

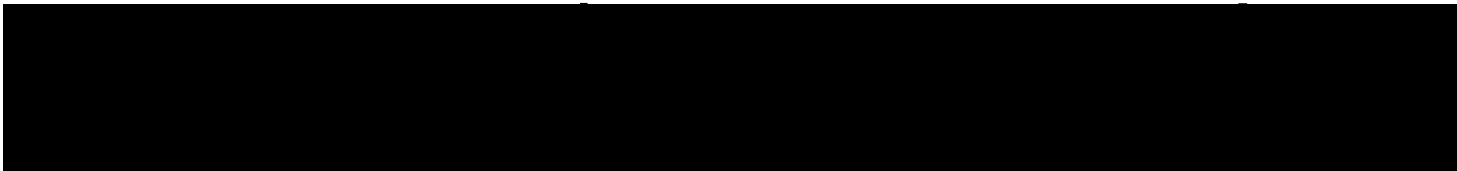
GCC它干什么了事

——《A Introduction to GCC》

PLCT-gnu-toolchain: 史玉龙
邮箱: yulong@nj.iscas.ac.cn

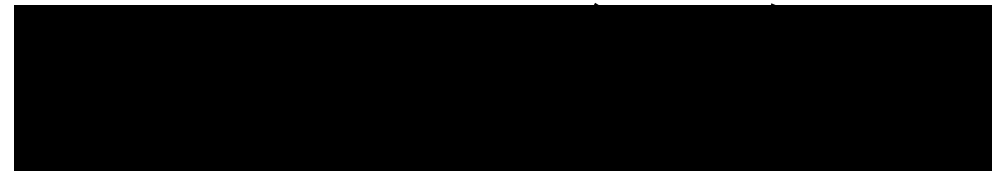
- GCC是个啥?
- GCC干了啥?
- GCC基本编译器选项

·GCC是个啥？

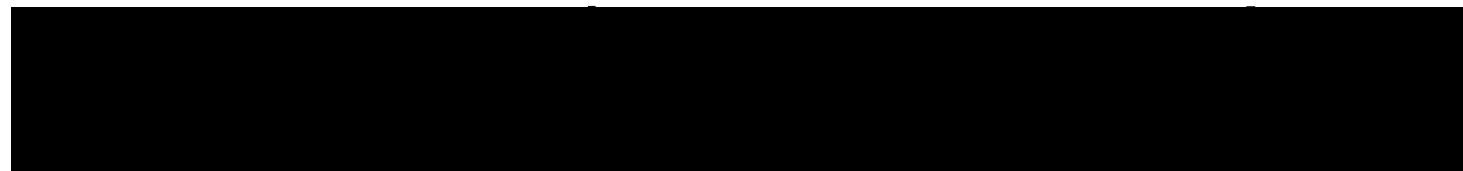


GCC是个啥？

- 他就是个锤子(我想说它其实是个工具，GNU项目开发的自由软件之一)，用来加工源代码，变成机器能够认识的机器码；
- 1987年发行的第一版本只支持C语言的编译；1992年的版本加入了C++的支持；后陆续支持了更多高级语言的编译。如Fortran, ADA, Java等。
- 能够在大部分平台上运行，能为许多类型的CPU提供可执行文件
- 除了个人PC机，还可以在微控制器，DSP和64位CPU上运行
- 跨平台编译器，既可以本地编译，也可以编译在其他平台上运行的可执行文件
- 模块化设计，可以加入对新语言和新CPU架构的支持。
- 自由软件，即使用自由，修改自由，发行自由。



·GCC干了啥？

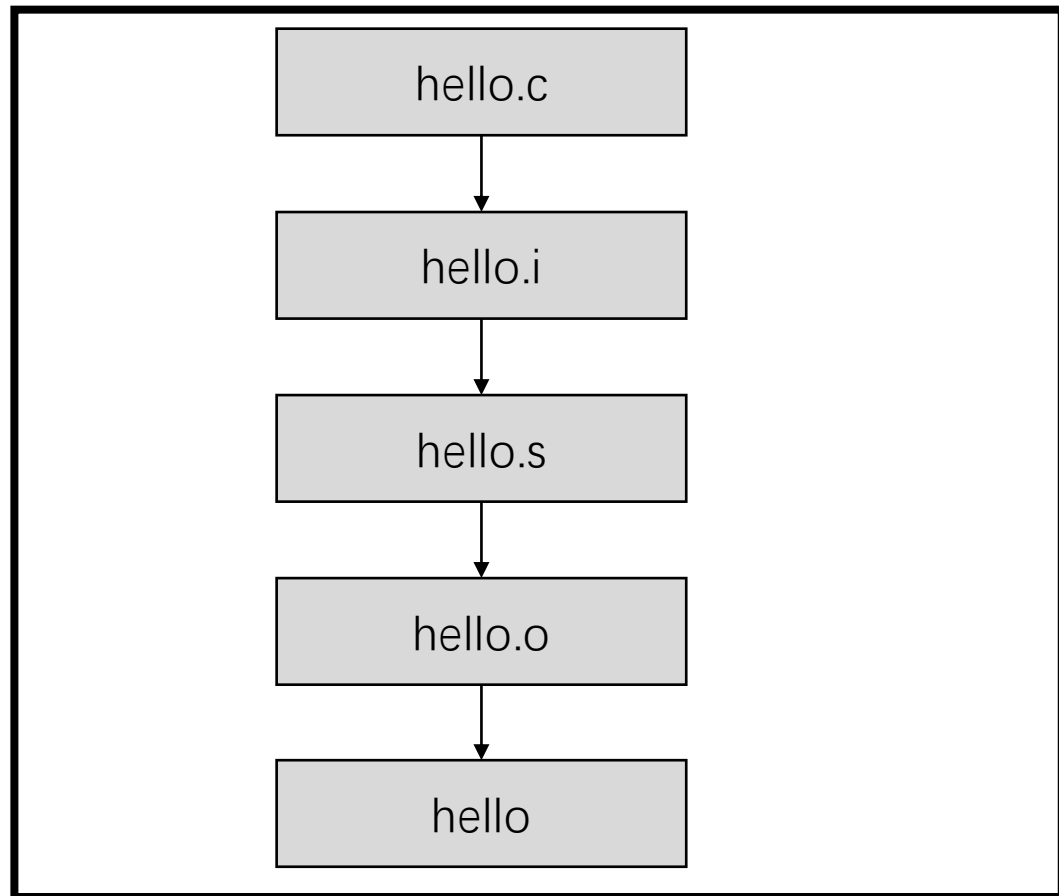


GCC干了啥？(以C语言为例)

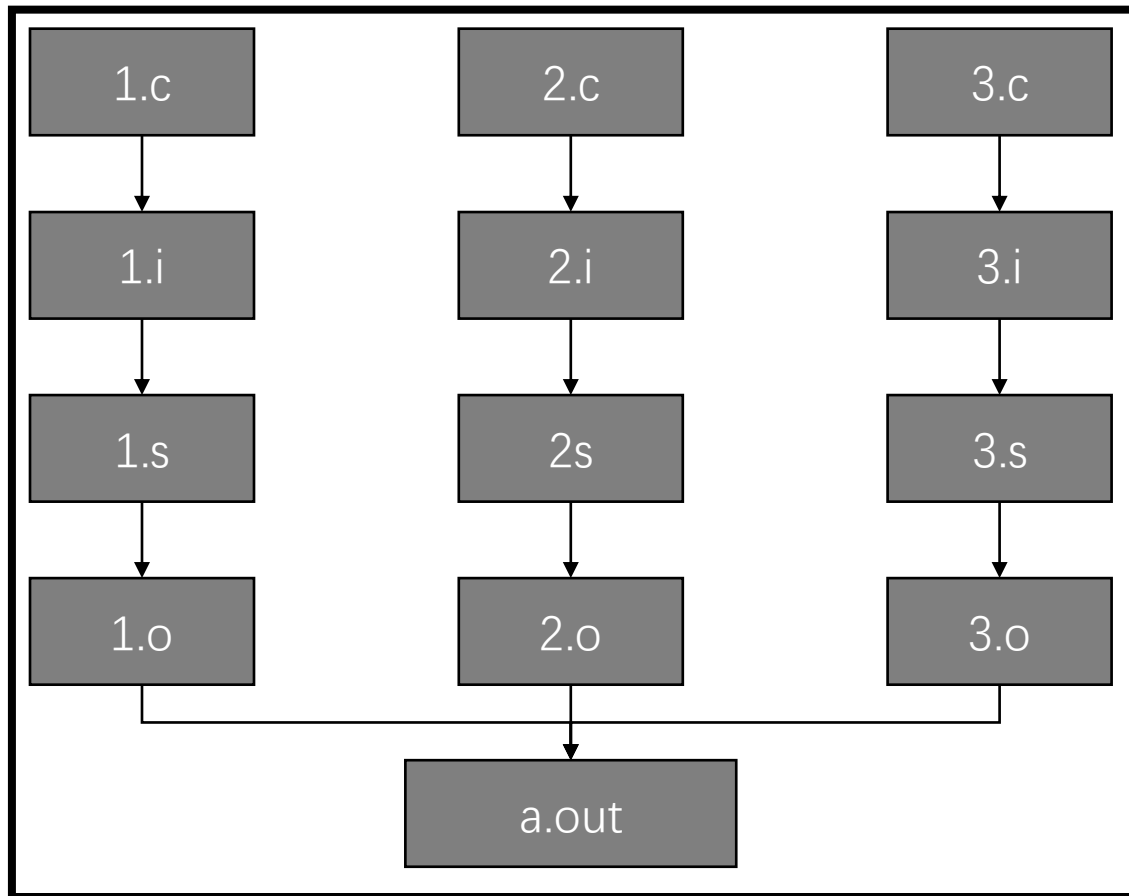
- 将高级语言翻译成机器能够识别的机器语言。
- 执行过程为：预处理→编译→汇编→链接。
- 预处理：预处理器来扩展宏和被包含的头文件，变成纯正的C代码；
- 编译：编译器将预处理后的源代码翻译成特定处理器的汇编语言；
- 汇编：汇编器将汇编语言转变成机器码并生成对象文件(obj文件)；
- 链接：连接器将所有的对象文件链接生成可执行文件，并补全汇编时保留的外部函数或变量引用的地址。

GCC干了啥? (以C语言为例)

·编译单个源文件



·编译多个源文件



预处理

- 预处理的过程是将.c源文件中的宏定义，include等内容全部展开。但是字符串中的宏不会替换展开
- 执行cpp hello.c > hello.i命令得到的就是包含所有宏被替换以及include后的C代码

编译

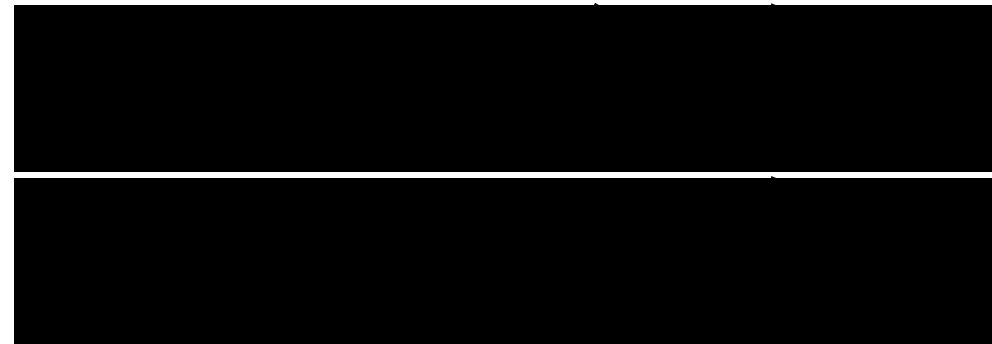
- 将源代码转换成汇编代码，不生成对象文件
- 执行`gcc -S hello.i`即可得到包含汇编代码的.s文件

汇编

- 将汇编语言转换成机器码，并生成对象文件(obj文件)，如果汇编文件中有对外部函数或变量的引用，汇编器保留外部函数的地址处于未定义状态，有连接器填写。
- 执行gcc hello.s -o hello.o命令得到hello.o对应的对象文件文件，这里包含所有机器码

链接

- 将所有的对象文件链接生成可执行文件，即可在对应cpu架构的系统上运行。否则会报错：
- 执行gcc hello.o即可得到默认名为a.out的可执行文件；同样可以通过-o选项指明输出到具体的文件，如gcc hello.o -o hello



·GCC基本编译器选项

GCC基本编译器选项

- 基本编译器选项的合理使用可以帮助我们在开发时优化和调试我们的代码；也可以有利于报错问题的定位解决。常见的选项有以下几种：
- -Wall：开启常用的编译告警功能，能够捕捉到许多编程中的常见错误，如输出格式符。
- -E：预处理
- -S：编译为汇编代码
- -c：编译为目标文件，但不链接
- -o：输出内容到指定文件，若该文件已存在，则覆盖里面的内容，默认输出到a.out
- -g：产生供GDB调试用的可执行文件内，加入调试信息

GCC基本编译器选项

- -I和-L（都大写） 设置头文件和库的搜索路径(编译报错FILE.h:No such file or directory时用此加入路径)

默认情况是在/usr/local/include/和/usr/include中搜索头文件，在/usr/local/lib和/usr/lib中搜索库，编译时加上该选项，如：gcc -Wall -I/opt/syl/include -L/opt/syl/lib xxxx.c（当然你也可以设置shell环境变量来实现）

- -O：编译优化，默认不优化即-O0，-O1，-O2，-O3分别是优化的等级，-Os可缩减可执行文件的大小
- -Dname：在命令行上定义宏（你也可以在.c文件中直接定义）