Principles of Compilers
General Lab description

February 23, 2006

Our objective in the Labs of "Principles of Compilers" will be the specification of a small programming language, and the implementation of a compiler for it, using tools like the lex and yacc programs.

The language to be specified should contain at least the following elements (additional features are optional):

> A program in this language consists of a text file with a list of declarations, which may be variables, or subprograms. All variables and subprograms must be declared before actually used. Variable declaration include the type of the variable, and a possible initial value. Subprogram declaration defines whether it is a function or a procedure, and the parameters needed to call it. In the case of function it must also specify the type of the returned value. Parameters in subprogram declaration should be consistent with parameters in the call. Only integers and arrays of integers are allowed as data types. The language must allow for plain function and procedure definitions (no nested, no recursion). Local declarations in each subprogram are possible. Simple parameters are passed by value, array parameters are passed by reference. Statements should include an assignment, function or procedure calls, a conditional, a while iteration and a compound statement. A return statement for ending subprograms is also needed. An input and an output statements should be predefined in the global environment. Conditions should include the usual comparison operators, combined with the usual boolean operators. Comments can start anywhere, and can span through multiple lines.

Todo list for delivering the project:

1. specify the grammar of your programming language (in some formal grammar language, like BNF). This is the first Assignment due 9/3/2006.
2. specify the list of tokens
3. implement a scanner of your language using Lex
4. implement a parser of your language using Yacc
5. design the symbol table data structure
6. extend your parser with actions for type checking