
The lexical analyzer generator — Lex

Exercise 1.

a) Write and compile the following specification file :

```
/* just like Unix wc */
%{
int chars = 0;
int words = 0;
int lines = 0;
}%
%%
[a-zA-Z]+ { words++; chars += yyleng; }
\n      { chars++; lines++; }
.       { chars++; }
%%
main(int argc, char **argv)
{
yyin=fopen("lex.yy.c", "r");
yylex();
printf("lines=%d\n words=%d\n words=%d", lines, words, chars);
fclose(yyin);
}
```

To compile a Lex program, we follow these steps:

- 1) Save the specification file with the extension `.l` (example : `tp1lex.l`)
- 2) Compile the file `tp1lex.l` with the **Lex compiler** using the command:
`lex tp1lex.l` (The result of this compilation is a C specification file called `lex.yy.c`)
- 3) Compile the file `lex.yy.c` with the **GCC compiler** using the command:
`gcc lex.yy.c -o tp1lex.exe -fl` (The result of this compilation is an executable file called `tp1lex.exe`)
- 4) Run the executable file `tp1lex.exe` with the command :
`./tp1lex.exe`