Politenico di Milano

Dipartimento Elettronica, Informazione e Bioingegneria

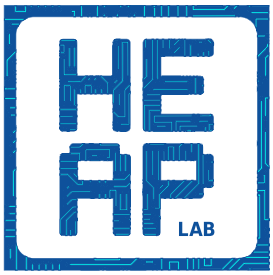
HEAPLab Project Report

Binary executable code size visualizer

Author:
Devi Radhakrishnan
Gayathri Prakash Menakath

Supervisor:
Federico Terraneo

September 3, 2021



Abstract

In the code size visualizer of a binary executable project, we aim at starting from a dump of every symbol in a Miosix executable and design a tool to show a breakdown of the code size.

1 Introduction

Our Goal is to design a tool to show a breakdown of the code size of a Miosix executable. We wrote C++ code which takes a .map file as input.

The frontend is done with PyQt5. PyQt is a library that lets you use the Qt GUI framework from Python. PyQt5 is a comprehensive set of Python bindings for Qt v5. The user can upload a .map file and the breakdown of the code size will be displayed. The application supports a chart view of libraries and non-libraries present in the executable.

2 Design and Implementation

2.1 Frontend

The PyQt5 application helps the user to upload a .map file and display the breakdown of the code size in kB. The breakdown is based on the contents that belong to the library and non-library. The application provides a high-level chart view of the contents of the library and non-library components. On selecting a particular library, a more detailed table view including the size of each file inside that library will be shown. The breadcrumb on the navigation bar allows the user to go back and forth to select and expand different libraries/non-libraries or their contents. The donut chart view is switched to table view when the number of items in a particular section selected exceeds 6.

2.2 MAP file Parsing

The python script for parsing the MAP file summarizes the code size which accounts to the sections such as .text section and the .data of the MAP file. Several sections such as .comment, .debug are disregarded and not taken into account for calculating the size.

3 Results

The following are the various snapshots of the resulting PyQt5 application displaying the various breakdown of code size.

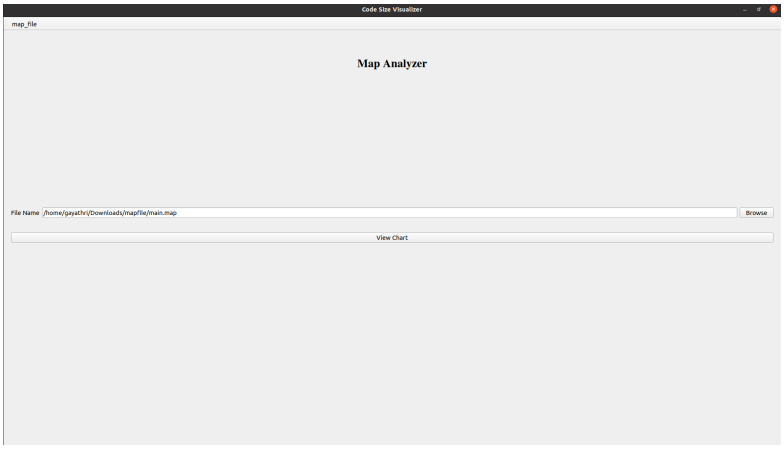


Figure 1: Home screen

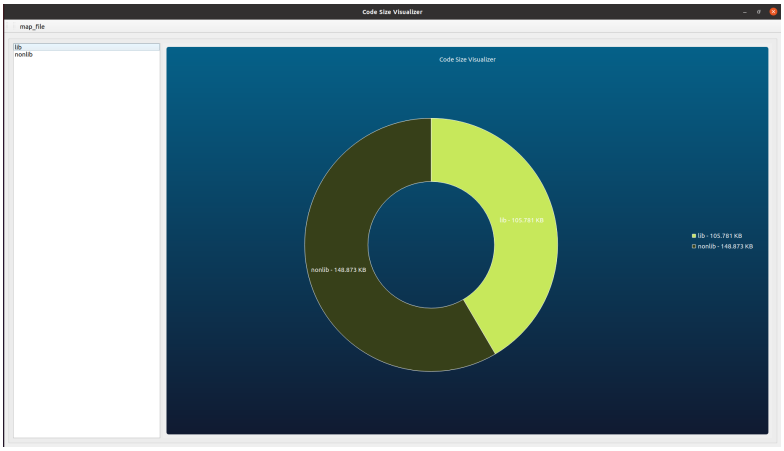


Figure 2: Donut chart view of library and non library section

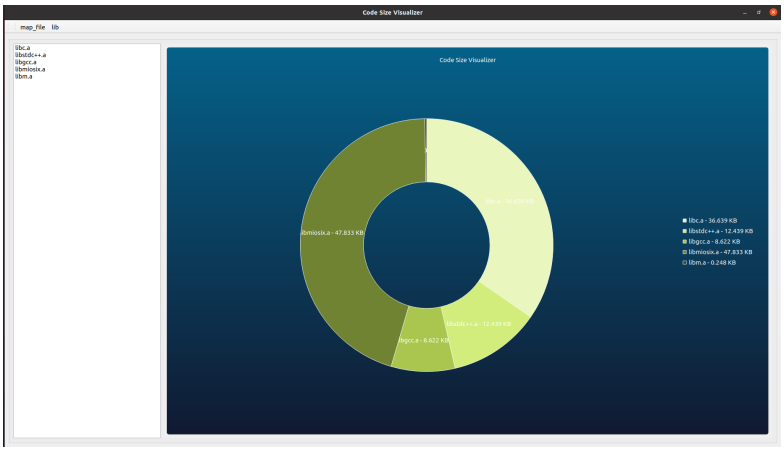


Figure 3: Donut chart view of code size breakdown of each library

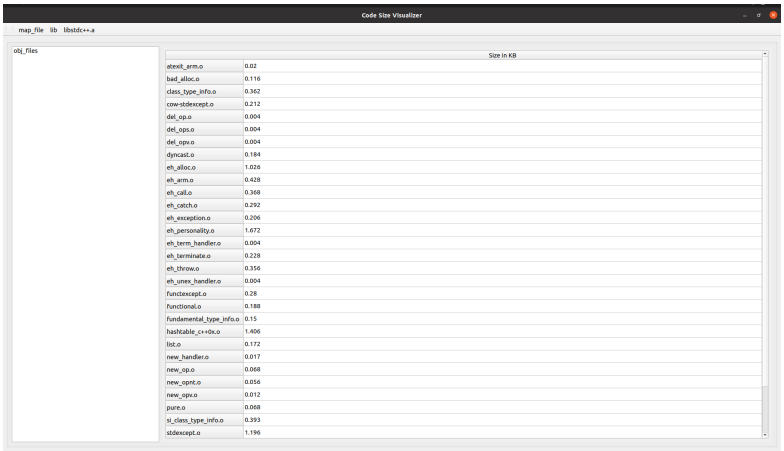


Figure 4: Table view of code size breakdown of a selected library

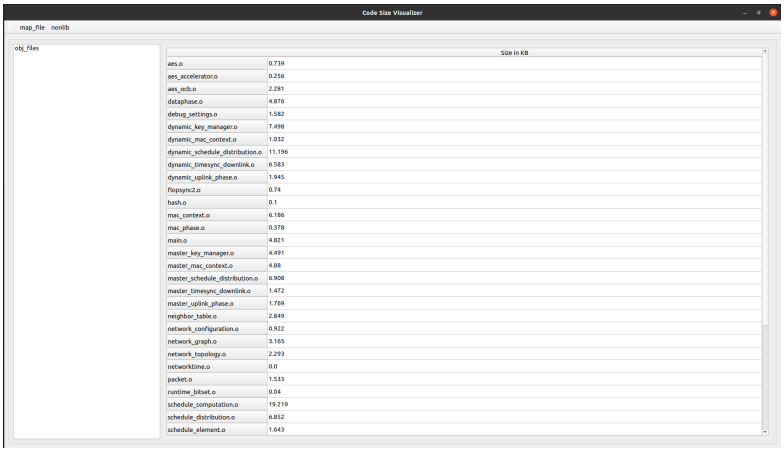


Figure 5: Table view of code size breakdown of non-library section

4 Conclusions

We developed a C++/PyQt5 application that takes as input a Miosix executable file and displays the breakdown of the code size with the help of table views and chart views.