

CDSS MINI PROJECT

COMMAND LINE CALCULATOR



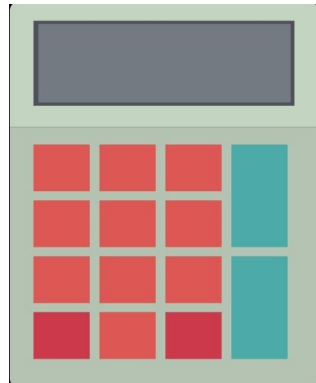
BY:
ROHIT K
SAHANA H P

CONTENTS

1. INTRODUCTION
2. PROBLEM STATEMENT
3. OBJECTIVES
4. REQUIREMENTS
5. LEXER
6. PARSER
7. IMPLEMENTATION / DEMONSTRATION
8. RESULTS
9. CONCLUSION
10. REFERENCES

INTRODUCTION

A command line calculator which supports mathematical expressions with scientific functions is very useful for most developers. The calculator available with Windows does not support most scientific functions. The most difficult part I found when designing such a calculator was the parsing logic. Later while working with .NET, the runtime source code compilation made the parsing logic easy and interesting. It uses runtime compilation and saves the variables by serializing in a file. Thus you can get the values of all the variables used in the previous calculation.



PROBLEM STATEMENT

The calculate function calculates an expression. It uses the saved variables. I have generated code which has a declaration of the variables

- To Evaluate the given expressions.
- To perform basic calculations

OBJECTIVES

The command line calculator is to be capable of parsing a human-readable mathematical expression with units, return the value if it can be evaluated and inform the user about the position of an error if not.

REQUIREMENTS

SOFTWARE REQUIREMENTS :

Windows/Ubuntu Operating System

C Programming Language

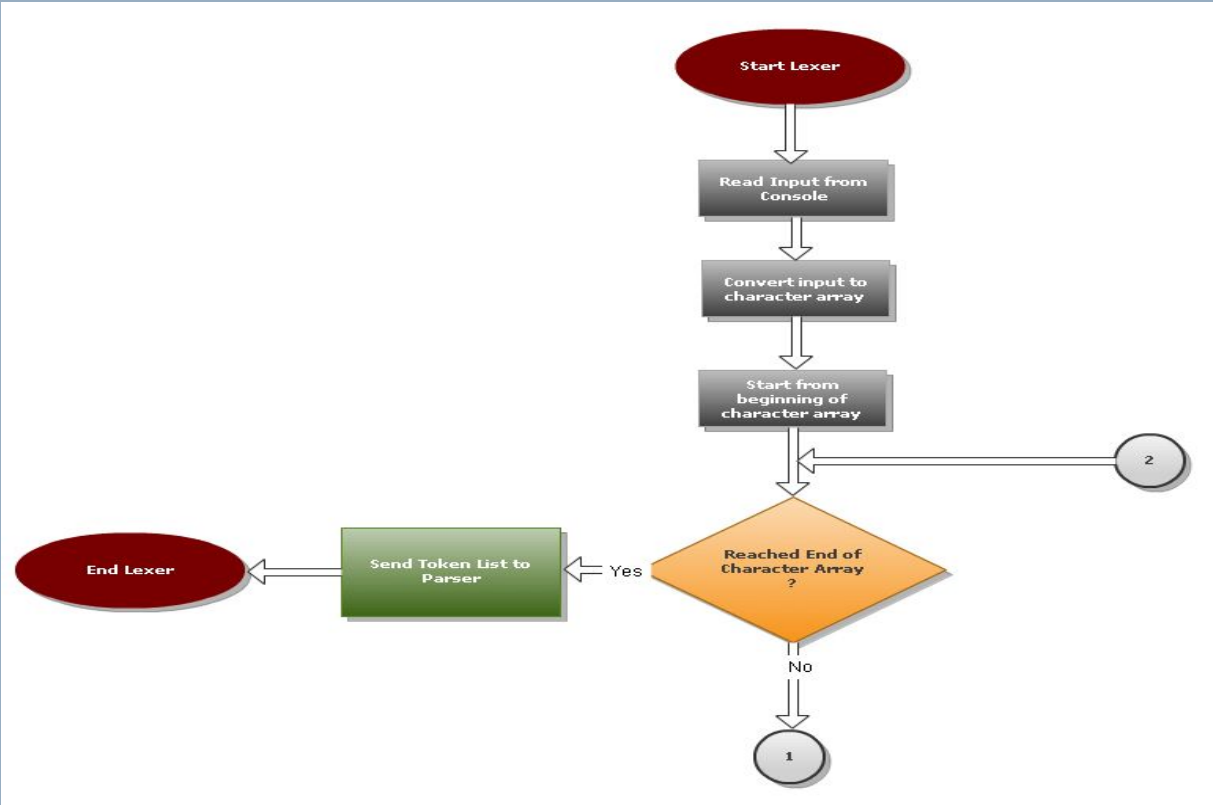
HARDWARE REQUIREMENTS :

Minimum of 4GB RAM

LEXER

- A lexer is a software program that performs lexical analysis.
- Lexical analysis is the process of separating a stream of characters into different words, which in computer science we call 'tokens' .
- For Example -While reading we are performing the lexical operation of breaking the string of text at the space characters into multiple words.

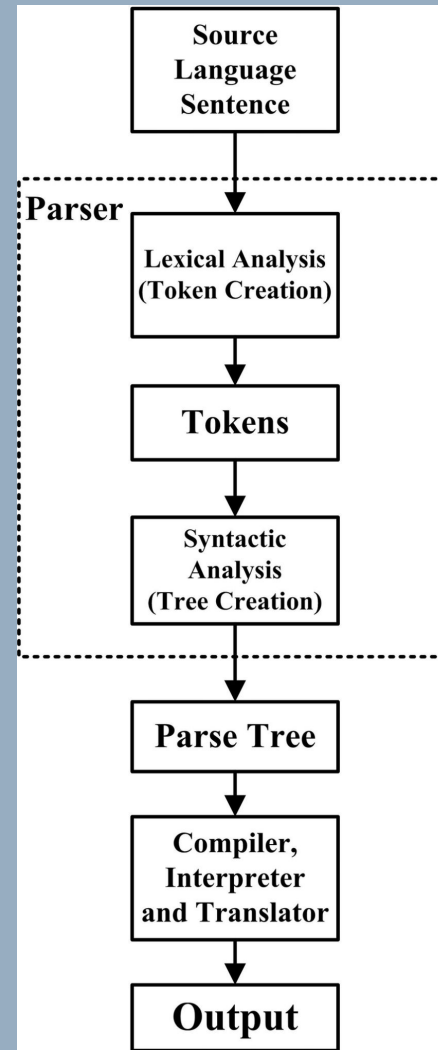
LEXER



PARSER

- A parser goes one level further than the lexer and takes the tokens produced by the lexer and tries to determine if proper sentences have been formed.
- Parsers work at the grammatical level, lexers work at the word level.
- Generally yacc is used to parse language syntax.
- The name "yacc" stands for "Yet Another Compiler Compiler" .
- Yacc uses a parsing technique known as LR-parsing or shift-reduce parsing.

PARSER



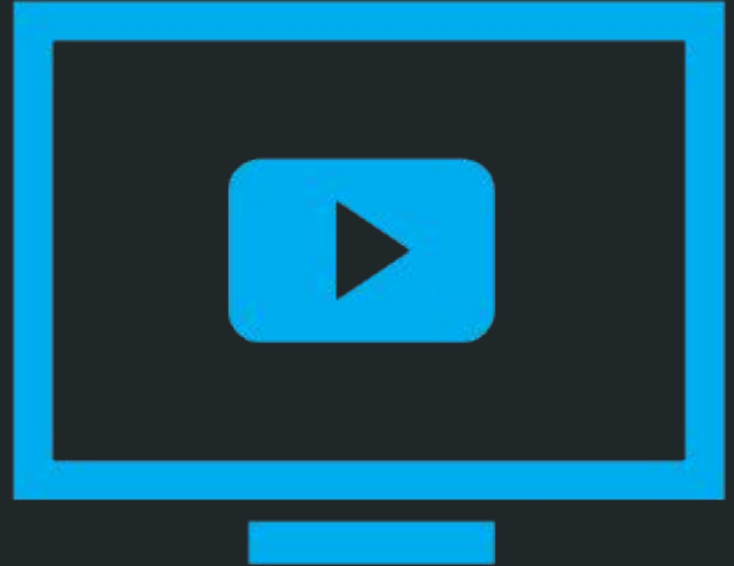
IMPLEMENTATION



IMPLEMENTED IN C

—

DEMONSTRATION



—

RESULTS

```
C:\Users\saich\Downloads\calc.exe
Use Control-C to quit.
> 5+5
parse_expr()
  parse_term()
    parse_factor()
      parse_num_op()
    parse_rest_term()
  parse_rest_expr()
    parse_term()
      parse_factor()
        parse_num_op()
      parse_rest_term()
    parse_rest_expr()
10.000000
> 5*3
parse_expr()
  parse_term()
    parse_factor()
      parse_num_op()
    parse_rest_term()
  parse_factor()
    parse_num_op()
  parse_rest_term()
  parse_rest_expr()
15.000000
> 10/2
parse_expr()
  parse_term()
    parse_factor()
      parse_num_op()
    parse_rest_term()
  parse_factor()
    parse_num_op()
  parse_rest_term()
  parse_rest_expr()
5.000000
> 10/2
```

RESULTS

C:\Users\saich\Downloads\calc.exe

[illegible]

RESULTS

C:\Users\HP\Downloads\calc.exe

[illegible]

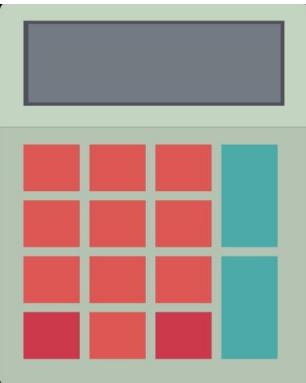
RESULTS

C:\Users\HP\Downloads\calc.exe

```
        parse_factor()
            parse_num_op()
            parse_rest_term()
                parse_rest_expr()
                    parse_rest_term()
                        parse_factor()
                            parse_num_op()
                            parse_rest_term()
                                parse_rest_expr()
                                    parse_term()
                                        parse_factor()
                                            parse_num_op()
                                            parse_rest_term()
                                                parse_rest_expr()
                                                    parse_term()
                                                        parse_factor()
                                                            parse_num_op()
                                                            parse_rest_term()
                                                                parse_factor()
                                                                    parse_num_op()
                                                                    parse_rest_term()
                                                                        parse_rest_expr()
-34.000000
> (3-4*4+6/2)+3-7-5*4_
```


CONCLUSION

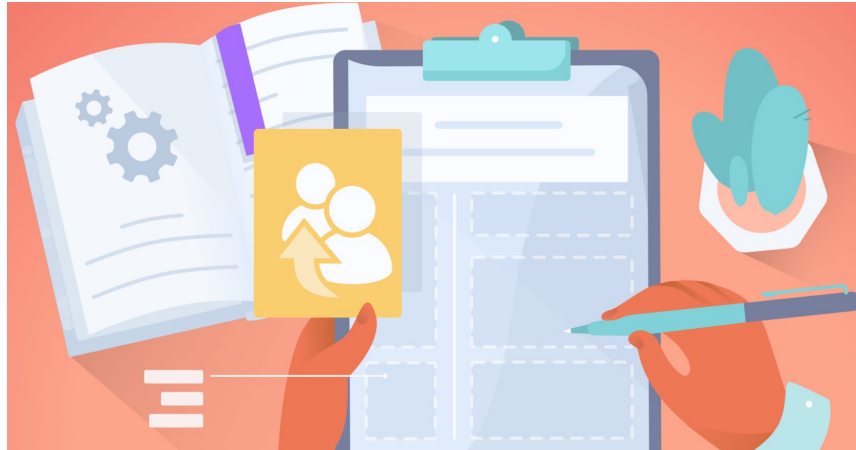
This is a powerful and versatile command-line calculator that really lives up to your expectation. Preloaded on all modern Linux distributions, this can make your number crunching tasks much easier to handle without leaving your terminals. Besides, if your shell script requires floating point calculation, can easily be invoked by the script to get the job done. All in all, CLC should definitely be in your productivity tool set.



REFERENCES

[1] - <https://www.codeproject.com/Articles/12395/A-Command-Line-Calculator>

[2] - <https://fedoramagazine.org/bc-command-line-calculator/>



THANK YOU

