C Program Compilation Process

C compilation is the process of converting source code written in the C programming language into machine code that can be executed by a computer. The following steps are typically involved in the C compilation process:

1. Pre-processing: The first step in C compilation is pre-processing. The pre-processor reads the source code and performs operations such as expanding macros, including header files, and removing comments.

2. Compilation: Once pre-processing is complete, the compiler reads the pre-processed source code and translates it into assembly language, which is a low-level representation of the program.

3. Assembly: The assembly code is then translated into machine code by an assembler. This produces an object file, which contains the machine code and other data required by the linker.

4. Linking: The linker combines the object file with other necessary libraries to create an executable file. The linker resolves external references, such as function calls and variable declarations, and generates the final binary executable.

During the compilation process, the compiler performs several checks on the source code to ensure that it is syntactically and semantically correct. If errors are found, the compiler generates error messages indicating the location and nature of the errors.

Once the compilation process is complete and an executable file is generated, it can be run on the machine.

Github repository link : https://github.com/im-sahiljain/Dos-2023