

# Regular scanner vs Lex

Jesus Alvarez

**Abstract**—This paper will try to determine if the use of lex is faster than a c code implementation for a basic scanner

## I. INTRODUCTION

When trying to create a compiler one of the first steps to be able to make one is to generate a lexical analyzer which purpose is to scan through an input set of characters and be able to classify them.

## II. PROBLEM

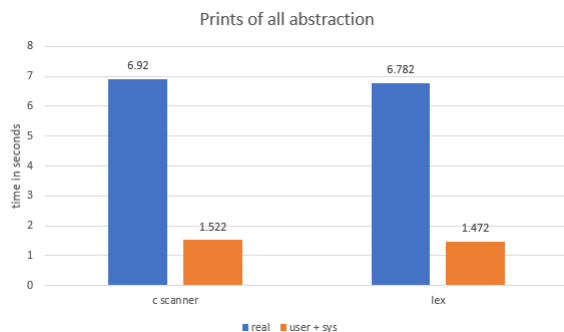
While creating a basic scanner is quite simple in a c code, because the scanner is one of the fundamentals and it has to interpret the input as whole, optimization is something that is very critical in this part as a little bit of difference in time can affect performance of the compiler in production use. Lex offers a way to generate a scanner that can do the same as a c code and more by using regular expressions. The main issue to discuss is if lex really offers an advantage over a plain c code.

## III. SOLUTION

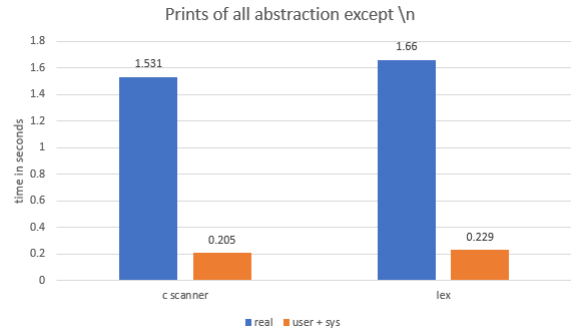
In order to prove this it is necessary to test the time it takes to process a file in which both do the same level of analysis and abstraction. The analyzed input will be a set of randomly generated code snippets that are analyzed by both. Time command will be used to know how much total time as well as CPU time (user + sys) it takes for each one to execute with the same input.

## IV. RESULTS

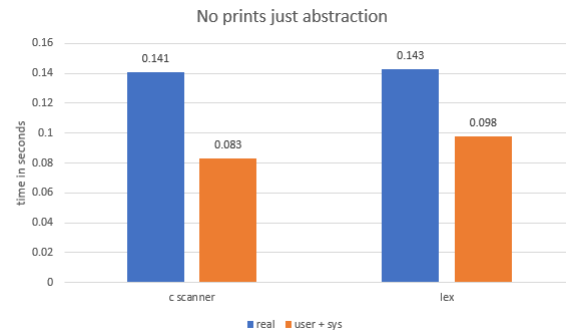
### A. Execution with prints to console



### B. Execution with prints to console without line breaks



### C. Execution with no prints



## V. CONCLUSIONS

As seen in the graphics we can conclude that the time difference of lex and a c code is almost non-existent to the point where lex takes a tiny amount of time more. While this is true the power and ease of use of lex makes it a better option to use especially when doing more than a scanner as using it paired with YACC allows to develop a more efficient basis for the compiler than doing it all in c.

## REFERENCES

- [1] Hubert, B. (2004, September 20). Lex and YACC primer/HOWTO. Retrieved February 23, 2019, from <https://ds9a.nl/lex-yacc/cvs/lex-yacc-howto.html>