

Guide for MulticoreTemplate Notebook

[Axel Rubini]

November 27, 2025

Contents

1	Description	2
2	Prerequisites	2
3	Initial Setup	2
3.1	Opening the Notebook	2
3.2	Runtime Configuration	2
3.3	Connecting to Runtime	2
4	Notebook Structure	2
4.1	Section 1: [Initial Setup and Future Exercises]	2
4.2	Section 2: [Loading Your Own Code]	3
4.3	Section 3: [Executing Code and Viewing Results]	3
4.4	Section 4: [Saving Results and Uploading to GitHub]	3

1 Description

This guide provides detailed instructions for using the `MulticoreTemplate` notebook in Google Colab. The notebooks are already present in the repository where this guide is located.

2 Prerequisites

- Google Account (to access Google Colab)
- GitHub (optional)
- Access to Google Drive

3 Initial Setup

3.1 Opening the Notebook

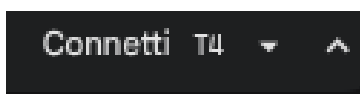
1. Go to <https://colab.research.google.com/>
2. Upload the notebook from your computer:
 - Click on `File > Upload notebook`
 - Select the `Upload` tab and choose the `MulticoreTemplate.ipynb` file from your computer

3.2 Runtime Configuration

1. Go to `Runtime > Change runtime type`
2. Select the following settings:
 - **Runtime type:** Python 3
 - **Hardware accelerator:** [GPU - T4]

3.3 Connecting to Runtime

- Click on `Connect` in the top right
- Wait until the runtime is ready (green icon with checkmark)



4 Notebook Structure

4.1 Section 1: [Initial Setup and Future Exercises]

Purpose: [This section of the notebook contains the scripts that create the repositories for the exercises. A main directory will be created that will contain the subdirectories for the current exercise and future exercises.]

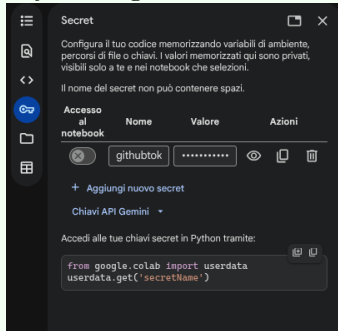
Cells to execute:

- [1] - [The first cell is used to create the main directory if it doesn't already exist, and to create the git repository for the current exercise]
- [2] - [This cell contains a script that allows you to view the GPU characteristics.]

- [3 (optional)] - [This cell configures git and creates a repository on your GitHub account. Access to GitHub must be done via a personal token, which needs to be entered each time the cell is executed for the first time in the runtime and remains valid until the runtime is restarted.]

Note

If you want to use GitHub, I recommend creating a token and saving it in the appropriate key manager in Colab



4.2 Section 2: [Loading Your Own Code]

Purpose: [Load your own .cu file into the directory of the current exercise.]

Cells to execute:

- [4] - [This cell contains a script that allows you to upload your own .cu file into the current exercise directory, or use files already existing in the project.]

Note

If you want to upload a file that has the same name as an existing one, it is recommended to delete the old file manually through the work terminal or with the side tree present in CoLab.

4.3 Section 3: [Executing Code and Viewing Results]

Purpose: [The following cells are used to compile, clean, execute and view the results of your own code within the notebook.]

Cells to execute:

- [5] - [Compiles the .cu source code. The pre-configured parameters are those for running on GPU T4. If you want to change GPU type or other parameters, you can modify the cell.]
- [6] - [Performs cleaning of the working directory, removing output files generated by previous executions.]
- [7] - [Executes the compiled code.]

4.4 Section 4: [Saving Results and Uploading to GitHub]

Purpose: [The following cells are used to generate some documentation and synchronize the repository on GitHub. The description is omitted for brevity.]