

# 《GCC技术参考大全》

## 图书基本信息

书名：《GCC技术参考大全》

13位ISBN编号：9787302088431

10位ISBN编号：7302088438

出版时间：2004-7-1

出版社：清华大学出版社

作者：Arthur Griffith,胡恩华

页数：516

译者：胡恩华

版权说明：本站所提供下载的PDF图书仅提供预览和简介以及在线试读，请支持正版图书。

更多资源请访问：[www.tushu000.com](http://www.tushu000.com)

# 《GCC技术参考大全》

## 内容概要

GCC技术参考大全，ISBN：9787302088431，作者：（美）Arthur Griffith著；胡恩华译；胡恩华译

## 作者简介

Arthur Griffith 从1997开始他的第一份编程工作，开发过编译程序、解释器、边接程序以及汇编器，曾作为小组成员为某些特殊用途的计算机开发过汇编器和连接程序。之后他加入PL/EXUS语言的编译程序的维护组。装下来的项目就是为SATS语言开发了交互式的解释器和编译程序。

## 书籍目录

第1部分 自由软件编译程序

第1章 GCC简介

第2章 查询与安装编译程序

第2部分 使用编译程序集合

第3章 预处理程序

第4章 编译C程序

第5章 编译C++程序

第6章 编译Objective-c

第7章 编译Fortran程序

第8章 编译Java

第9章 编译Ada程序

第10章 混合语言

第3部分 外设和内设

.....

第4部分 附录

.....

# 《GCC技术参考大全》

## 编辑推荐

在本书中，计算机咨询师Arthur Giffith提供了最重要的开源软件——GCC的大量信息，囊括GCC为几乎所有平台和语言开发部署各类软件的技术细节。全书分为3个部分，当中解释了每种语言的命令行选项，详细描述了如何成功地综合使用多种语言开发应用程序，并总结了各类工具软件的使用法，此外还近索了一些令人兴奋的可能实现的新编程方法，以及新加入GCC的将Java代码的功能。

## 精彩短评

- 1、 工具书，一个晚上可以读完
- 2、 有什么很郁闷的问题可以看一下。。。
- 3、 一般般.....
- 4、 Linux下学C编程的看过来,这本书不错,值得一看.
- 5、 从图书馆借来就翻看了自己感兴趣的那几章，读起比较乏味。
- 6、 苦逼工科男，Start，2013年5月7日。虽然看不很懂，但感觉GCC很牛叉的样纸。。 Done，2013年9月14日
- 7、 基本是废话==
- 8、 一般般吧
- 9、 不只是一本GCC手册，同时还举了很多实用的例子，深入了解GCC的好读物。

# 《GCC技术参考大全》

## 精彩书评

1、人生掌握你所用语言的一个编译器足矣；如果你用C编程，使用GCC套装吧；如果你用C++编程，使用GCC套装吧；如果你用Java编程，使用GCC套装吧；如果你用Fortran编程，使用GCC套装吧；如果你用Ada编程，使用GCC套装吧；如果你用Go编程，使用GCC套装吧；. . . . .一旦你掌握了GCC (minGW)，其他的同一种语言的编译器你都可不在学习，因为它已经强大的不能再强大了。而这本书，是我见过的介绍GCC最为详细和完成的书！

## 章节试读

### 1、《GCC技术参考大全》的笔记-第58页

感觉解释不合理：

图 创建共享库

这里的意思就是说后缀.so起到了决定性的作用，而-shared只是防止出错。

读者使用.sso后缀进行测试，然后mv \*.sso \*.so，此时生成的共享库也是能够正常工作的。

即便不进行更名，直接

gcc main.c \*.sso也是能够正常编译和运行的。

另外，gcc的帮助信息中-shared就是说生成静态库。

因此，读者感觉此处的解释不合理。

-----  
如有错误，欢迎指正。

### 2、《GCC技术参考大全》的笔记-第201页

运行时载入库

一旦程序被连接，要使用共享库，就必须在运行的时候能够找到共享库的位置。因为是通过名字定位库，而不是通过目录定位库，因此可以在连接程序的时候使用一个库，而在运行的时候使用另一个库。如果改变了库的版本，而没有改变程序版本，可能出现的问题——这就是大多数库会把版本号作为名字的一部分（libm.so.6）。

无论何时载入程序并打算运行，共享库应该位于以下位置：

环境变量LD\_LIBRARY\_PATH列出的所有用分号分割的目录

文件/etc/ld.so.cache中找到的库的列表，由工具ldconfig维护

目录/lib

目录/usr/lib

想知道载入哪个库，以及确切了解应用程序使用的库，使用工具ldd。

```
$ gcc -fpic -shared sayhello.c saysomething.c -o libsayfn.so
```

P205 ldd命令选项

工具ldd会遍历二进制可执行代码的目标文件，或者是命令行指定共享库中的目标文件，然后列出所有共享库的依赖关系。例如，列出linux系统上bash shell程序使用的共享库：

```
$ ldd /bin/bash
```

每行列出的第一个名字是共享库出现在函数内部的名字，第二个名字就是磁盘上找到的实际运行库的路径名。库被载入内存的地址出现在行末。

### 3、《GCC技术参考大全》的笔记-第213页

COFF格式将目标分成不同的章节。其中的.text章节包含了可执行的代码，而.data章节包含的是具有初始化值的变量，.bss部分的则是未被初始化的数据。这样划分的本质原因是如果该程序同时运行多个进程，可在内存中共享相同的.text部分，而且.data章节也可作为一个单独的程序块载入内存，其中都是初始值，而.bss章节只作为独立的数字（大小）就可存在文件中，可在加载程序时将其扩展到



正确尺寸。

## 4、《GCC技术参考大全》的笔记-第201页

翻译错误

原文：

Most often the system libraries are stored in the directories /lib and /usr/lib, so these two directories are automatically searched. You can specify other directories to be searched by using one or more -L options. For example, the following command instructs the linker to look in both the current directory and the directory named /home/fred/lib for any libraries that are “ not found on ” the default search path.

## 5、《GCC技术参考大全》的笔记-第57页

翻译错误：

原文：使用下列命令可将这两个源文件编译成一个目标文件：

```
gcc -c -fpic shellofirst.c shellosecond.c
```

原文：These two source files can be compiled into object files with the following command

此处命令会为每个源文件分别生成一个目标文件，而不是翻译中所说“可将这两个源文件编译成一个目标文件”。

## 6、《GCC技术参考大全》的笔记-第43页

下面是一个特征和规则列表，可应用于#include指示字：

尖括号包围的文件名会令查找由使用-I选项说明的目录开始，然后继续查找系统目录的标准集合。由引号包围的文件名会导致查找当前工作目录（包含正在处理的源文件的目录），然后继续查找由带尖括号的指示字指定的目录。

在UNIX系统中，系统目录的标准集合如下：

```
/usr/local/include
```

```
/usr/lib/gcc-lib/target/version/include
```

```
/usr/target/include
```

```
/usr/include
```

相对路径名可用作文件名。例如，如果指定#include<sys/time.h>，会在所有标准的子目录sys中查找头文件time.h

## 7、《GCC技术参考大全》的笔记-第51页

排版错误（导致严重错误）：

翻译：

产生make程序的描述文件

原文：

```
trick.o: trick.c /usr/include/stdio.h /usr/include/features.h \
```

```
/usr/include/sys/cdefs.h /usr/include/gnu/stubs.h \
```

```
/usr/lib/gcc-lib/i386-redhat-linux/2.96/include/stddef.h \
```

```
/usr/include/bits/types.h /usr/include/bits/pthreadtypes.h \
```

```
/usr/include/bits/sched.h /usr/include/libio.h /usr/include/_G_config.h \  
/usr/include/wchar.h /usr/include/bits/wchar.h /usr/include/gconv.h \  
/usr/lib/gcc-lib/i386-redhat-linux/2.96/include/stdarg.h \  
/usr/include/bits/stdio_lim.h barrow.h
```

## 8、《GCC技术参考大全》的笔记-第40页

可变的宏是具有可变数目参数的宏。这些参数由省略号代表，被保存在一个由逗号分隔的字符串中作为变量\_\_VA\_ARGS\_\_，它会在宏的内部扩展。

```
#define errout(a,b,...) \  
    fprintf(stderr, "File %s Line %d \n",a,b); \  
    fprintf(stderr, __VA_ARGS__)  
  
errout(__FILE__, __LINE__, "unexpected termination.\n");
```

## 9、《GCC技术参考大全》的笔记-第198页

静态库包含目标文件，叫做文档archive文件，由工具ar构造并维护。文档文件的名称前缀为lib后缀为.a。下面的命令序列编译三个目标文件，并将它们复制到库中，然后连接程序会使用目标文件main.o和库中的内容构造可执行程序。

```
$ gcc -c inlet.c outlet.c genspru.c  
$ ar -r libspin.a inlet.o outlet.o genspru.o  
$ gcc main.c libspin.a -o spinner
```

下面一对命令使用ar工具的-q选项可将文件快速附加在现存文档后面，而不需要更新索引，然后使用ranlib更新索引反应文档当前的状态：

```
$ ar -q libpin.a mongul.o strop.o klbrgr.o  
$ ranlib libspin.a
```

表12-1 指明操作的ar选项

## 版权说明

本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问:[www.tushu000.com](http://www.tushu000.com)