What's in this folder

This folder contains template files for the last two phases of the JACK compiler.

Download the ‘Compiler files.zip’ file to your computer and unzip it. In the resulting folder, you will find the following files:

lexer.h: This is the same header file that you used to develop your lexer and parser modules.

parser.h: This is an updated version of the same header file that you used in your parser module. You will notice that the SyntaxErrors enumerated type has been augmented with two new items corresponding to two types of possible semantic errors: undecIdentifier (undeclared identifier) and redecIdentifier (redeclaration of identifier). You should augment your parser functions with the necessary code to detect any occurrence of these semantic errors and return the appropriate error type.

- symbols.h: use this file to define your own types and function prototypes for the symbol tables. You will notice that this file does not contain any declarations as it will be entirely up to you to decide how to implement the symbol table(s) functionality of the compiler.

symbols.c: use this file to implement your symbol tables and any functions needed to manipulate and query them. Again, this file is almost blank as you will decide how to implement the symbol table functionality.

- compiler.h: contains declaration of three functions that you need to implement to run the entire compiler:

- InitCompiler (): initialises the compiler
- compile (char* dir_name): compiles a JACK program in directory dir_name (remember that a JACK program is comprised of one or more JACK files in one directory).
- StopCompiler (): does any necessary clean-up before the program terminates.

- compiler.c: use this file to implement the above three compiler functions.

The zipped folder also contains a sample of valid JACK programs (e.g. Pong) that you can use to test your symbol table functionality.

The folder also contains a self-grader for the symbol table. Only two types of semantic checks are required in this case (undeclared identifiers or redeclared identifiers). You can use it to check that your compiler can correctly detect these.

Note that since a JACK program is defined in multiple classes residing in different files, your compiler cannot determine if a symbol (e.g. method) has not been declared until it has parsed the entire program.

To run your compiler, copy your own lexer.c and parser.c files to the same directory as other files (don’t copy lexer.h and parser.h), then compile the entire package using this command:

cc -std=c99 lexer.h parser.h symbols.h compiler.h lexer.c parser.c symbols.c compiler.c