A Mini-Project Report

ON

Development of Parser for C

BY

Bhavani B - 1PI11IS027

Anushree Prasanna Kumar - 1PI11IS017

B Swathi Manaswini - 1PI11IS022

Guide

Prof. Nypunya D.

Department of IS & E

PESIT

Bangalore

August 2013 - Dec 2013

DEPARTMENT OF INFORMATION SCIENCE & ENGINEERING
PES INSTITUTE OF TECHNOLOGY
100 FEET RING ROAD, BANASHANKARI III STATE
BANGALORE-560085

TITLE :

Development of Parser for C/any scripting language .

ABSTRACT :

In computer technology, a parser is a program that receives input in the form of sequential source program instruction and command line instruction and breaks them up into parts that can then be managed by other programming components of a compiler.

Parsing or syntactic analysis is the process of analysing a string of symbols in computer languages, according to the rules of a formal grammar specified in the parsing program.

A parser constructs an intermediate representation, produces meaningful error messages, and performs context-free syntax analysis. It also guides sensitive analysis and attempts error correction.

INTRODUCTION :

The minproject employs LALR Parser for implementation which is the most oftenly used technique. It is based on LR(0) set of items and has many fewer states compared to other typical parsers that are based on LR(1) items . LALR

parser helps remove shift/reduce conflicts that arise in LR(1).However it doesnt eliminate reduce-reduce conflicts completely.

ALGORITHM/APPROACH :

1. Input the expression
2. Based on the regular expression involing multiple states, a state diagram and a state table is
   generated in the file parse.out
3. The input expression is the parsed using the parser table generated.
4. If an error in the input expression (improper syntax) exists then an error is thrown
   else the expression is considered valid .
5.The corresponding output is displayed on the screen.

Steps involved in Construction of the Parser:

1. Use of Python2.7-Module-ply

2. Method = LALR parser

3. Regular expressions for all types of syntax written as individual functions

4. Command Shell like environment for input

Example :

Input1 :

input statement > int a,b,c=10;

Output 1:

None // indicating valid statement.

Input 2 :

input statement > int a,b

Output 2:
Error in the code
// indicating invalid statement