第5章

GCC 的安装

学习本章要达到的目标

(1)对GCC有一定的了解。

(2) 学会使用 GCC 源代码来安装 GCC 的方法。

(3) 掌握 Linux 下软件安装的基本流程。

5.1 GCC 简介

GCC 是整个 GNU 项目中的一个重要组成部分,它的全称是 the GNU Compiler Collection,即 GNU 编译工具集合,其标志如图 5.1 所示。GCC 包括了预编译(cpp)、编译(gcc),它与生成(make)、连接(ld)、调试(gdb)等多个编译过程的常用工具组成了GNU 开发工具链。

GCC 最早的正式版本 1.0 于 1987 年 5 月 23 日发布, 历经二十余 年的发展, GCC 已经成为了一款功能强大的世界级优秀编译工具。 GCC 支持多种语言的编译,包括 C 语言、C++、Ada、Fortran、Java 等,并且包含这些语言相应的开发支持库。GCC 具有较高的代码编译 质量,这一点为计算机软件开发人员所称道。GCC 编译生成的可执行 文件,与微软等其他软件公司出品的编译器相比较,具有代码长度短、 执行效率高等特点。GCC 还具有较强的灵活性和高度的可移植性,只 要在安装时做出简单的设置或对源代码进行少量的修改,就可以使



图 5.1 GCC 标志

GCC完成在不同指令系统间的交叉编译,为嵌入式设备的软件开发提供了极大的便利。此外,GCC的灵活性也方便了GCC和其他软件或协议的对接。嵌入式开发工具Embest IDE、苹果公司的Xcode开发工具等许多集成开发环境,都采用了GCC作为底层支持环境。GCC支持OpenMP并行程序接口。

GCC 的最新版本 4.9.2, 于 2014 年 10 月 30 日发布。

5.2 解压缩工具 tar

tar 是 GNU 项目中的一个工具,用来将多个文件打包成一个归档文件。

在 GNU 环境下的软件是由多个文件组成的,这个特点给软件在互联网上的传播造成 了麻烦。因为,这样用户若想使用一个软件就不得不分别下载软件中的每个小文件,这样 不仅费时费力,而且用户还容易将它们放错位置导致软件无法运行。使用 tar 工具就可轻 松地解决这个问题。开发人员将开发的软件打包成一个文件,然后上传到互联网上。用户 下载软件包文件, 解包后进行简单的安装就可以正常使用了。为了缩短网络传输时间, 通常在打包的同时还要对文件进行压缩。tar 支持多种压缩格式,常用的压缩格式有 gzip 和 bzip2 两种。后者拥有更高的压缩比。从扩展名上可以分辨压缩文件的类型:".tar.gz"表示用 gzip 压缩格式压缩的文件,".tar.bz2"表示用 bzip2 压缩格式压缩的文件。

tar 同其他的 Linux 命令类似,其语法格式如下:

tar [参数]… [文件]…

参数说明:

- -c 创建一个新的归档文件
- -r 向归档文件中添加文件
- -t 列出归档文件的内容
- -x 从归档文件中提取文件,是创建文件的逆过程
- -S 有效处理稀疏文件
- -O 提取文件到标准输出
- -f 使用指定的目标文件或设备
- -a 根据文件的后缀自动选择解压缩工具
- -j 使用 bzip2 压缩/解压缩文件
- -z 使用 gzip 压缩/解压缩文件
- -v 打印正在操作的文件的文件名

通常,在对软件包的归档文件进行解包时,用到 xzvf(针对 gzip 压缩文件)和 xjvf(针 对 bzip2 压缩文件)两种参数组合。

5.3 在 Linux 下使用源代码安装软件的基本步骤

在 GNU 环境下,所有的软件都具有高度的可移植性。为了保证这种可移植性,Linux 下几乎所有的软件都采用源代码的形式进行发布,这也充分地体现了自由软件的精神。采 用源代码形式发布软件的安装过程同 Windows 下的软件安装有着较大的区别。安装过程大 体上分为以下几个步骤。

(1)将软件包的归档文件解压缩到一个目录下,以便进行下面的操作。

(2)用户需要对源代码进行配置。配置是对源代码本地化的过程,这一步是为编译软件做准备。配置通常包括设置安装路径、设置目标环境(指定编译器)、设置软件的功能(常用在嵌入式软件安装中,又称软件裁减)等工作。配置通常使用软件开发人员事先通过软件开发工具生成的配置工具来进行,这个工具通常是在源代码所在目录下,命名为 configure 的可执行文件。该文件携带大量开关参数,用户通过这些开关参数进行配置。一些较为友好的软件开发者会提供一些如文本菜单、图形界面等交互方式的配置工具。经过配置会生成一个"Makefile"文件,它记录了配置的细节和对应的软件编译流程。

(3)进行编译。编译通常使用 make 工具完成, make 工具是 GNU 环境中用来管理源 代码工程的工具。它根据当前目录下的 Makefile 文件的内容进行编译。关于 Makefile 文件, 将在第6章进行较为详细的阐述。

(4) 进行安装。安装在 Linux 下其实仅仅是一个简单复制的过程。经过编译,源代码

。 第 5

章

Linux 操作系统应用与开发教程

和可执行文件是混合放在安装目录下的。通过安装,将可执行文件和相应的库文件等复制 到目标目录,并建立相对应的链接文件以方便访问。这些过程也是通过 make 工具完成的, 常使用的命令是 make install。

下面开始一步步地完成 GCC 的安装。

5.4 获得 GCC 软件包

GCC 是自由软件,可以从互联网上下载,它的大小在 50~120M 之间。首先访问 GCC 的首页。打开 "Firefox 网络浏览器",在地址栏中输入 gcc.gnu.org,按 Enter 键,出现 GCC 的首页,如图 5.2 所示。



图 5.2 GCC 首页

单击右侧 Download 下的 Mirror Sites, 跳转到下载镜像服务器列表页面, 如图 5.3 所示。



图 5.3 GCC 下载镜像服务器列表

在列表中任选择一个网站。通常从任何一个站点都可以下载到 GCC 的源代码,但考虑到网络速度,建议选择一个访问速度最快的网站。这里以 GNU FTP server and its mirrors 为例进行说明。单击进入该服务器的文件列表,如图 5.4 所示。在图 5.4 中找到 China 服务器的列表,选择一个进入即可,在此选择 ftp://mirrors.usts.edu.cn/gnu/,单击进入页面,如图 5.5 所示。

70

| 🔕 😑 🗈 GNU Mirror List - GNU Project - Free Software Foundation - Mozilla Firefox | | | |
|--|--|-------------------|-----------|
| 💖 GNU Mirror List - G 🗙 🕂 | 😞 🖨 🗉 ftp://mirrors.ustc.edu.cn/gnu/ 的索引 | - Mozilla Firefox | |
| 🗲 🖲 www.gnu.org/prep/ftp.html 🛛 👻 🖓 Google 🔍 😭 😫 | 👳 ftp://mirrors.ustc.e 🗙 📫 | | |
| rsync://mirrors.ispros.com.bd/gnu | • Ttp://mirrors.ustc.edu.cn/gnu/ | ▼ C 😫 ▼ Google | Q 🏠 🗎 🖁 |
| | 🚞 garpd | 2010年12月07日 | 00时00分00秒 |
| China | 🚞 gawk | 2014年04月09日 | 00时00分00秒 |
| http://mirror hust edu.cn/gnu/ | 🚞 gcal | 2013年03月30日 | 00时00分00秒 |
| http://mirrors.ustc.edu.cn/gnu/ | 📄 gcc | 2014年12月19日 | 21时05分00秒 |
| ftp://mirrors.ustc.edu.cn/gnu/ | 🚞 gcide | 2012年03月04日 | 00时00分00秒 |
| rsvnc://mirrors.ustc.edu.cn/gnu/ | 冲 gcl | 2014年10月28日 | 22时05分00秒 |
| 0 | 🚞 gcompris | 2003年08月02日 | 00时00分00秒 |
| India | 🚘 gdb | 2015年02月21日 | 01时50分00秒 |

图 5.4 GCC 镜像站点列表

图 5.5 GNU 软件列表

在 GNU 软件目录列表中,找到需要的 GCC,单击进入 gcc 列表文件夹,进入如图 5.6 所示的页面,可以看到页面中列出了很多不同版本的 GCC。这里选择下载最新版本,单击 gcc-4.9.2 进入目录,出现如图 5.7 所示的页面。可以看到,GCC 4.9.2 的文件列表中的 gcc-4.9.2.tar.bz2 和 gcc-4.9.2.tar.gz 都是 GCC 的完整源代码文件包,它们的区别仅仅是压缩 格式不同。按照个人喜好选择它,当然为了下载能够快一些,这里推荐使用体积较小的前者,它的体积是 89.9M。单击该文件会出现下载提示。选择"保存文件",然后单击"确定" 按钮,如图 5.8 所示。

| ftp://mirrors.ustc.edu.cn/gnu/gcc/ f ftp://mirrors.ustc.e × + | 的东引 - Mozi | illa Firefox | | G ftp://mirrors.ustc.edu.cn/gn Ftp://mirrors.ustc.e × + | ɪ/gcc/gcc-4.9.2/ 的算 | 索引 - Mozilla Firefo | x |
|--|------------|----------------------------|------------------------|--|---------------------|---------------------|--------------|
| Ftp://mirrors.ustc.edu.cn/gnu/gcc/ | ~ C | 🖞 🛛 🔻 Google | Q 🏠 🗎 | The second | cc/gcc-4.9.2/ 🔻 | C S - Google | Q ☆ ∎ |
| 📄 gcc-4.8.2 📄 gcc-4.8.3 | | 2013年10月16日 2014年05月22日 | 00时00分00秒 00时00分00秒 | ▲ 回到上一层文件夹 | au.cn/ghu/g | cc/gcc-4.9.2/ | 的系引 |
| 📄 gcc-4.8.4 | | 2014年12月19日 | 21时06分00秒 | 名称 | 大小 | 修改时 | 间 |
| 🚞 gcc-4.9.0 | | 2014年04月22日 | 00时00分00秒 | 🔤 gcc-4.9.1-4.9.2.diff.bz | 409 KB | 2014年10月30日 | 17时53分(|
| 🚞 gcc-4.9.1 | | 2014年07月16日 | 00时00分00秒 | 📄 gcc-4.9.1-4.9.2.diff.bz | .sig 1 KB | 2014年10月30日 | 17时53分 |
| acc-4.9.2 | | 2014年10月30日 | 18时01分00秒 | gcc-4.9.1-4.9.2.diff.gz | 446 KB | 2014年10月30日 | 17时53分 |
| acc-vms-1 42 tar oz | 837 KB | 1992年11月24日 | 00時100分00秒 | 📄 gcc-4.9.1-4.9.2.diff.gz. | ig 1 KB | 2014年10月30日 | 17时53分(|
| and this issues 1.0 has an | 470 KD | 1002年02日00日 | 008+00/\005# | gcc-4.9.2.tar.bz2 | 87832 KB | 2014年10月30日 | 17时53分 |
| gnu-objc-issues-1.0.cal.gz | 170 KB | 1993年03月08日 | 000000000000 | gcc-4.9.2.tar.bz2.sig | 1 KB | 2014年10月30日 | 17时53分 |
| sed-2.05.bin.README | 1 KB | 1999年08月11日 | 00时00分00秒 | gcc-4.9.2.tar.gz | 112976 KB | 2014年10月30日 | 17时53分 |
| sed-2.05.bin.hpux10 | 304 KB | 1997年07月22日 | 00时00分00秒 | gcc-4.9.2.tar.gz.sig | 1 KB | 2014年10月30日 | 17时53分 |

图 5.6 GCC 软件列表

图 5.7 GCC 4.9.2 软件列表

下载完毕后,可以在桌面上找到该文件。但该文件放在桌面上使用起来不够方便,需 要把它复制到/tmp 文件夹中。选择启动器上面的"主文件夹"打开一个新的窗口,然后点 击右方列表中的"计算机",可以看到根目录下面的所有文件夹。进入/tmp 目录,然后通 过拖曳的方式将文件移动或复制到该目录下,如图 5.9 所示。

| (//mirrors.ustc.edu.cn/gm/gcc/gc ftp://mirrors.ustc.edu. の到上一贯文件夹 名が のcc4.9.1-49.2.diffbz2 のcc4.9.1-49.2.diffbz2 のcc4.9.1-49.2.diffgz.sig のcc4.9.1-49.2.diffgz.sig のcc4.9.1-49.2.diffgz.sig のcc4.9.2.tarbz2.sig のcc4.9.2.tarbz2.sig のcc4.9.2.tarbz2.sig のcc4.9.2.tarbz2.sig のcc4.9.2.tarbz2.sig 0.cc4.9.2.tarbz2.sig | 4.9.2/ ・C (3・coogle) Q 山 山 山 二 二 ● (1) 在式打すgcc4.9.2.1.8rb2 法 三 - 二 | 位置 ○ 最近使用的 ↑ Home ● ● 下教 く > ↑ 主文件夹 下 位置 ○ 最近使用的 ↑ Home ■ 桌面 | gcc-4.9.2.tar.bz2 gcc-4.9.2.tar.bz2 |
|--|--|--|--|
| 因50 | CCC 的下栽坦云 | 図 50 CCC 下裁 会比 | 收壹有判到"/tmn" |

5.5 解压缩软件包

72

现在,需要建立安装环境,所谓安装环境就是一个目录,用来存放安装过程中产生的 一系列文件。

首先,启动"终端"。使用如下命令建立并进入目录。

```
mkdir ~/exper/exp5
cd ~/exper/exp5
```

利用这个目录作为工作环境,然后使用 tar 命令解压缩 GCC 源代码包的归档文件。

```
tar -vxjf /tmp/gcc-4.9.2.tar.bz2
```

如果下载的是 gzip 格式的压缩文件,使用下面的命令。

```
tar -vxzf /tmp/gcc-4.9.2.tar.gz
```

解压缩会持续几分钟的时间。解压缩完毕以后,在当前目录下会生成一个新的名为gcc-4.9.2的目录,如图 5.10 所示。



图 5.10 GCC 软件包解压缩完毕

5.6 对源文件进行配置

这是一步重要的工作,它决定要将 GCC 编译成一个什么样的 GCC。配置前,为了使 源代码文件所在的目录在安装过程中保持整洁,需要新建一个用来存放编译结果的目录。

mkdir gcc-build cd gcc-build

之后进行配置。配置源文件需要使用 gcc-4.9.2 目录下的 configure 工具。首先,对配置中可使用的参数进行简要的介绍:

| -h | 显示帮助信息 |
|------------------|---------------------------------|
| prefix=PATH | 安装体系结构独立文件的目标路径,默认路径为/usr/local |
| exec-prefix=PATH | |
| | 安装体系结果非独立文件的目标路径, 默认路径与独立文件的 |
| | 目标路径相同 |
| disable-FEATURE | |
| | 禁止某项功能 |
| enable-FEATURE | 打开某项功能 |
| host=HOST | 指定编译器运行的主机环境(常用于交叉编译) |

--target=TARGET 指定编译器目标的指令系统(常用于交叉编译)

--enable-languages=LANGS

设定 GCC 支持的语言,若不设置,则安装所有语言

此外还有其他的一些参数,感兴趣的读者可以自行查阅帮助信息。根据常用的情况, 使用下面的命令对 GCC 进行配置。"-disable-multilib"当前系统是 64 位系统,所以需要选择,视不同情况而定。

```
../gcc-4.9.2/configure --prefix=/usr/local/gcc-4.9.2 --disable-multilib
--with-gmp=/usr/local/gmp-5.0 --with-mpfr=/usr/local/mpfr-3.0.0 --with-
mpc=/usr/local/mpc-1.0.2
```

这里设定目标路径主要目的是防止新安装的 GCC 和系统中现有的 GCC 所冲突。当出 现如图 5.11 所示的界面时,且没有报告任何错误信息,表明配置已经顺利结束。这时发现 新建的目录下会多出一些文件,最主要的文件是 Makefile 文件。所有的安装过程几乎都需 要这个文件来调度。

| checking for readelf no | |
|---|---|
| checking for readelf readelf | |
| checking for strip no | |
| checking for strip strip | |
| checking for windres no | |
| checking for windres no | |
| checking for windmc no | |
| checking for windmc no | |
| checking where to find the target ar host tool | |
| checking where to find the target as host tool | |
| checking where to find the target cc just compiled | |
| checking where to find the target c++ just compiled | |
| checking where to find the target c++ for libstdc++ just compiled | |
| checking where to find the target dlltool host tool | |
| checking where to find the target gcc just compiled | |
| checking where to find the target gcj just compiled | |
| checking where to find the target gfortran just compiled | |
| checking where to find the target gccgo host tool | |
| checking where to find the target ld host tool | |
| checking where to find the target lipo host tool | |
| checking where to find the target nm host tool | |
| checking where to find the target objdump host tool | |
| checking where to find the target ranlib host tool | |
| checking where to find the target readel 🛣 🔿 🚳 | |
| checking where to find the target strip. | |
| checking where to find the target windres host tool | |
| checking where to find the target windmc host tool | |
| checking whether to enable maintainer-specific portions of Makefiles no | |
| configure: creating ./config.status | |
| config.status: creating Makefile | |
| user@localhost:~/exper/exp5/gcc-build\$ | |
| | _ |

图 5.11 GCC 配置成功

在配置过程中,控制台会提示需要额外安装 M4、GMP、MPFR、MPC 软件包。首先 需要安装 M4 软件包,在软件中心中搜索 M4,然后单击"安装"按钮即可,如图 5.12 所示。



GCC 的安装

Linux 操作系统应用与开发教程

M4 安装完成之后,参考图 5.5 所示的页面分别下载 GMP、MPFR、MPC 的软件包, 链接地址为 ftp://mirrors.ustc.edu.cn/gnu/。选择下载 gmp-5.0、mpfr-3.0.0、mpc-1.0.2 等软件 包,下载结果如图 5.13 所示,然后参考 GCC 编译的步骤分别执行下面命令,进行编译安 装,安装结果如图 5.14 所示。

74

安装 GMP:

```
./gmp-5.0.0/configure --prefix=/usr/local/gmp-5.0
make
sudo make install
```

安装 MPFR:

```
./mpfr-3.0.0/configure --prefix=/usr/local/mpfr-3.0.0 --with-gmp=/usr/
local/gmp-5.0
make
sudo make install
```

安装 MPC:

```
./mpc-1.0.2/configure --prefix=/usr/local/mpc-1.0.2 --with-gmp=/usr/local/
gmp-5.0 --with-mpfr=/usr/local/mpfr-3.0.0
make
```

sudo make install



5.7 编译 GCC

这个过程虽然用户参与十分简单,但是却是一个漫长的过程,一般来说整个编译过程 需要持续一个小时以上,如果计算机性能不是很好,可能时间需要更长(要做好4个小时 以上的准备)。输入如下命令,编译 GCC。

make

编译完成后,会发现在目录 gcc-build 下会产生很多文件,如图 5.15 所示。

| tore_4o cas_4o exch_4o fa | dd_4o fsub_4o fand_4o fio | r_4o fxor_4o fnand_4o tas_4o ' |
|---------------------------------|---------------------------------|---------------------------------|
| ore_8o cas_8o exch_8o fad | d_8o fsub_8o fand_8o fior | _8o fxor_8o fnand_8o tas_8o l |
| ore_16o cas_16o exch_16o | fadd_16o fsub_16o fand_16 | o fior_16o fxor_16o fnand_16o t |
| d_16_1o store_16_1o cas_16_ | 1o_exch_16_1o_fadd_16_1o | fsub_16_1o fand_16_1o fior_16_1 |
| .o fnand_16_1o tas_16_1o | | |
| libtool: link: ranlib .libs/lib | atomic.a | |
| libtool: link: (cd ".libs" && | rm -f "libatomic.la" && ln -s " | /libatomic.la" "libatomic.la") |
| true DO=all multi-do # make | | |
| make[4]:正在离开目录 `/home/del | l/exper/exp5/gcc-build/x86 64-u | nknown-linux-gnu/libatomic' |
| make[3]:正在离开目录 `/home/del | l/exper/exp5/gcc-build/x86_64-u | nknown-linux-gnu/libatomic' |
| make[2]:正在离开目录 `/home/del | l/exper/exp5/gcc-build/x86_64-u | nknown-linux-gnu/libatomic' |
| make[1]:正在离开目录 `/home/del | l/exper/exp5/gcc-build' | 5. |
| user@localhost:~/exper/exp5/gcc | -build\$ ls | |
| build-x86_64-unknown-linux-gnu | prev-fixincludes | stage1-intl |
| сомраге | prev-gcc | stage1-libbacktrace |
| config.log | prev-intl | stage1-libcpp |
| config.status | prev-libbacktrace | stage1-libdecnumber |
| fixincludes | prev-libcpp | stage1-libiberty |
| gcc | prev-libdecnumber | stage1-lto-plugin |
| intl | prev-libiberty | stage1-x86_64-unknown-linux-gnu |
| libbacktrace | prev-lto-plugin | stage1-zlib |
| libcpp | prev-x86_64-unknown-linux-gnu | stage_current |
| libdecnumber | prev-zlib | stage_final |
| libiberty | serdep.tmp | stage_last |
| lto-plugin | stage1-fixincludes | x86_64-unknown-linux-gnu |
| Makefile | stage1-gcc | zlib |
| user@localhost:~/exper/exp5/gcc | -buildS | |

图 5.15 GCC 编译成功

5.8 安装 GCC

实际上,当完成了编译 GCC 时,已经可以使用 GCC 了。但是,经过编译操作的 gcc-build 目录中存在很多编译产生的中间文件,这对于使用 GCC 来说比较困难。因此,将编译好 的 GCC 按照配置时设定的方式复制到安装目录会方便使用。

安装的方法很简单,依然采用 make 命令。由于需要在系统使用的目录下建立目录、 复制文件,所以需要切换到根用户。整个安装过程会持续几分钟。

su *你的密码* make install

安装完成后,可以看到目录/usr/local/gcc-4.9.2 下出现了 GCC 安装后的文件结构,如 图 5.16 所示。其中,目录 bin 中存放了 GCC 的可执行文件,即命令; lib 中存放编译好的 库文件; include 中存放软件开发所用到的头文件。



图 5.16 GCC 安装成功



5.9 测试 GCC 安装结果

76

首先,进入目录/usr/local/gcc-4.9.2/bin,这里可以找到命令 gcc。输入"./gcc"和"./gcc -v",如果出现如图 5.17 所示的情况,说明 GCC 可以正常工作。

| root@localhost | :/home/dell/ | exper/exp5/gcc- | build# cd /usr/local/gcc-4.9.2 | /bin/ | |
|---|--|-----------------|--------------------------------|---------------------------------------|--|
| root@localhost | :/usr/local/ | gcc-4.9.2/bin# | ls | | |
| aot-compile | gcc-nm | gjar | grmiregistry | x86_64-unknown-linux-gnu-gcc | |
| C++ | gcc-ranlib | gjarsigner | gserialver | x86_64-unknown-linux-gnu-gcc-4.9.2 | |
| срр | gcj | gjavah | gtnameserv | x86_64-unknown-linux-gnu-gcc-ar | |
| g++ | gcj-dbtool | gkeytool | jcf-dump | x86_64-unknown-linux-gnu-gcc-nm | |
| gappletviewer | gcjh | gnative2ascii | jv-convert | x86_64-unknown-linux-gnu-gcc-ranlib | |
| gc-analyze | gcov | gorbd | rebuild-gcj-db | x86_64-unknown-linux-gnu-gcj | |
| gcc | gfortran | grmic | x86_64-unknown-linux-gnu-c++ | x86_64-unknown-linux-gnu-gfortran | |
| gcc-ar | gij | grmid | x86_64-unknown-linux-gnu-g++ | | |
| root@localhost 使用内建 specs | :/usr/local/ | gcc-4.9.2/bin# | ./gcc -v | | |
| COLLECT GCC=./ | åcc | | | | |
| COLLECT LTO WR | APPER=/usr/l | ocal/occ-4.9.2/ | libexec/acc/x86 64-unknown-lin | ux-gnu/4.9.2/lto-wrapper | |
| 目标:x86_64-unknown-linux-gnu | | | | | |
| 記賞为:/acc | -4.9.2/confi | oureprefix=/ | usr/local/gcc-4.9.2disable- | multilibwith-gmp=/usr/local/gmp-5.0 - | |
| -with-mpfr=/usr/local/mpfr-3.0.0with-mpc=/usr/local/mpc-1.0.2 | | | | | |
| 线程模型:postx | | | | | |
| acc 版本 4.9.2 (GCC) | | | | | |
| root@localhost | root@localhost:/usr/local/gcc-4.9.2/bin# | | | | |

图 5.17 初步检查 GCC

由于 GCC 并不在系统的查找目录(即变量 "PATH"所指定的目录,这些目录用来存放 Shell 命令)中,因此每次使用都需要输入新的 GCC 的完整路径。

有两种办法解决上面的问题:(1)可以将目录/usr/local/gcc-4.9.2/bin 添加到环境变量 PATH 中,使用如下的命令。但是这有一个缺点,就是你系统中原来的 GCC 和新的 GCC 的命令名称是相同的,会引起混淆,因此并不建议这样做。

export PATH=\$(PATH):/usr/local/gcc-4.9.2/bin

建立连接,在目录/usr/local/bin 或变量 PATH 所指定的其他目录中建立一个指向这些新的 GCC 命令的连接文件。

```
cd /usr/local/bin
ln -s /usr/local/gcc-4.9.2/bin/gcc gcc4.9
ln -s /usr/local/gcc-4.9.2/bin/g++ g++4.9
ln -s /usr/local/gcc-4.9.2/bin/gcj gcj4.9
```

建立好连接以后,可以通过如下几个命令查看是否建立成功。建立成功后的界面如图 5.18 所示。

```
ls *4.9
ls -1 *4.9
gcc4.9
```

上面只是从表面上查看所安装的 GCC 是否能够被访问。下面需要编写一个简单的 C 语言程序,测试它是否能够正常的编译。使用 su 命令回到普通用户状态,使用 cd 命令回 到实验目录,使用 vi 建立文件 test.c。

| root@localhost:/usr/local/bin# ln -s /usr/local/acc-4.9.2/bin/acc acc4.9 |
|--|
| root@localhost:/usr/local/bin# ln -s /usr/local/gcc-4.9.2/bin/g++ g++4.9 |
| rootAlocalhost:/usr/local/bin# ln -s /usr/local/gcc-4.9.2/bin/gci gci4.9 |
| rootelocalbost:/usr/local/bin# 1s *4 9 |
| |
| |
| |
| lrwxrwxrwx 1 root root 28 3月 28 08:50 g++4.9 -> /usr/local/gcc-4.9.2/bin/g++ |
| 'lrwxrwxrwx 1 root root 28 3月 28 08:50 gcc4.9 -> /usr/local/gcc-4.9.2/bin/gcc |
| lrwxrwxrwx 1 root root 28 3月 28 08:50 gcj4.9 -> /usr/local/gcc-4.9.2/bin/gcj |
| root@localhost:/usr/local/bin# gcc4.9 -v |
| 使用內建 specs |
| COLLECT GCC=acc4.9 |
| COLLECT LTO WRAPPER=/usr/local/gcc-4.9.2/libexec/gcc/x86 64-unknown-linux-gnu/4. |
| 9.2/lto-wrapper |
| 目标:x86 64-unknown-linux-gnu |
| 配置为:/gcc-4.9.2/configureprefix=/usr/local/gcc-4.9.2disable-multilib |
| with-gmp=/usr/local/gmp-5.0with-mpfr=/usr/local/mpfr-3.0.0with-mpc=/usr/ |
| local/mpc-1.0.2 |
| 线程模型:posix |
| |
| geet (local best (local /bis# |
| rocucocaciosc./dsi/cocac/bcn# |

图 5.18 为 GCC 建立访问链接

```
exit
cd ~/exper/exp5
vi test.c
```

编写一个简单的 Hello World 程序,该程序在终端中输出"Hello World!"。程序如下:

```
#include <stdio.h>
int main()
{
    printf("Hello World!\n");
    return 0;
}
```

保存文件,退出 vi。然后使用如下命令编译该程序并执行。

```
gcc4.9 test.c -o test
./test
```

程序正常运行,终端输出"Hello World!",如图 5.19 所示。

| root@localhost:/usr/local/bin# exit exit | |
|---|------------------|
| user@localhost:~\$ cd ~/exper/exp5/ | |
| user@localhost:~/exper/exp5\$ vi test.c | |
| user@localhost:~/exper/exp5\$ gcc4.9 test.c -o test | |
| user@localhost:~/exper/exp5\$ ls | |
| gcc-4.9.2 gcc-build gmp-5.0 mpc-1.0.2 mpfr-3.0 tes | t test.c test.c~ |
| user@localhost:~/exper/exp5\$./test | |
| Hello World! | |
| user@tocathost:~/exper/expss | |

图 5.19 GCC 成功编译 "Hello World"程序

5.10 思考与练习

- (1)为什么 Linux 下的软件大部分采用源代码的形式发布。
- (2) Linux 下的软件安装与 Windows 有何区别。
- (3)为什么要将 GCC 的目标路径设定在目录/usr/local/gcc-4.9.2?
- (4) 尝试使用源代码安装其他 Linux 软件包,如 ld 和 gdb。

77 第 5 章