EECS6083: Compiler Theory

Max Thrun

April 24, 2014

1 Description

This project implements a recursive decent parser-generator compiler for the language specified in `projectLanguage.pdf`. The main compiler source code is written in Python and can be found in the `src` directory. It is broken up into three major parts: token construction (`src/scanner.py`), parsing (`src/parser.py`), and code generation (`src/gen.py`).

Memory is provided via the runtime as an integer array. Storing floats is accomplished by simply `memcpy`ing them into a slot in the memory array which makes the assumption that both floats and integers are 4 bytes. Strings are defaulted to a max length of 100 characters and all strings allocate the max length regardless of their actual size. Strings input by the user are truncated at the 100 byte mark. All strings and arrays are passed via pointers to and from procedures.

2 Usage

A wrapper script, `compiler.py`, is provided in the root directory of the project which provides an easy way to both compile the test programs and optionally run them after. Both the intermediate `.c` file and the final executable have the same name and are placed in the same directory as the input file. An example of compiling and running the `square_array.src` test file is shown below:

```
$ ./compiler.py -r tests/square_array.src
0 1 4 9 16 25 36 49 64 81
```

The full usage of `compiler.py` is shown below.

```
usage: compiler.py [-h] [-c] [-r] filename

EECS 6083 Compiler

positional arguments:
  filename      input .src file

optional arguments:
  -h, --help    show this help message and exit
  -c, --c_only  only generate .c file, do not compile it
  -r, --run     run the program after compiling it
```

3 Error Messages

A lot of time was spent trying to achieve the best possible error messages. There are currently 57 unique error and warning messages. Message styling and some phrasing were inspired by Clang. An example of some error messages from the test file `tests/errors.src` are shown below.

```
tests/errors.src:2:21: error: expected positive integer specifying array size
integer numbers[-1];
                ^

tests/errors.src:3:5: error: expected type mark
intger res[10];
^~~~~
tests/errors.src:5:45: error: expected keyword 'in' or 'out'
procedure square_array(integer array[10], integer result[10] out)
                                                ^

tests/errors.src:10:11: error: unsupported character '='
result[i] = array[i] * array[i];
          ^

tests/errors.src:14:11: error: expected ';' after statement
i := -1
      ^
```