

Review

1. What is the first file in the C compilation process?

Answer:

The .c file, which contains the original C source code written by the programmer.

2. What is the C preprocessor?

Answer:

The preprocessor processes C source code **before compilation**. It handles directives such as #include, #define, and #ifdef.

3. What does #include do?

Answer:

#include tells the preprocessor to include the contents of a header file in the source code.

Example:

```
#include <stdio.h>
```

Review

4. What is a .i file?

Answer:

A .i file contains the **preprocessed C source code**. It is produced by the preprocessor from the .c file.
.c → .i

5. What does the compiler do?

Answer:

The compiler translates the **preprocessed C code (.i)** into **assembly language (.s)**.
.i → .s

6. What is an .s file?

Answer:

An .s file is a **text file containing assembly-language instructions** for a particular computer architecture.

Review

7. What does the assembler do?

Answer:

The assembler translates the assembly-language file (.s) into an **object file (.o)** containing machine code.

.s → .o

8. What does the linker do?

Answer:

The linker combines object files and required libraries to create the **final executable program**.

.o + libraries → executable

9. What is the complete C compilation process?

Answer:

.c → .i → .s → .o → executable

More specifically:

C source



Preprocessor



.i



Compiler



.s



Assembler



.o



Linker



Executable

Review

10. What is the difference between a compiler and an assembler?

Answer:

- Compiler:** Translates C code into assembly language.
- Assembler:** Translates assembly language into machine/object code.

C code → Compiler → Assembly

Assembly → Assembler → Object code