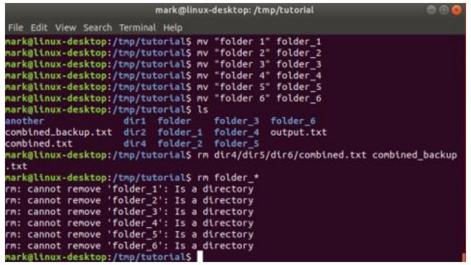
LINUX INTRODUCTION

TLDR:

- Linux is an OS LINUX
- We'll use a distribution (version) of Linux called Ubuntu ubuntu
- Linux can be used **with** a UI

- We'll learn to use it **without** a UI, only through a terminal

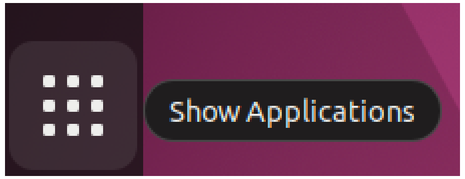

mark@linux-desktop: /tmp/tutorial
File Edit View Search Terminal Help

```
mark@linux-desktop:/tmp/tutorial$ mv "folder 1" folder_1
mark@linux-desktop:/tmp/tutorial$ mv "folder 2" folder_2
mark@linux-desktop:/tmp/tutorial$ mv "folder 3" folder_3
mark@linux-desktop:/tmp/tutorial$ mv "folder 4" folder_4
mark@linux-desktop:/tmp/tutorial$ mv "folder 5" folder_5
mark@linux-desktop:/tmp/tutorial$ mv "folder 6" folder_6
mark@linux-desktop:/tmp/tutorial$ ls
another             dir1  folder    folder_3  folder_6
combined_backup.txt  dir2  folder_1  folder_4  output.txt
combined.txt        dir4  folder_2  folder_5
mark@linux-desktop:/tmp/tutorial$ rm dir4/dir5/dir6/combined.txt combined_backup
.txt
mark@linux-desktop:/tmp/tutorial$ rm folder_*
rm: cannot remove 'folder_1': Is a directory
rm: cannot remove 'folder_2': Is a directory
rm: cannot remove 'folder_3': Is a directory
rm: cannot remove 'folder_4': Is a directory
rm: cannot remove 'folder_5': Is a directory
rm: cannot remove 'folder_6': Is a directory
mark@linux-desktop:/tmp/tutorial$
```

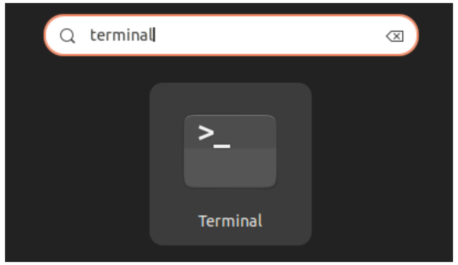
OPEN TERMINAL

OPTION 1: Open it through **Applications**

a. In the bottom left, click on ‘Show Applications’



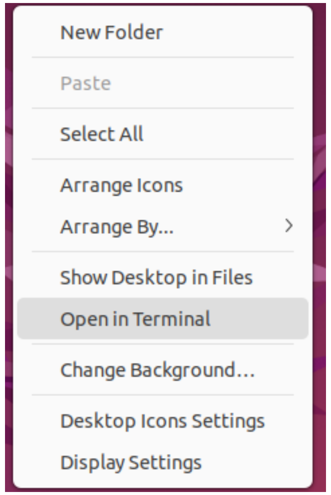
b. Search ‘terminal’ in the search-bar and open it



OPTION 2: Open it with a **hotkey**

CTRL + ALT + T

OPTION 3: Open it by **right-clicking** a location



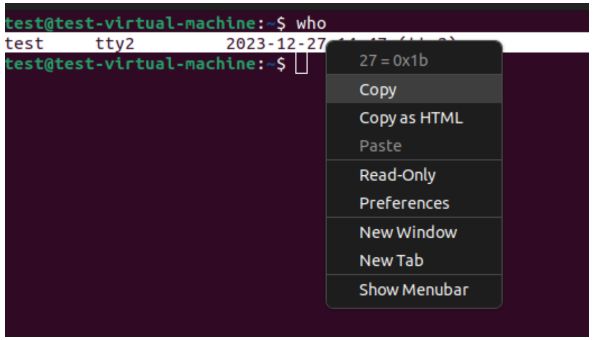
***NOTE: this opens that terminal and starts you from where you opened it**

Ex: Open from Desktop → you’ll start in the Desktop/ directory, not in the Home directory like the other options

COPY & PASTING IN TERMINAL

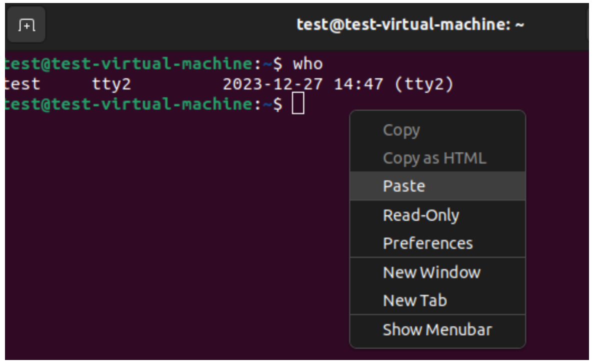
With MOUSE:

COPY: select, right-click and Copy



```
test@test-virtual-machine:~$ who
test      tty2         2023-12-27
test@test-virtual-machine:~$ 
```

PASTE: right-click and Paste



```
test@test-virtual-machine:~$ who
test      tty2         2023-12-27 14:47 (tty2)
test@test-virtual-machine:~$ 
```

Without MOUSE:

COPY: select and then…

CTRL + SHIFT + C

PASTE:

CTRL + SHIFT + V

GIT & GITHUB

TLDR:

- **GIT** is a system that allows **multiple users** to make changes to a **common folder**

James

Cooking Pot

Stephan

James is adding the eggs and Stephan is adding the flour, both to the same pot

James → Developer #1

eggs → Function #1

Stephan → Developer #2

flour, → Function #2

pot → Project (folder/repository)

- **Github** is a "hub" for GIT

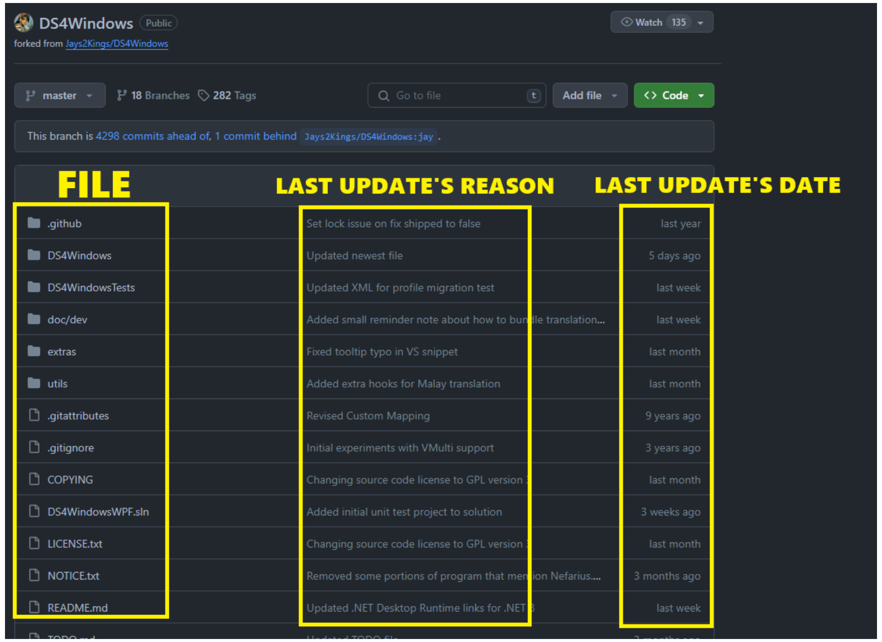
  It's where the projects are **stored**, it's the **kitchen** where the *cooking pot* is

***NOTE: There are other "kitchens" (ex: Bitbucket and GitLab), but GitHub is the most popular**

GITHUB CONCRETE EXAMPLE

1. Let’s look at the DS4 Windows project: https://github.com/Ryochan7/DS4Windows

There are multiple folders/files that make up the project:



DS4Windows Public

forked from Jays2Kings/DS4Windows

Watch 135

master | 18 Branches | 282 Tags | Go to file | Add file | Code

This branch is 4298 commits ahead of, 1 commit behind `Jays2Kings/DS4Windows:jay` .

FILE	LAST UPDATE'S REASON	LAST UPDATE'S DATE
.github	Set lock issue on fix shipped to false	last year
DS4Windows	Updated newest file	5 days ago
DS4WindowsTests	Updated XML for profile migration test	last week
doc/dev	Added small reminder note about how to bundle translation...	last week
extras	Fixed tooltip typo in VS snippet	last month
utils	Added extra hooks for Malay translation	last month
.gitattributes	Revised Custom Mapping	9 years ago
.gitignore	Initial experiments with VMulti support	3 years ago
COPYING	Changing source code license to GPL version 3	last month
DS4WindowsWPF.sln	Added initial unit test project to solution	3 weeks ago
LICENSE.txt	Changing source code license to GPL version 3	last month
NOTICE.txt	Removed some portions of program that mention Nefarius....	3 months ago
README.md	Updated .NET Desktop Runtime links for .NET 8	last week
TODO.md	Updated TODO file	3 months ago

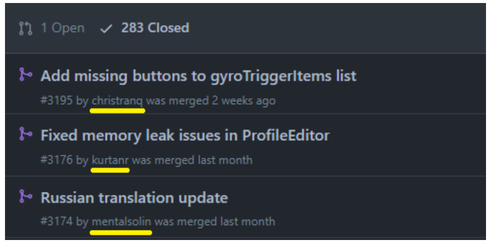
2. Who makes all those changes?

One, two, dozens, even hundreds of users. Even you can contribute.

3. How?

1. **Download** the project
2. Make your **changes locally**
3. Make a **request** to have those changes added to the real project
4. Get your changes **approved**
5. Your changes are **merged** into the real project

If we look at this page, we see the **approved** and **merged** changes people made. These changes have been added to the real project. As you can see in yellow, they’ve been made by different users.



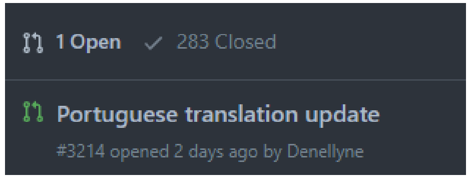
1 Open ✓ 283 Closed

Add missing buttons to gyroTriggerItems list
#3195 by christranq was merged 2 weeks ago

Fixed memory leak issues in ProfileEditor
#3176 by kurtanr was merged last month

Russian translation update
#3174 by mentalsolin was merged last month

If we look at this page, we see the ones **pending**. Someone decided to make a Portugese translation and he’s waiting for it to get reviewed and approved.



1 Open ✓ 283 Closed

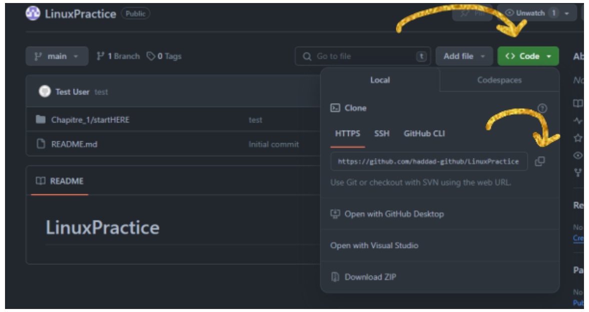
Portuguese translation update
#3214 opened 2 days ago by Denellyne

USING GITHUB

1. Create a Github account (https://github.com/signup)

2. Go to this lesson’s folder (repository) link: https://github.com/haddad-github/LinuxPractice

3. Click on “<> Code” and then copy the HTTPS link shown, to the clipboard:



https://github.com/haddad-github/LinuxPractice

Use Git or checkout with SVN using the web URL.

4. Make sure GIT is installed on your Linux machine by entering the command in your terminal:

```
sudo apt-get install git
```

5. Download the lesson’s folder (called “cloning”) by entering this command in your terminal:

```
git clone https://github.com/haddad-github/LinuxPractice.git
```

COMMAND OPTIONS

You want to know **what** a command does or what are its **possibilities**?

OPTION 1: --help

```
[COMMAND] --help
```

Example, you want to know what the command ’’**ls**’’ does:

```
ls --help
```

OPTION 2: man (*slightly more detailed, doesn’t print out in the terminal, navigate using arrow keys and press Q to quit)

```
man [COMMAND]
```

Example, you want to know what the command ’’**ls**’’ does:

```
man ls
```

THINGS TO KNOW

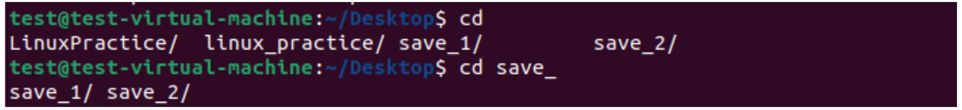
General things to know that will be used down the line

1. Use TAB to **suggest/auto-complete** file or directory names

When using a Linux command and you're about to enter a file name or file directory, you can press TAB to display what is available.

If there are multiple matching names (ex: save_1/, save_2/), you can type 'save' and then press TAB to get matching options.

Ex: In both instances, I pressed TAB on my keyboard, at first, I got all possibilities, then I narrowed them down by adding ''save''

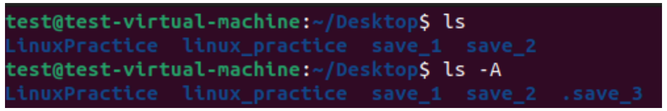


```
test@test-virtual-machine:~/Desktop$ cd
LinuxPractice/  linux_practice/ save_1/         save_2/
test@test-virtual-machine:~/Desktop$ cd save_
save_1/ save_2/
```

2. **Hidden** files can start with a period

Files can be hidden in certain contexts, with the use a period at the start of the file or directory name.

Ex: I listed all files using a command, and then I added an option to the command to list all files including those that are hidden

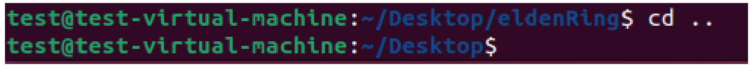


```
test@test-virtual-machine:~/Desktop$ ls
LinuxPractice  linux_practice  save_1  save_2
test@test-virtual-machine:~/Desktop$ ls -A
LinuxPractice  linux_practice  save_1  save_2  .save_3
```

3. **Going backwards** by one directory is a folder named ''..''

Technically, there's always a super-hidden ''..'' directory and that's the one you use to go backwards.

Ex: Here I'm in the ~/Desktop/eldenRing directory and I want to go backwards by one directory which puts it in ~/Desktop

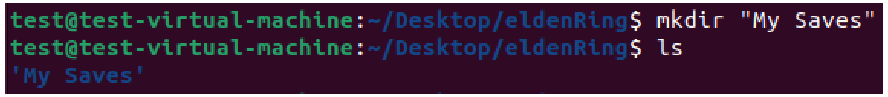


```
test@test-virtual-machine:~/Desktop/eldenRing$ cd ..
test@test-virtual-machine:~/Desktop$
```

4. Use quotation marks when referring to a file or directory name with **spaces**

Names with spaces can be tricky, it's best to use quotation marks when referring to them.

Ex: I want to create a 'My Saves' directory, so instead of referring to My Saves (which would actually create 2 folders: 'My' and 'Saves'), I use ''My Saves''

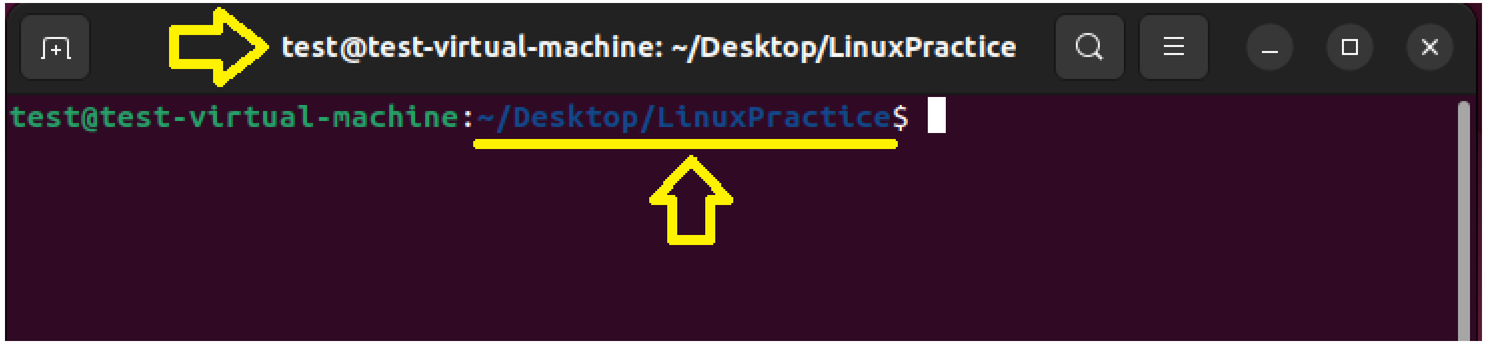


```
test@test-virtual-machine:~/Desktop/eldenRing$ mkdir "My Saves"
test@test-virtual-machine:~/Desktop/eldenRing$ ls
'My Saves'
```

WHERE AM I?

You want to know **where** you currently are?

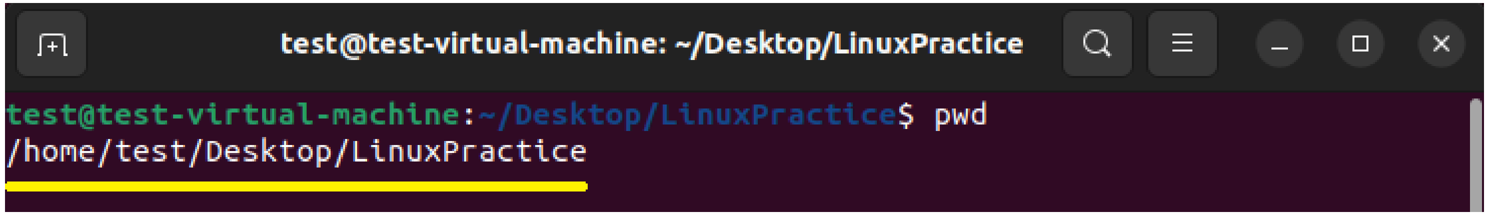
OPTION 1: Visually in the terminal



```
test@test-virtual-machine:~/Desktop/LinuxPractice$ 
```

OPTION 2: **pwd**

```
pwd
```



```
test@test-virtual-machine:~/Desktop/LinuxPractice$ pwd
/home/test/Desktop/LinuxPractice
```

WHAT’S IN A DIRECTORY?

You want to know **what** files and folders there are inside a directory?

COMMAND: ls

STANDS FOR : list, as in list the files

Ex 1: lists what’s inside the current directory

```
ls
```

Ex 2: lists what’s inside another directory

```
ls pictures/wedding
```

Ex 3: lists what’s inside more than one directory

```
ls pictures/wedding pictures/parc
```

HOW DO I NAVIGATE?

You want to **navigate** in the system, from folder to folder?

COMMAND: cd

STANDS FOR : change directory

Ex 1: move to a directory that is accessible from your current one

```
cd pictures
```

Ex 2: move to a directory that is within another accessible directory

```
cd pictures/weddings/2023
```

Ex 3: move to your home directory

```
cd ~
```

HOW DO I CREATE OR DELETE FOLDERS?

You want to **create** or **delete** a directory?

COMMANDS: mkdir / rmdir

STANDS FOR : make directory / remove directory

Ex 1: creating a directory

```
mkdir screenshots
```

Ex 2: creating a directory within a deeper

```
mkdir pictures/weddings/2023/JohnAndElena
```

Ex 3: removing a directory

```
rmdir screenshots
```

HOW DO I CREATE OR DELETE FILES?

You want to **create** or **delete** a file?

COMMANDS: touch / rm

STANDS FOR : touch (as in touching) / remove

Ex 1: creating a file

```
touch groceries.txt
```

Ex 2: removing a file

```
rm groceries.txt
```

Ex 3: removing 2 files

```
rm groceries.txt eldenring_script.py
```

HOW DO I COPY OR MOVE FILES?

You want to **copy-paste** a file somewhere or **move** it?

COMMANDS: cp / mv

STANDS FOR : copy / mv

Ex 1: copy-pasting a file from a deeper directory into the current one

```
cp eldenRing/steam.ini .
```

Ex 2: moving a file from the current directory to a deeper one

```
mv steam.ini eldenRing
```

Ex 3: moving a file from the current directory to one folder back

```
mv steam.ini ..
```

HOW DO I ZIP OR UNZIP?

You want to **ZIP** files or **UNZIP** a file?

COMMANDS: zip / unzip

STANDS FOR : .zip format

Ex 1: zip files into a .zip

```
zip backup.zip data.bin settings.config
```

Ex 2: unzip a .zip file in the current directory

```
unzip backup.zip
```

Ex 3: list the files inside a zip file (without unzipping)

```
unzip -l backup.zip
```

HOW DO I LOOK FOR SOMETHING?

You want to **look for** a file, a folder, or even of text inside files?

COMMANDS: find / grep

STANDS FOR : find / global regular expression print

Ex 1: find a file, from the current directory

```
find -name travel.pdf
```

Ex 2: find any directory that ends with part01, starting from the current directory

```
find -type d –name ‘*part01’
```

Ex 3: find the file(s) where ‘legal’ is mentionned

```
grep -R legal
```

HOW DO I SEE THE PROCESSES THAT ARE RUNNING?

Do you want to see which **processes** are running?

COMMAND: top

STANDS FOR : table of processes

Ex: list all processes

```
top
```

HOW DO I CHECK HOW MUCH RAM IS BEING USED?

You want to see how much **RAM** is being used in the system?

COMMAND: free

STANDS FOR : free memory

Ex: how much memory used

```
free
```

HOW DO I CHECK HOW LONG THE MACHINE HAS BEEN UP?

You want to see **how long** you’ve been running?

COMMAND: uptime

STANDS FOR : uptime

Ex: uptime

```
uptime
```

HOW DO I TEST IF I CAN CONNECT TO A SEVER?

You want to know if you can **reach** or **connect** to a server or a website?

COMMAND: ping

STANDS FOR : packet internet groper

Ex: check if Facebook is up (or at least for you)

```
ping facebook.com
```

HOW DO I MANAGE FILE/FOLDER PERMISSIONS?

You want to manage **who** can do something to a file or folder?

COMMAND: chmod

STANDS FOR : change mode

Ex: make a file unaccessible to all users

```
chmod a-r backup.zip
```

HOW DO I DOWNLOAD PACKAGES OR FILES FROM ONLINE?

You want to **download** a package (feature) or a software?

COMMANDS: apt-get / wget

STANDS FOR : advanced-package-tool get / world-wide-web get

Ex 1: installing a package called *curl*

```
apt-get install curl
```

Ex 2: downloading a file from the web

```
wget https://cdn.jagex.com/Jagex%20Launcher%20Installer.exe
```

HOW TO INSTALL A DEBIAN PACKAGE (.DEB)?

You **downloaded** a **.deb** and you want to **install** it? (UBUNTU SPECIFIC)

COMMAND: dpkg

STANDS FOR : Debian package manager

Ex: installing the debian package

```
dpkg -i osrs_for_linux.deb
```