

CSC254 Assignment 4: Disassembler

Team: Yuesong Huang, Wentao Jiang

E-mail: yhu116@u.rochester.edu, wjiang20@u.rochester.edu

Introduction:

- Our Ruby program, `disassem`, that takes a `name` of the executable as input, and uses the output of `llvm-dwarfdump --debug-line` and `objdump -d` to construct web pages that contain side-by-side source and assembly language versions of a given **multi-file** program with additional web functionalities (extra credits).
***Note:** Our `disassem` program orchestrates all the works itself - the user doesn't need to perform any separate steps. The `name` (s) of the source file(s) from which an executable are gleaned from the `llvm-dwarfdump` output.

How to Run the Project:

- Run the `disassem` program with a given `*name`:

```
ruby disassem.rb *name
```

***Note:** `*name` should be a C executable file, compiled by GCC with flag `-g3`.

- Run the provided tests using `ascii.c` with a different compiled `*name` (`test_name`):

```
make test
```

- Run the provided Tests using `ascii.c` with a different optimization flages:

```
make test_01  
make test_02  
make test_03
```

How to Clean the Project:

- Clean all test executable, `*dump.txt`, and `*.html` files:

```
make clean
```

Feature:

- **Extra Credit Implementation:**

- i. Extended our program to accommodate executables compiled by multiple source files and generate their corresponding `disassem` HTMLs.
 - ii. Augmented the HTML pages so that up and down keys move to the (source or assembly) line above or below the currently selected line (if any), avoiding the need to click on the number of that line.
- The program will generate a HTML file for each source file, and the HTML file will be named as `[*source_file_name]_disassem.html` . For example, if the source file name is `ascii.c` , then the HTML file name will be `ascii_disassem.html` .
***Note:** If the source file does **NOT** have any corresponding assembly instruction, then the HTML file name will **NOT** be generated.

Program Experience:

1. Ruby is a powerful tool to handle system call and file I/O.
2. We used **Embedded-Ruby (ERB)** to generate HTML files. **ERB** is a powerful tool to use in this way. (could be find in `disassem.erb`)
3. Using several Regular Expression to match the pattern of the assembly code and source code, and then use the matched pattern of `llvm-dwarfdump` to generate the HTML files. (could be find everywhere in three main functions in `disassem.rb`)