Experiment in Compiler Construction

Scope management

ONE LOVE. ONE FUTURE.

Scope management

- Checking duplicate object declaration
- Checking reference to object

Checking fresh identifier

- A fresh identifier is an identifier that is new (has not been used) in current scope
- Checking fresh identifier is task of function

```
void checkFreshIdent(char *name);
```

Checking fresh identifier

- Checking fresh identifier is performed in
  - Constant declaration
  - User-defined type declaration
  - Variable declaration
  - Parameter declaration
  - Function declaration
  - Procedure declaration

Checking declared constant

- Performed when there is a reference to a constant, e.g:
  - When analysing an unsigned constant
  - When analysing an constant
- If a constant is not declared in current block, search in outer blocks.
- The value of declared constant will be the value of the constant that we are dealing with
  - Share the value
  - Do not share the value → **`duplicateConstantValue`**

Checking declared type

- Performed when there is a reference to a type, e.g: when analysing a type in function **`compileType`**
- If a type is not declared in current block, search in outer blocks
- The actual type of refered type name will be used to create the type we are dealing with
  - Share type
  - Do not share type → **`duplicateType`**

Checking declared variable

- Performed when there is a reference to a variable, e.g:
  - In for statement
  - When analysing factor
- If a variable is not declared in current block, search in outer blocks.

Checking declared LHS

- An identifier that appears in the left-hand side of an assign statement or in a factor possibly is:
  - Current function
  - A declared variable
    - If the variable's type is array type, the array index must follow the variable's name.
- Variable is different from parameters and current function.

Checking a declared procedure

- Performed when a procedure is referred, e.g:
  - In CALL statement
- If a procedure is not declared in current block, search in outer blocks.
- Global procedures: WRITEI, WRITEC, WRITELN

List of error codes

- ERR_UNDECLARED_IDENT
- ERR_UNDECLARED_CONSTANT
- ERR_UNDECLARED_TYPE
- ERR_UNDECLARED_VARIABLE
- ERR_UNDECLARED_PROCEDURE
- ERR_DUPLICATE_IDENT

Function lookupObject: Most closely scope rule

```
Object* lookupObject(char *name)

{  Scope* scope = symtab->currentScope;
  Object* obj;
  while (scope != NULL) {
    obj = findObject(scope->objList, name);
    if (obj != NULL) return obj;
    scope = scope->outer;
  }
  obj = findObject(symtab->globalObjectList, name);
  if (obj != NULL) return obj;
  return NULL;
}
```

Project organization

#	Filename	Task
1	Makefile	Project
2	scanner.c, scanner.h	Token reader
3	reader.h, reader.c	Read character from source file
4	charcode.h, charcode.c	Classify character
5	token.h, token.c	Recognize and classify token, keywords
6	error.h, error.c	Manage error types and messages
7	parser.c, parser.h	Parse programming structure
8	debug.c, debug.h	Debugging
9	symtab.c symtab.h	Symbol table construction
10	semantics.c. semantics.h	Analyse the program's semantic
11	main.c	Main program

Assignments

- Complete the following functions in semantics.c
  - **`checkDeclaredIdent`**
  - **`checkDeclaredVariable`**
  - **`checkDeclaredProcedure`**
  - **`checkDeclaredLValueIdent`**
- Insert code to the functions marked TODO or or other places (if necessary) to check when an object is declared or used.
- Compile and test with sample examples.