

第17章 C语言编程工具

如果你需要下列问题的一个快速解决方案	请查阅节号
安装C语言编译器	17.2.1
安装基本C语言函数库	17.2.2
安装内核函数库	17.2.3
安装build管理器	17.2.4
安装C语言预处理器	17.2.5
安装文件比较工具软件	17.2.6
安装源代码补丁修补软件	17.2.7
寻找专业函数库	17.2.8
建立一个简单的制作文件	17.2.9

17.1 概述

许多Linux操作系统的用户是因为喜欢编程而选择这个操作系统的，而另外的一些人通过学习更多与他们工作有关的知识，通过为他们的系统编译自己下载来的源代码，也逐步成长为跨越了初级水平的程序员。对编程了解得越深，就越能体会到它对自己的重要性，就更能掌握如何对之进行升级，因为终究会有一天，用户自己下载的源代码就会要求用户使用它们。

17.1.1 C语言编程工具

C语言这样的编程语言是编译型而不是解释型的，也就是说最少需要有一个编译器才能发挥这种语言的作用。在此基础上，C语言的一个巨大优势是它可以使用共享函数库。这些函数库包括了大量的代码，使程序员不必为完成一个基本任务而从一点一滴开始做起。对于诸如计算机电话通讯、函数库之类的各种专业方向的编程工作来说，也有相应的专业软件库。有些专业软件库是随着发行版本的CD-ROM光盘发行的；另外一些则在因特网上，或者免费下载，或者收钱销售。

注意 C语言是一个很复杂的工具。本章并不是想教你如何使用它。我们的重点是向你介绍都有哪些工具以及如何找到并安装它们。

17.1.2 Linux操作系统中的C语言编程

如果你来自微软公司的Windows编程环境，熟悉使用项目（project）文件，那么就需要一些工夫来适应在Linux操作系统中的C语言编程。现在你必须学习如何手动编写一些制作文件（Makefile）。在17.2.9节中有为初学者准备的例子。

17.2 快速解决方案

17.2.1 安装C语言编译器

请按照下面的方法安装GNU的C语言编译器：

- 1) 以根用户身份登录进入系统。
- 2) 在CD-ROM光驱中放入发行版本的CD-ROM光盘。
- 3) 使用mount命令挂装发行版本的CD-ROM光盘。
- 4) 使用cd命令把路径切换到软件包目录中。两种发行版本中，这个目录分别是：
 - Red Hat发行版本CD-ROM光盘的/mnt/cdrom/RedHat/RPMS目录。
 - Caldera发行版本CD-ROM光盘的/mnt/cdrom/Packages/RPMS目录。
- 5) 输入“ls egcs*”命令查看哪个软件包是基本的egcs软件包。这个基本的软件包没有任何附带文件，看起来应该是egcs-1.1.2-12.i386.rpm的样子。
- 6) 输入“rpm -ivh egcs-1.1.2-12.i386.rpm”命令安装这个软件包。如果用户的版本号与这里列出的版本号不一致，请使用相应的内容替换它。

相关解决方案	请查阅节号
挂装到文件系统上	9.2.2
从文件系统上卸载	9.2.3
安装一个RPM软件包	15.2.1

17.2.2 安装基本C语言函数库

请按照下面的方法安装C语言编译器使用的基本函数库：

- 1) 以根用户身份登录进入系统。
- 2) 在CD-ROM光驱中放入发行版本的CD-ROM光盘。
- 3) 使用mount命令挂装发行版本的CD-ROM光盘。
- 4) 使用cd命令把路径切换到软件包目录中。两种发行版本中，这个目录分别是：
 - Red Hat发行版本CD-ROM光盘的/mnt/cdrom/RedHat/RPMS目录。
 - Caldera发行版本CD-ROM光盘的/mnt/cdrom/Packages/RPMS目录。
- 5) 输入“rpm -ivh glibc-devel*”命令安装标准C语言函数库。

17.2.3 安装内核函数库

内核源代码函数库是从内核本身开始编译许多C语言程序所必须的，请按照下面的方法安装内核源代码函数库：

- 1) 以根用户身份登录进入系统。
- 2) 在CD-ROM光驱中放入发行版本的CD-ROM光盘。
- 3) 使用mount命令挂装发行版本的CD-ROM光盘。
- 4) 使用cd命令把路径切换到软件包目录中。两种发行版本中，这个目录分别是：
 - Red Hat发行版本CD-ROM光盘的/mnt/cdrom/RedHat/RPMS目录。
 - Caldera发行版本CD-ROM光盘的/mnt/cdrom/Packages/RPMS目录。

5) 输入 “ rpm -ivh kernel-head* kernel-source* ” 命令安装包含了内核的C语言头文件和它的源代码的软件包。

17.2.4 安装build管理器

make程序的作用相当于一个管理器，用来管理许多以源代码形式下载的程序编译过程，请按照下面的方法安装 make 程序：

- 1) 以根用户身份登录进入系统。
- 2) 在CD-ROM光驱中放入发行版本的CD-ROM光盘。
- 3) 使用mount命令挂装发行版本的CD-ROM光盘。
- 4) 使用cd命令把路径切换到软件包目录中。两种发行版本中，这个目录分别是：
 - Red Hat发行版本CD-ROM光盘的/mnt/cdrom/RedHat/RPMS目录。
 - Caldera发行版本CD-ROM光盘的/mnt/cdrom/Packages/RPMS目录。
- 5) 输入 “ rpm -ivh make* ” 命令安装make程序。

17.2.5 安装C语言预处理器

C语言预处理器（cpp）提供了允许建立代替小段C语言代码的宏命令的工具、提供了条件编译器选项、提供了与头文件类似的各种include（包含）项目的定义和数据。请按照下面的方法安装C语言预处理器：

- 1) 以根用户身份登录进入系统。
- 2) 在CD-ROM光驱中放入发行版本的CD-ROM光盘。
- 3) 使用mount命令挂装发行版本的CD-ROM光盘。
- 4) 使用cd命令把路径切换到软件包目录中。两种发行版本中，这个目录分别是：
 - Red Hat发行版本CD-ROM光盘的/mnt/cdrom/RedHat/RPMS目录。
 - Caldera发行版本CD-ROM光盘的/mnt/cdrom/Packages/RPMS目录。
- 5) 输入 “ rpm -ivh cpp* ” 命令安装这个软件包。

17.2.6 安装文件比较工具软件

文件比较工具软件diffutils软件包包括了诸如diff程序(你可以使用这个程序建立源代码补丁)之类的应用程序，请按照下面的方法安装它：

- 1) 以根用户身份登录进入系统。
- 2) 在CD-ROM光驱中放入发行版本的CD-ROM光盘。
- 3) 使用mount命令挂装发行版本的CD-ROM光盘。
- 4) 使用cd命令把路径切换到软件包目录中。两种发行版本中，这个目录分别是：
 - Red Hat发行版本CD-ROM光盘的/mnt/cdrom/RedHat/RPMS目录。
 - Caldera发行版本CD-ROM光盘的/mnt/cdrom/Packages/RPMS目录。
- 5) 输入 “ rpm -ivh diffutils* ” 命令安装这个软件包。

17.2.7 安装源代码补丁修补软件

源代码补丁修补软件patch程序可以在源代码中打补丁，这样就可以对源代码进行升级。

请按照下面的方法安装它：

- 1) 以根用户身份登录进入系统。
- 2) 在CD-ROM光驱中放入发行版本的CD-ROM光盘。
- 3) 使用mount命令挂装发行版本的CD-ROM光盘。
- 4) 使用cd命令把路径切换到软件包目录中。两种发行版本中，这个目录分别是：
 - Red Hat发行版本CD-ROM光盘的/mnt/cdrom/RedHat/RPMS目录。
 - Caldera发行版本CD-ROM光盘的/mnt/cdrom/Packages/RPMS目录。
- 5) 输入“rpm -ivh patch*”命令安装这个软件包。

17.2.8 寻找专业函数库

如果用户对编制程序很投入，并对寻找一些专业化的函数库很感兴趣的话，可以试试下面给出的一些方法：

- 检查随发行版本而来的RPM包，看看那里有没有包括什么专业化的函数库。输入“ls *lib*”命令也是一个快速查看已安装函数库的好办法，然后可以输入“rpm -qip packagename”命令查看关于某个具体的软件包本身的信息。
- 查看列出在站点www.linuxprogramming.com上的函数库清单。
- 查看metalab.unc.edu/LDP/HOWTO/Commercial-HOWTO.html处的用于Linux操作系统的商业化HOW-TO文件，那里有可以通过商业渠道获得的用来完成各种任务的函数库。

17.2.9 建立一个简单的制作文件

本节假定用户知道如何编写必要的头(.h)文件和C语言代码(.c)文件，并且已经准备好开始进行其第一次在Linux操作系统中的程序制作过程。因为程序制作在调试阶段通常都会反复进行不止一次，所以现在编写一个制作文件会比每次都手工完成整个程序制作过程更有效率得多。请按照下面的方法建立一个简单的制作文件：

- 1) 把路径切换到用户打算编译的源代码所在的目录中去。
- 2) 输入“vi Makefile”命令使用vi文本编辑器程序建立这个Makefile制作文件。
- 3) 按下“i”键进入vi编辑器的插入编辑模式。
- 4) 一个制作文件中的每一个数据项都有如下所示的格式：

```
target : prerequisite (s)
      command
```

每一个制作文件都可以被分为一系列这样的语句。这些语句在制作文件中的输入顺序是不重要的，但是出于使用方便的考虑，许多人都会按照最复杂到最不复杂的顺序依次排列各个语句。需要建立的第一个目标(target)通常是“all”目标，这个目标比较特别，因为它不包含任何命令，看起来像下面所示的样子：

```
all : main_target
```

其中的main_target项目一般就是你想建立制作的二进制程序的名称。

- 5) 接下来开始建立main_target语句，这个语句看起来像下面所示的样子：

```
main_target : target1.o target2.o target3.o
      cc -o main_target target1.o target2.o target3.o
```

- 6) 现在，为每一个更小一些的目标建立语句，这些语句可能会是下面的样子：

```
target1.o : target1.c
```

```
cc -c target1.c
```

```
target2.o : target2.c
```

```
cc -c target2.c
```

```
target3.o : target3.c
```

```
cc -c target3.c
```

请记住，上面语句中的命令并不需要一定是一个 C 语言编译器的命令。它可以是在 make 程序制作处理阶段中需要执行的任何东西。

7) 输入 “ :wq ” 命令保存文件并退出。

8) 输入 “ rpm -q make ” 命令检查 make 软件包是否已经安装好了。如果它还没有安装好，请阅读 17.2.4 节中的内容。

9) 输入 “ make all ” 或者 “ make main_target ” 命令开始制作程序。