链滴

GCC 编译过程

作者：someone756
原文链接：https://ld246.com/article/1445656550072
来源网站：链滴

<h4>GCC所支持后缀名</h4>
<table border="1" cellspacing="0" cellpadding="0">
<tbody>
<tr><th width="85">后 缀 名</th><th valign="top" width="230">所对应的语言</th><th width="85">后 缀 名</th><th valign="top" width="173">所对应的语言</th></tr>
<tr>
<td width="77">
.c
</td>
<td valign="top" width="181">
C原始程序
</td>
<td width="85">
.s/.S
</td>
<td valign="top" width="173">
汇编语言原始程序
</td>
</tr>
<tr>
<td width="77">
.C/.cc/.cxx
</td>
<td valign="top" width="181">
C++原始程序
</td>
<td width="85">
.h
</td>
<td valign="top" width="173">
预处理文件（头文件）
</td>
</tr>
<tr>
<td width="77">
.m
</td>
<td valign="top" width="181">
Objective-C原始程序
</td>
<td width="85">
.o
</td>
<td valign="top" width="173">
目标文件
</td>
</tr>
<tr>
<td width="77">
.i
</td>
<td valign="top" width="181">
已经过预处理的C原始程序
</td>

<td width="93">
.a/.so
</td>
<td valign="top" width="173">
编译后的库文件
</td>
</tr>
<tr>
<td width="77">
.ii
</td>
<td valign="top" width="181">
已经过预处理的C++原始程序
</td>
<td valign="top" width="85"> </td>
<td valign="top" width="173"> </td>
</tr>
</tbody>
</table>
<h4>编译过程</h4>
GCC的编译过程分为四个步骤，分别为：

预处理（Pre-Processing）
编译（Compiling）
汇编（Assembling）
链接（Linking）

（1）预处理阶段
在该阶段，编译器将上述代码中的stdio.h编译进来，可以以下命令进行编译
<blockquote>
gcc -E hello.c -o hello.i
</blockquote>
编译后的文件后缀是.i ，此时文件很大，因为stdio.h被编进来的原因。
（2）编译阶段
在这个阶段，gcc首先要检查代码的规范性、语法是否有等，以确定代码实际要做的工作，检查无误后将.i文件翻译成汇编语言，可以用以下命令进行编译</spn>
<blockquote>
gcc -S hello.i -o hello.s
</blockquote>
编译后的文件后缀是.s。
（3）汇编阶段
汇编阶段就是把编译阶段生成的.s文件转成目标文件，可用以下命令进行编译
<blockquote>
gcc -c hello.s -o hello.o
</blockquote>
（4）链接阶段
在成功编译之后，就进入了链接阶段。在这里涉及到一个要的概念：函数库。
现在我们可以重新查看这个小程序，在这个程序中并没有义”printf”的函数实现，且在预编译中包含进的”stdio.h”中也只有该

数的声明，而没有定义函数的实现，那么，是在哪里实现”printf”函数的呢？最后的案是：系统把这些函数实现都被做到名为libc.so.6的库文件中去了，在没有特别指定时，Gcc会到系默认的搜索路径”/usr/lib”下进行查找，也就是链接到libc.so.6库函数中去，这样就能现函数”printf”了，而这也就是链接的作用。
函数库一般分为静态库和动态库两种。静态库是指编译链时，把库文件的代码全部加入到可执行文件中，因此生成的文件比较大，但在运行时也就不再需要库件了。其后缀名一般为”.a”。动态库与之相反，在编译链接时并没有把库文件的代码入到可执行文件中，而是在程序执行时由运行时链接文件加载库，这样可以节省系统的开销。动态库般后缀名为”.so”，如前面所述的libc.so.6就是动态库。Gcc在编译时默认使用动态库
可以用以下命令进行链接
<blockquote>
gcc hello.o –o hello
</blockquote>
（5）运行
第五步运行就行了。
<blockquote>
./hello
</blockquote>

原文链接：GCC 编译过程