

# AI编译器系列

# GCC编译过程和原理



ZOMI



# About 关于本内容

## 1. 传统编译器

- History of Compiler - 编译器的发展
- **GCC process and principle – GCC 编译过程和原理**
- LLVM/Clang process and principle – LLVM 编译过程和原理

## 2. AI编译器

- History of AI Compiler – AI编译器的发展
- Base Common architecture – AI编译器的通用架构
- Different and challenge of the future – 与传统编译器的区别，未来的挑战与思考

# What is Compiler 编译器是什么

- In computing, a compiler is a computer program that translates computer code written in one programming language into another language. The name "compiler" is primarily used for programs that translate source code from a high-level programming language to a lower level language to create an executable program.



# GNU Compiler Collection, GCC



RMS

Richard M. Stallman



GNU's Not Unix!

GNU



GNU Compiler Collection  
(GNU C Compiler)

# GNU Compiler Collection, GCC

- first released in 1987 by Richard Stallman, GCC 1.0 was named the GNU C Compiler since it only handled the C programming language.
- The GNU Compiler Collection (GCC) is an optimizing compiler produced by the GNU Project supporting various programming languages, hardware architectures and operating systems.

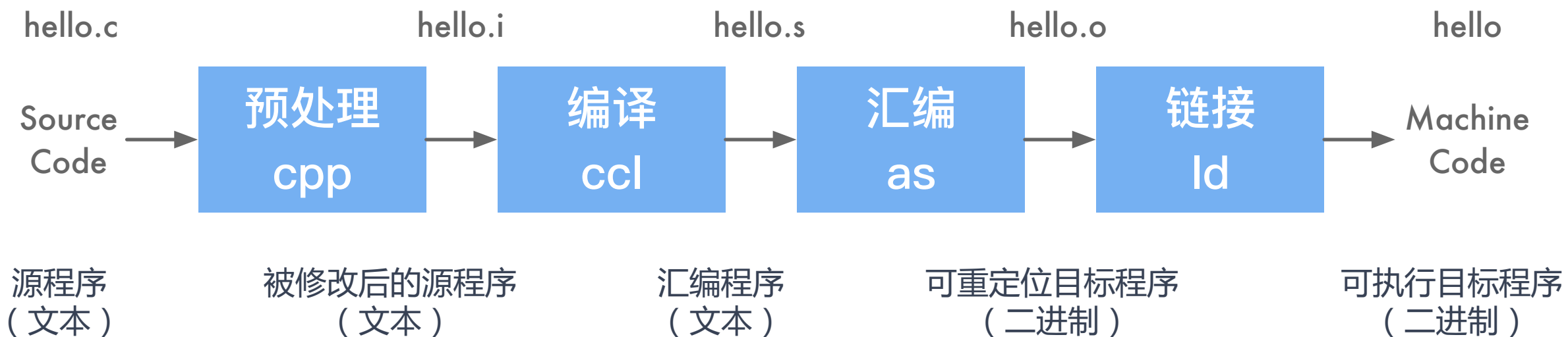


# GCC Main Feature

- GCC是一个可移植的编译器，支持多种硬件平台；
- GCC不仅仅是本地编译器，它还能跨平台交叉编译；
- GCC有多种语言前段，用于解析不同的语言；
- GCC模块化设计，可加入新语言和新CPU架构支持；
- GCC是开源自由软件，可免费使用。

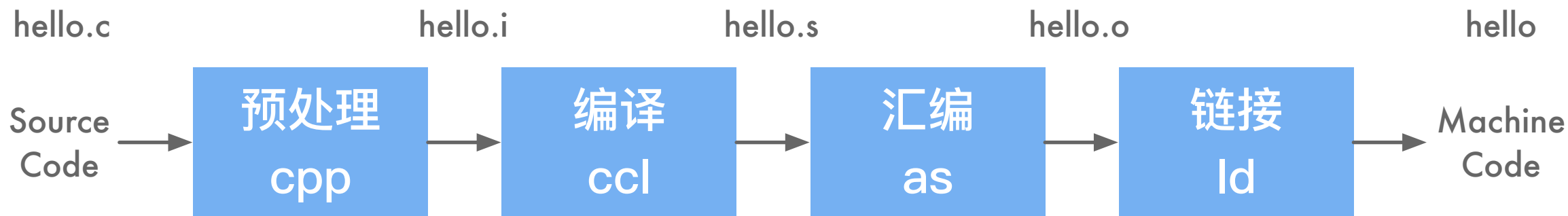


# GCC Compile Process



# GCC Compile Process

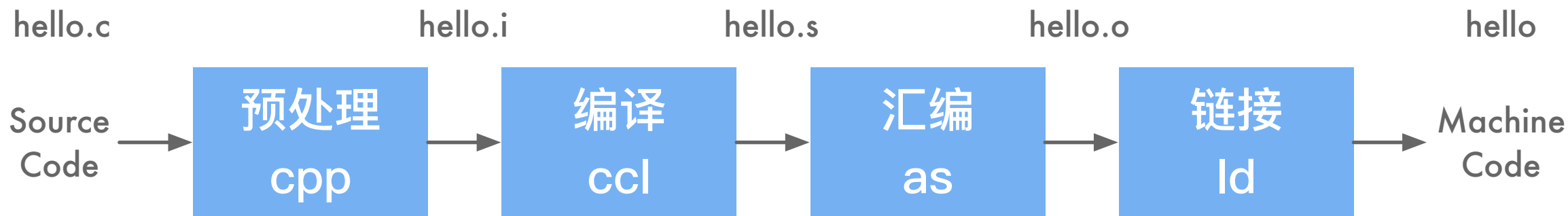
- **预处理 ( Pre-Processing )** : 包括宏定义, 文件包含, 条件编译三部分。预处理过程读入源代码, 检查包含预处理指令的语句和宏定义, 并对其进行响应和替换。预处理过程还会删除程序中的注释和多余空白字符。最后会生成 .i 文件。





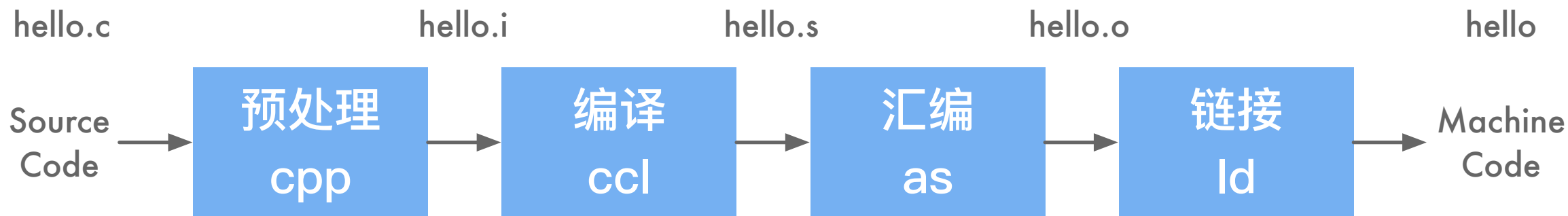
# GCC Compile Process

- **编译器 ( Compiling )** : 编译器会将预处理完的 .i 文件进行一系列的语法分析, 并优化后生成对应的汇编代码。会生成 .s 文件。



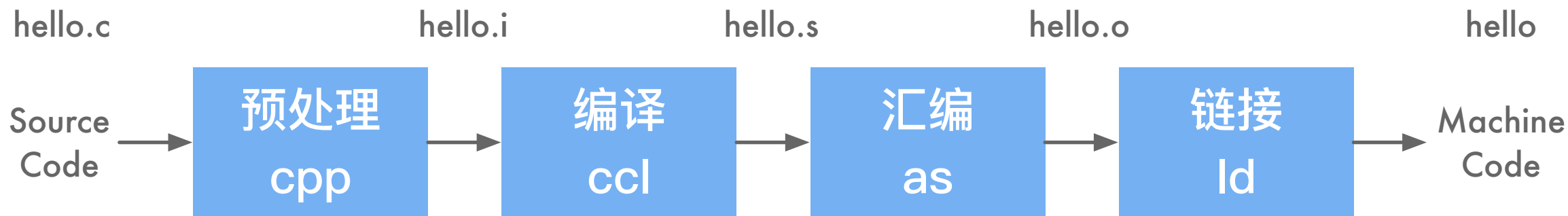
# GCC Compile Process

- **汇编器 ( Assembling )** : 汇编器会将编译器生成的 .s 汇编程序汇编为机器语言或指令，也就是可以机器可以执行的二进制程序。会生成 .o 文件。

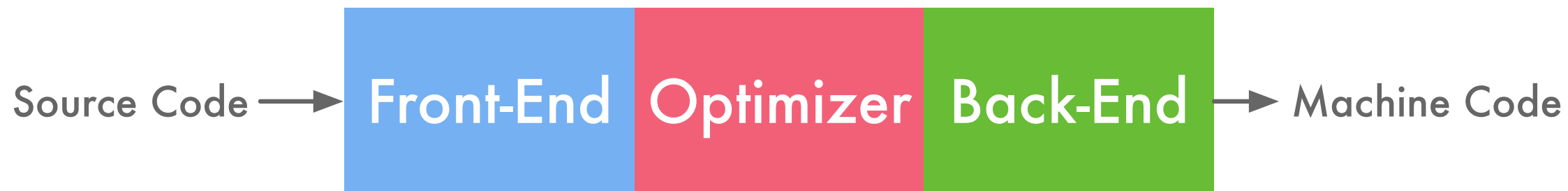


# GCC Compile Process

- **链接器 ( Linking )** : 链接器会来链接程序运行的所需要的目标文件, 以及依赖的库文件, 最后生成可执行文件, 以二进制形式存储在磁盘中。



# Compiler basic constitution 编译器基本构成



# Advantage

1. 支持 JAVA/ADA/FORTRAN
2. GCC 支持更多平台
3. GCC 更流行，广泛使用，支持完备
4. GCC 基于 C，不需要 C++ 编译器即可编译

# Shortcoming

1. GCC 代码耦合度高，很难独立，如集成到专用 IDE 上，模块化方式来调用 GCC 难；
2. GCC 被构建成单一静态编译器，使得难以被作为 API 并集成到其他工具中；
3. 从1987年发展到2022年35年，越是后期的版本，代码质量越差；
4. gcc大约有1500万行代码，是现存最大的自由程序之一；



BUILDING A BETTER CONNECTED WORLD

THANK YOU

Copyright©2014 Huawei Technologies Co., Ltd. All Rights Reserved.

The information in this document may contain predictive statements including, without limitation, statements regarding the future financial and operating results, future product portfolio, new technology, etc. There are a number of factors that could cause actual results and developments to differ materially from those expressed or implied in the predictive statements. Therefore, such information is provided for reference purpose only and constitutes neither an offer nor an acceptance. Huawei may change the information at any time without notice.