

循序渐进，学习开发一个 RISC-V 上的操作系统



第 3 章 编译与链接

软件所智能软件中心 PLCT 实验室 汪辰

➤ **GCC 介绍**

➤ **ELF 介绍**

➤ **GCC 介绍**

- GCC 简介
- GCC 的命令格式
- GCC 的主要执行步骤
- GCC 涉及的文件类型

➤ **ELF 介绍**

➤ **GCC (GNU Compiler Collection)**

- <https://gcc.gnu.org/>



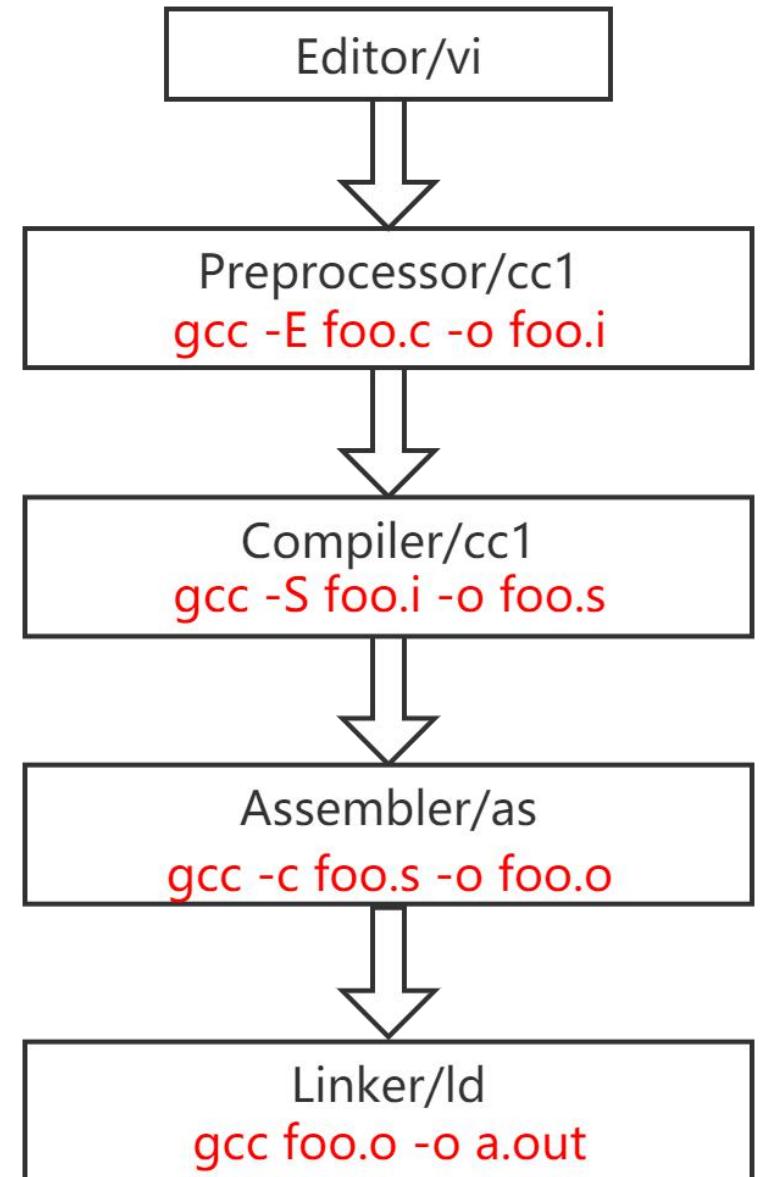
- 由 GNU 开发的，遵循 GPL 许可证发行的编译器套件。
- 支持 C、C++、Objective-C、Fortran、Ada 和 Go 语言等多种语言前端，已被移植到多种计算机体系架构上，如 x86、ARM、RISC-V 等。
- GCC 的初衷是为 GNU 操作系统专门编写一款编译器，现已被大多数 “Unix-like” 操作系统（如 Linux、BSD、MacOS 等）采纳为标准的编译器。

gcc [options] [filenames]

常用选项	含义
-E	只做预处理
-C	只编译不链接，生成目标文件 “.o”
-S	生成汇编代码
-o file	把输出生成到由 file 指定文件名的文件中
-g	在输出的文件中加入支持调试的信息
-v	显示输出详细的命令执行过程信息

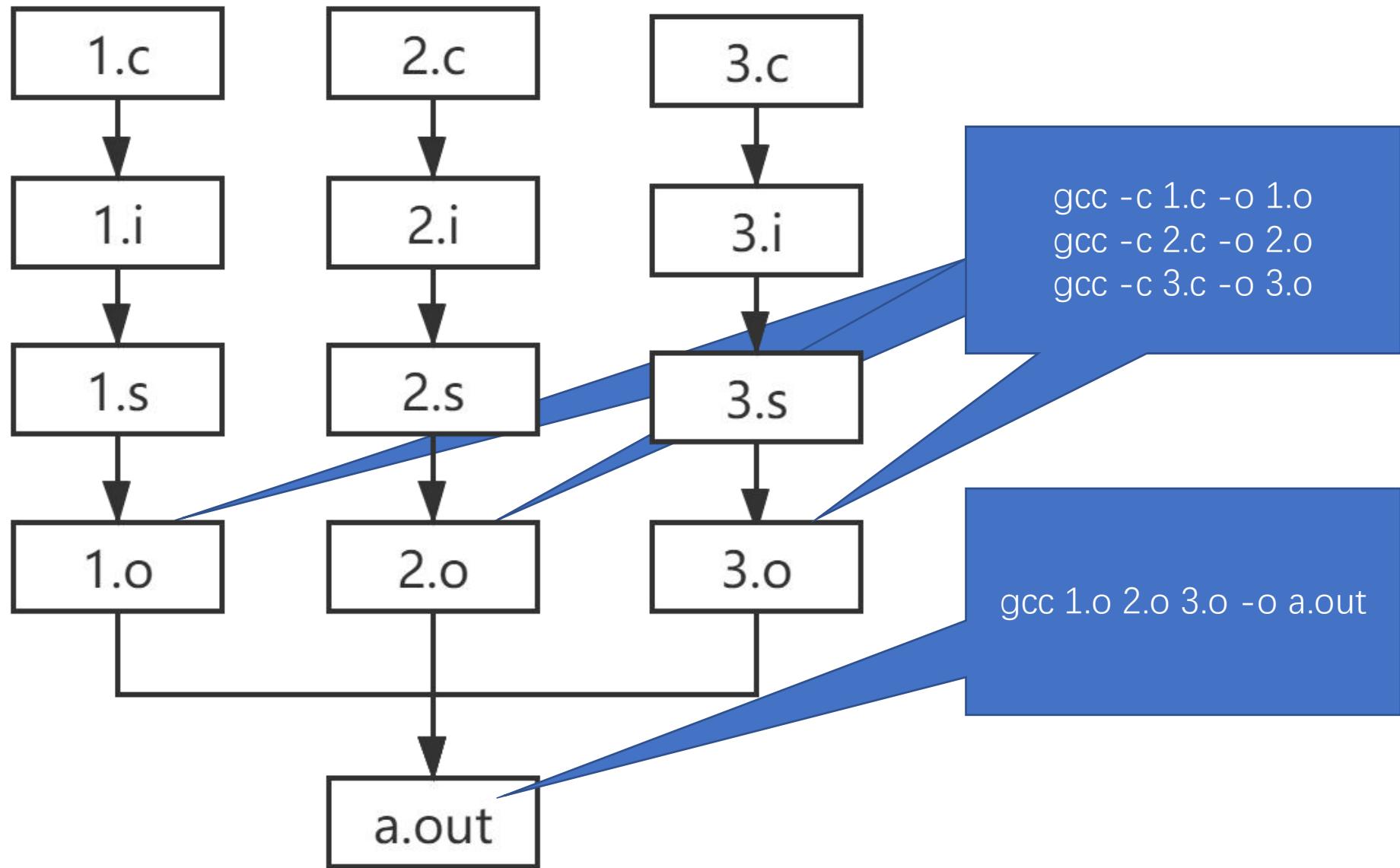
GCC 的主要执行步骤

- **编译 (cc1, 这里针对 C 语言, 不同的语言有自己的编译器)** : 编译器完成“预处理”和“编译”, “预处理”指处理源文件中以“#”开头的预处理指令, 譬如 `#include`、`#define` 等; “编译”则针对预处理的结果进行一系列的词法分析、语法分析、语义分析, 优化后生成汇编指令, 存放在 `.o` 为后缀的目标文件中。
- **汇编 (as)** : 汇编器将汇编语言代码转换为机器 (CPU) 可以执行的指令。
- **链接 (ld)** : 链接器将汇编器生成的目标文件和一些标准库 (譬如 `libc`) 文件组合, 形成最终可执行的应用程序。



- **.c**: C 源文件
- **.cc/.cxx/.cpp**: C++ 源文件
- **.i**: 经过预处理的 C 源文件
- **.s/.S**: 汇编语言源文件
- **.h**: 头 (header) 文件
- **.o**: 目标 (object) 文件
- **.a/.so**: 编译后的静态库 (archive) 文件和共享库 (shared object) 文件
- **a.out**: 可执行文件

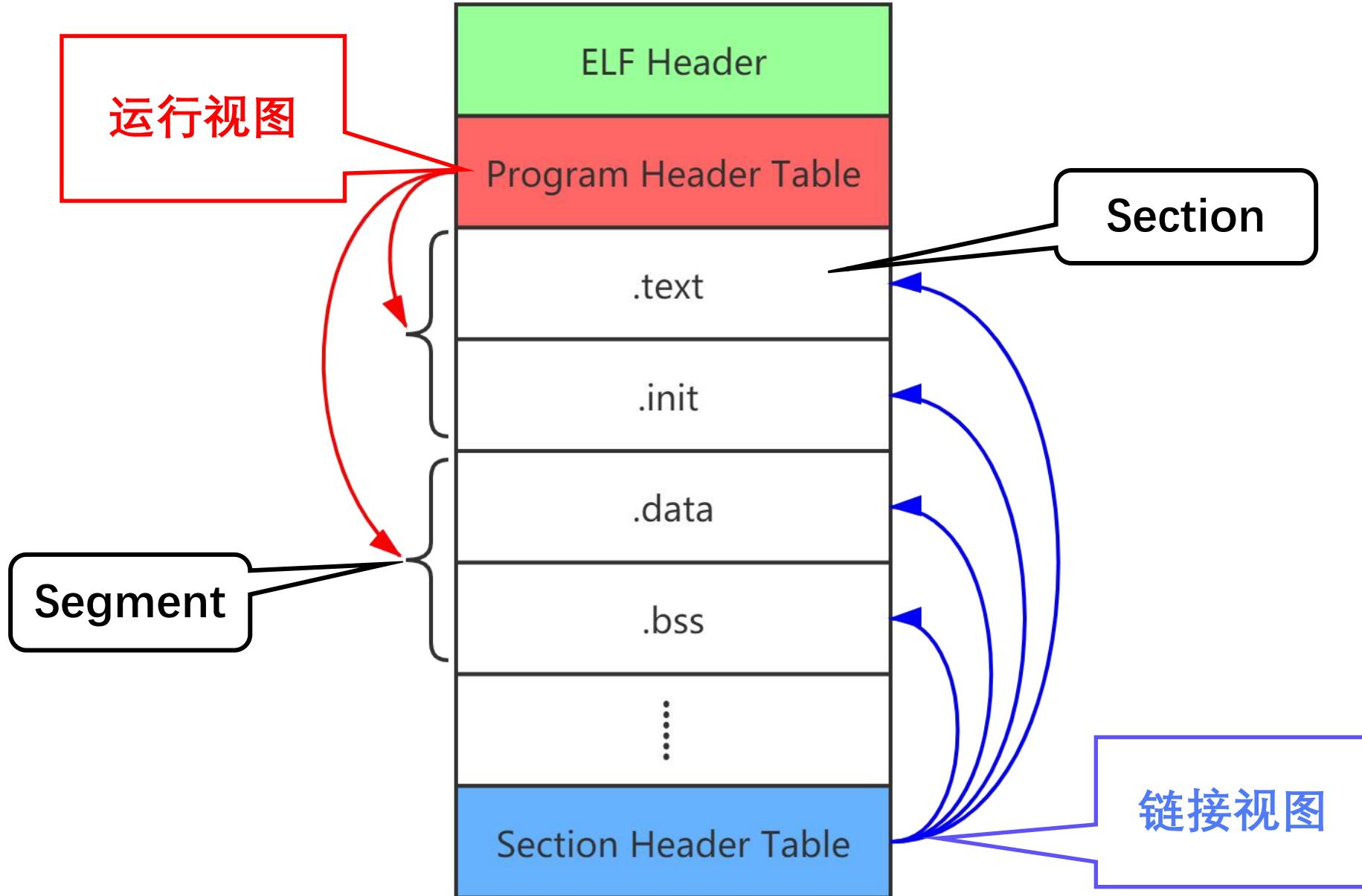
针对多个源文件的处理



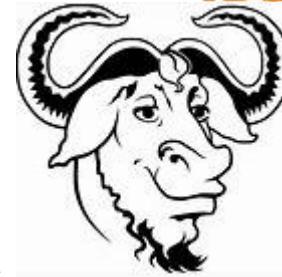
- **GCC 介绍**
- **ELF 介绍**
 - ELF 简介
 - ELF 文件格式
 - ELF 文件处理相关工具：Binutils

- ELF (Executable Linkable Format) 是一种 Unix-like 系统上的二进制文件格式标准。
- ELF 标准中定义的采用 ELF 格式的文件分为 4 类：

ELF 文件类型	说明	实例
可重定位文件 (Relocatable File)	内容包含了代码和数据，可以被链接成可执行文件或共享目标文件。	Linux 上的 .o 文件
可执行文件 (Executable File)	可以直接执行的程序	Linux 上的 a.out
共享目标文件 (Shared Object File)	内容包含了代码和数据，可以作为链接器的输入，在链接阶段和其他的 Relocatable File 或者 Shared Object File 一起链接成新的 Object File；或者在运行阶段，作为动态链接器的输入，和 Executable File 结合，作为进程的一部分来运行	Linux 上的 .so
核心转储文件 (Core Dump File)	进程意外终止时，系统可以将该进程的部分内容和终止时的其他状态信息保存到该文件中以供调试分析。	Linux 上的 core 文件



➤ <https://www.gnu.org/software/binutils/>



- **ar**: 归档文件, 将多个文件打包成一个大文件。
- **as**: 被 gcc 调用, 输入汇编文件, 输出目标文件供链接器 **ld** 连接。
- **ld**: GNU 链接器。被 gcc 调用, 它把目标文件和各种库文件结合在一起, 重定位数据, 并链接符号引用。
- **objcopy**: 执行文件格式转换。
- **objdump**: 显示 ELF 文件的信息。
- **readelf**: 显示更多 ELF 格式文件的信息 (包括 DWARF 调试信息) 。
-

- **有利于程序员优化程序的性能。**譬如：switch vs if-else；函数调用的开销，传参数时传变量还是指针，.....
- **理解并解决编译链接时出现的错误。**譬如：“error: XXXXXX redefined”，“error: cannot find XXXXXX”，.....
- **写出更健壮的程序。**缓冲区溢出，非法访问，“Segmentation fault”，.....

谢 谢

欢迎交流合作