

6.3 gcc或者g++

2021年9月13日 16:29

鹏哥c语言有讲到 预处理等的一个过程

code.c -> bin(可执行程序) -> 程序的翻译(编译)

翻译:

1. 预处理 (a. 头文件展开 b. 去注释 c. 宏替换 d. 条件编译)
2. 编译 (把C变成汇编语言)
3. 汇编 (汇编变成.o目标二进制文件, 二进制不可执行!)
4. 链接 (本质引入我们在代码中使用的第三方库, c库)

\$ gcc -E test.c -o test.i #第一步走完, 停下! 临时内容写入.i文件
\$ gcc -S test.i -o test.s #第二步走完, 停下! 临时内容写入.s文件
\$ gcc -c test.s -o test.o #第三步走完, 停下! 临时内容写入.o文件
\$ gcc test.o -o mybin #gcc会根据文件, 进行默认连接(编译器和文件共同决定)

libc-2.17.so : 动态库
libc.a : 静态库
库真正名字, 是去掉前缀lib, 去掉.XXXX, 剩下的就是库名字!!!

-E -S -c -o ESc -o
test.i test.s test.o .iso #Centos.XXXX.iso, 镜像文件

多文件情况, 最好是将多文件先统一变编译.o, *o链接生成bin

gcc选项

- -E 只激活预处理, 这个不生成文件, 你需要把它重定向到一个输出文件里面
- -S 编译到汇编语言不进行汇编和链接
- -c 编译到目标代码
- -o 文件输出到 文件
- -static 此选项对生成的文件采用静态链接
- -g 生成调试信息。GNU 调试器可利用该信息。
- -shared 此选项将尽量使用动态库, 所以生成文件比较小, 但是需要系统由动态库。
- -O0
- -O1
- -O2

- -O3 编译器的优化选项的4个级别, -O0表示没有优化, -O1为缺省值, -O3优化级别最高
- -w 不生成任何警告信息。
- -Wall 生成所有警告信息。

gcc选项记忆

- esc, iso例子

7]\$ gcc -E test.c -o test.i

这一句的意思是把gcc做完预处理 -o重命名为test.i

多文件情况

多文件情况, 最好是将多文件先统一变编译.o, *o链接生成bin

make/makefile

解决文件多的情况

在这里涉及到一个重要的概念:函数库

- 我们的C程序中,并没有定义“printf”的函数实现,且在预编译中包含的“stdio.h”中也只有该函数的声明,而没有定义函数的实现,那么,是在哪里实现“printf”函数的呢?
- 最后的答案是: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 会到系统默认的搜索路径“/usr/lib”下进行查找,也就是链接到 libc.so.6 库函数中去,这样就能实现函数“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。

- 静态库是指编译链接时,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文件了。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加入到可执行文件中,而是在程序运行时由运行时链接文件加载库,这样可以节省系统的开销。动态库一般后缀名为“.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。完成了链接之后,gcc 就可以生成可执行文件,如下所示。`gcc hello.o -o hello`
- gcc默认生成的二进制程序,是动态链接的,这点可以通过 file 命令验证。