

GNU C Compiler & Makefile

2018 年 10 月 23 日

0.1 编译过程简介

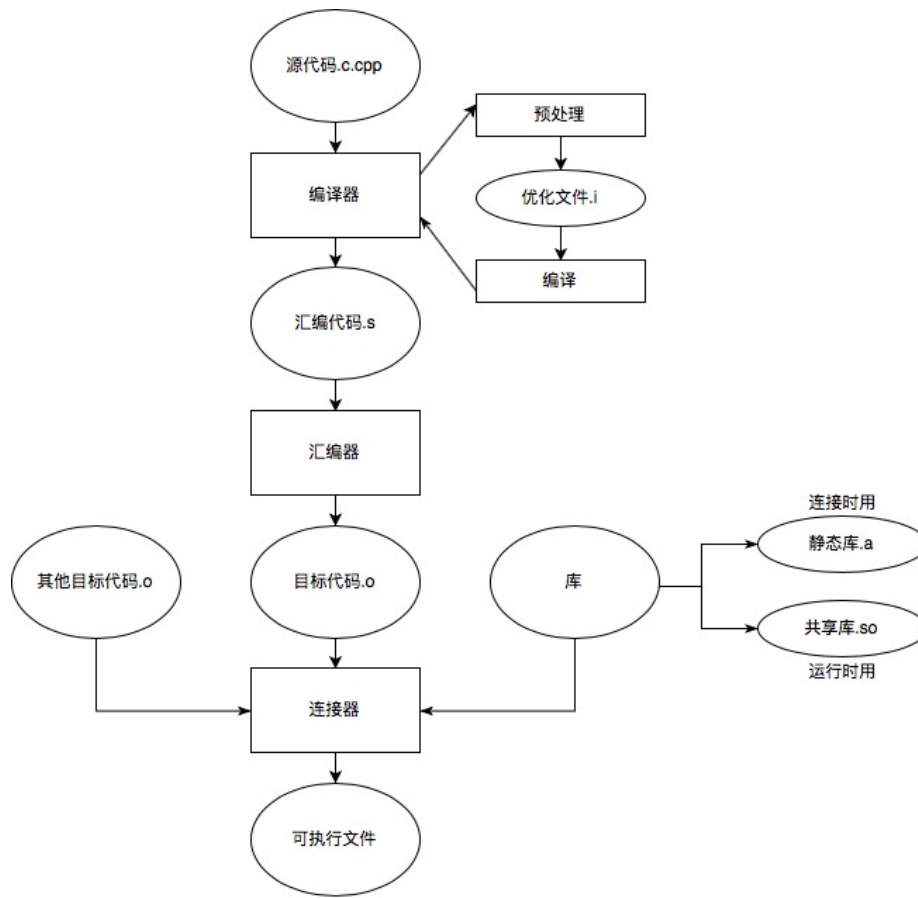


图 1: 编译基本过程

0.2 GNU C Compiler(gcc/g++)

命令格式:

gcc/g++ [command] [filename] [command] [filename]

其中 command 可以是:

- c (compiler): 编译, 可以将源代码编译成目标代码, 或将汇编代码编译到目标代码.

- o (object file): 指示目标文件.
- E: 只生成预处理结果
- S: 只生成汇编代码
- Wall: 显示警告
- I: 指示 include 文件
- L: 指示 lib 文件
- l: 指示依赖项.

0.3 关于 Makefile

文件格式:

执行文件: 目标文件 1.o 目标文件 2.o # 指示依赖关系

tab¹ gcc 目标文件 1.o 目标文件 2.o -o 执行文件 # 指示具体操作

目标文件 1.o : 源文件 1.c

tab gcc -c 源文件 1.c -o 目标文件 1.o

... # 其他目标文件

...

clean:

rm -rf *.o 执行文件 # 删除目标文件 (.o) 删除执行文件

... # 删除其他中间文件

有用的函数:

1. wildcard: 通配符:SOURCE = \$(wildcard *.c *.cpp), 其中 SOURCE 为所有.c.cpp 文件的列表.* 为通配符. 参数之间用空格隔开.
2. patsubst: 替换
 - 1.OBJECTS = \$(patsubst %.c,%.o,\$(SOURCE)), 将所有.c 文件换成.o 文件.% 用于匹配字符,%.c 代表所有.c 结尾的文件.
 - 2.OBJECTS = \$(patsubst %.c,%.o,\$(patsubst,%.cpp,%.o,\$(SOURCE))),

¹'tab' 表示这里需要一个制表符

替换所有.c.cpp 文件.

特殊符号:

1. %: ” 在依赖文件.o: 和源文件.c” 处, 可用%.o:%.c 代替.
2. \$@ 表示目标文件集。
3. \$< 依赖目标中的第一个目标名字。如果依赖目标是以模式（即”%”）定义的，那么”\$<” 将是符合模式的一系列的文件集。注意，其是一个一个取出来。所有比目标新的依赖目标的集合。以空格分隔。
4. \$ 所 有的依赖目标的集合。以空格分隔。如果在依赖目标中有多个重复的，那个这个变量会去除重复的依赖目标，只保留一份。