

• GCC 使用入门（一）(1)

• <http://developer.51cto.com> 2006-09-26 16:52 宇文 51CTO.com 我要评论(1)

本文将向读者介绍在 Linux 平台下应用程序的编译过程,以及编译程序 GCC 在编译应用程序的过程的具体用法,同时详细说明了 GCC 的常用选项、模式和警告选项。

【51CTO.com 独家特稿】一、GCC 简介

通常所说的 GCC 是 GUN Compiler Collection 的简称,除了编译程序之外,它还含其他相关工具,所以它能把易于人类使用的高级语言编写的源代码构建成为计算机能够直接执行的二进制代码。GCC 是 Linux 平台下最常用的编译程序,它是 Linux 平台编译器的标准。同时,在 Linux 平台下的嵌入式开发领域,GCC 也是用得最普遍的一种编译器。GCC 之所以被广泛采用,是因为它能支持各种不同的目标体系结构。例如,它既支持基于宿主的开发(简单讲就是要为某平台编译程序,就在该平台上编译),也支持交叉编译(即在 A 平台上编译的程序是供平台 B 使用的)。目前,GCC 支持的体系结构有四十余种,常见的有 X86 系列、Arm、PowerPC 等。同时,GCC 还能运行在不同的操作系统上,如 Linux、Solaris、Windows 等。



• [详解 GCC 的下载和安装](#)

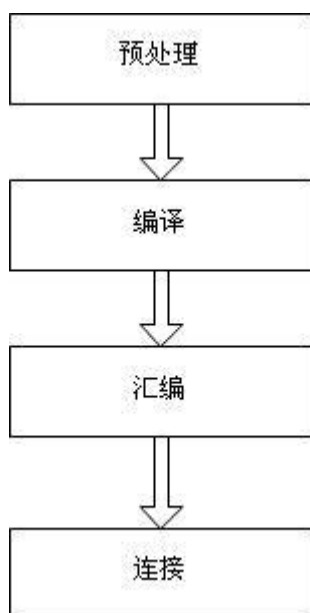
• [Linux 编程起步 GCC 基本用法](#)

除了上面讲的之外,GCC 除了支持 C 语言外,还支持多种其他语言,例如 C++、Ada、Java、Objective-C、FORTRAN、Pascal 等。

本系列文章中,我们不仅介绍 GCC 的基本功能,还涉及到一些诸如优化之类的高级功能。另外,我们还考察 GCC 的一些映像操作工具,如 size 和 objcopy 等,这将在后续的文章中加以介绍。

二、程序的编译过程

对于 GUN 编译器来说,程序的编译要经历预处理、编译、汇编、连接四个阶段,如下图所示:



从功能上分，预处理、编译、汇编是三个不同的阶段，但 GCC 的实际操作上，它可以把这三个步骤合并为一个步骤来执行。下面我们以 C 语言为例来谈一下不同阶段的输入和输出情况。

在预处理阶段，输入的是 C 语言的源文件，通常为*.c。它们通常带有.h 之类头文件的包含文件。这个阶段主要处理源文件中的`#ifdef`、`#include`和`#define`命令。该阶段会生成一个中间文件*.i，但实际工作中通常不用专门生成这种文件，因为基本上用不到；若非要生成这种文件不可，可以利用下面的示例命令：

```
gcc -E test.c -o test.i
```

在编译阶段，输入的是中间文件*.i，编译后生成汇编语言文件*.s。这个阶段对应的 GCC 命令如下所示：

```
GCC -S test.i -o test.s
```

在汇编阶段，将输入的汇编文件*.s 转换成机器语言*.o。这个阶段对应的 GCC 命令如下所示：

```
GCC -c test.s -o test.o
```

最后，在连接阶段将输入的机器代码文件*.s（与其它的机器代码文件和库文件）汇集成一个可执行的二进制代码文件。这一步骤，可以利用下面的示例命令完成：

```
GCC test.o -o test
```

上面介绍了 GCC 编译过程的四个阶段以及相应的命令。下面我们进一步介绍常用的 GCC 的模式。

三、GCC 常用模式

这里介绍 GCC 追常用的两种模式：编译模式和编译连接模式。下面以一个例子来说明各种模式的使用方法。为简单起见，假设我们全部的源代码都在一个文件 `test.c` 中，要想把这个源文件直接编译成可执行程序，可以使用以下命令：

```
$ GCC -o test
```

这里 `test.c` 是源文件，生成的可执行代码存放在一个名为 `test` 的文件中（该文件是机器代码并且可执行）。`-o` 是生成可执行文件的输出选项。如果我们只想让源文件生成目标文件（给文件虽然也是机器代码但不可执行），可以使用标记 `-c`，详细命令如下所示：

```
$ GCC -c test.c
```

默认情况下，生成的目标文件被命名为 `test.o`，但我们也可以为输出文件指定名称，如下所示：

```
$ GCC -c test.c -o
```

上面这条命令将编译后的目标文件命名为 `mytest.o`，而不是默认的 `test.o`。

迄今为止，我们谈论的程序仅涉及到一个源文件；现实中，一个程序的源代码通常包含在多个源文件之中，这该怎么办？没关系，即使这样，用 GCC 处理起来也并不复杂，见下例：

```
$ GCC -o test first.c second.c third.c
```

该命令将同时编译三个源文件，即 `first.c`、`second.c` 和 `third.c`，然后将它们连接成一个可执行程序，名为 `test`。

• Linux 编程起步 GCC 基本用法

-
- <http://developer.51cto.com> 2008-10-28 17:39 孤独的 blog 百度空间 [我要评论\(0\)](#)

初学时最好从命令行入手，这样可以熟悉从编写程序、编译、调试和执行的整个过程。编写程序可以用 vi 或其它编辑器编写。编译则使用 GCC 命令。要往下学习首先就得熟悉 GCC 命令的用法。

初学时最好从命令行入手，这样可以熟悉从编写程序、编译、调试和执行的整个过程。编写程序可以用 vi 或其它编辑器编写。编译则使用 GCC 命令。要往下学习首先就得熟悉 GCC 命令的用法。

GCC 命令提供了非常多的命令选项，但并不是所有都要熟悉，初学时掌握几个常用的就可以了，到后面再慢慢学习其它选项，免得因选项太多而打击了学习的信心。

一. 常用编译命令选项

假设源程序文件名为 test.c。

1. 无选项编译链接

用法: #gcc test.c

作用: 将 test.c 预处理、汇编、编译并链接形成可执行文件。这里未指定输出文件，默认输出为 a.out。

2. 选项 -o

用法: #gcc test.c -o test

作用: 将 test.c 预处理、汇编、编译并链接形成可执行文件 test。-o 选项用来指定输出文件的文件名。

3. 选项 -E

用法: #gcc -E test.c -o test.i

作用: 将 test.c 预处理输出 test.i 文件。

4. 选项 -S

用法: #gcc -S test.i

作用: 将预处理输出文件 test.i 汇编成 test.s 文件。

5. 选项 -c

用法: #gcc -c test.s

作用: 将汇编输出文件 test.s 编译输出 test.o 文件。

6. 无选项链接

用法: #gcc test.o -o test

作用: 将编译输出文件 test.o 链接成最终可执行文件 test。

7. 选项-O

用法: `#gcc -O1 test.c -o test`

作用: 使用编译优化级别 1 编译程序。级别为 1~3, 级别越大优化效果越好, 但编译时间越长。

二. 多源文件的编译方法

如果有多个源文件, 基本上有两种编译方法:

[假设有两个源文件为 `test.c` 和 `testfun.c`]

1. 多个文件一起编译

用法: `#gcc testfun.c test.c -o test`

作用: 将 `testfun.c` 和 `test.c` 分别编译后链接成 `test` 可执行文件。

2. 分别编译各个源文件, 之后对编译后输出的目标文件链接。

用法:

`#gcc -c testfun.c //将 testfun.c 编译成 testfun.o`

`#gcc -c test.c //将 test.c 编译成 test.o`

`#gcc -o testfun.o test.o -o test //将 testfun.o 和 test.o 链接成 test`

以上两种方法相比较, 第一中方法编译时需要所有文件重新编译, 而第二种方法可以只重新编译修改的文件, 未修改的文件不用重新编译。