



GNU C/C++、Fortran 编译器的使用

李会民

hmli@ustc.edu.cn, lihm@qibebt.ac.cn

中国科学院青岛生物能源与过程研究所 超级计算中心

2009 年 12 月

- 1 GNU C/C++、Fortran 编译器简介
- 2 GNU C/C++ 编译器用法
- 3 GNU Fortran 编译器用法
- 4 串行程序调试: gdb
- 5 联系信息

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

- GNU 编译器编译 C、C++ 源程序的命令分别为 gcc 和 g++
- 编译 Fortran 77 和 9x 源程序的命令分别为 g77 和 gfortran:
 - gfortran 可以直接编译 Fortran 77 源程序
 - g77 不可编译 Fortran 9x 源程序

注意：

- HP RX2600 平台上的 GNU 编译器的版本是 3 系列的，不支持 Fortran 9x
- 联想集群未装 gfortran
- 以下将以科大联想集群上安装的 4.1.2 版本介绍，其余平台的有些选项未必适合

- 1 GNU C/C++、Fortran 编译器简介
- 2 GNU C/C++ 编译器用法
- 3 GNU Fortran 编译器用法
- 4 串行程序调试: gdb
- 5 联系信息

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

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



编译选项对运行速度编译的兼容性等有影响，以下仅仅介绍部分重要选项，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。

- **-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。
- **-v:** 详细模式，显示在每个命令执行前显示其命令行。

控制文件类型的选项 II



- `-###`: 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。



- `-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。



- -g: 包含调试信息。
- -ggdb: 包含利用 gbd 调试时所需要的信息。



- `-O[level]`: 设置优化级别。优化级别 `level` 可以设置为 0、1、2、3、s。

- -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。

- `-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。



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



- `-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。

- 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`

- 1 GNU C/C++、Fortran 编译器简介
- 2 GNU C/C++ 编译器用法
- 3 GNU Fortran 编译器用法**
- 4 串行程序调试: gdb
- 5 联系信息

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.i90	自由格式的 Fortran 源文件	被 Fortran 编译器编译
filename.F90	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



编译选项对运行速度编译的兼容性等有影响，以下仅仅介绍部分重要选项，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。

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, 默认从 Fortran 90 起的源文件为自由格式, 之前的 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 并程序。
- `-std=<std>`: 指明 Fortran 标准, `std` 可以为 `f95`、`f2003`、`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。

- `-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。



- `-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。



- -g: 包含调试信息。
- -ggdb: 包含利用 gbd 调试时所需要的信息。



- `-O[level]`: 设置优化级别。优化级别 `level` 可以设置为 0、1、2、3、s。

- -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。

- `-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。



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

- **-fno-automatic**: 将每个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。

- 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`

- 1 GNU C/C++、Fortran 编译器简介
- 2 GNU C/C++ 编译器用法
- 3 GNU Fortran 编译器用法
- 4 串行程序调试: gdb
- 5 联系信息



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]`

- `-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`: 以文本模式运行
- `-write`: 支持写入可执行和 `core` 文件
- `-x FILE`, `-command=FILE`: 指定启动 `gdb` 后执行的操作命令

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 gbb 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

- 中国科大超算中心：
 - 主页: <http://scc.ustc.edu.cn>
 - 电话: 0551-3602248
 - 信箱: sccadmin@ustc.edu.cn
- 青能所超算中心：
 - 当前主页: <http://124.16.151.186>
 - 将来域名: <http://scc.qibebt.cas.cn>
 - 电话: 0532-80662613
 - 信箱: scc@qibebt.ac.cn
- 李会民：
 - 主页: <http://staff.ustc.edu.cn/~hmli/>
 - 电话: 0532-80662613
 - 信箱: hmli@ustc.edu.cn、lihm@qibebt.ac.cn

欢迎指出错误和改进意见。