1_2

Chapter 1: A Tour of Computer System

1.2 Programs Are Translated by Other Programs into Different Forms

```
本节主要介绍了，源文件被什么程序通过什么处理过程逐渐成为一个可执行程序的。
```

Why we need to translate a program

- The hello program begins life as a high-level C program because it can be read and understood by human beings in that form.
- In order to run hello.c on the system, the individual C statements must be translated by other programs into a sequence of low-level machine-language instructions.
- These instructions are then packaged in a form called an executable object program and stored as a binary disk file. Object programs are also referred to as executable object files.
- On a Unix system, the translation from source file to object file is performed by a compiler driver.

How to translate a program from a source file to an executable object file

![](./1_2.assets/Screenshot 2023-09-26 at 13.38.51.png)

- 4 phases from source file (hello.c) to an executable object file (hello):
  - Preprocessing (hello.c -> hello.i) - 预处理
    - modifies the original C program according to directives that begin with the '#' character.
    - The result is another C program, typically with the .i suffix.

    ```
    gcc -E hello.c  // show hello.i in console
    or
    gcc -save-temps -S hello.c // save all temp files (.i, .s)
    ```

  - Compilation (hello.i -> hello.s) - 编译
    - The compiler (cc1) translates the text file hello.i into the text file hello.s, which contains an assembly-language program.

    ```
    main:
            subq    $8, %rsp
            movl    $.LC0, %edi
            call    puts
            movl    $0, %eax
            addq    %8, %rsp
            ret
    ```

    ```
    gcc -S hello.c // keep .s file
    ```

    - Understanding assembly-language is key to machine-level execution model.
  - Assembly (hello.s -> hello.o) - 汇编

  - the assembler (as) translates hello.s into machine-language instructions, packages them in a form known as a relocatable object program, and stores the result in the object file hello.o.

    ```
    gcc -c hello.c  // keep object file
    ```

  - Disassembling object file:

    ```
    objdump -d hello.o
    ```

- Linking (hello.o -> hello) - 链接
  - hello program calls the printf function, which is part of the standard C library provided by every C compiler.
  - The printf function resides in a separate precompiled object file called printf.o, which must somehow be merged with our hello.o program.
  - The linker (ld) handles this merging.