

**Source File**

* This is the C program you write, typically saved with a .c extension (e.g., program.c).
* Example:

#include <stdio.h>

int main() {

printf("Hello, World!\n");

return 0;

}

**2. Preprocessor**

* **Input**: The source code file (.c).
* **Output**: Preprocessed file (.i file).
* The preprocessor handles **preprocessor directives** (lines starting with # like #include, #define).
  + Includes the contents of header files (e.g., <stdio.h>).
  + Expands macros.
  + Removes comments.
* Result: A new file with expanded code ready for compilation.

**Compiler**

* **Input**: The preprocessed file (.i file).
* **Output**: Assembly code file (.s file).
* The compiler translates the high-level C code into **assembly language**, which is a low-level, human-readable representation of the program.
* Example of assembly code:

mov eax, 4

int 0x80

**Assembler**

* **Input**: The assembly code file (.s file).
* **Output**: Object code file (.o file).
* The assembler converts the assembly language into **machine code** (binary), which is stored in an object file.
* This object file is not yet an executable—it’s just a compiled version of your code without external libraries.

**Linker**

* **Input**: Object code file (.o file).
* **Output**: Executable file.
* The linker combines your object file with other necessary **library files** (like the Standard C Library for functions like printf).
* Resolves external references (e.g., linking printf to its implementation).
* Produces a complete **executable file** that can be run directly.

**Final Output: Executable File**

* The final file is a binary executable (.exe on Windows or no extension on Linux/Unix).
* You can now run your program to see the output.

**Key Summary:**

1. **Preprocessor**: Prepares the code for compilation by expanding macros and including headers.
2. **Compiler**: Translates preprocessed code into assembly language.
3. **Assembler**: Converts assembly code into machine-readable object code.
4. **Linker**: Combines object code with libraries to create an executable.