Universidade de Aveiro

DETI

Report for the first practical project

Implementation of a Double Linked List (DLL) GCC, Make and C modularity

Course of SETR at UAveiro
March of 2025

Teacher:

Professor Paulo Pedreiras

Autors:

Pedro Ramos
p.ramos@ua.pt
n.º 107348
Masters in Robotics
and Intelligent Systems

Rafael Morgado
rafa.morgado@ua.pt
n.º 104277
Masters in Robotics
and Intelligent Systems

First Project

1. Introduction

This project consists on the implementation of a Double Linked List (DLL) in the C programming language. The objective of this project is to get more familiar with the C programming language and demonstrate the use of tools such as GCC, Make, Doxygen, Git and Github. This project also serves as a way to practice the development of C code as easily to integrate modules;

2. Project links

Github: https://github.com/P-Ramos16/SETR_Proj1

3. How to execute the project

Download and compile the project:

```
$ git clone https://github.com/P-Ramos16/SETR_Proj1
$ cd SETR_Proj1/src/build
$ cmake ..
$ make
```

Run the test and the main executable:

```
$ cd $PROJECT_ROOT/src/build
$ ./tests
$ ./main
```

Generate and examine the doxygen documentation:

```
$ cd $PROJECT_ROOT
$ doxygen Doxyfile
$ firefox docs/html/index.html
```

Assuming that the user is inside the 'PROJECT_ROOT', all compiled files are generated inside the 'src/build' folder and all documentation is generated inside the 'docs' folder.

4. Main Features

The main features of the program are:

- Define a static maximum number of elements and maximum data size for the elements;
- Insert elements with key and data;
- Find elements by key;
- Edit elements by key;
- Remove elements by key;
- Get the next element from the last one used;
- Get the previous element from the last one used;
- Print the table with varying length based on the maximum data size chosen;
- Clear the full list.

5. Aditional features

This project has some additional features such as:

- Ability to edit items without affecting the data order;
- Ability to recursively clear the list;
- Implementation of a test executable that guarantees the correct functionality of most of the main module.

6. Test program

The featured test program tests the following data workflows:

- Initializing the data structure;
- Adding normal items, items that are too long for the maximum data size and more items than can fit on the list;
- Tests the printing functionality;
- Finds items with keys that exist and do not exist;
- Finds the next and previous items even when we are at the head or end of a list;
- Edits items that exist and do not exist;
- Removes items that exist and do not exist;
- Removes every item from a list;
- Clears all items on the list.

These tests were created in order to test as much of the implemented functionality as possible, especially some edge-cases that might be harder to replicate manually. The test file also provides a quick way of telling if a change negatively affected the functionality of our module.