Practical - 6

Aim: Study of YACC / Bison tool for syntax analyzer.

Theory:

In compiler design, syntax analysis (or parsing) is the second phas of compilation, following lexical analysis. It checks whether a given sequence of tokens follows the syntactic rule of programming language. If the syntax ~~of~~ is correct, the parser generates a parse tree or abstract syntax tree (AST), which sources as an intermediate representation of the program. Syntax analyzer use context-free grammars (CFGs) which consist of a set of production rule that defines the valid syntax of a programming language. These rules are processed by parser generators like YACC and Bison, which automates the creation of syntax analyzers.

YACC: Yet Another Compiler Compiler is a parser generator, which is a program that takes as its input a specification of the syntax of the programming langauage, and produce as its output a parse procedure for the language whose name is yyparse(). The notation used for preparing this specification is a context free grammar (CFG). YACC is a LALR parser generator developed at the beginning of the 1970's by Stephen C Johnson for the Unix Operating system. YACC palys an important role in compiler and interpreter development since it provides a means to specify the grammar of a language and to produce parsers that either interpret or compile code written in that language.

YCCE

Page No.

Date

Key Concepts and Features of YACC

1) Grammar Specialization : The input to YACC is a CFG that describes the syntax rules of the language it parses

2) Parser Generation : YACC translates the grammar into a C function that could perform an efficient parsing of input text according to such predefined rules.

3) LALR(1) Parsing : This is a bottom-up parsing method that makes use of a single token lookahead in determining the next action of parsing.

4) Semantic Actions : These are the grammar production that are associated with an action ; this enables the execution of code, usually in C, used in the construction of abstract syntax tree, the generation of intermediate representation or error handling.

5) Attribute Grammar: These grammar consist of non-terminal grammar symbols with attributes, which through semantic actions are used in the construction of parse tree or the output of code.

6) Integration with Lex: It is often used along with Lex, a tool that generates lexical analyzers, which breaks input into tokens that are then processed by the YACC parser.

YCCE

Page No.

Date

Bison :

Bison is a parser generator that translates a context-free grammar into a C-based parser. It is the GNU version of YACC and is used for generating syntax analyzer in compilers, interpreters and language processing tools.

Features of Bison :

* Bison uses Look-Ahead Left to Right, Rightmost Derivation (LALR(1)) parsing technique which balances efficiency and grammar handling power.

* It supports YACC syntax and is often a drop-in-replacement

* Provides advanced error handling using % error-verbose and user-defined error messages

* Supports multiple independent parser instances, making it useful in modern applications.

* Generates parsers in C, C++ and other languages (multi language support)

Advantages:

→ Automates Syntax Analysis : Saves time compared to manual parser writing

→ Handles Complex Grammar : Supports nested and recursive structure.

→ Provides error handling support.

→ Works seamlessly with lexical analyzer for tokenization.

→ Efficient and optimised : LALR(1) parsing makes it lightweight and fast.

Disadvantages:

1) Limited to LALR(1) Grammar: Cannot handle full LR(1) grammar.
2) Error handling complexity : Custom error recovery may require extra effort.
3) No Built in AST generation: Requires additional logic to construct an abstract syntax tree.

Application:

1) Used in programming language compilers design like GCC
2) Used in scripting language parsor.
3) Used in scientific computing applications where mathematical expression's calculation is required.
4) Configuration file parsor parses structured data like JSON or XML

Conclusion : We have studied the YACC / Bison tools. They automate syntax analysis by generating efficient parsers. Bison enhances YACC with better error handling. These tools are essential for compilers and language processing development.