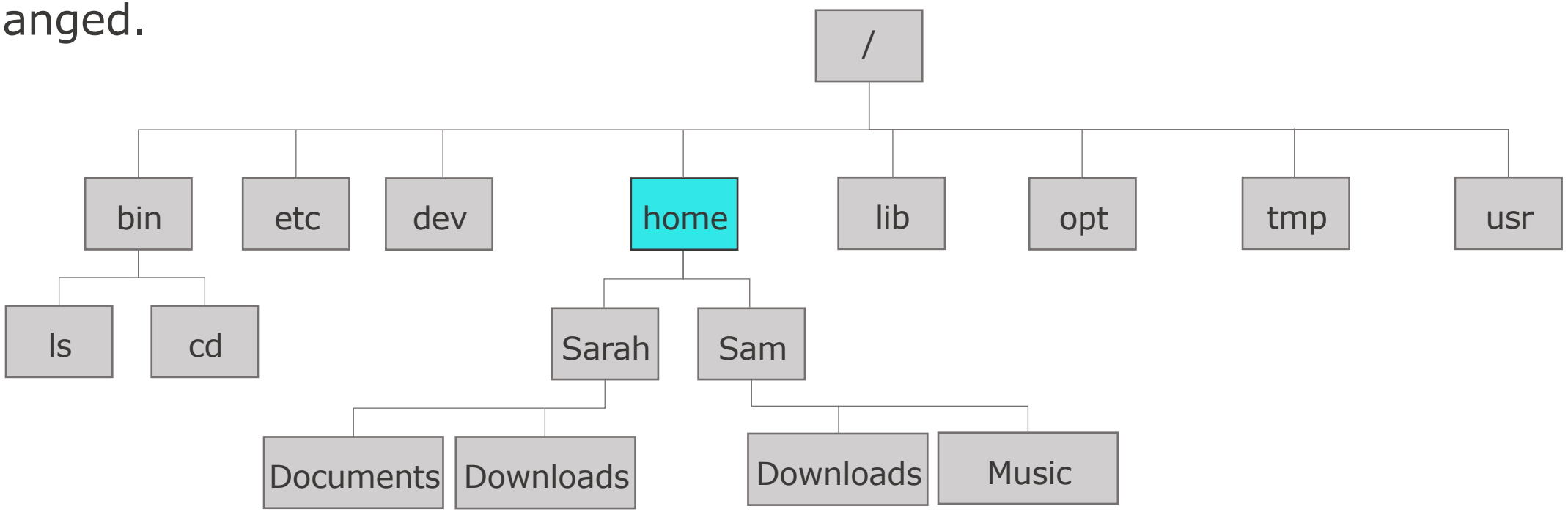
Directory Tree

- When you log on the Linux OS using your username, you are automatically located in your home directory.
- **/bin** Important Linux commands available to the users.
- **/dev** All device drivers. Device drivers are the files that your Linux system uses to talk to your hardware.
- **/etc** System Configuration files.



Directory Tree

- When you log on the Linux OS using your username, you are automatically located in your home directory.
- **/lib** System libraries. Libraries are just bunches of programming code that the programs on your system use to get things done.
- **/tmp** Temporary files and storage space. Don't put anything here that you need.
- **/usr** Programs and data that can be shared across many systems and don't need to be changed.



Basic Commands

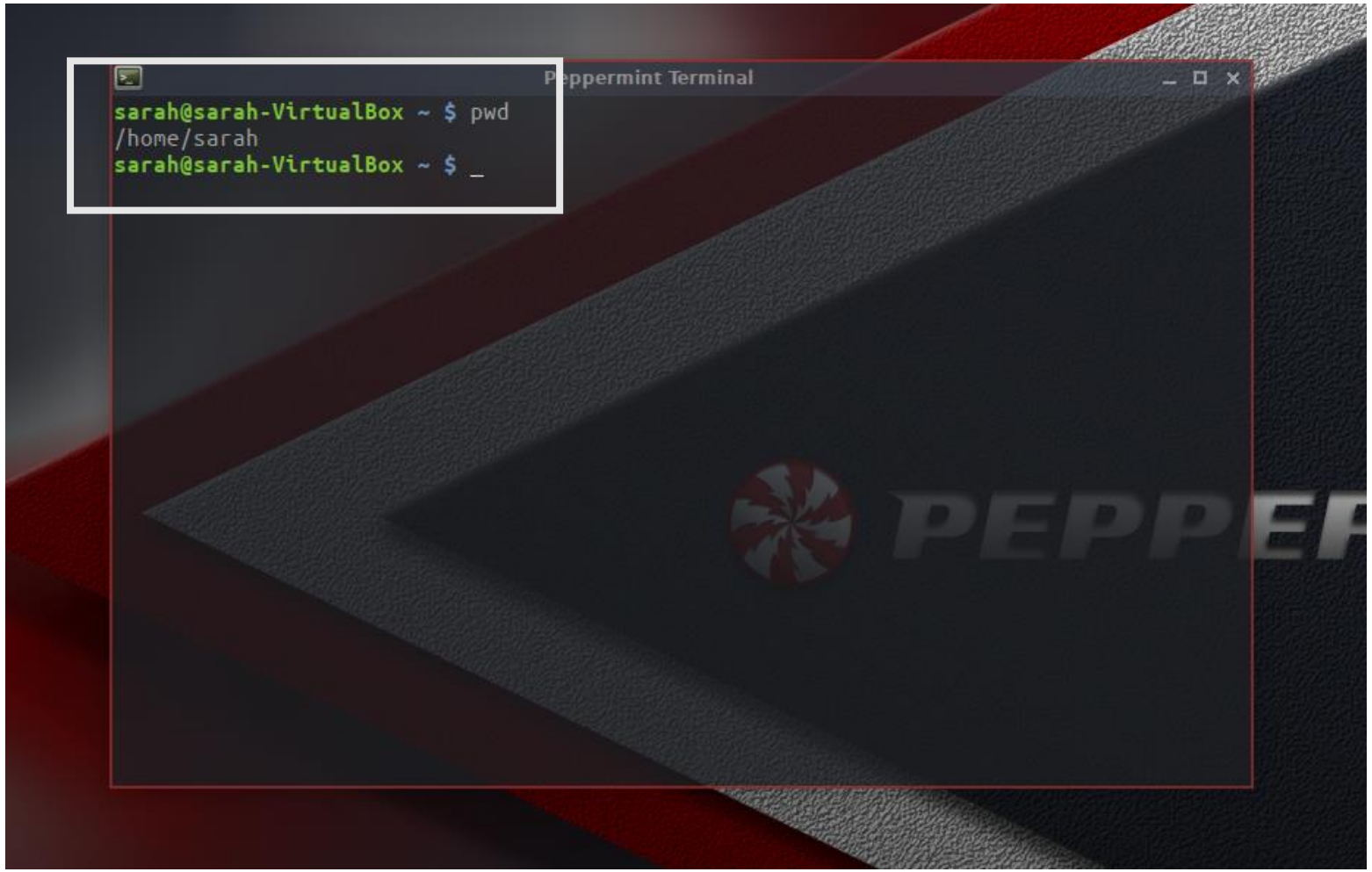
- **pwd** To know which directory you are in. it gives an absolute path.
- **ls** To know what files are in the directory you are in.
- **cd** To go to a directory
- **mkdir** To create a new directory
- **rmdir** To delete an empty directory
- **rm** To delete files and directories
- **touch** To create a file (from an empty txt file to an empty zip file)
- **man** To know more about a command and how to use it.
- **cp** To copy files
- **mv** To move files through the command lines
- **cat** To display the contents of a file
- **nano, vi** The text editors in Linux (The nano command is a good editor)

Basic Commands

- **sudo** Stands for "SuperUser Do". if you want any command to be done with administrative or root privileges, you can use the sudo command.
- **apt-get** To install packages. This requires root privileges, so use the sudo command with it.
- **df** To see the available disk space in each of the partitions in the system.
- **du** To see the disk usage of a file in your system.
- **chmod** To change permission granted to a file.
- **chown** To change the owner and the group owner of a file.
- **echo** To print a string of text to the terminal window.
- **ps** To list the running processes.
- **hostname** To know your name in your host or network.

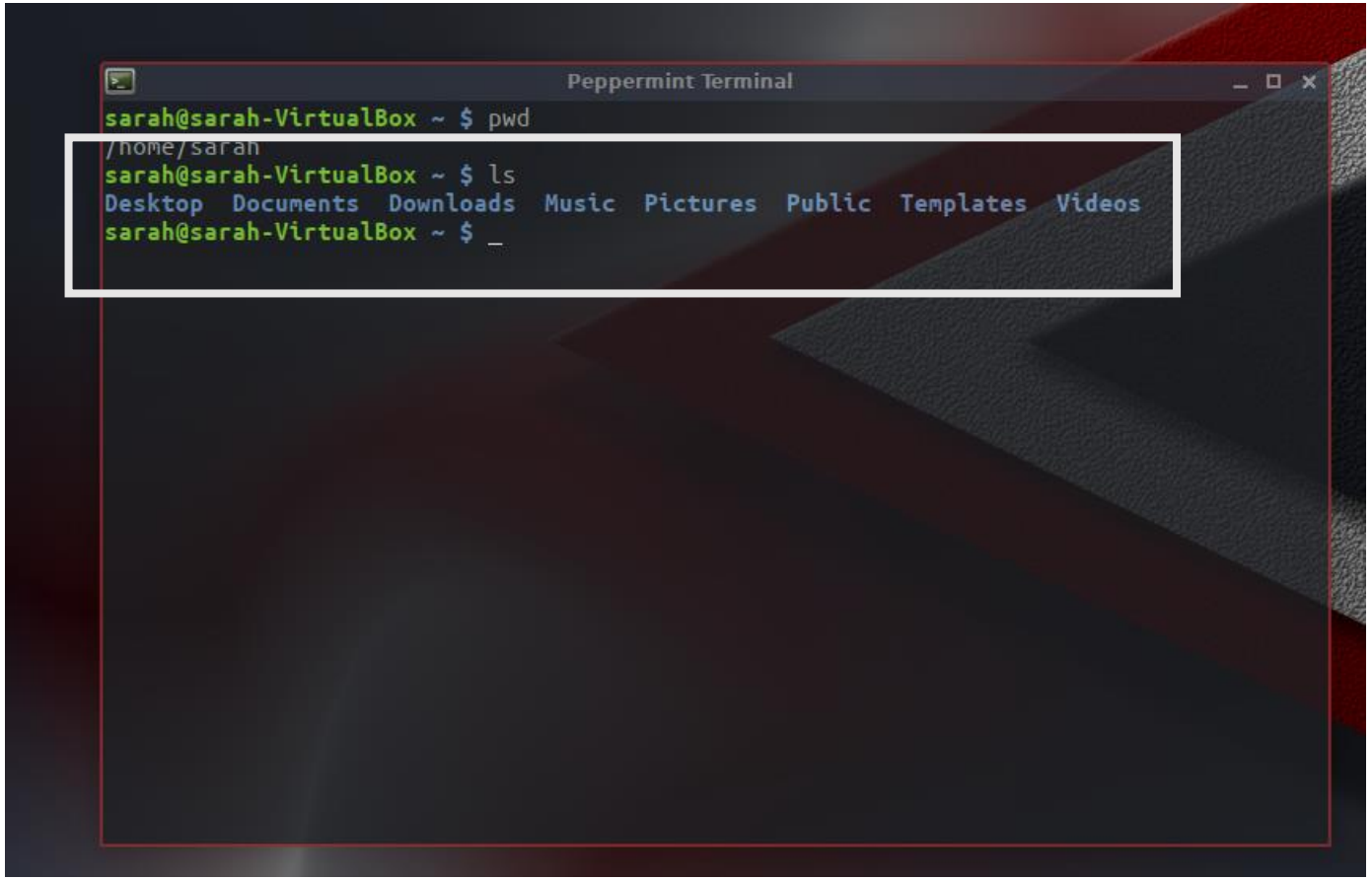
Basic Commands

- **pwd** To print the current working directory.



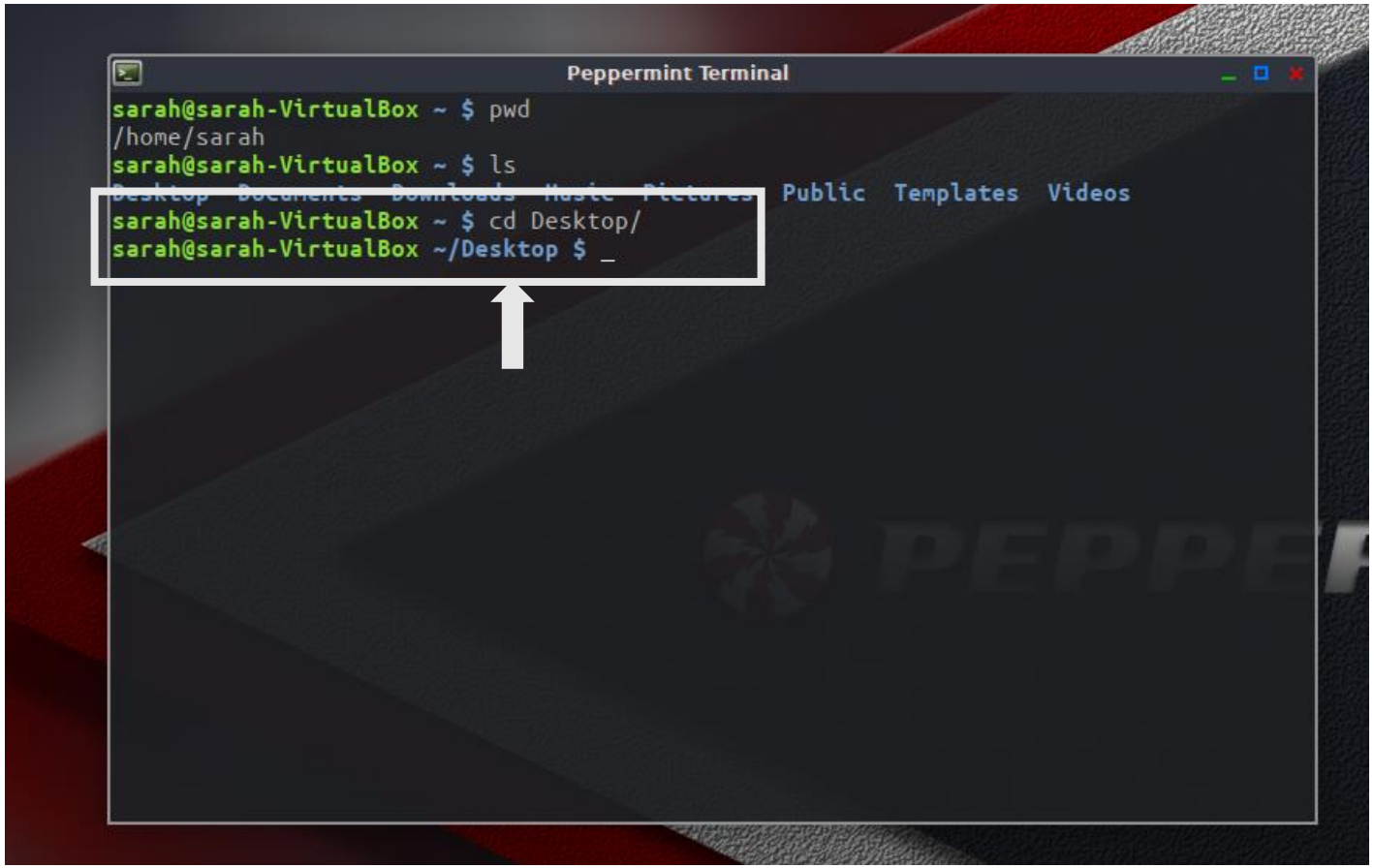
Basic Commands

- **ls** To list the contents of a directory.



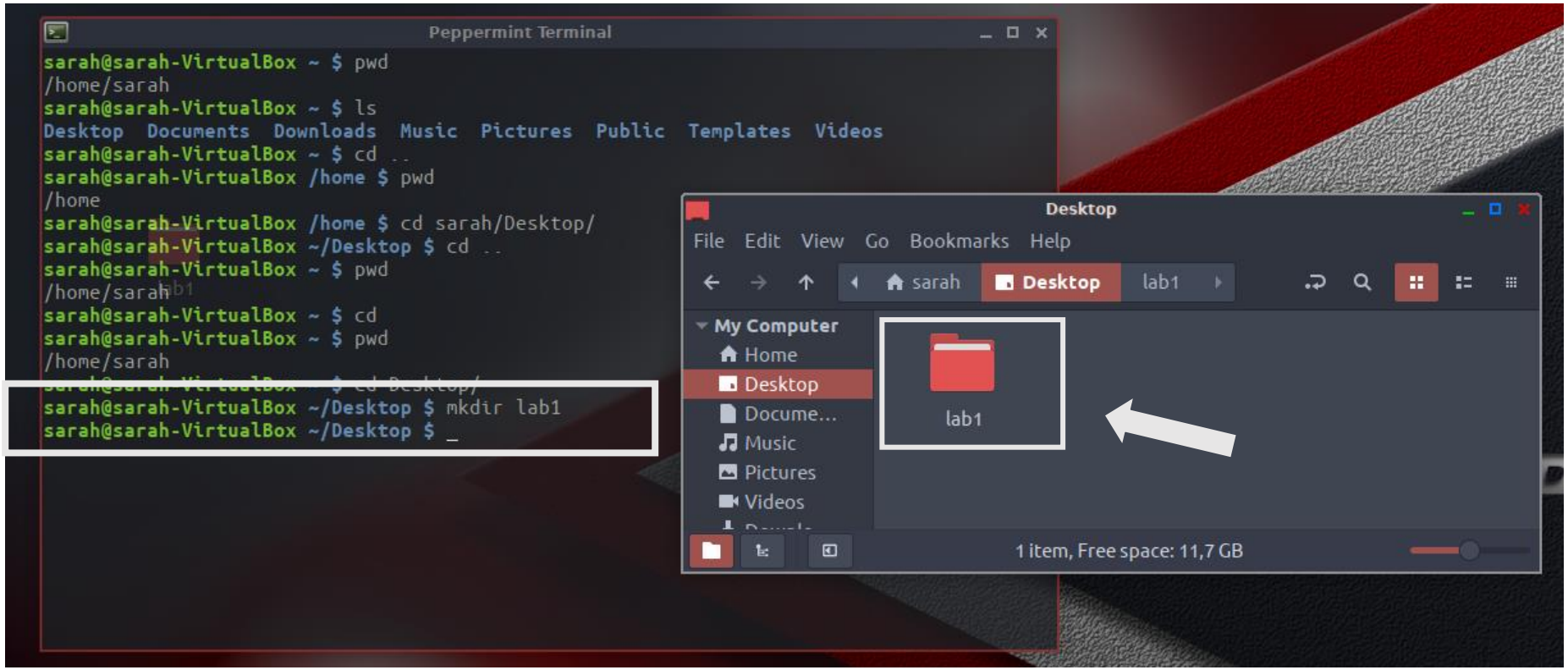
Basic Commands

- **cd** To enter the directory x.



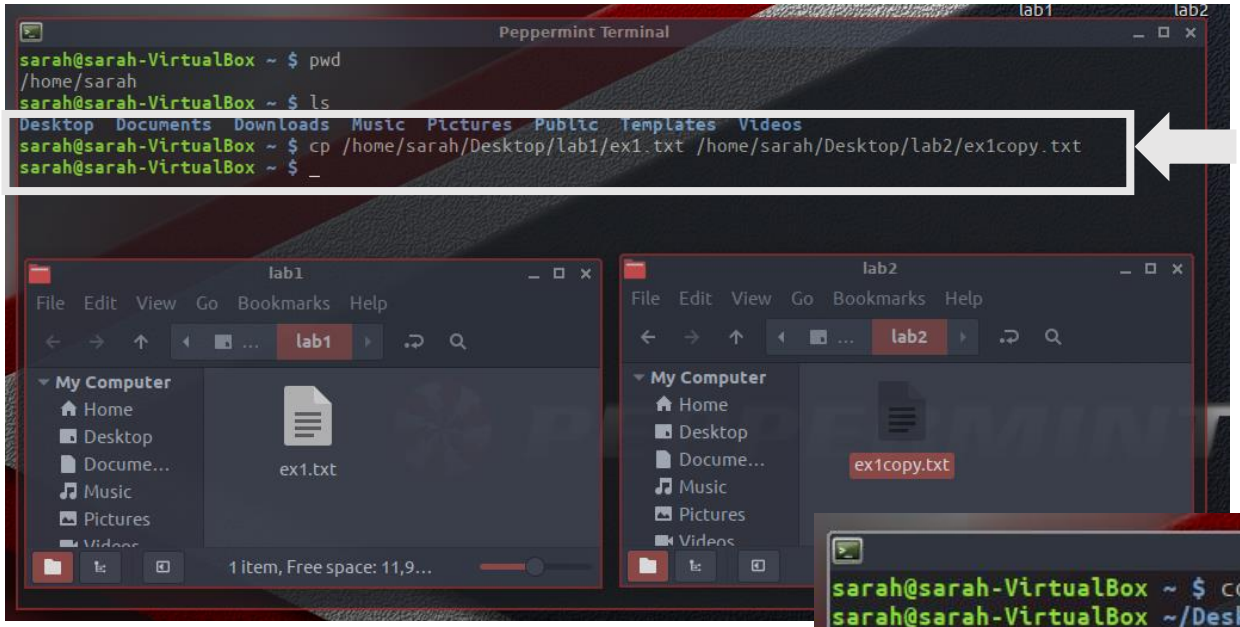
Basic Commands

- **mkdir** To create a new directory.

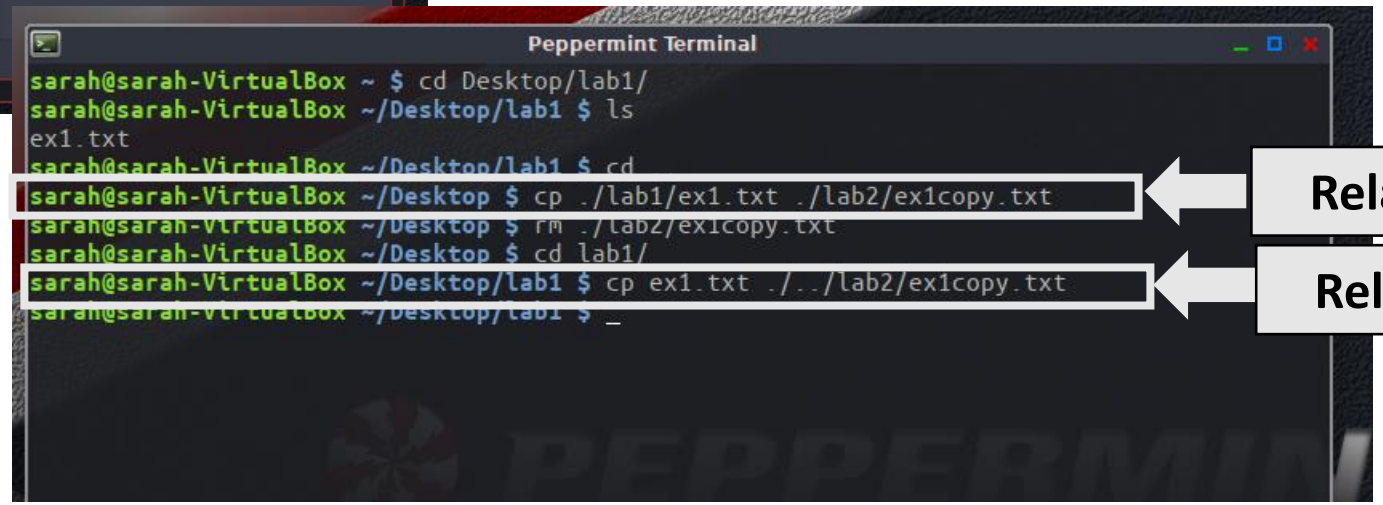


Basic Commands

- **cp** To copy files and directories (absolute and relative paths).



Absolute path



Relative path

Relative path

Basic Commands

- **chmod** To modify the access right of the file.

To whom permissions apply:

u >> owner (file's owner)

g >> group (users who are members of the file's group)

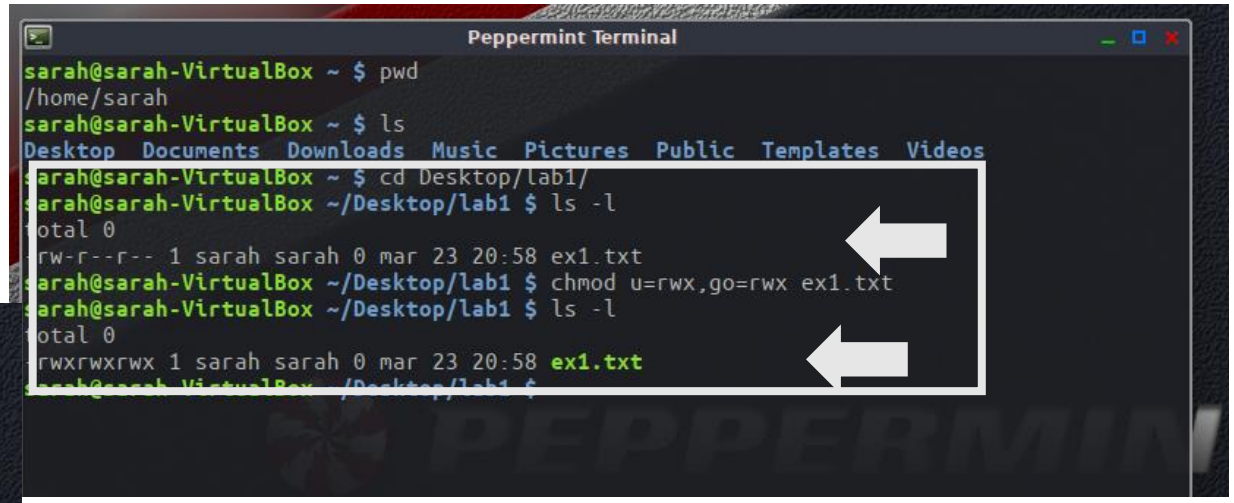
o >> others (who are neither the owner or member of file's group)

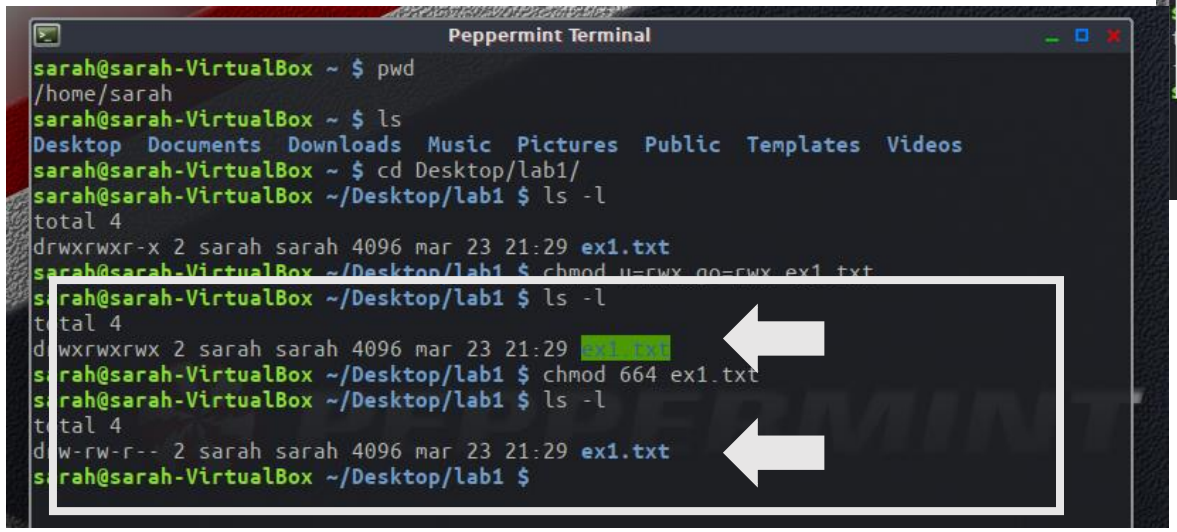
Access rights are:

read >> r

write >> w

executable >> x





Access rights are:

read >> 4

write >> 2

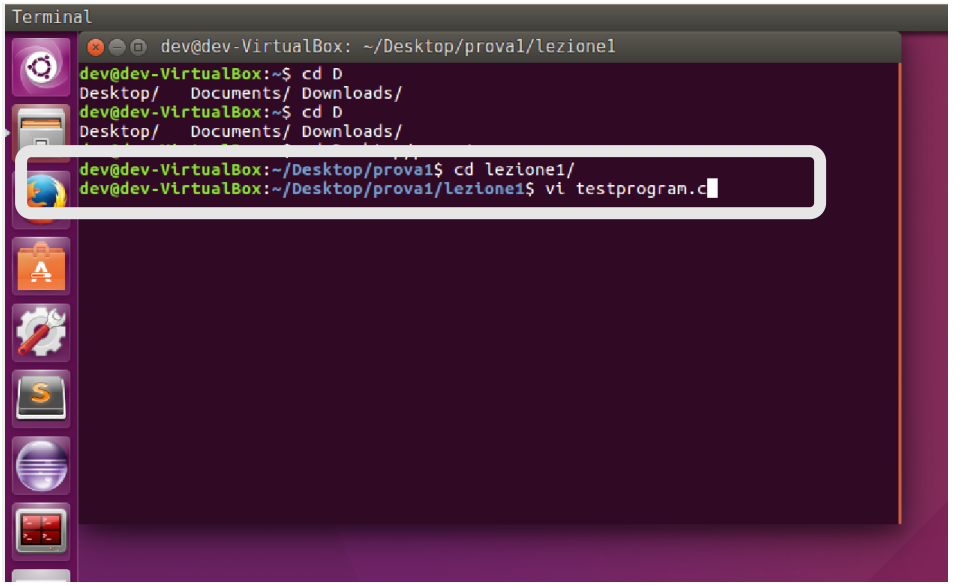
executable >> 1

without right >> 0

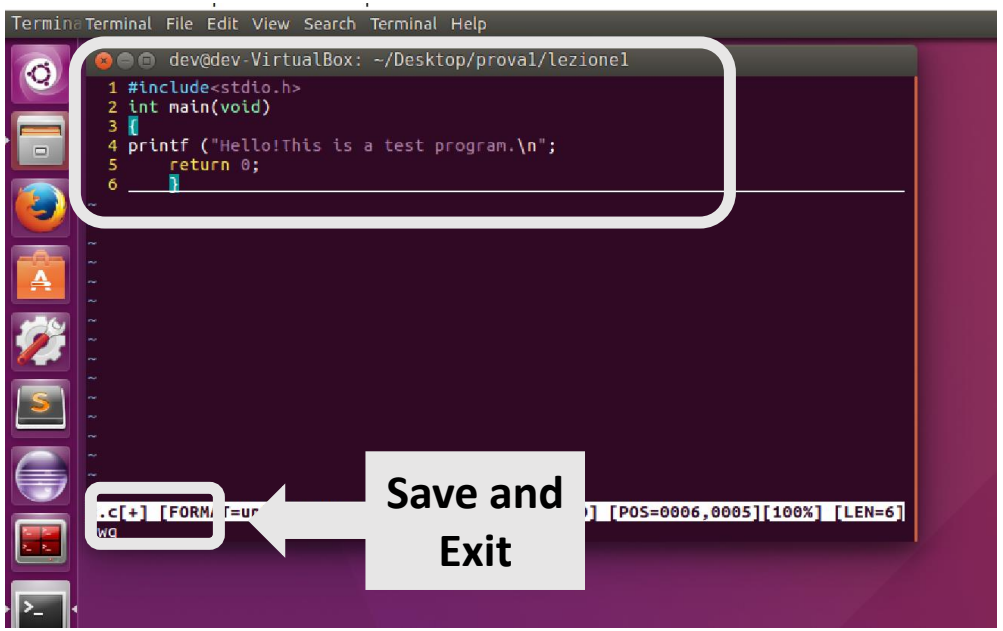
Basic Commands

- **gcc** To compile and execute a code C in Linux

Using an editor to create a file



Write a simple C program



Basic Commands

- **gcc** To compile and execute a code C in Linux

Generate an executable file using:

gcc program-name.c -o program-name

Launch the executable file using:

./program-name