

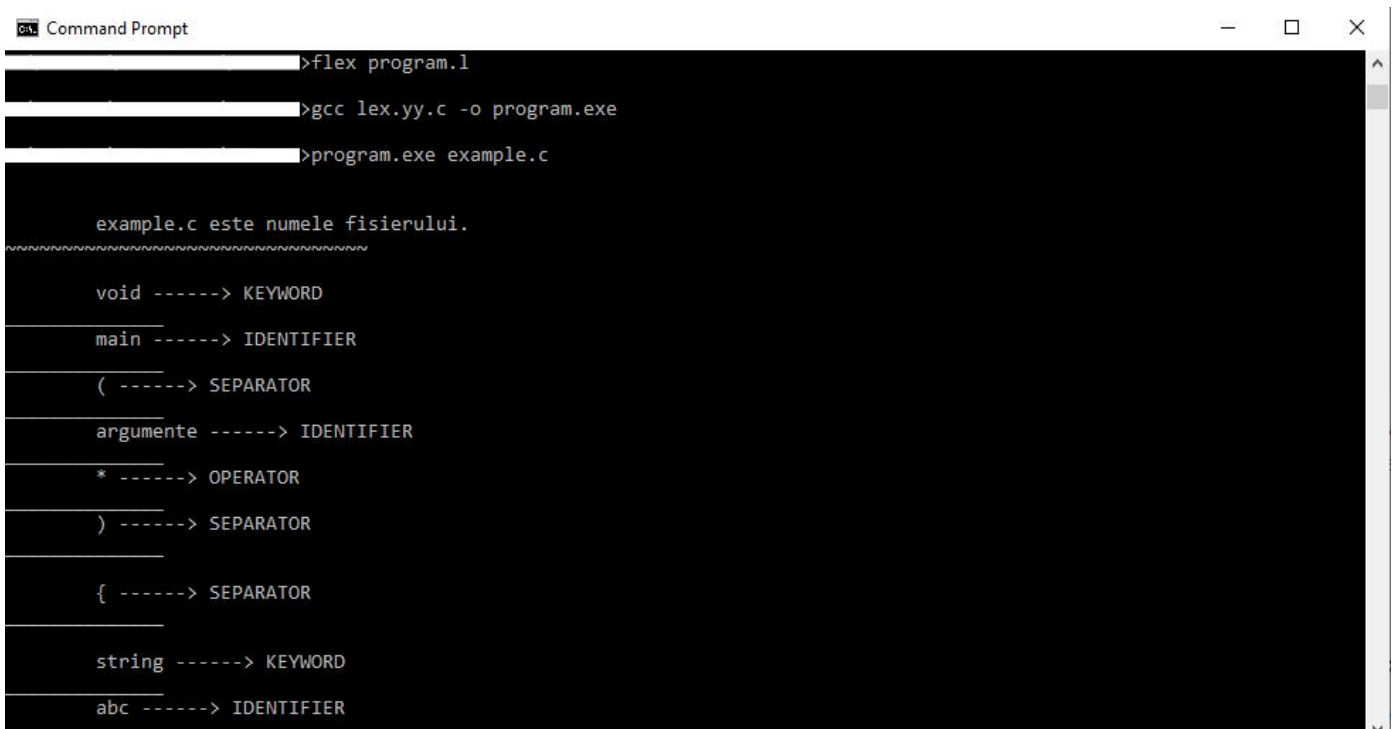
Tehnici de Compilare

Tema 2 : 1.7 – Analizor lexical folosind Lex

Tema propusa are in vedere utilizarea software-ului Lex, pentru a scrie si a utiliza in mod corect un analizor lexical pentru limbajul C/C++. Acestea fiind spuse, procesul se imparte in mai multe subprocesse :

- 1) Dezvoltarea programului cu extensie Lex (extensie “.l”), deci scrierea specificatiei Lex a limbajului vizat;
- 2) Utilizarea programului “flex” pentru a genera codul sursa al analizorului propriu-zis ce rezulta din specificatia Lex amintita mai sus; din acest pas va rezulta un fisier “lex.yy.c”;
- 3) Fisierul rezultat din pasul 2 se va compila folosind gcc, rezultand un executabil;
- 4) Executabilul de la pasul 3 este rulat, **avand ca argument un fisier tinta**, argument ce reprezinta programul pe care se face analiza lexicala;

Exemplu analiza lexicala pe programul “example.c”



```
Command Prompt
>flex program.l
>gcc lex.yy.c -o program.exe
>program.exe example.c

example.c este numele fisierului.
~~~~~
void -----> KEYWORD
main -----> IDENTIFIER
( -----> SEPARATOR
argumente -----> IDENTIFIER
* -----> OPERATOR
) -----> SEPARATOR
{ -----> SEPARATOR

string -----> KEYWORD
abc -----> IDENTIFIER
```

Referinte si site-uri utile :

- [https://en.wikipedia.org/wiki/Lex_\(software\)](https://en.wikipedia.org/wiki/Lex_(software))
- <https://regex101.com>
- <http://bmohanrajcse.blogspot.com/2018/01/3-implementation-of-lexical-analyzer.html>
- <https://cs6612compilerlab.blogspot.com/2017/12/implementation-of-lexical-analyzer.html>
- <https://www.geeksforgeeks.org/flex-fast-lexical-analyzer-generator/>