Pascal Inspired Compiler

Zhengdong Wang
2017 CVUT FIT

Introduction

PIC is a very little pascal-like compiler based on LLVM. As the semestral work of BIE-PJP, this project is just a first try to make a compiler. The syntax and semantics are inspired by pascal, but there are lots of modification. This documentation is aimed to help readers know the features and drawbacks of this simple project.

File structrue

```
.
├── ast.cpp
├── ast.h             # Abstract Syntax Tree
├── input.cpp
├── input.h           # Input function
├── lexan.cpp
├── lexan.h           # Lexical Analyser
├── main_lexan.cpp    # Lexical Analyser Debugger
├── main_parser.cpp   # Parser Debugger
├── Makefile
├── parser.cpp
├── parser.h          # Parser
├── sfe.cpp           # LLVM Stuff: Codegen, Optimize..
├── tabsym.cpp
└── tabsym.h          # Symbol Table
```

Usage

Debugger

```
make lexan      # create excutable lexical Analyser
make parser     # create excutable parser
```

Combine with LLVM

Put all things (**without** `Makefile`) under `llvm-src/projects/sfe`. Just Make it! :)

Features

- Main function
- Printing numbers: use `WRITE`
- Reading numbers: use `READ`, `READ x` not `READ(X)`.
- **Global variable support**: Every variables declared in main block is global.
- Expression, Assignments.
- **Number constants in decadic, hexadecimal and octal base**: `20`, `0x3F`, `033`...
- **If, While, For (with exit)**:
  - Use `break` keyword instead of `exit`, just an alias. Similarly in functions.
  - `else` part is optional.;
  - If you want to use more than one statment(`;` , :P) in the block, use `begin...end` wrapper.
  - `For` **supports to and downto**, but actually it's just a translation to `while`.
- **Nested Block**: Illustrated as above.
- **Statically allocated arrays**: Just like `array[-11...30]`, but the second integer should larger than the first one.
- **Functions, Local variables**:
  - Functions should be declared in main block and defined immediately. That's a very big shortcoming of PIC. There is no difference between declaration and definition.
  - Every variable declared in sub-functions‘ block is local.
- **Parameters of functions**
- **Recursion**
- **Indirect recursion** : This feature is implemented in a special way. Because functions should only be declared and defined in main block. So in code geneartion, function's own code will genearted after every function declared. So it works.

Provided Test Code

Acknowledgement: they are based on examples on gitlab of CVUT FIT

```
expression2.p
# print,read,variable,constants,expressions,assignment...

arrayTest.p
# array, nested block, for to, downto..

sortBubble.p
# if, for, array, algorithm...

gcd.p
# while, if, functions, break, recursion, parameters, local variable
s...

indirectrecursion.p
# indirect recursion, break...
```