Installation tutorial of OpenMC via Windows using Linux subsystem

Presentation · April 2023

DOI: 10.13140/RG.2.2.17608.60161/2

2 authors:

Willian Vieira de Abreu
Federal University of Rio de Janeiro
54 PUBLICATIONS **100** CITATIONS
SEE PROFILE

Artur Silva
Federal University of Rio de Janeiro
4 PUBLICATIONS **0** CITATIONS
SEE PROFILE

INSTALLATION TUTORIAL FOR OPENMC VIA WINDOWS USING LINUX SUBSYSTEM

Willian Vieira de Abreu

Artur Souza e Silva

Rio de Janeiro

April 2023

Introduction

Figure 1. OpenMC logo.

OpenMC is a community-developed code for Monte Carlo neutron and photon transport, capable of performing various types of calculations on models built with different geometries. It has a flexible tally system and can run in parallel on supercomputers. The code has rich and extensible programming interfaces in Python and C/C++, allowing for pre-and post-processing, data processing, and visualization. A supporting infrastructure ensures the quality and accuracy of the code over time, with continuous integration testing and benchmark simulations. You can use it to simulate Fission AND Fusion processes. For free. Big deal.

That being said, in order to run this code, one usually needs to use a Linux-based system. However, from Windows 10 onwards, it is possible to use this magnificent code directly from the Microsoft OS without needing a platform as a service (PaaS. e.g., Docker or similar software) which, in turn, usually isn't so friendly for beginners. Therefore, this document pretends to explain how to do that as quickly as possible, following 23 steps (or less). We also use a lot of images to facilitate comprehension.

Linux Subsystem Activation and Ubuntu Installation

1. First, you need to check if the Virtualization of your PC is enabled in the Bios. Press "Ctrl + Alt + Del" to open the Task Manager. Go to the "Performance" tab. At the bottom of the screen, you'll find if it is enabled or disabled:

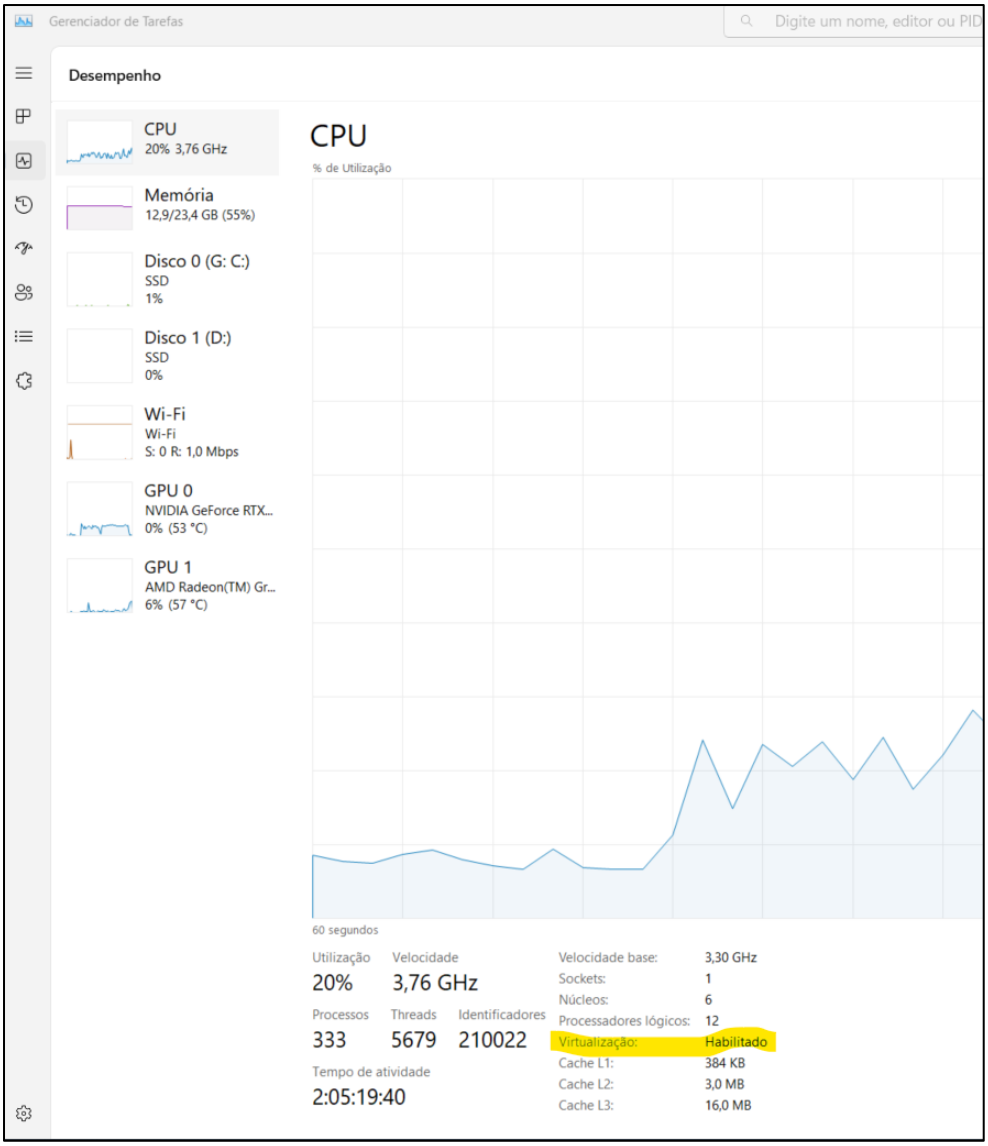

Figure 2. Task manager (Performance Tab).

2. If it shows "Disabled," you must activate it via Bios. Reboot your computer. Right when the computer is coming up from the black screen, press F10, Delete, Esc, F1, F2, or F4. Each computer manufacturer uses a different key. If you miss it the first time, reboot and try again. It helps tap the key twice a second when the computer is coming up. Consult your computer's manual if you cannot enter the BIOS via this method.
3. Inside the Bios, go to the security settings.

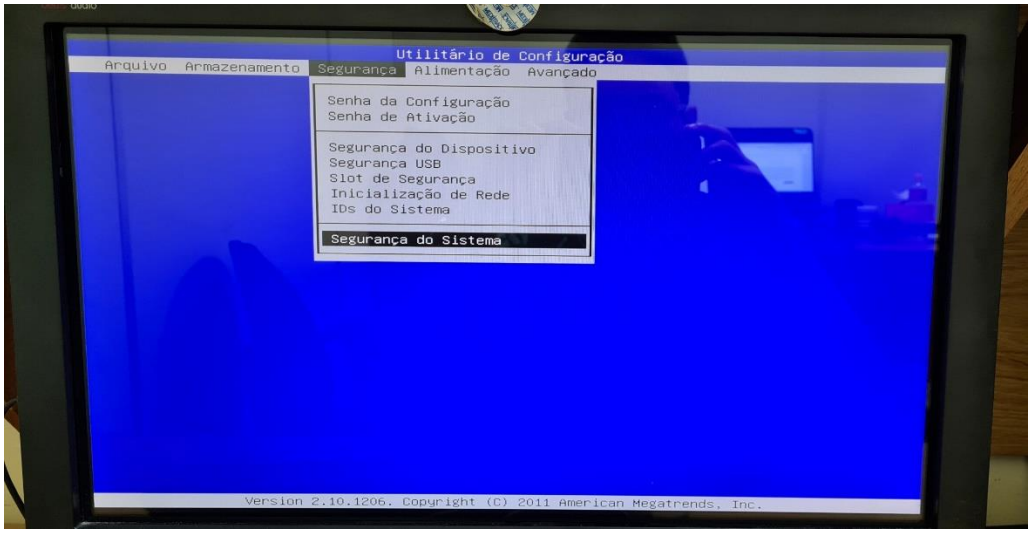

Figure 3. Bios.

4. Enable the Virtualization (Vtx/Vtd). Save and exit Bios. Reboot your PC twice and rerun the Task Manager to see if the Virtualization is enabled. These steps can change from PC to PC. However, it is not a difficult task to find where it will be.

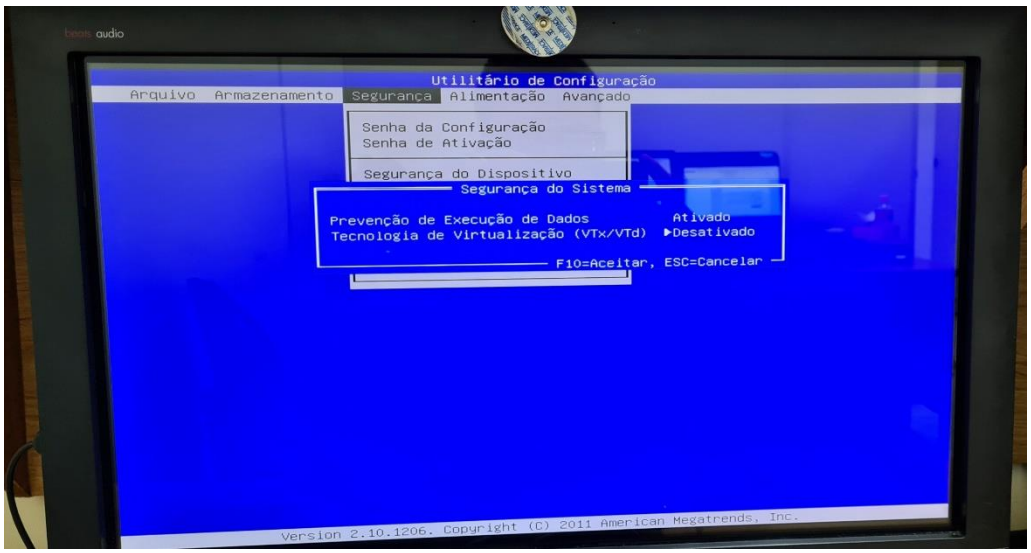

Figure 4. Virtualization is disabled.

Figure 5. Virtualization enabled.

5. If you use Windows 10 version 2004 and higher (Build 19041 and higher) or Windows 11, the wsl and Ubuntu installation is effortless. If this is not your case, advance to step 7. Open PowerShell or Windows Command Prompt in administrator mode by right-clicking and selecting "Run as administrator" enter the wsl --install command, then restart your machine;

```
wsl --install
```

6. By default, the installed Linux distribution will be Ubuntu. You can opt for another one using the -d flag. The list of options is available if one uses the following command line;

```
wsl --list -online
```

Figure 6. Linux distribution: list of options.

7. If steps 5 and 6 were enough for your case, proceed to step 12. If not (if you use an older version than Windows 10 2004), Right-click on the Windows icon. Launch up "Apps and Features";

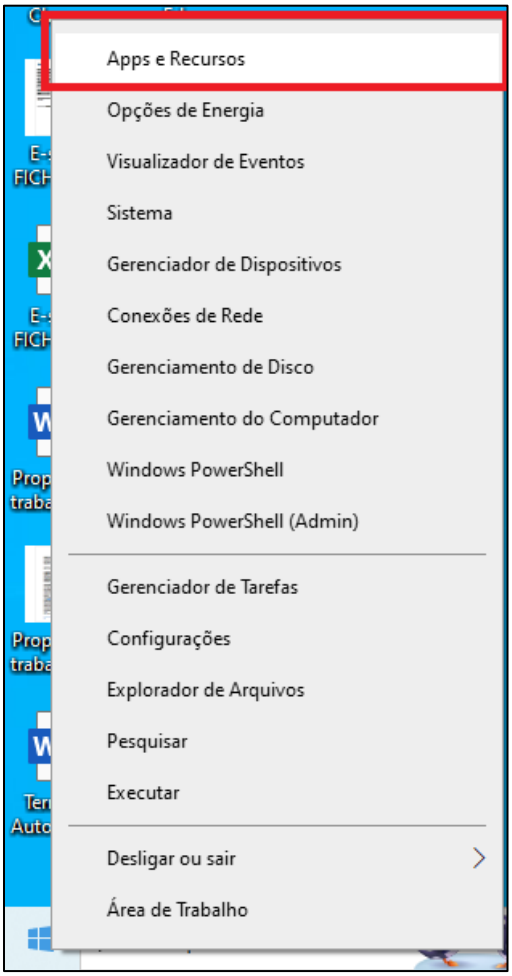

Figure 7. Apps and Features (Portuguese).

8. Click "Program and Features" on Related settings;

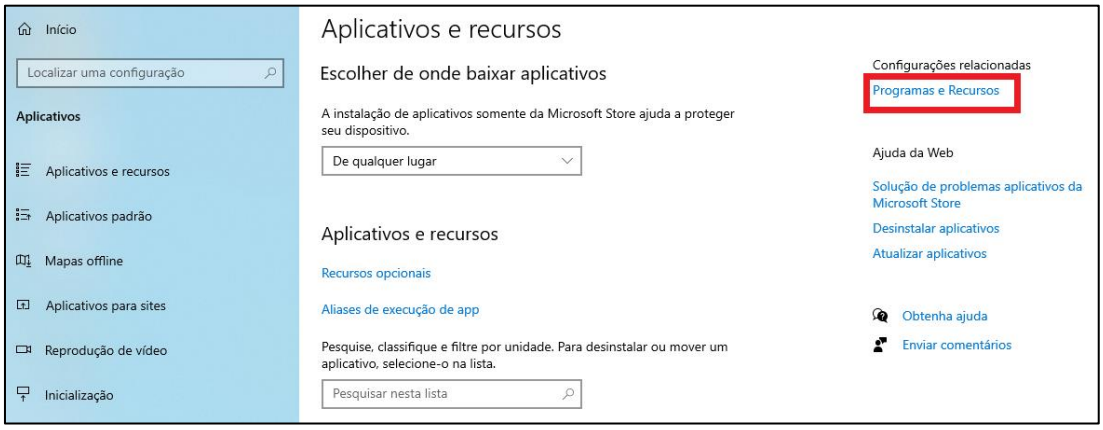

Figure 8. Program and Features (Portuguese).

9. Click "Turn Windows features on or off";

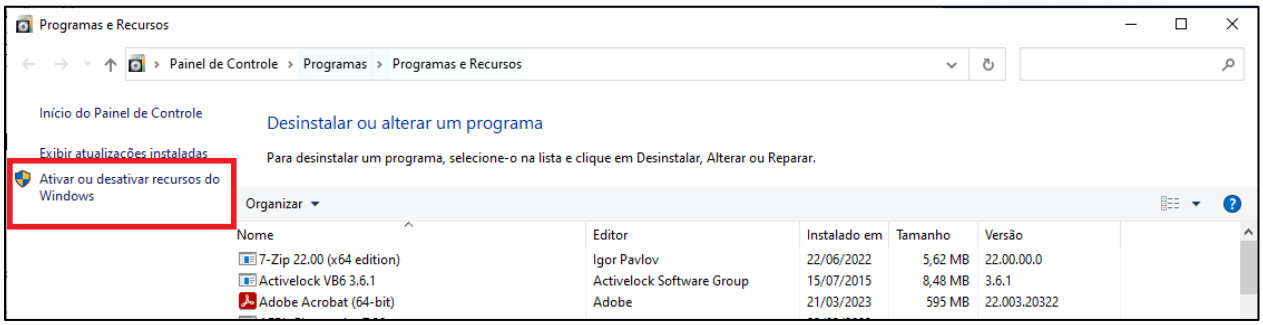

Figure 9. Turn Windows features on or off (Portuguese).

10. Certify that the option "Windows Subsystem for Linux" is checked. Press "OK" and wait some minutes. After this, restart your Windows OS;

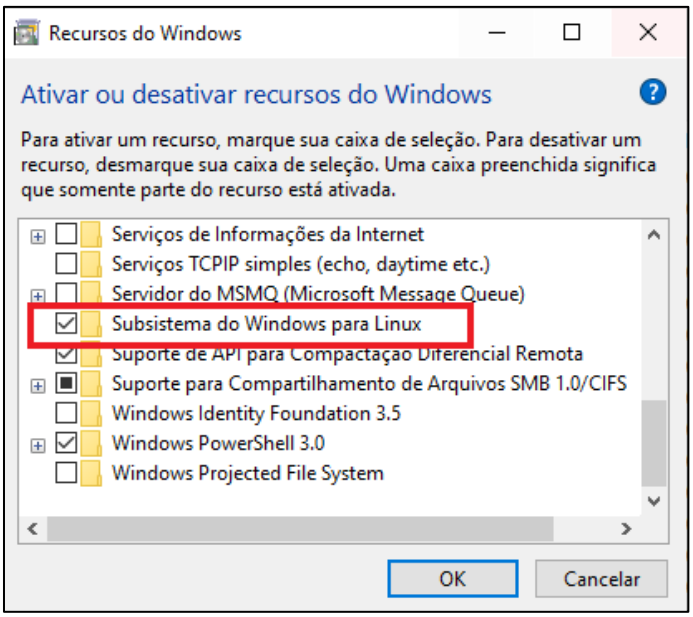

Figure 10. Windows Subsystem for Linux (Portuguese).

11. Go to Microsoft Store and search for "Ubuntu". Click "GET" to install it;

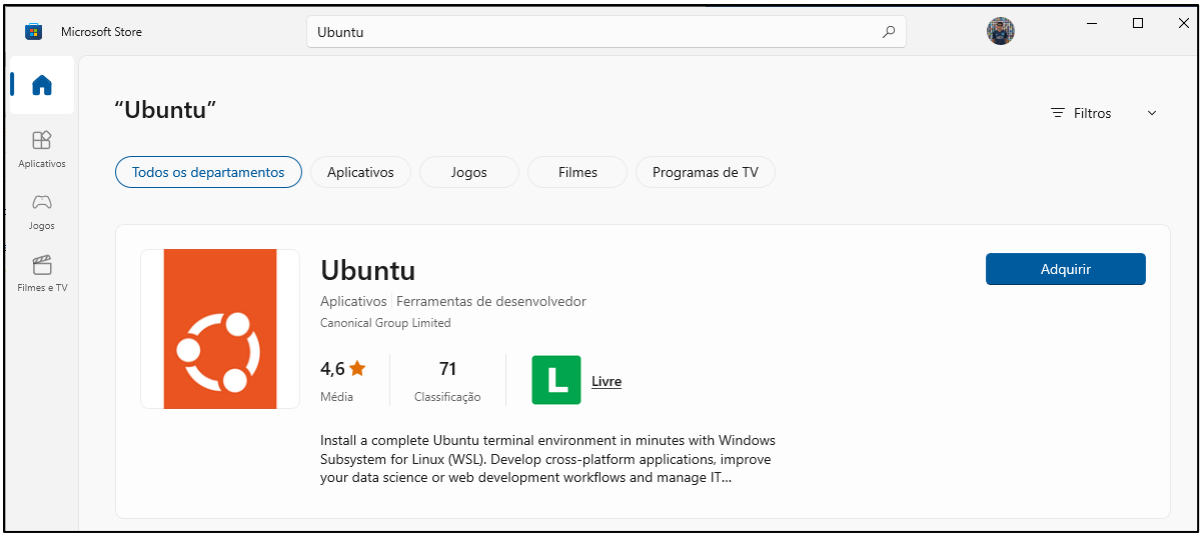

Figure 11. Ubuntu in the Microsoft Store (Portuguese).

OpenMC Installation

12. Access the CONDA webpage;

13. Download Miniconda for LINUX (remember, you'll install it on the Linux Subsystem);

14. Choose the latest version (e.g., Miniconda3 Linux 64-bit - April/2023);

15. Navigate to the folder where the downloaded file is located via the "File Explorer". Usually, it goes to the "Downloads" folder, but you can move the file to any location you want;

16. Type "wsl" in the navigation bar;

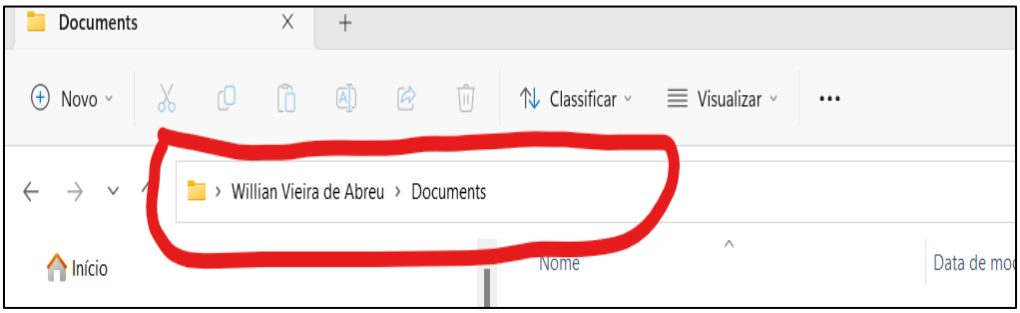

Figure 12. Navigation bar.

17. A Linux terminal will open. From this point on, the OpenMC installation process is the same as if you were using a Linux OS;

18. From the open Linux terminal, the first necessary step is to install Miniconda, which was previously downloaded. For this, you need to copy the line that is present on the conda website, which tends to be something like this:

```
bash filename
```

At the time this humble tutorial was created (April 2023), the command would be as follows:

```
bash Miniconda3-latest-Linux-x86_64.sh
```

19. Once Miniconda is installed, we suggest the user ensure they will not work in the "base" environment for security (and organizational) reasons. It is a "best practice" to avoid installing additional packages into your base software environment. When you are in the base, the term "(base)" appears at the beginning of the line. To make our suggestion, run the following commands (via wsl):

```
conda deactivate base
```

```
conda config --set auto_activate_base False
```

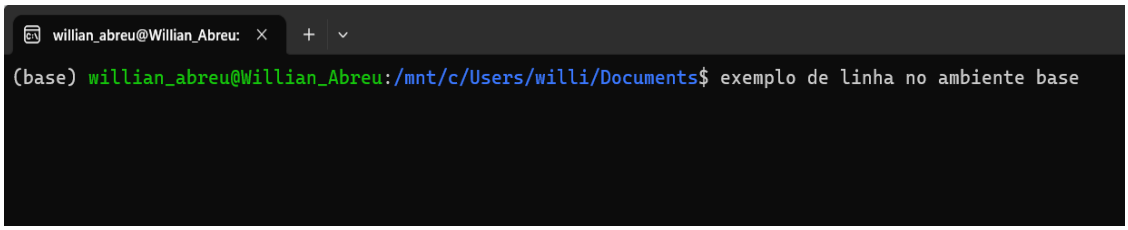

Figure 13. Example of "base" environment.

20. The next step is to make conda see the repository where OpenMC is hosted (condaforge). To do this, enter the following command line:

```
conda config --add channels conda-forge
```

21. The next step is to create an environment inside conda and install the packages that will be used: OpenMC, NodeJS (optional), and Jupyter Lab (optional). The code line for this is:

```
conda create --name createyourname
python=3.10 openmc nodejs jupyterlab
```

Note that the latest version of Python may be more prone to errors. If the latest version of Python has just been released, install the previous version. This error will not always occur.

22. Finally, OpenMC itself is already installed. To work on it via Jupyter Lab, type the following command line:

```
conda activate createyourname
```

```
jupyter lab
```

23. After the Jupyter lab command, two links will be available. You can access them by copying and pasting them into the browser or simply by pressing "ctrl" plus clicking with the left mouse button.

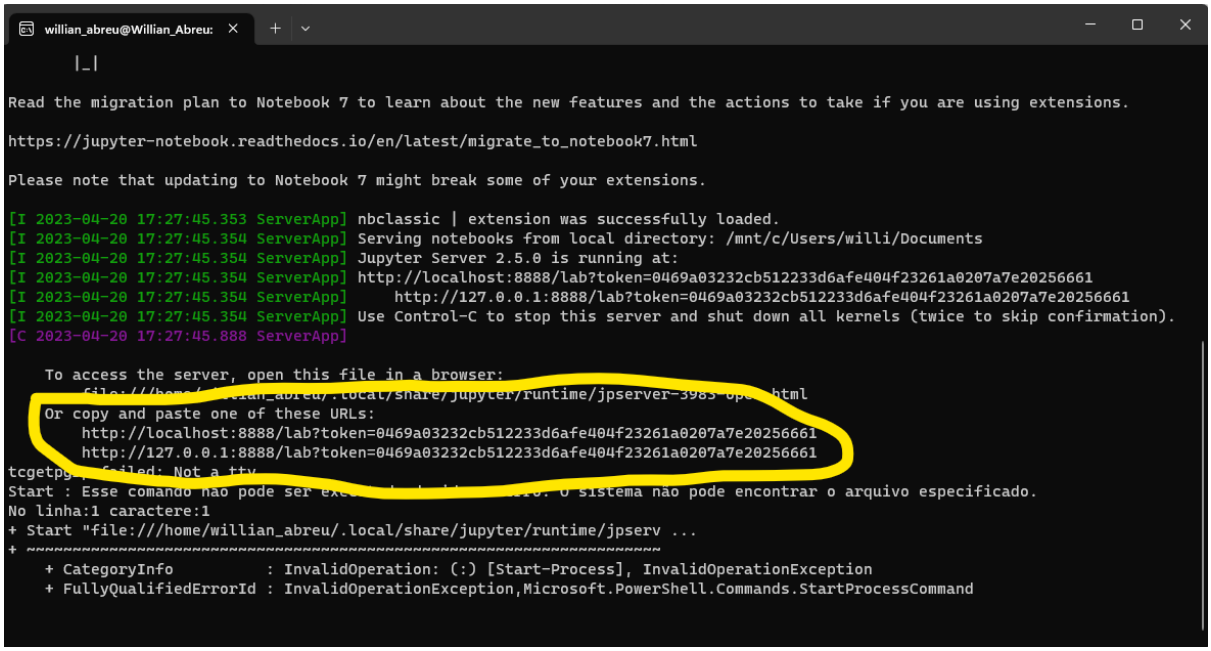

Figure 14. Links are available after running the Jupyter command.