**Scheme of Course Project**

**Name of the course :** Compiler Design

**Year / Semester :** III- B. Tech I-semester

**Project Title :** Parser for a Text Editor.

**Aim :** To create a Parser for C Language that accepts the Specified Grammar.

Team Members:

1.Challa Vishnu Vardhan Reddy - 15071A05J0

2.Varun Patlola - 15071A05L5

3.Swaroop Manchala - 15071A05K6

4.Nikhil Alapati - 15071A05I1

**Stages involved:**

**1.** Identification of all the Tokens in the Language.

**2.** Construction of finite Automata and Transition table for it Using Symbol Table.

**3.** Understanding the Concept of Parsing.

**4**. Construction of Parse tree for a given grammar and checking whether it is grammatically correct or not.

**5.** Display of errors in the code.

**Features:**

**1.** A lexical Analyzer to identify all the tokens in the Language.

**2.** Generation of Regular expression for all the tokens and Construction of Finite Automata.

**3.** Construction of Transition Table for Obtained Finite Automata.

**4.** From the grammar obtained construction of parse tree is done to check whether the language statement is grammatically correct or not.

Typical Time line of course project (after completion of theory portion of proposed project in the course):

Steps For Compilation: -

$ lex Compiler.l

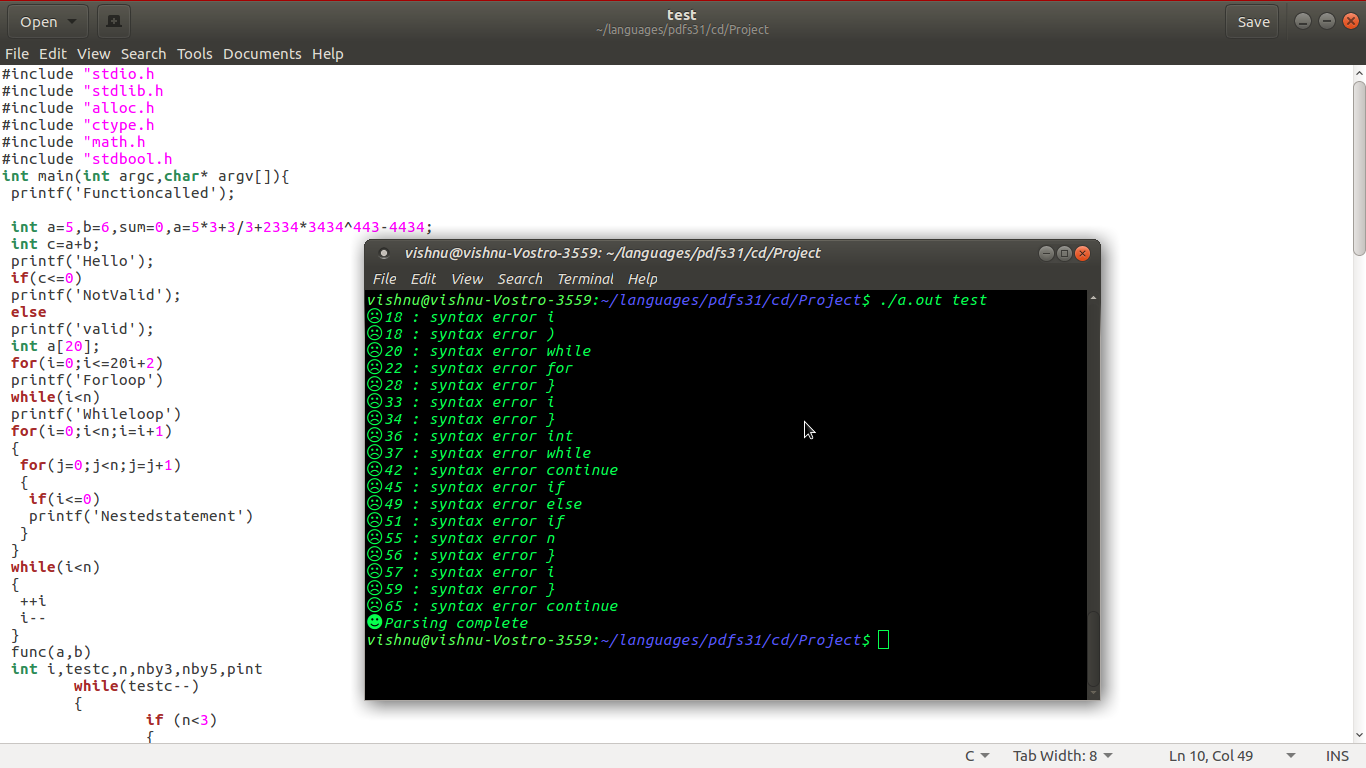
$ yacc Compiler.y

$ gcc y.tab.c -ll -ly

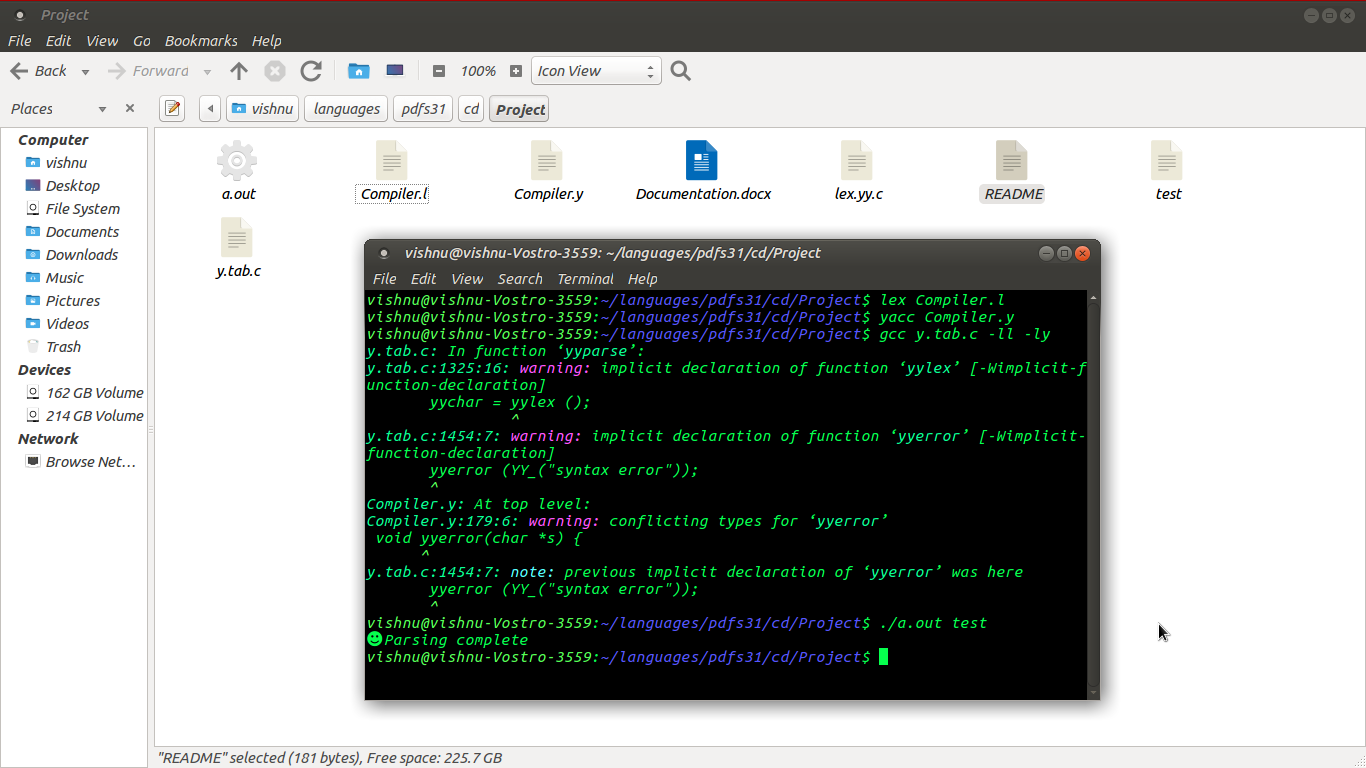
$ ./a.out [filename]

Overview

1.If the Code has errors then the Terminal Screen appears as Below.



2.If there are No Errors in the Code the Terminal Screen Appears as Below.



Result : Hence a Parser Has Been Implements Using Lex and Yacc.