Tuna Han Salih MERAL 2012400120

Documentation Of Project 1

In project, every nonterminal given in project description is a function itself, either calling its subfunctions or creates output.

Program reads given mylang code and stores the whole code in a string, and tokenizes the code. After tokenizing, Main function calls the “start” function and starts the conversion by calling the appropriate function.

This is the list of my functions and what they do is:

- int getNum(stringstream& iss);
  Reads integer from the string and returns an integer

- string toLowerStr(string str);
  Assembly is a noncase-sensitive language. I changed all letters to lowercase in order to prevent any conflicts deriving from case-sensitivity. For example, if user creates an integer with the same name with some case conversion, I should assign this variable only once in assembly.

- void consumeSpace(stringstream& iss);
  While reading the *.my files and creating tokens, I used this function in order to get next char without no more than one space. My code can tokenize the given program even if paranthesis, “;”. Only the words should be written seperately.

- void multiply(stringstream& iss, ofstream& ofs);
  Converts given multiplication in mylang code into assembly

- void divide(stringstream& iss, ofstream& ofs);
  Converts given division in mylang code into assembly

- void modulo(stringstream& iss, ofstream& ofs);
  Converts given modulo operation in mylang code into assembly

- void addition(stringstream& iss, ofstream& ofs);
  Converts given addition in mylang code into assembly

- void subtraction(stringstream& iss, ofstream& ofs);
  Converts given subtraction in mylang code into assembly

- void factor(stringstream& iss,ofstream& ofs);
  A nonterminal given in Mylang BNF description.

- void morefactors(stringstream& iss,ofstream& ofs);
  A nonterminal given in Mylang BNF description.

- void moreterms(stringstream& iss,ofstream& ofs);
  A nonterminal given in Mylang BNF description.

- void term(stringstream& iss,ofstream& ofs);
  A nonterminal given in Mylang BNF description.

- void expr(stringstream& iss,ofstream& ofs);
  A nonterminal given in Mylang BNF description.

- void print_stm(stringstream& iss,ofstream& ofs);
  Reads the statement and reates “write” code in assembly

- void read_stm(stringstream& iss,ofstream& ofs);
  Creates code for reading an ID in assembly

- void if_stm(stringstream& iss,ofstream& ofs);
  Creates labels and statements in Assembly in order to convert If statements in Mylang

- void while_stm(stringstream& iss,ofstream& ofs);
  Creates labels and statements in Assembly in order to convert while statements in Mylang

- void init_stm(stringstream& iss,ofstream& ofs, string s);
  Initialization or assingnment statement

- bool stm(stringstream& iss,ofstream& ofs);
  A function for statements in Mylang, Function differentiates which statement is given and calls appropriate function. It returns true if there is a semicolon at the end of statement, which means there is more functions in statementlist

- void stmt_list(stringstream& iss,ofstream& ofs);
  Calls multiple statements

- void opt_stmts(stringstream& iss, ofstream& ofs);
  Calls statements if any statements is given

- void start(stringstream& iss,ofstream& ofs);
  Starts the conversion by calling the stm function.