

Lab Five

John DeFalco

john.defalco1@marist.edu

October 2019

GOAL

Work with symbol tables.

EXAMPLES FROM THE READINGS

1 CRAFTING A COMPILER

Below are the examples listed on the requirements document for the lab from the *Crafting a Compiler* textbook.

1.1 PROBLEM 8.1

The two data structures most commonly used to implement symbol tables in production compilers are binary search trees and hash tables. What are the advantages and disadvantages of using each of these data structures for symbol tables?

Binary search trees and hash tables have different advantages and disadvantages that come with them. Starting with hash tables, the advantage that comes with this data structure is quickly searching is optimized for the symbol table. Insertion and lookup into the symbol table can reach speeds of $O(1)$ with each entry having a unique hash key. However, the disadvantage is this data structure is difficult to implement into the source code, and can be complicated. On the other hand, binary trees are easy to implement into the compiler, however the speed they reach is $O(\log_2 n)$ on average.

1.2 PROBLEM 8.3

Describe two alternative approaches to handling multiple scopes in a symbol table, and list the actions required to open and close a scope for each alternative. Trace the sequence of actions that would be performed for each alternative during compilation of the program in Figure 8.1.

The two different alternatives for handling multiple scopes are the approaches to the symbol table. A user can either create a symbol table for each scope, or one table for all scopes.

If one were to make individual tables for each scope, they would have to have a scope stack of symbol tables to maintain. Every time a scope is opened, the stack would have to push a new table into the stack, and when the scope is closed, that table would be popped.

If one were to have just one table for all scopes, all levels of scope must be entered with each symbol. This is easier for the program to search for a variable at a certain scope, as the program does not have to search through a stack of different tables.