

Recreating the Lisa Pascal Compiler

USING FLEX AND BISON

EQUIPO 10

Introduction

- For our final project, we were tasked with recreating the Lisa Pascal Compiler using Flex and Bison based on a given user manual. This project involved understanding the workings of lexical analysis and syntax parsing, constructing an effective symbol table, and ensuring that our Pascal compiler would successfully parse and analyze Pascal programs.

Project Description

- The Lisa Pascal Compiler project aimed to create a compiler capable of lexical analysis, syntax analysis, and semantic analysis for Pascal programs. Flex was utilized to generate the lexical analyzer, which tokenizes the input source code. Bison was used to build the parser, applying grammar rules to the sequence of tokens produced by the lexical analyzer.

Lexer Specification

- Defined in 'lisa.l', includes tokens for Pascal keywords, operators, identifiers, and numerical values.
- Handled comments and white spaces effectively.
- Multi-line comments and line number tracking for better error reporting.


Parser Specification

- Defined in 'lisa.y', governs the structure of valid Pascal programs.
- Token types and precedence rules are defined.
- Semantic actions populate the symbol table with parsed identifiers.
- Covers various Pascal constructs such as program headers, block structures, and statements.

Symbol Table Implementation

- Implemented as a hashmap to store information about identifiers.
- Stored details: names, scope (ambito), type, line number.
- Allowed quick lookups for identifiers, ensuring semantic correctness.

Project Challenges

1. Understanding Flex and Bison: Required time to understand their syntax and capabilities.
 2. Error Handling: Ensuring meaningful feedback for syntax and semantic errors.
 3. Memory Management: Managing dynamic memory for symbol table entries, ensuring no memory leaks or segmentation faults.
- 



Learning Outcomes

1. Compiler Construction: Practical experience in building compilers, learning about lexical analyzers and parsers.
2. Flex and Bison Proficiency: Improved understanding of these tools for lexical and syntax analysis.
3. Problem-Solving Skills: Enhanced problem-solving abilities, especially in debugging complex compiler issues.
4. Team Collaboration: Learned effective communication and project management techniques.

Conclusion

- Recreating the Lisa Pascal Compiler was challenging yet rewarding. It provided deep insights into compiler design and enhanced our technical capabilities in using Flex and Bison. The project underscored the importance of thorough testing and robust error handling in software development. The project equipped us with practical skills and knowledge for future endeavors.