Compiler Documentation

Thomas Maloney
net-id tmaloney@iastate.edu

February 2021

1 Part 0 Documentation

1.1 main.c

This file contains the entry point of the compiler. It passes the arguments from the command line to a method that parses them (described in section 1.2) and returns a `struct` that contains info about what flags were passed and what the possible output file might be (should that have been given). I then check which flag got passed in and whether the output should be written to a provided file or just `stdout`. The only `mode` flag that currently gets handled is `-0`.

1.2 args_parser

The header file contains a `struct`, an `enum` (both described in section 1.3), and a method signature that gets implemented in `args_parser.c`. Here I use `getopt` from `unistd.h` to read/parse each flag and then record which flag and whether there was an output file to a `struct` that I return. The potential output file gets passed back as an out-parameter. If no flags were passed in, or one that doesn't exist, I print out the string that describes how to use the program to `stderr`.

1.3 Data Structures

There is one `struct` and one `enum` that have been defined so far:

- (`struct`) `parsed_args_t`: This data type holds what `mode` flag gets passed in by way of the `mode_e enum`. It also contains a flag that signals if there is a specific output file that should be written to instead of `stdout`.
- (`enum`) `mode_e`: This just represents the possible `mode` flags that can be passed in plus an extra one that is set by default to make it easier to check if none of the required flags were passed in.