GNU C/C++、Fortran编译器的使用

河南师范大学 高性能计算中心

2015-12-5

GNU C/C++、Fortran编译器简介

GNU C/C++ Fortran编译器（GCC）为系统自带的默认编译器，用户无需特殊设置即可使用

- 编译C、C++源程序的命令：分别为gcc和g++
- 编译Fortran 77和9x[1]、200x[2]源程序的命令：分别为g77和gfortran：
  - gfortran：属于GCC 4系列，可以编译Fortran 77及9x、200x源程序
  - g77：属于GCC 3.4系列，不可编译Fortran 9x、200x源程序

[1]指的为Fortran 90、95标准

[2]指的是Fortran 2003和2008标准

编译命令基本格式

基本格式:

- C: *gcc␣[options]␣file1␣[file2␣...]*
- C++: *g++␣[options]␣file1␣[file2␣...]*

注意:

- *[␣]*表示是其内部的选项可选
- 文件名和选项区分大小写

输入文件后缀与类型的关系

编译器默认将按照输入文件的后缀判断文件类型，编译时也可以用编译选项强制指定。

文件名	解释	动作
filename.c	C源文件	传给编译器
filename.C filename.CC filename.cc filename.cpp filename.cxx	C++源文件	传给编译器
filename.a filename.so	库文件	传递给链接器
filename.i	已预处理的文件	传递给标准输出
filename.o	目标文件	传递给链接器
filename.s	汇编文件	传递给汇编器

输出文件后缀与类型的关系

编译器默认将输出按照文件类型与后缀相对应。

文件名	解释
filename.i	已预处理的文件，由使用-p选项生成
filename.o	目标文件，由添加-c选项生成
filename.s	汇编文件，由添加-s选项生成
a.out	默认生成的可执行文件

重要编译选项

- 编译选项对运行速度、编译的兼容性等有影响
- 建议仔细看看编译器手册，多加测试，选择适合自己程序的编译选项以提高性能
- 以下仅仅介绍部分重要选项

控制文件类型的选项 I

- -x language：明确指定而非让编译器判断输入文件的类型。language可为：
  - c、c-header、c-cpp-output
  - c++、c++-header、c++-cpp-output
  - objective-c、objective-c-header、objective-c-cpp-output
  - objective-c++ objective-c++-header objective-c++-cpp-output
  - assembler、assembler-with-cpp
  - ada
  - f95、f95-cpp-input
  - java
  - treelang

  当language为none时，禁止任何明确指定的类型，其类型由文件名后缀决定
- -c：仅编译成目标文件（.o文件），并不进行链接
- -o file：指定生成的文件名

控制文件类型的选项 II

- -v：详细模式，显示在每个命令执行前显示其命令行
- -###：显示编译器、汇编器、链接器的调用信息但并不进行实际编译，在脚本中可以用于捕获驱动器生成的命令行
- –help：显示帮助信息
- –target-help：显示目标平台的帮助信息
- –version：显示编译器版本信息

C语言选项

- -ansi：C模式时，支持所有ISO C90指令。在C++模式时，去除与ISO C++冲突的GNU扩展
- -std=standard：控制语言标准，standard可为c89、iso9899:1990、iso9899:199409、c99、c9x、iso9899:1999、iso9899:199x、gnu89、gnu99、gnu9x、c++98、gnu++98

警告选项

- -fsyntax-only：仅仅检查代码的语法错误，并不进行其它操作
- -w：编译时不显示任何警告，只显示错误
- -Wfatal-errors：遇到第一个错误就停止，而不尝试继续运行显示更多错误信息

调试选项

- -g：包含调试信息
- -ggdb：包含利用gbd调试时所需要的信息

优化选项

- -O[level]：设置优化级别。优化级别level可以设置为0、1、2、3、s，默认为-O0
- -Ofast：设置优化级别，相当于-O3

预处理选项

- -C：预处理时保留C源文件中的注释
- -D name：预处理时定义宏name的值为1
- -D name=def：预处理时定义name为def
- -U name：预处理时去除的任何name初始定义
- -undef：不预定义系统或GCC声明的宏，但标准预定义的宏仍被定义
- -dD：显示源文件中定义的宏及其值到标准输出
- -dI：显示预处理中包含的所有文件，包括文件名和定义时的行号信息
- -dM：显示预处理时源文件中定义的宏及其值，含定义时文件名和行号
- -dN：与-dD类似，但只显示源文件中定义的宏，而不显示宏值
- -E：预处理.c文件，将结果发给标准输出，不进行编译、汇编或链接
- -I<头文件目录>：指明头文件的搜索路径
- -M：打印make的依赖关系到标准输出
- -MD：打印make的依赖关系到文件file.d，file是编译文件的根名字
- -MM：打印make的依赖关系到标准输出，但忽略系统头文件
- -MMD：打印make的依赖关系到文件file.d，其中file是编译的文件的根名字，但忽略系统头文件
- -P：预处理每个文件，并保留每个file.c文件预处理后的结果到file.i

链接选项

- -pie：在支持的目标上生成位置无关的可执行文件
- -s：从可执行文件中去除所有符号表
- -rdynamic：添加所有符号表到动态符号表中
- -static：静态链接所需的库
- -shared：生成共享目标而不是可执行文件，必须在编译每个目标文件时使用-fpic选项
- -shared-libgcc：使用共享libgcc库
- -static-libgcc：使用静态libgcc库
- -u <symbol>：确保symbol未定义，强制链接一个库模块来定义它
- -I<头文件目录>：指明头文件的搜索路径
- -l<库文件>：指明所链接的库名，如库为libxyz.a，可用-lxyz指定
- -L<库目录>：指明库的搜索路径
- -B<路径>：设置寻找可执行文件、库、头文件、数据文件等路径

CPU平台相关选项

- -mtune=cpu-type：设置优化针对的CPU类型，可为：generic、core2、corei7、corei7-avx、core-avx-i、core-avx2、k8、opteron等
- -march=cpu-type：设置指令针对的CPU类型，可为：generic、core2、corei7、corei7-avx、core-avx-i、core-avx2、k8、opteron等
- -mieee-fp和-mno-ieee-fp：浮点操作是否严格按照IEEE标准

约定成俗的选项

- -fpic：生成位置无关的代码以用于共享库
- -fPIC：如果目标机器支持，将生成位置无关的代码
- -fopenmp：编译OpenMP并行程序
- -fpie和-fPIE：与-fpic和-fPIC类似，但生成的位置无关代码只能链接到可执行文件中

串行及OpenMP并行程序编译举例

- 将C程序yourprog.c编译为可执行文件yourprog:
  gcc␣-o␣yourprog␣yourprog.c
- 将C程序yourprog.c编译为目标文件yourprog.o:
  gcc␣-c␣yourprog.c
- 将使用lapack库的C程序yourprog.c编译为可执行文件yourprog:
  gcc␣-o␣yourprog␣-L/opt/lib␣-llapack␣yourprog.c
- 将C程序yourprog.c静态编译为O3优化的可执行文件yourprog:
  gcc␣-O3␣-static␣-o␣yourprog␣yourprog.c
- 将C++程序yourprog.cpp编译为可执行文件yourprog:
  g++␣-o␣yourprog␣yourprog.cpp
- 将OpenMP指令并行的C程序yourprog-omp.c编译为可执行文件yourprog-omp:
  gcc␣-o␣yourprog-omp␣-fopenmp␣yourprog.c

编译时出错信息格式

```
netlog.c: In function 'main':
netlog.c:84:7: error: 'for' loop initial declarations are only allowed in C99 mode
netlog.c:84:7: note: use option -std=c99 or -std=gnu99 to compile your code
```

- 源文件名: 函数中
- 源文件名:行数:列数:错误类型:具体说明
- 源文件名:行数:列数:注解:解决办法

编译命令基本格式

基本格式：

- 4.x.y版本编译器：*gfortran␣[options]␣file1␣[file2␣...]*
- 3.x.y版本编译器：*g77␣[options]␣file1␣[file2␣...]*

注意：

- *[␣]*表示是其内部的选项可选
- 文件名和选项区分大小写
- 后续内容为关于4.x.y版本gfortran的，3.x.y版本的g77有所不同

输入文件后缀与类型的关系

编译器默认将按照输入文件的后缀判断文件类型，编译时也可以用编译选项强制指定。

文件名	解释	动作
filename.a	目标库文件	传给编译器
filename.f filename.for filename.ftn filename.i	固定格式的Fortran源文件	被Fortran编译器编译
filename.fpp filename.FPP filename.F filename.FOR filename.FTN	固定格式的Fortran源文件	自动被Fortran编译器预处理后再被编译
filename.f90 filename.f95 filename.f03 filename.f08 filename.i90 filename.i95 filename.i03 filename.i08	自由格式的Fortran源文件	被Fortran编译器编译
filename.F90 filename.F95 filename.F03 filename.F08	自由格式的Fortran源文件	自动被Fortran编译器预处理后再被编译
filename.s	汇编文件	传递给汇编器
filename.so	库文件	传递给链接器
filename.o	目标文件	传递给链接器

输出文件的后缀与类型的关系

编译器默认将输出按照文件类型与后缀相对应。

文件名	解释	生成方式
filename.o	目标文件	编译时添加-c选项生成
filename.so	共享库文件	编译时指定为共享型，如添加-shared，并不含-c
filename.mod	模块文件	编译含有MODULE声明时的源文件生成
filename.s	汇编文件	编译时添加-S选项生成
a.out	默认生成的可执行文件	编译时没有指定-c时生成

重要编译选项

- 编译选项对运行速度、编译的兼容性等有影响
- 建议仔细看看编译器手册，多加测试，选择适合自己程序的编译选项以提高性能
- gfortran支持所有gcc选项
- 以下仅仅介绍部分重要选项

控制Fortran语言类型的选项

- -ffree-form和-ffixed-form：声明源文件是自由格式还是固定格式，默认Fortran 9x、200x的源文件为自由格式，Fortran 77等的源文件为固定格式
- -fdefault-double-8：设置DOUBLE PRECISION类型为8比特
- -fdefault-integer-8：设置INTEGER和LOGICAL类型为8比特
- -fdefault-real-8：设置REAL类型为8比特
- -fno-backslash：将反斜线(\)当作正常字符（非转义符）处理
- -ffixed-line-length-<n>：设置固定格式源代码的行宽为n
- -ffree-line-length-<n>：设置自由格式源代码的行宽为n
- -fmax-identifier-length=<n>：设置名称的最大字符长度为n，Fortran 95和200x的长度分别为31和65
- -fimplicit-none：禁止变量的隐式声明，所有变量需显式声明
- -fcray-pointer：支持Cray指针扩展
- -fopenmp：编译OpenMP并行程序
- -std=<std>：指明Fortran标准，std可以为gnu、f95、f2003、f2008、legacy
- -M<dir>和-J<dir>：指定编译时保存生成的模块文件目录
- -fconvert=<conversion>：指定对无格式Fortran数据文件表示方式，其值可为：native，默认值；swap，在输入输出时从大端（big-endian）到小端（little-endian）交换比特，或相反；big-endian，用大端方式读写；little-endian，用小端方式读写。x86架构为小端，IBM POWER架构为大端。

一般选项

- -c：仅编译成目标文件（.o文件），并不进行链接
- -o file：指定生成的文件名
- -v：详细模式，显示在每个命令执行前显示其命令行
- -###：显示编译器、汇编器、链接器的调用信息但并不进行实际编译，在脚本中可以用于捕获驱动器生成的命令行
- –help：显示帮助信息
- –target-help：显示目标平台的帮助信息
- –version：显示编译器版本信息

警告选项

- -fsyntax-only：仅仅检查代码的语法错误，并不进行其余操作
- -w：编译时不显示任何警告，只显示错误
- -Wfatal-errors：遇到第一个错误就停止，而不尝试继续运行

调试选项

- -g：包含调试信息
- -ggdb：包含利用gbd调试时所需要的信息

优化选项

- -O[level]：设置优化级别。优化级别level可以设置为0、1、2、3、s，默认为-O0
- -Ofast：设置优化级别，相当于-O3

预处理选项

- -C：保留预处理的C源文件中的注释
- -D name：在预处理中定义宏name的值为1
- -D name=def：在预处理中定义name为def
- -U name：去除预处理中的任何name初始定义
- -undef：不预定义系统或GCC声明的宏，但标准预定义的宏仍被定义
- -dD：显示源文件中定义的宏及其值到标准输出
- -dI：显示预处理中包含的所有文件，包括文件名和定义时的行号
- -dM：显示预处理时源文件中定义的宏及值，含定义时文件名和行号
- -dN：与-dD类似，但只显示源文件中定义的宏，而不显示宏值
- -E：预处理各文件，将结果发给标准输出，不进行编译、汇编或链接
- -I<头文件目录>：指明头文件的搜索路径
- -M：打印make的依赖关系到标准输出
- -MD：打印make的依赖关系到文件file.d，file是编译文件的根名字
- -MM：打印make的依赖关系到标准输出，但忽略系统包含
- -MMD：打印make的依赖关系到文件file.d，其中file是编译的文件的根名字，但忽略系统头文件
- -P：预处理每个文件，并保留每个file.c文件预处理后的结果到file.i

链接选项

- -pie：在支持的目标上生成位置无关的可执行文件
- -s：从可执行文件中去除所有符号表
- -rdynamic：添加所有符号表到动态符号表中
- -static：静态链接所需的库
- -shared：生成共享目标而不是可执行文件，必须在编译每个目标文件时使用-fpic选项
- -shared-libgcc：使用共享libgcc库
- -static-libgcc：使用静态libgcc库
- -u <symbol>：确保符号symbol未定义，强制连接一个库模块定义它
- -I<头文件目录>：指明头文件的搜索路径
- -l<库文件>：指明需链接的库名，如库为libxyz.a，则可用-lxyz指定
- -L<库目录>：指明库的搜索路径
- -B<路径>：设置寻找可执行文件、库、头文件、数据文件等路径

平台相关选项

- -mtune=cpu-type：设置优化针对的CPU类型，可为：generic、core2、corei7、corei7-avx、core-avx-i、core-avx2、k8、opteron等
- -march=cpu-type：设置指令针对的CPU类型，可为：generic、core2、corei7、corei7-avx、core-avx-i、core-avx2、k8、opteron等
- -mieee-fp和-mno-ieee-fp：浮点操作是否严格按照IEEE标准

约定成俗的选项

- -fno-automatic：将程序单元的本地变量和数组声明具有SAVE属性
- -ff2c：与g77和f2c生成的代码兼容
- -fno-underscoring：不在名字后添加_。注意：gfortran默认行为与g77和f2c不兼容，为了兼容需要加-ff2c选项。除非使用者了解与现有系统环境的集成，否则不建议使用-fno-underscoring选项
- -funderscoring：对外部名字没有_的加_，以与一些Fortran编译器兼容
- -fsecond-underscore：默认gfortran对外部名称添加一个_，如果使用此选项，那么将添加两个_。此选项当使用-fno-underscoring选项时无效。此选项当使用-ff2c时默认启用
- -fpic：生成位置无关的代码以用于共享库
- -fPIC：如果目标机器支持，将生成位置无关的代码
- -fpie和-fPIE：与-fpic和-fPIC类似，但生成的位置无关代码只能链接到可执行文件中

串行及OpenMP并行程序编译举例

- 将Fortran 77程序yourprog.for编译为可执行文件yourprog:
  gfortran␣-o␣yourprog␣yourprog.for
- 将Fortran 90程序yourprog.f90编译为可执行文件yourprog:
  gfortran␣-o␣yourprog␣yourprog.f90
- 将使用lapack库的Fortran 90程序yourprog.f90编译为可执行文件yourprog:
  gfortran␣-o␣yourprog␣-L/opt/lib␣-llapack␣yourprog.f90
- 将Fortran 90程序yourprog.f90编译为目标文件yourprog.o:
  gfortran␣-c␣yourprog.f90
- 将Fortran 90程序yourprog.f90静态编译为O3优化的可执行文件yourprog:
  gfortran␣-O3␣-static␣-o␣yourprog␣yourprog.f90
- 将OpenMP指令并行的Fortran 90程序yourprog-omp.f90编译为可执行文件yourprog-omp:
  gfortran␣-o␣yourprog-omp␣-fopenmp␣yourprog.f90

编译时出错信息格式

```
NOlihm.f90:146.14:

   n2nd=0;    npr=0
                  1
Error: Symbol 'npr' at (1) has no IMPLICIT type
```

- 源文件名:行数:列数:
- 源文件代码
- 1指示出错位置
- 错误类型: 具体说明

GNU程序调试器gdb简介

- GNU调试器gdb可用于调试C/C++、Fortran等语言编写的程序
- 程序编译时需要添加-g编译选项
- 基本启动方式:
  - *gdb␣[−help]␣[−nx]␣[−q]␣[−batch]␣[−cd=dir]␣[−f]*
    [−b␣bps]␣[− tty =dev]␣[−s␣ symfile]␣[−e␣prog]␣[−se␣prog]
    [−c␣core]␣[−x␣cmds]␣[−d␣dir]␣[prog␣[core|procID]]
  - *gdb␣[options]␣−−args␣prog␣[arguments]*
- 注意：程序本身含有参数时，前面需有–args，否则不起作用，如:
  gdb␣−−args␣yourprog␣in . dat

gdb启动参数

- -b bps：指定远程调试时的线速，波特率或者比特/每秒
- -batch：以批处理模式运行-x FILE（和.gdbinit）中的命令，退出值为0时表示成功运行指定的所有命令，否则表示有错误发生
- -c FILE, -core=FILE：指定使用core文件
- -cd=directory：以directory目录为其工作目录
- -e FILE, -exec=FILE：指定使用的可执行文件
- -h, -help：显示帮助信息
- -se=file：从文件file中读取符号表，并当可执行文件使用
- -q, -quiet：安静模式，启动时不显示版权等信息
- -tty=device：以设备device做为标准输入输出设备
- –args：将可执行文件名后面的参数传递给此可执行文件
- -tui：以带有专门源代码显示窗口的模式运行
- -write：支持写入可执行和core文件
- -x FILE, -command=FILE：指定启动gbd后执行的操作命令

进入gdb后的操作

进入gdb后的命令非常多，这里仅仅解释几个主要的，详细的请进入gdb后利用help命令查看，里面的信息以及命令比man gdb多得多。

- break [file:]function：设置[file中的]名字为function的函数断点
- run [arglist]：[以arglist参数列表]启动程序
- bt：显示程序的堆栈stack
- print expr：打印表达式的值
- c：继续运行因为断点等终止的程序
- next：执行程序中的下一行语句，如此行是函数，将执行完此函数
- edit [file:]function：编辑当前停止行的代码
- list [file:]function：打印当前停止行附近的代码
- step：执行行程序中的下一行语句，如此行是函数将进入函数内
- help [name]：显示[GDB name命令或]一般GDB信息
- quit：退出GDB

联系信息

河南师范大学高性能计算中心：

- 电话：0373-3326142
- 信箱：cxq@htu.cn
- QQ群：171803133
- 主页：http://www.htu.cn/hpcc/
- 办公室：河南师大琢玉楼3楼