The Compilation Process of C Program

A modern C compiler by default compile code and generate an executable file. We can roughly split the process into four stages:

1. Pre-processing
2. Compilation
3. Assembly
4. Linking

1. Pre-processing: - Pre-processing is the first stage of compilation. In this stage lines start with # character are interpreted as the preprocessor or define macros or remove comments from program. Program also remove #include line or expand header file code or include it in our source file.

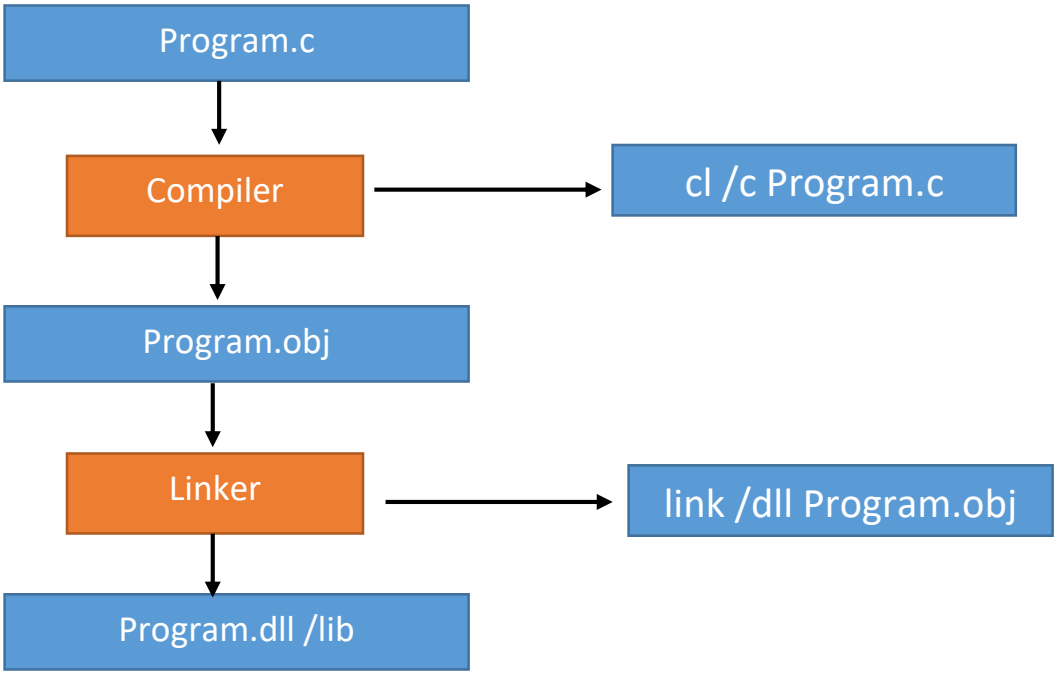
2. Compilation: - Compilation is the second stage of compiling C program. In this stage, the preprocessed code is translated to assembly instructions specific to the target processor architecture.

3. Assembly: - Assembly is the third stage of compilation. In this stage an assembler is used to translate the assembly instructions to object code. The output consists of actual instructions to be run by the target processor. Assembly process generate an object source file with .obj or .o extension.

4. Linking: - Linking is the last stage of compilation process where linker program arrange all the functions, addresses and calling procedure. Linker program can generate an executable, lib or dll depends on given parameter. If no parameter than by default it will generate an executable file.

Steps for compiling Assignment 2 library(dll) file :-

1. Create a “library.h” file and declare the functions information or prototype.
2. Create a “library.c” file and include library.h
3. Write functions code inside library.c
4. Compile library.c file with following command. This command will generate an object file (.obj) : - **cl /c library.c** .
5. ‘/c’ is for only generate an object file.
6. Link object file with following command :- link /DLL library.obj
7. ‘dll’ is for generate a dll file. That mean this program does not contain main() function.



Compilation process of a C program or generate a DLL file.