赛题 79

编译器的验证测试移植到 Compass-CI 平台

直播导师：董长春

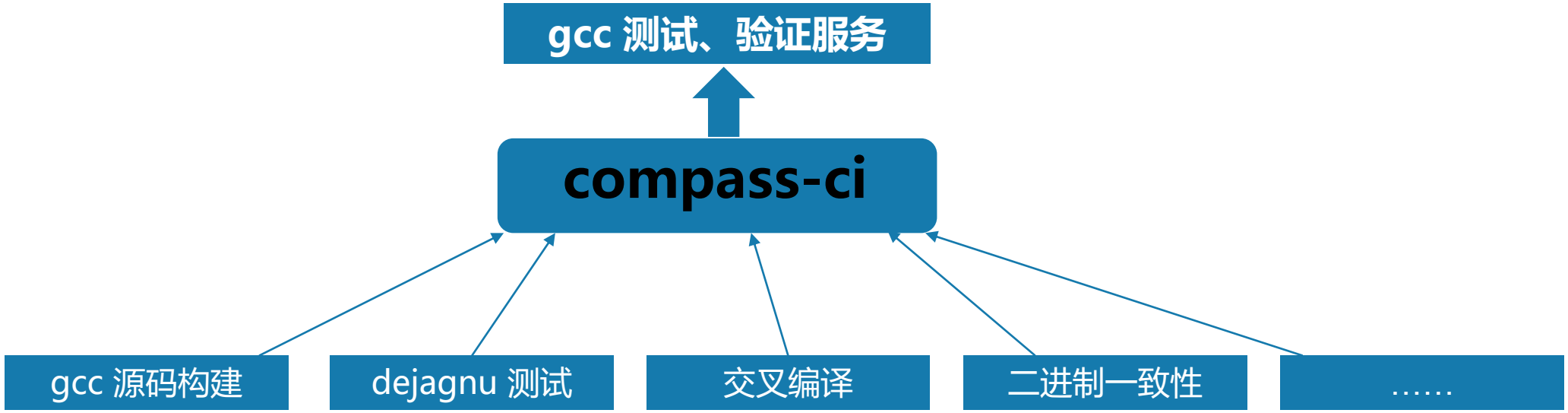
openEuler QA-sig 成员

项目背景

- 项目名称：编译器的验证测试移植到 compass-ci 平台
- 应用场景：实现对不同版本的 gcc 的自动化测试，例如 gcc 项目在 compass-ci 平台的注册、自举、dejagnu 的自动化测试等
- 技术价值：通过 compass-ci 打造一套面向开发提供调测服务的平台，为 gcc 开发者提供 gcc 测试、验证方面的便利

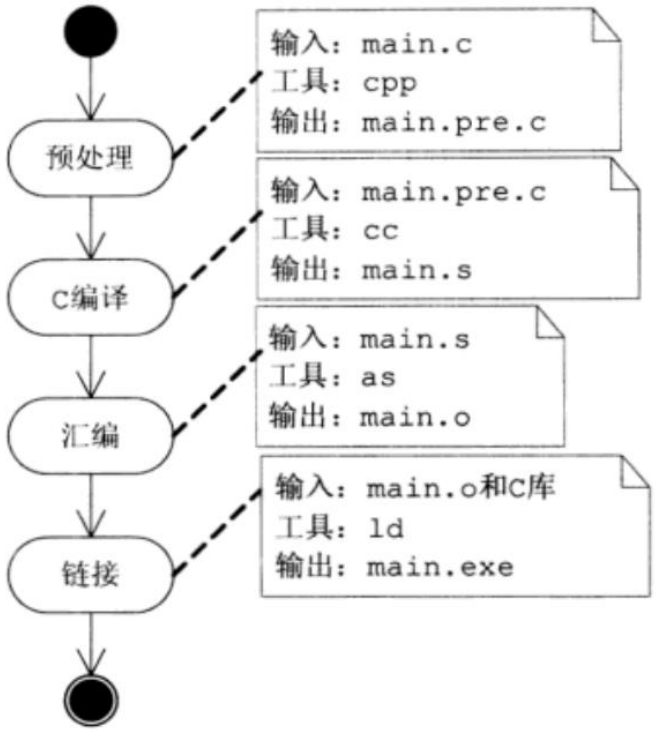
题目介绍

- 通过 compass-ci 打造一套面向开发提供调测服务的平台，实现为 gcc 开发者提供 gcc 自动化测试、验证服务



技术要求（1）

- 掌握 gcc 调测的基础知识，具备一定 gcc 编译、调优方面的经验
- 建议先学习掌握 gcc 编译四个阶段：预处理、编译、汇编、链接，以及相关的常用编译选项



gcc实用选项：

-S：生成汇编

-I：指定头文件路径

-g：包含调试信息

-On：n=0～3，编译器优化，n越大优化等级越高

-Wall：提示更多的警告信息

-D：编译器时定义宏，注意-D和宏之间没有空格

-E：生成预处理文件

-c:只编译，不链接

-C *告诉预处理器不要丢弃注释。配合`-E'选项使用。*

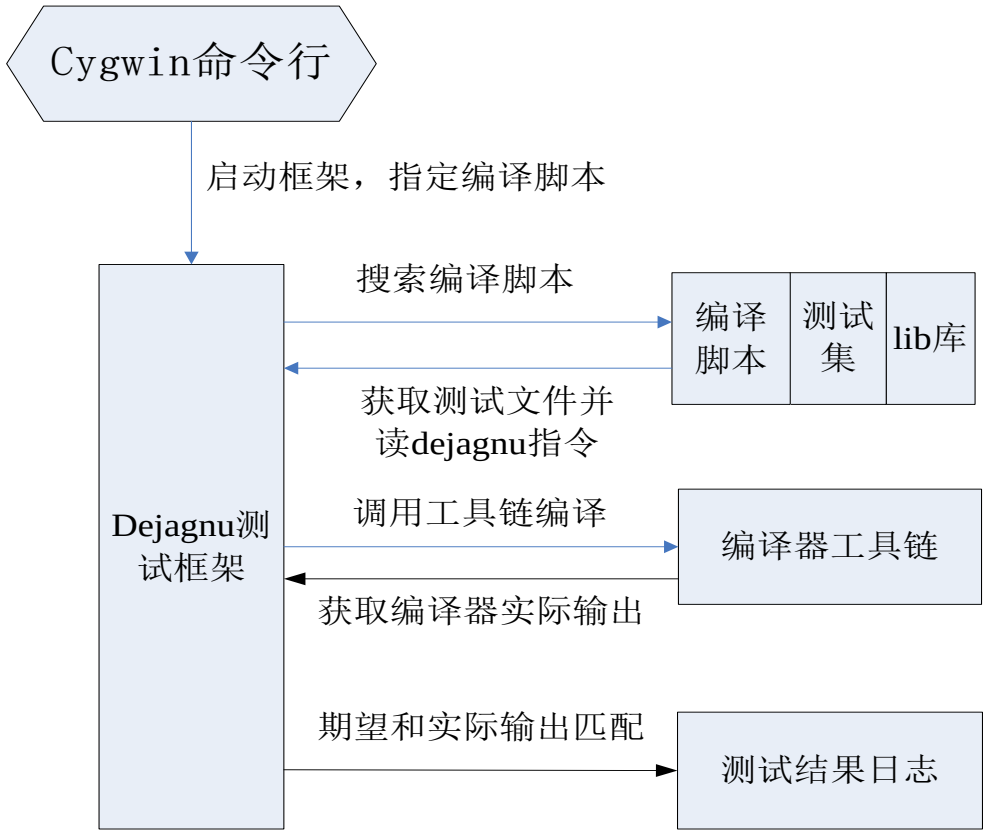
-M：生成与.c和头文件的依赖关系以用于Makefile，包括系统库的头文件

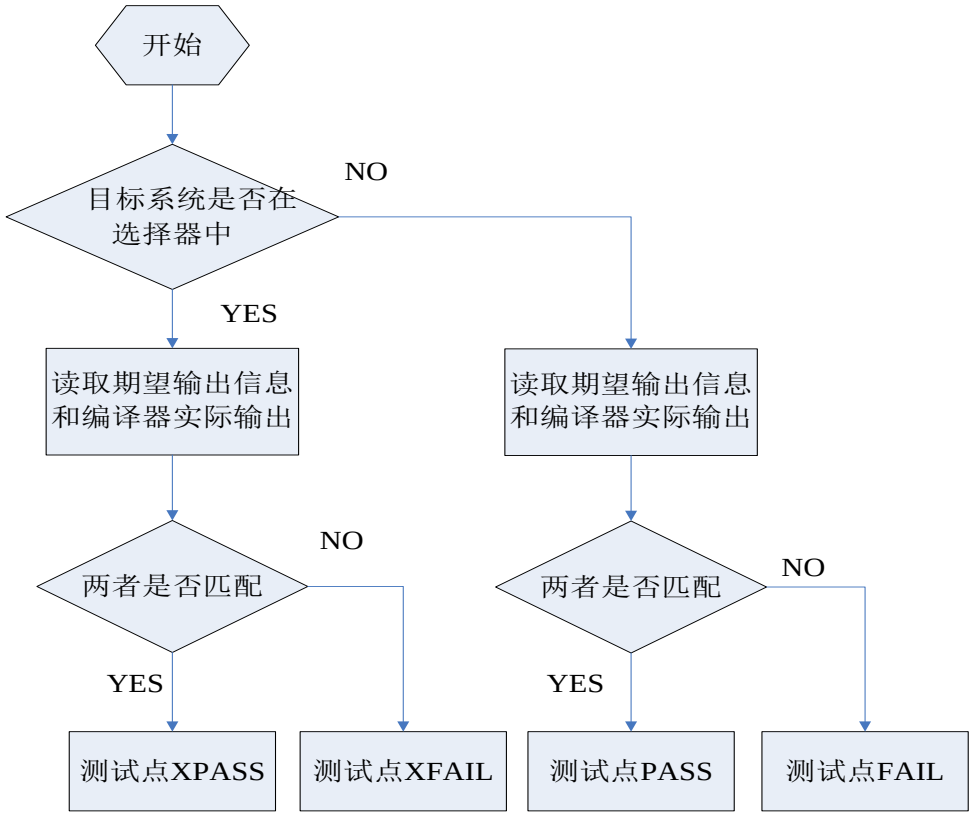
-MM：生成与.c和头文件的依赖关系以用于Makefile，不包括系统库的头文件

-o：生成目标文件。eg： -o filename。将经过gcc处理的结果保存为filename。如果这个选项被忽略，生成的可执行文件默认为a.out，假设源文件为file.c，目标文件默认为file.o，汇编文件默认为file.s。

技术要求（2）

- 掌握dejanu常见测试方法
- 可参考学习 gcc 社区的 dejagnu 及相关用例调用、结果分析方法





技术要求（3）

- 掌握 compass-ci 的基本技术架构及使用，可参考学习：https://gitee.com/openeuler/compass-ci
- 了解其余常见的编译器测试方法：自举、交叉编译、汇编一致性等

自举

编译器编译自己的源码构建出二进制，对于新生成的编译器，能够跑通全量用例

交叉编译

在 gcc 编译时，指定主机与目标机处理器型号不一致（比如目标机是 ARM 处理器，而主机是 X86 处理器），则生成的编译器就是交叉编译器（cross compiler），即生成的编译器将在指定主机上进行编译活动，但在编译器生成的程序却运行于目标机

汇编一致性

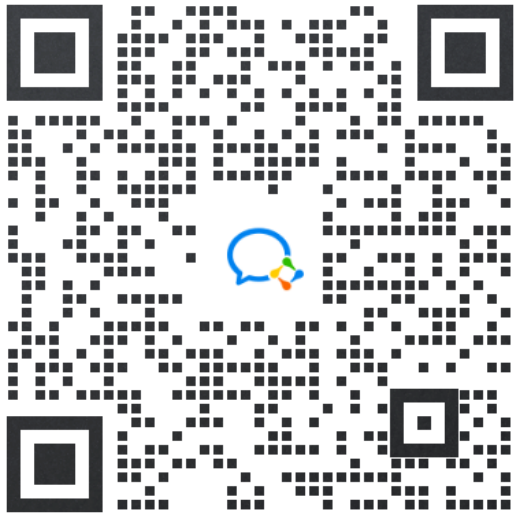
在编译器不变的情况下，确保编译器前后两次编译的二进制代码行为一致，不同操作系统下编译器出来的二进制代码一致，主要方法是编译生成的汇编代码进行对比，需要严格一致

基本功能 & DEMO 设计

- 基本功能：实现在 compass-ci 平台上以任务提交的方式进行对 gcc 源码的 dejanu 测试，并反馈正确的测试结果

- 此外，最好能够实现：用户能够根据需求，选择不同的测试方法以及在一种方法中加入个性化的测试因子（例如用户可以添加自己设计的编译优化选项或测试用例等）

DEMO 设计（可参考）：



赛事交流群