National University of Sciences and Technology (NUST)
School of Electrical Engineering and Computer Science

Department of Computing

CS 354: Compiler Construction

Class: BSCS-5AB

Lab [03]: Tokens and Hash Values

Date: 27th Sep, 2018

Time: [09:00 – 12:00hrs &14:00 – 16:50 hrs]

Instructor: Dr. Rabia Irfan

Lab Engineer: Mr. Azaz Farooq

National University of Sciences and Technology (NUST)
School of Electrical Engineering and Computer Science

Lab [03] : Tokens and Hash Values

Introduction

This lab comprises the implementation of tokenization and creating hash values.

Objectives

1. Successful understanding/implementation symbol table in C/C++/Java

Tools/Software Requirement

1. gcc, g++, GNU Make or Visual Studio C++

Description

Symbol table stores the symbols of the source program as the compiler encounters them. The symbol table is populated incrementally during the analysis phase. This information is used during the synthesis phase for target code generation. Each entry contains the symbol name plus a number of parameters describing what is known about the symbol. Reserved words (if, then, else, etc.) maybe stored in the symbol table as well.

Recall that a token is a pair consisting of a

token name: an abstract symbol representing lexical unit e.g. ID, NUM, OP and
attribute/token value: e.g. x, 5, +.

A hash value is a numeric value of a fixed length that uniquely identifies data.

Lab Tasks

1. create a program to make a symbol table with three columns, no comments allowed in the output.
2. Remember to add a hash value column (2 marks)
3. You have to read a separate file (leapyear.c, available on LMS) to extract symbols, tokens and generate hash values, as was asked in previous lab. Your output should be generate in the following pattern:

Token Name	Token value	Hash Value

Deliverables

You are required to upload your task (Sources & Word/PDF document) using the link created on LMS followed by a viva. Your word file should contain screenshot of your output.

Note: You are not allowed to use the built-in functions/libraries to perform the above tasks.