

**Benchmarking scripts**

for multi

-

core CPUs



**By Deac Dan Cristian**

**Content**

1. **Introduction** ................................................................................ 1
2. **Bibliographic research** ...............................................................1
   1. Clock() .............................................................................1
   2. Threads ...........................................................................1
3. **Analisys** ………....…………..………………………………………..2
   1. Make the scripts in C/C++ …………………………………2
   2. User interface ……………………………………………….2
4. **Design** ……………………….….……………………………………..2

# Introduction

This project is a benchmarking application that performs several measurements on multicore processors. The project will be done on a linux machine using the available tools. The

development of this project as well as this document is achieved using open source software. The script is intended to work in any operating system and will be available on github for

download.

**>git clone https://....**

Compilation(Linux):

**>g++ script.cpp** –**o executable**

**>chmod +x executable**

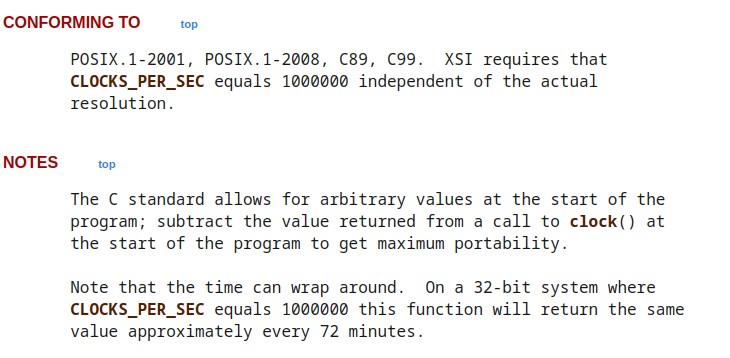
**>./executable**

# Bibliographic research

2.1 Looking in the man7 proves to be very useful understanding the libraries and the use

of functions that attempt to measure the performance of a CPU. One example is the time.h which enables the use of the clock() function.

Linux Manual page: <https://man7.org/linux/man-pages/man3/clock.3.html>



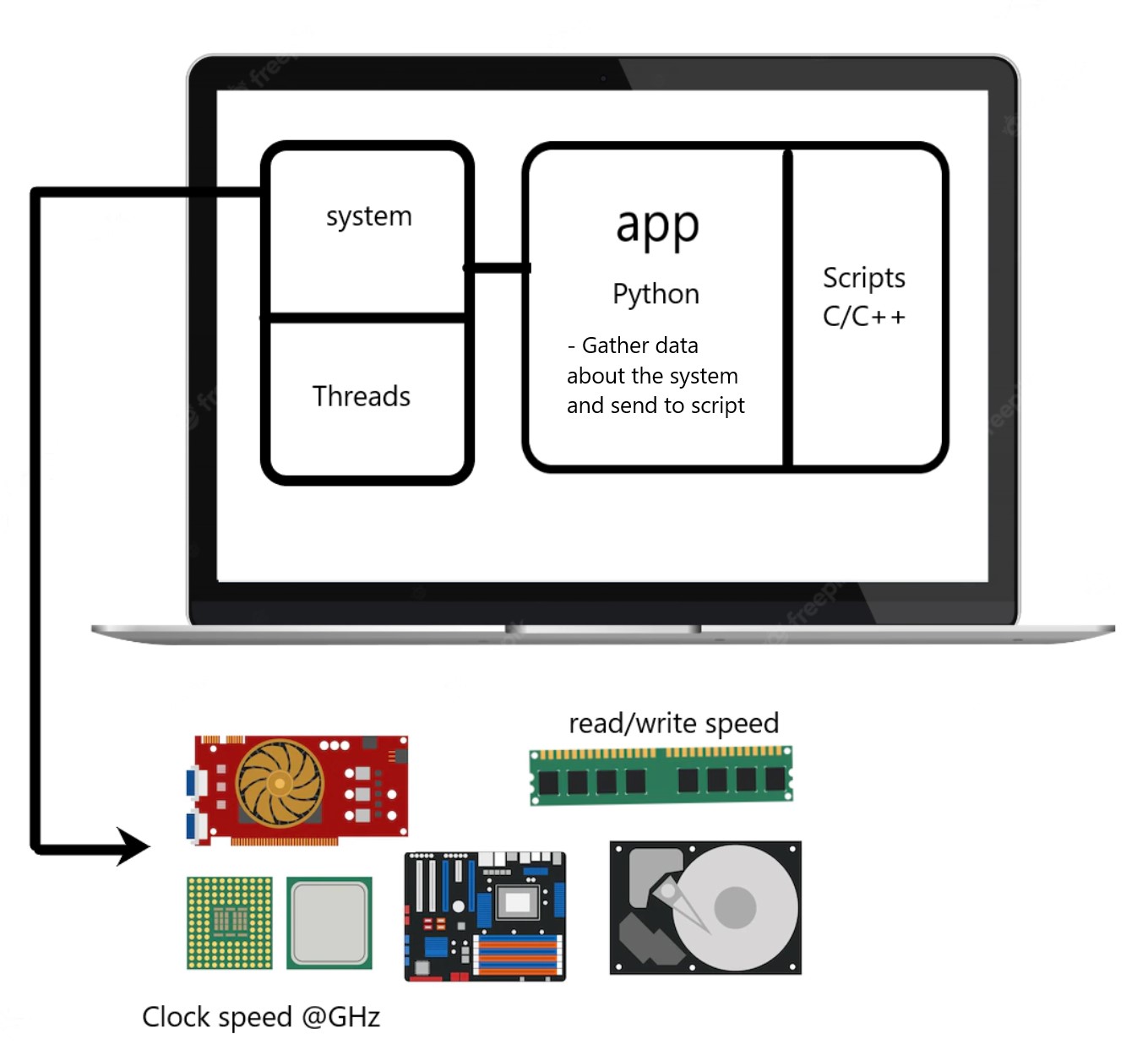
## Analysis

The first part of the project will be focused on the development of the functional part. The core should be based on several C/C++ scripts that are executed by the user interface, or their results are stored in several files. Then the files are interpreted by the user interface and shown with some kind of graphical representation.

The interface should run the script with certain arguments that are read by the front end of the application. A library will be used to gather this information and then the user interface will wait for the script to run, and then display the results. An other solution would be to generate the results in pdf files and open them automatically in the user interface after the scripts were run successfully.

The second part will be the development of the user interface. I would prefer to use C# for windows, but if this project will be developed with linux in mind I am thinking of switching on something python based with tkinter library.

## Design



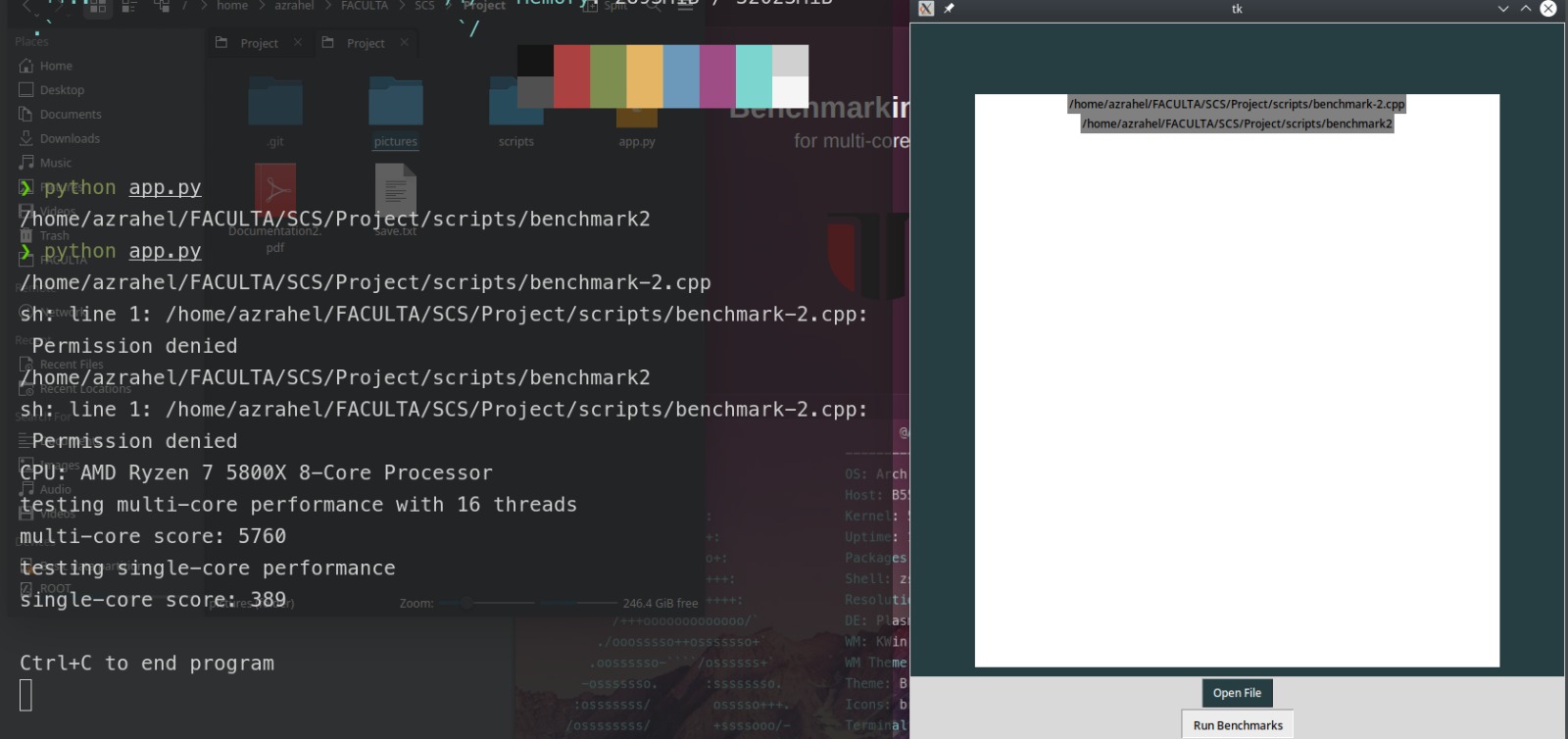
**Figure 1 – design**

Application Interface:

* Should be able to select which tests to be ran
* Start benchmark / Stop benchmark
* Open results button (which could open a generated pdf file by the script)

Script

* Get data from arguments (probably only used for displaying it in the chart)
* Run Test that was given by the argument - Put the results in a chart – pdf file



**Figure 2 - UI**

This version of the app is based on tkinter framework and is able to open a folder and select from the system the desired benchmarks that can be ran. In the future version I am thinking of switching completely to PyQt5. It offers a broader range of tool for UI design.

Functionality wise, it uses right now a default “stress\_test” that is a prewritten function which tests the performance of the single core and the multicores. Then it times them using functions from the chrono library and prints the “score” in the terminal in this state.

The implementation uses some assembly code to check for a certain place in memory to get the brand of the CPU and other useful information. That is the only things so far that runs on assembly code.