

Compiler Basics

Compiling Process

- Aim: convert high level source code to machine instructions
 - Also generates a list of errors and warnings for the programmer
- Several stages
 - One pass over the input file
 - Several passes over the input file

Stages in the Compilation Process

- Lexical analysis
 - Scan the source file and break into tokens using the basic delimiters
 - Tokens can be of several types (data types, user specified such as variable names, operators)
 - Each token type is recorded and put into a symbol table
- Syntax checking
 - Checks whether the token sequences are compliant with the language grammar

Stages in the Compilation Process

- Syntax checking
 - The parsing process generates a parse tree
 - The parse tree is a representation of the high level instructions that is independent of the source language
 - Syntactically correct programs comply with the rules of the language but may not be valid

Stages in the Compilation Process

- Semantic analysis
 - Checks whether the parsed program is compliant with the semantic rules of the language
 - Common semantic errors: type incompatibility, undeclared identifiers (depending on the language!)
- Optimization
 - Attempts to generate a compact, very efficient version of the code
 - Uses the machine specific features to improve speed
- Object code generation and linking