

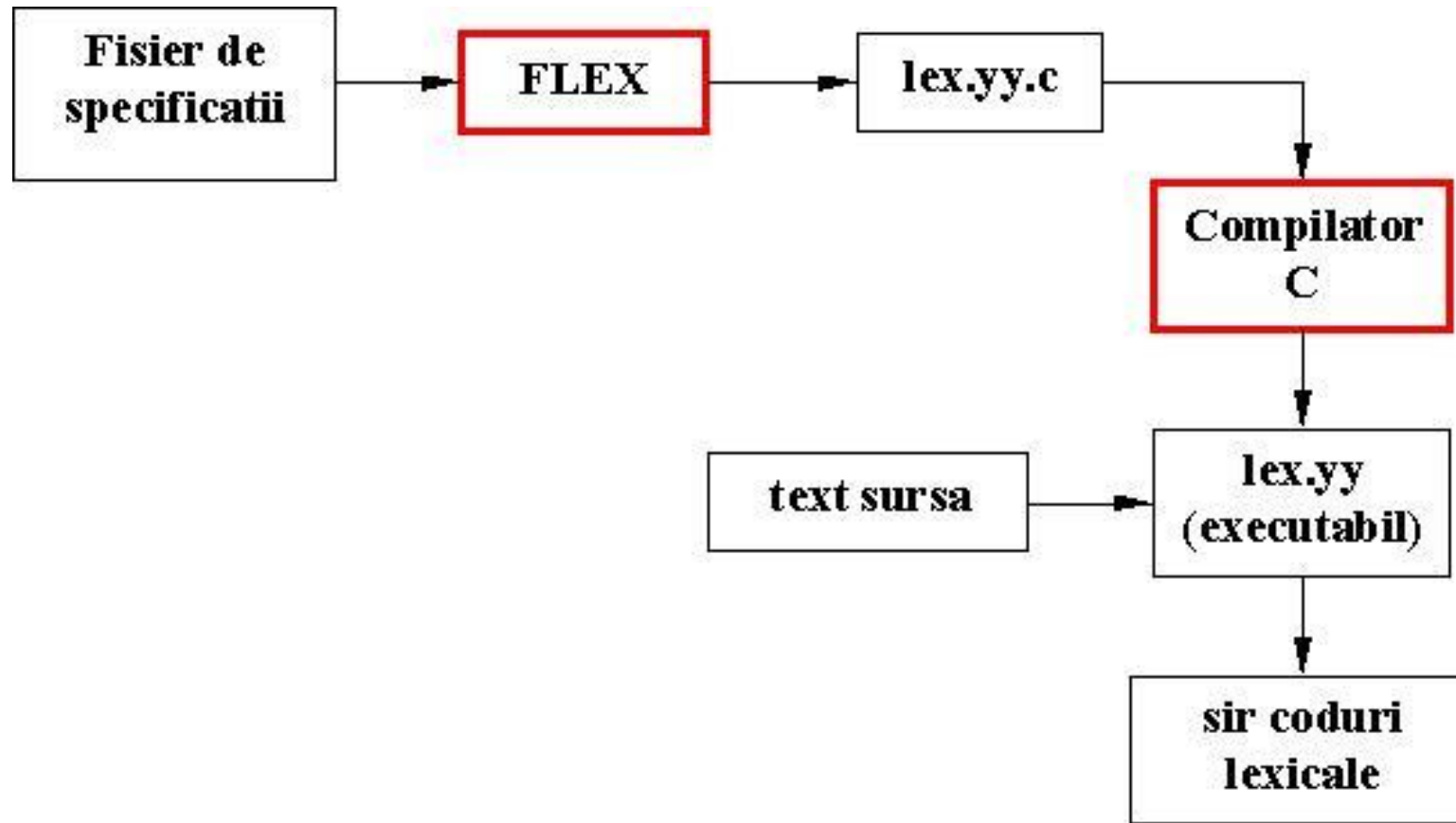
# LEX & YACC

1. Have you heard about these tools?
2. Have you used any of them?

# Scanning & Parsing Tools

- Scanning => lex
- Parsing => yacc

# Lex – Unix utility (flex – Windows version)



# INPUT FILE FORMAT

- The file containing the specification is a text file, that can have any name. Due to historic reasons we recommend the extension **.lxi**.
- Consists of 3 sections separated by a line containing %%:

definitions

%%

rules

%%

user code

### *Example 1:*

`%s`

```
username printf( "%s", getlogin() );
```

**specifies a scanner that, when finding the string  
“username”, will replace it with the user login name**

## Definition Section:

- C declarations

+

- declarations of simple *name definitions* (used to simplify the scanner specification), of the form

`name definition`

- where:
  - **name** is a word formed by one or more letters, digits, '\_' or '-', with the remark that the first character MUST be letter or '\_' and must be written on the FIRST POSITION OF THE LINE.
  - **definition** is a regular expression and is starting with the first nonblank character after name until the end of line.
  - declarations of *start conditions*.

## Rules Section

- to associate semantic actions with regular expressions. It may also contain user defined C code, in the following way:

**pattern action**

where:

- **pattern** is a regular expression, whose first character MUST BE ON THE FIRST POSITION OF THE LINE;
- **action** is a sequence of one or more C statements that MUST START ON THE SAME LINE WITH THE PATTERN. If there are more than one statements they will be nested between {}. In particular, the action can be a void statement.

## User Defined Code Section:

- Is optional (if is missing, then the separator %% following the rules section can also miss). If it exists, then its containing user defined C code is copied without any change at the end of the file lex.yy.c.
- Normally, in the user defined code section, one may have:
  - function *main()* containing call(s) to *yylex()*, if we want the scanner to work autonomously (for ex., to test it);
  - other called functions from *yylex()* (for ex. *yywrap()* or functions called during actions); in this case, the user code from definitions section must contain: either prototypes, either **#include** directives of the headers containing the prototypes



Launching the execution:

`lex [option] [name_specification _file]`

where *name\_specification \_file* is an input file (implicitly, stdin)

**\$ lex spec.lxi**

**\$ gcc lex.yy.c -o your\_lex**

**\$ your\_lex<input.txt**

**options:** <http://dinosaur.compilertools.net/flex/manpage.html>

# Example