

# **C PROGRAMMING**

## **Lecture 10**

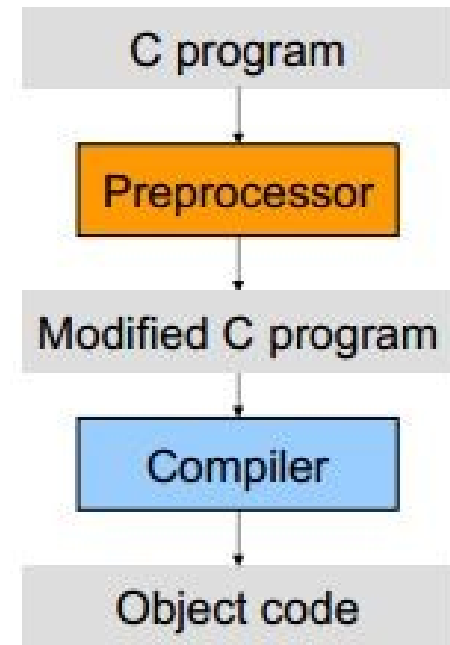
**1st semester 2023 - 2024**

# Example

```
#include <stdio.h>
#include <string.h>
int main (int argc, char* argv[]) {
    char linie[1000], *p, *q;
    int propozitii = 0;
    FILE* f = fopen(argv[1], "r");
    for (;;) {
        p = fgets(linie, 1000, f);
        if (p == NULL) break;
        for (p = linie, q = NULL;; ) {
            q = strchr(p, '.');
            if (q == NULL) break;
            propozitii++;
            p = q + 1;
        }
    }
}
```

# C Preprocessor

- Before a C program is compiled, the source code file is first handled by a preprocessor
- The output of preprocessor (C code) is the actual input for the compiler



# Preprocessor Role

- The actions:
  - Removes comments and whitespaces from the source code
  - Acts according to preprocessor directives
  - Expands some kind of characters
  - ! Does not compile code
- Preprocessor directive:
  - Command recognized only by preprocessor
  - Each directive prefixed by #

# Preprocessor Role

- Directives and facilities provided for:
  - File inclusion
    - #include
  - Macro expansion
    - #define
  - Conditional compilation
    - #if, #ifdef, #ifndef, ...

# File Inclusion

- Specifies that the preprocessor should read in the contents of the specified file
  - usually used to read in type definitions, prototypes, etc.
  - proceeds recursively
    - #includes in the included file are read in as well
- Two forms:
  - #include <filename>
    - searches for filename from a predefined list of directories
    - the list can be extended via “gcc -I *dir*”
  - #include “filename”
    - looks for *filename* specified as a relative or absolute path

# File Inclusion

`#include <filename>`

This form is used to include files from the standard library.

The search for the file is performed in the system directories of the compiler.

This form is used when including standard header files, such as those from the C Standard Library.

`#include <stdio.h>`

# File Inclusion

- `#include filename`

This form is used to include project-specific or local files.

The search for the file begins in the current source directory and then extends to other project-specific directories.

This form is used when including header files defined within the project or in other parts of the code.

```
#include "myheader.h"
```



# File Inclusion

- a header file has file-extension '**.h**'
  - these header files typically contain “public” information
    - type declarations
    - macros and other definitions
    - function prototypes
  - Most of the time the public information associated with a code file **f.c** will be placed in a header file **f.h**
  - these header files are included by files that need that public information

# File Inclusion – Examples

```
/* Load system header for stdio library */  
#include <stdio.h>
```

```
/* Load system header for math library */  
#include <math.h>
```

```
/* Load user header file */  
#include "file_header.h"
```

```
/* Loads user header file */  
#include "../My_Lib/mylib.h"
```

# File Inclusion

- What happens if we try to include the same file multiple times?

# File Inclusion

- In C, if you attempt to include the same file multiple times using the `#include` directive, the preprocessor ensures that the contents of the file are included only once.
- This is achieved through the use of include guards or, more recently, the **`#pragma once`** directive.

# File Inclusion

- Include Guards:
- In the header file that you want to include, you include conditional compilation directives to check whether a particular symbol (commonly the file name in uppercase with underscores) has been defined. If it hasn't, you define it and include the contents of the file; otherwise, you skip the inclusion.
- `// myheader.h`
- `#ifndef MYHEADER_H`
- `#define MYHEADER_H`
- `// Contents of the header file`
- `#endif // MYHEADER_H`
-

# File Inclusion

- `#pragma once`:

An alternative to include guards is to use `#pragma once` at the beginning of the header file. This directive ensures that the file is included only once during compilation.

```
// myheader.h
```

- `#pragma once`

```
// Contents of the header file
```

# File Inclusion

- `// math_functions.h`
- `#ifndef MATH_FUNCTIONS_H`
- `#define MATH_FUNCTIONS_H`  
`// Declaration of the addition function`
- `int add(int a, int b);`  
`#endif // MATH_FUNCTIONS_H`

# File Inclusion

- `// math_functions.c`
- `#include "math_functions.h"`
- `// Definition of the addition function`
- `int add(int a, int b) {`
- `return a + b;`
- `}`



# File Inclusion

- `// main.c`
- `#include <stdio.h>`
- `#include "math_functions.h"`
- `int main() {`
- `// Using the addition function from the header file`
- `int result = add(5, 3);`
- `// Displaying the result`
- `printf("The sum is: %d\n", result);`
- `return 0;`
- `}`