DEREK TROM AND ELENA CORPUS

FINAL DELIVERY

Mini Pascal Compiler Written in C

IO MODULE

- Responsible for reading input the pascal file and writing output to the scanner.
- The IO Module reads from the file character by character and is able to put the character back into a file stream.
- This gets the lexemes from the scanner needed for the lexical analysis and receives errors from the classes for the Lexer.

SCANNER

- Is responsible for the lexical analysis in the program.
- This scans the input from the IO module and translates the input into lexemes to be used by the parser.

PARSER

- Using Flex and Bison as automated generators for lexical analysis and parsing (LL1) will be used to verify the scanner and parser.
- In addition, the current parser generates an abstract syntax tree.

SYMBOL TABLE

- This keeps track of all the lexemes and its values.
- The parser is now able to add entries into the symbol table, with context, verify that there are no conflicting symbols, and create multiple levels of symbol tables for different blocks within the program, e.g. functions, procedures, etc.
- Print statements are shown for each entrance into a function call for proof of concept and its error messages as well.

INTERMEDIATE CODE GENERATION / CODE GENERATOR

- Generate machine dependent code
- Using the abstract syntax tree to generate the MIPS code