

嵌入式Linux 程序设计与应用案例

王进德 编著
王佩芳 姚庆良 改编



中国电力出版社
www.infopower.com.cn

全華



TP316.89

113

2007

嵌入式技术丛书

嵌入式Linux

程序设计与应用案例

王进德 编著

王佩芳 姚庆良 改编



中国电力出版社

www.infopower.com.cn

内 容 简 介

本书以 24 个典型实例、一个综合实例和 10 道习题的形式详细讲解了 Linux C 的编译方法、文件处理、进程、信号、程序通信、Linux RS-232 程序设计、Socket 程序设计、网络多任务与网络广播、监控分布式模块、CGI 等知识,旨在培养大家的 Linux C 和 Linux IPC 多元化程序设计的能力,了解 Linux 程序语言的应用。

本书讲解深入浅出,理论通俗易懂,适合对嵌入式 Linux 程序设计有兴趣的程序员、高等院校相关专业的老师和学生及培训班学员使用。

图书在版编目(CIP)数据

嵌入式 Linux 程序设计与应用案例 / 王进德编著. —北京: 中国电力出版社, 2007.5

ISBN 978-7-5083-5399-9

I. 嵌… II. 王… III. Linux 操作系统—程序设计 IV. TP316.89

中国版本图书馆 CIP 数据核字(2007)第 040594

北京市版权局著作权合同登记号 图字: 01-2006-5851 号

版权声明

本书中文简体字版由台湾全华科技图书股份有限公司独家授权,仅限于中国大陆境内出版发行,不含台湾、香港、澳门。

责任编辑: 夏华香

责任校对: 崔燕菊

责任印制: 李文志

丛 书 名: 嵌入式技术丛书

书 名: 嵌入式 Linux 程序设计与应用案例

编 著: 王进德

改 编: 王佩芳 姚庆良

出版发行: 中国电力出版社

地址: 北京市三里河路 6 号 邮政编码: 100044

电话: (010) 68362602 传真: (010) 68316497

印 刷: 北京市同江印刷厂

开本尺寸: 185 × 233

印 张: 17.25

字 数: 395 千字

书 号: ISBN 978-7-5083-5399-9

版 次: 2007 年 5 月北京第 1 版

印 次: 2007 年 5 月第 1 次印刷

印 数: 0001 — 4000

定 价: 28.00 元

敬告读者

本书封面贴有防伪标签,加热后中心图案消失

本书如有印装质量问题,我社发行部负责退换

版 权 专 有 翻 印 必 究

序

为迎接互联网蓬勃发展及嵌入式系统 (Embedded System) 时代的到来, 对现代工程师而言, 学习相关的嵌入式 Linux 程序设计课程, 是提升自我实力的最佳途径; 对从事职业技术教学的老师而言, 开设相关的嵌入式 Linux 程序设计课程, 培育更多的嵌入式系统设计人才, 是提升学校教学质量的不二法门。

本书编写的目的, 在于培养学生具备 Linux C 程序设计、Linux IPC 程序设计、Linux 通信程序设计及 Linux CGI 程序设计等能力, 并具有嵌入式 Linux 应用系统设计的能力; 本书所采用的教学平台为 Linux 操作系统、Red-Builder 开发软件、I-7000 分布式控制模块及 ECM-3412 嵌入式模块; 所采用的程序语言则为 Linux C 语言。

由于这是一本专为教师教学、学生学习而写的书, 所以我们在书中加入了 24 个实例、1 个综合实例及 10 道习题, 以方便教师讲授实例时使用。本书的所有范例都经笔者上机调试过, 力求程序范例的准确性, 且每个实例都附上参考程序解答, 以激发学生的应用能力。

本书之所以可以完成, 要感谢各界的帮助, 使笔者可以带领一群优秀的学生: 文洲、兆铭、金璋、瀚威、文昌、光贤、智玄、永迪、坤钟、彦宏、南亦、固霆、宏俊等, 共同进入 Linux 的世界, 回想起那段一起设计实例、一起编写程序、一起做综合实例的日子, 至今仍记忆犹新。

本书在编写期间, 承蒙孙炳阳顾问及台湾全华科技图书股份有限公司的鼓励与支持, 使本书得以顺利出版, 在此谨向他们致以最衷心地感谢。同时, 我更要将完成本书的喜悦之情献给我最亲爱的父母、最心爱的太太明珠, 及我最疼爱的两个儿子允翬、允亮。

虽然我们力求以最佳的书奉献给读者, 但若在阅读本书时发现有任何疏漏之处, 还请多加批评指正, 本人将不胜感激!

王进德

E-mail: jdwang66@ms7.hinet.net

本书阅读方法

- ※ 本书大部分的实例只需一台装有 Linux 操作系统的 PC 即可，Linux 操作系统版本不限，只要有 gcc 编译器，可编译 Linux C 程序即可。
- ※ 本书部分实例需要有 I-7000 分布式模块及刷卡机的配合。
- ※ 如果想做本书有关嵌入式系统的实习，则需有 Red-Builder 开发软件、嵌入式模块的配合，且 Linux 操作系统最好是 Red Hat Linux 7.3。
- ※ 教师授课时，可以用 1 小时时间介绍概念，1 小时时间讲解本书所附的实例程序，再以 1 小时时间给学生上机练习，或是另外出一个练习题给学生上机练习。
- ※ 本书附录 B 附有 10 道习题，教师可参考作为期中、期末考试之用。

目 录

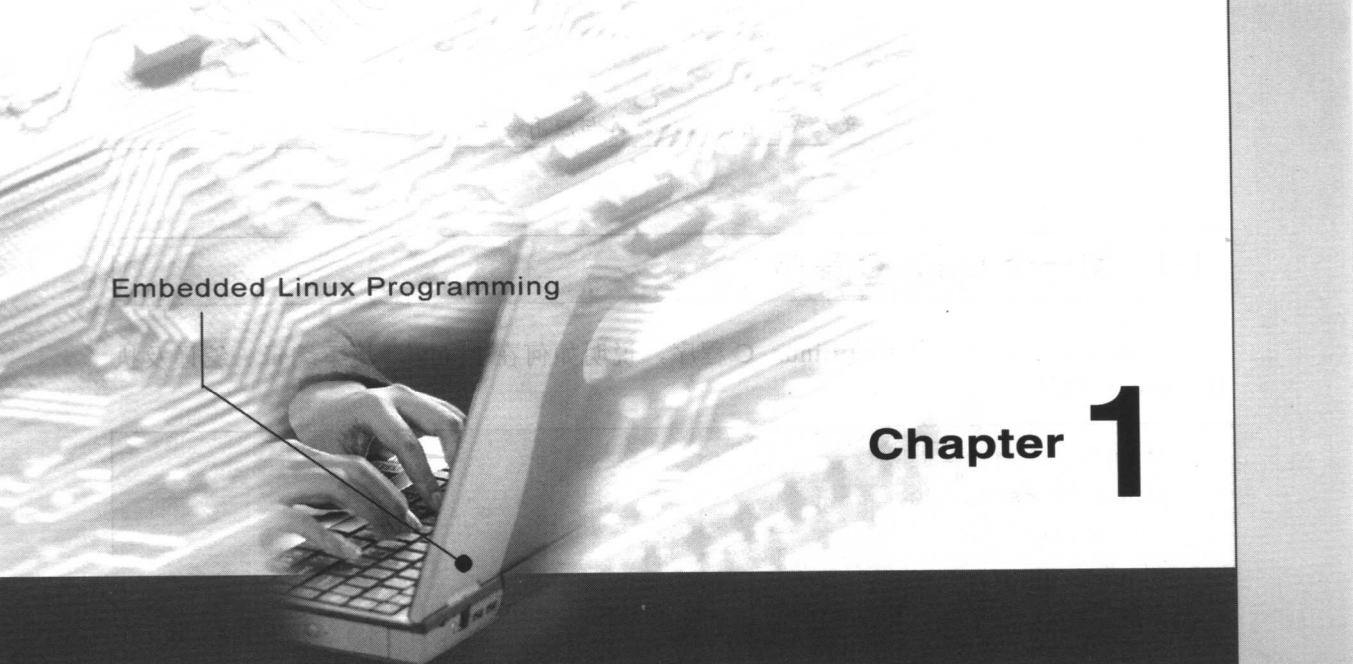
序

第 1 章	编译 Linux C.....	1
1-1	第一个 Linux C 程序.....	2
1-2	实例 1: 编译 Linux C.....	6
1-3	文件分割.....	7
1-4	实例 2: 编译多个 Linux 程序.....	9
1-5	使用 GNU gcc 编译器.....	12
1-6	Make 命令与 makefiles.....	12
1-7	实例 3: 建立 makefile.....	14
1-8	随机数的产生.....	16
1-9	makefile 变量的使用.....	17
1-10	实例 4: makefile 变量的使用.....	19
1-11	数组、指针的应用.....	22
1-12	本章重点.....	23
第 2 章	Linux 文件处理.....	25
2-1	Linux 文件系统.....	26
2-2	文件命名.....	27
2-3	文件权限.....	28
2-4	文件目录修改权限.....	29
2-5	文件权限屏蔽.....	30
2-6	实例 5: 改变系统的文件权限.....	31
2-7	低级文件处理函数.....	34
2-8	实例 6: Copy 程序.....	36
2-9	本章重点.....	39
第 3 章	进程.....	41
3-1	进程简介.....	42
3-2	进程的相关函数.....	43
3-3	实例 7: 复制进程.....	47
3-4	一些字符串处理函数.....	49
3-5	zombie 进程.....	50
3-6	实例 8: zombie 进程.....	52
3-7	函数指针.....	55
3-8	本章重点.....	56

第 4 章	信号	57
4-1	信号简介	58
4-2	信号的处理	59
4-3	实例 9: SIGINT 信号处理	60
4-4	结构数据类型	62
4-5	信号相关函数	63
4-6	实例 10: SIGALRM 信号处理	64
4-7	使用 typedef 来分割名称	66
4-8	本章重点	67
第 5 章	程序通信	69
5-1	管道	70
5-2	低级管道函数	70
5-3	实例 11: 读写 PIPE	72
5-4	高级管道函数	74
5-5	实例 12: popen 读写 pipe	75
5-6	二进制文件的访问	78
5-7	有名管道: FIFO	79
5-8	FIFO 函数	79
5-9	实例 13: 读写 FIFO	82
5-10	本章重点	85
第 6 章	Linux RS-232 程序设计	87
6-1	终端机界面	88
6-2	Termios 结构	89
6-3	终端机相关函数	96
6-4	实例 14: RS-232 通信	99
6-5	实例 15: 刷卡机应用	103
6-6	本章重点	106
第 7 章	Socket 程序设计	107
7-1	TCP/IP 简介	108
7-2	Socket 基本概念	108
7-3	IPv4 Socket 寻址结构	109
7-4	设置 IPv4 Socket 寻址结构	110
7-5	Socket 相关函数	111
7-6	设计 TCP 网络程序	114
7-7	TCP 输入输出函数	117
7-8	实例 16: TCP 程序设计	119
7-9	命令行自变量	125
7-10	设计 UDP 网络程序	125

7-11	UDP 输入输出函数	128
7-12	实例 17: UDP 程序设计	129
7-13	本章重点	135
第 8 章	网络多任务与网络广播	137
8-1	网络多任务	138
8-2	处理 SIGCHLD 信息	141
8-3	实例 18: 网络多任务程序	141
8-4	广播	148
8-5	设置 socket 属性	149
8-6	实例 19: UDP 广播程序	151
8-7	本章重点	157
第 9 章	Linux 监控分布式模块	159
9-1	分布式监控	160
9-2	分布式模块	160
9-3	Linux PC 监控分布式模块	161
9-4	I-7000 分布式模块命令	162
9-5	数字输入输出模块 7060	163
9-6	i7k 模块函数库	165
9-7	实例 20: 监控数字输入输出模块	168
9-8	模拟输入模块 7012	173
9-9	模拟输出模块 7021	174
9-10	实例 21: 监控 AD/DA 模块	176
9-11	实例 22: I-7000 模块综合练习	178
9-12	本章重点	181
第 10 章	Internet 程序设计: CGI	183
10-1	CGI 简介	184
10-2	环境变量	184
10-3	如何调用 CGI	185
10-4	在 CGI 中传送信息	186
10-5	窗体数据的编码原则	186
10-6	CGI 输出格式	187
10-7	CGI 译码程序	187
10-8	窗体应用: POST 方法	192
10-9	窗体应用: GET 方法	194
10-10	实例 23: 卡片阅读机 CGI 程序	196
10-11	本章重点	201
第 11 章	嵌入式 Linux 开发系统	203
11-1	嵌入式系统	204

11-2	嵌入式模块	204
11-3	嵌入式软件	205
11-4	使用 Red-Builder	206
11-5	构建嵌入式 Linux 系统	206
11-6	实例 24: 架设嵌入式 Server	212
11-7	本章重点	221
第 12 章	综合实例: Embedded Linux 监控分布式控制模块	223
12-1	研究目的	224
12-2	硬件配置	224
12-3	程序	227
12-4	将程序下载至嵌入式计算机	244
12-5	操作流程	252
12-6	综合实例实际设备图	255
12-7	结论	255
附录	本书习题	257



Embedded Linux Programming

Chapter 1

编译 Linux C

本章学习目标

- ※ 编辑、编译及运行 Linux C 程序。
- ※ 学习分割 Linux C 程序。
- ※ make 命令的使用。
- ※ makefile 文件的编写，自定义变量的使用方法。

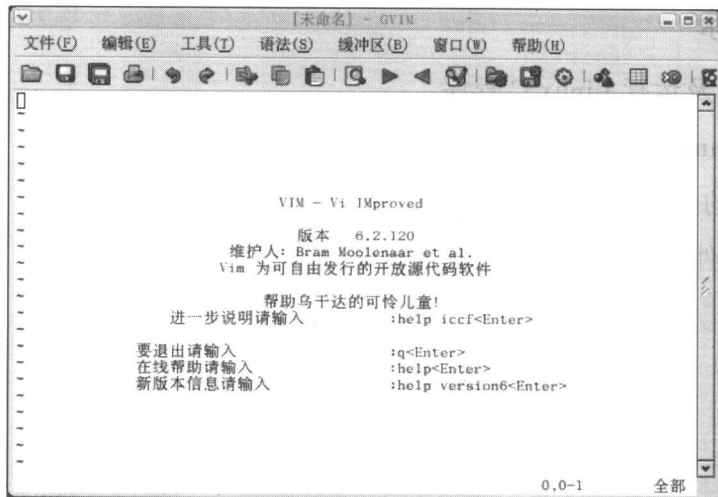
1-1 第一个 Linux C 程序

在本节中，以一个简单的 Linux C 程序，说明如何在 Linux 环境下编辑、编译及执行 Linux C 程序。

```
/* hello.c */
#include <stdio.h>
int main()
{
    printf("This is the first Linux C program.\n");
    return 0;
}
```

编写程序

在 Linux 操作系统下，打开文本文件编辑器，如 vim 来编辑 Linux C 程序。也可以在 Linux 的 X Window 下执行【主菜单】|【编程】| Vi Improved 命令，打开 vim 编辑器，打开后的界面如下图所示。



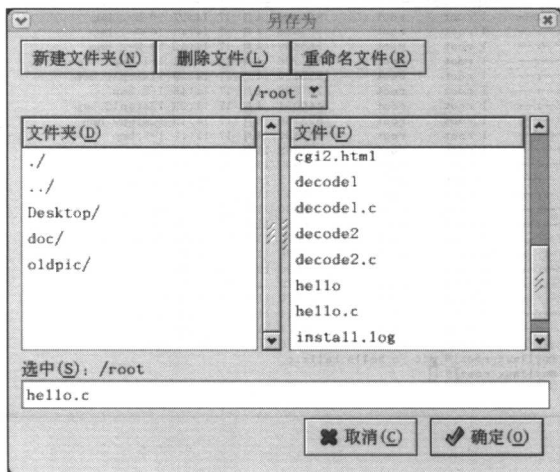
若要编辑文本文件，步骤如下：

依次按 ESC 键→i 键→输入文字内容

输入完 Linux C 程序后的界面如下图所示。



接着运行【文件】|【保存】命令，将程序保存起来，文件保存的对话框如下，在此我们将文件名取为“hello.c”，保存路径为“/root”。



编译 Linux C

要编译 Linux C 程序，先打开【终端机】，打开后输入“ls -l”命令，查看 hello.c 文件是否存在，界面如下图所示。

```

root@mylinux:~# ls -l
总用量 7806
-rw-r--r-- 1 root root 864854 4月 17 14:34 10-8-1.bmp
-rw-r--r-- 1 root root 864854 4月 17 14:37 10-8-2.bmp
-rw-r--r-- 1 root root 565866 4月 17 14:41 10-9-1.bmp
-rw-r--r-- 1 root root 565866 4月 17 14:42 10-9-2.bmp
-rw-r--r-- 1 root root 737206 4月 17 14:44 l-1.bmp
-rw-r--r-- 1 root root 825450 4月 17 14:29 l1step10.bmp
-rw-r--r-- 1 root root 737206 4月 17 14:45 l-2.bmp
-rw-r--r-- 1 root root 1440054 4月 17 14:23 l2step12.bmp
-rw-r--r-- 1 root root 825450 4月 17 14:13 l2step9.bmp
-rw-r--r-- 1 root root 467430 4月 17 14:47 l-3.bmp
-rw-r--r-- 1 root root 1363 11月 17 01:01 anaconda-ks.cfg
-rw-r--r-- 1 root root 693 11月 22 16:01 cgi1.html
-rw-r--r-- 1 root root 245 11月 21 17:14 cgi2.html
-rwxr-xr-x 1 root root 8055 11月 22 17:54 decode1
-rw-r--r-- 1 root root 3590 11月 22 17:53 decode1.c
-rwxr-xr-x 1 root root 8055 11月 22 16:03 decode2
-rw-r--r-- 1 root root 3276 11月 22 16:01 decode2.c
dwxr-xr-x 2 root root 4096 11月 18 17:07 Desktop
dwxr-xr-x 2 root root 4096 11月 21 16:48 doc
-rwxr-xr-x 1 root root 4679 4月 17 11:08 hello
-rw-r--r-- 1 root root 110 11月 21 16:09 hello.c
-rw-r--r-- 1 root root 49311 11月 17 01:00 install.log
-rw-r--r-- 1 root root 4471 11月 17 00:59 install.log.syslog
-rwxr-xr-x 2 root root 4096 4月 17 14:07 oldpic
root@mylinux:~#

```

接着输入如下命令，将 hello.c 程序编译成可执行文件 hello。

```
gcc -o hello hello.c
```

若程序没有错误，编译时不会出现错误信息，界面如下图所示。

```

root@mylinux:~# ls -l
总用量 7806
-rw-r--r-- 1 root root 864854 4月 17 14:34 10-8-1.bmp
-rw-r--r-- 1 root root 864854 4月 17 14:37 10-8-2.bmp
-rw-r--r-- 1 root root 565866 4月 17 14:41 10-9-1.bmp
-rw-r--r-- 1 root root 565866 4月 17 14:42 10-9-2.bmp
-rw-r--r-- 1 root root 737206 4月 17 14:44 l-1.bmp
-rw-r--r-- 1 root root 825450 4月 17 14:29 l1step10.bmp
-rw-r--r-- 1 root root 737206 4月 17 14:45 l-2.bmp
-rw-r--r-- 1 root root 1440054 4月 17 14:23 l2step12.bmp
-rw-r--r-- 1 root root 825450 4月 17 14:13 l2step9.bmp
-rw-r--r-- 1 root root 467430 4月 17 14:47 l-3.bmp
-rw-r--r-- 1 root root 1363 11月 17 01:01 anaconda-ks.cfg
-rw-r--r-- 1 root root 693 11月 22 16:01 cgi1.html
-rw-r--r-- 1 root root 245 11月 21 17:14 cgi2.html
-rwxr-xr-x 1 root root 8055 11月 22 17:54 decode1
-rw-r--r-- 1 root root 3590 11月 22 17:53 decode1.c
-rwxr-xr-x 1 root root 8055 11月 22 16:03 decode2
-rw-r--r-- 1 root root 3276 11月 22 16:01 decode2.c
dwxr-xr-x 2 root root 4096 11月 18 17:07 Desktop
dwxr-xr-x 2 root root 4096 11月 21 16:48 doc
-rwxr-xr-x 1 root root 4679 4月 17 11:08 hello
-rw-r--r-- 1 root root 110 11月 21 16:09 hello.c
-rw-r--r-- 1 root root 49311 11月 17 01:00 install.log
-rw-r--r-- 1 root root 4471 11月 17 00:59 install.log.syslog
-rwxr-xr-x 2 root root 4096 4月 17 14:07 oldpic
root@mylinux:~# gcc -o hello hello.c
root@mylinux:~#

```

运行程序

编译好 hello.c 程序后，若要执行 hello 文件，命令如下：

```
./hello
```

此时系统会出现如下的运行结果。

This is the first Linux C program.

终端机界面如下图所示：

```

root@mylinux:~# ls -l
total 12
-rw-r--r-- 1 root root 563846 4月 17 14:42 10-9-2.bmp
-rw-r--r-- 1 root root 737206 4月 17 14:44 1-1.bmp
-rw-r--r-- 1 root root 825430 4月 17 14:29 11step10.bmp
-rw-r--r-- 1 root root 737206 4月 17 14:45 1-2.bmp
-rw-r--r-- 1 root root 1440054 4月 17 14:23 12step12.bmp
-rw-r--r-- 1 root root 825430 4月 17 14:13 12step9.bmp
-rw-r--r-- 1 root root 467430 4月 17 14:47 1-3.bmp
-rw-r--r-- 1 root root 755046 4月 17 14:50 1-4.bmp
-rw-r--r-- 1 root root 755046 4月 17 14:51 1-5.bmp
-rw-r--r-- 1 root root 1383 11月 17 01:01 anaconda-ks.cfg
-rw-r--r-- 1 root root 993 11月 22 16:01 cgi1.html
-rw-r--r-- 1 root root 243 11月 21 17:14 cgi2.html
-rwxr-xr-x 1 root root 8055 11月 22 17:54 decode1
-rw-r--r-- 1 root root 3500 11月 22 17:53 decode1.c
-rwxr-xr-x 1 root root 8055 11月 22 16:03 decode2
-rw-r--r-- 1 root root 3276 11月 22 16:01 decode2.c
drwxr-xr-x 2 root root 4096 11月 18 17:07 Desktop
drwxr-xr-x 2 root root 4096 11月 21 16:48 doc
-rw-r--r-- 1 root root 4670 4月 17 14:51 hello
-rw-r--r-- 1 root root 110 11月 21 16:09 hello.c
-rw-r--r-- 1 root root 49311 11月 17 01:00 install.log
-rw-r--r-- 1 root root 4471 11月 17 00:59 install.log.syslog
-rw-r--r-- 2 root root 4096 4月 17 14:07 oldpic
[root@mylinux root]# gcc -o hello hello.c
[root@mylinux root]# ./hello
This is the first Linux C program.
[root@mylinux root]#

```

终端机常用命令

当在终端机下编译及运行 Linux C 程序时，常会使用到一些命令，如列出文件清单、建立新目录等，这些常用命令说明如下表所示。

常用命令	说明
pwd	显示当前工作目录
ls	列出当前工作目录的文件清单
ls -l	列出文件详细清单
mkdir 新目录名	建立新目录
cd 目录名	切换目录
cd ..	切换回上一层目录
rm 文件名	删除文件
rm -rf 目录名	删除目录及所属的所有文件
more 文件名	查看文件内容，若文件内容很长，按 Enter 键会一行一行地显示，按 Space 键会一页一页显示
cat -n 文件名	查看文件内容，并将显示内容加注行号

续表

常用命令	说明
touch 文件名	文件名称若为新的文件名，表示要建立这个新文件；若为旧文件名，表示要更改文件的编辑日期
cp 源文件名 目标文件名	复制文件
mv 源文件名 目标文件名	重新命名文件
ln 源文件名 连接文件名	硬连接文件，以指针指向源文件名，若源文件被删除，连接文件的内容并不会被删除
ln-s 源文件名 连接文件名	软连接文件，以指针指向源文件名，若源文件被删除，连接文件的内容也会消失
locate 文件名	列出文件所属的路径
find -name 文件名	由工作目录起，列出文件所属的路径

1-2 实例 1：编译 Linux C

目的

- (1) 练习 Linux C 程序的编译与执行。
- (2) 复习 C 语言的 printf()与 scanf()函数。
- (3) 复习 C 语言的 if-else if 结构。

要求

系统首先会要求输入一个成绩，若成绩大于等于 90 分，会显示“甲”；若成绩介于 80~90 分，会显示“乙”；若成绩介于 70~80，会显示“丙”；若成绩介于 60~70，会显示“丁”；而若成绩小于 60，则会显示“不及格”。

程序

```
/* score.c */
#include <stdio.h>
int main()
{
```

```
int score;
printf("输入成绩: ");
scanf("%d",&score);
if (score >= 90)
    printf("甲\n");
else if (score >= 80)
    printf("乙\n");
else if (score >= 70)
    printf("丙\n");
else if (score >= 60)
    printf("丁\n");
else
    printf("不及格\n");
}
```

编译与运行

```
[mylinux@mylinux ex01]$ gcc -o score score.c
```

```
[mylinux@mylinux ex01]$ ./score
```

输入成绩: 77

丙

```
[mylinux@mylinux ex01]$ ./score
```

输入成绩: 55

不及格

1-3 文件分割

当在建立大型程序时, 如果把已经建好的函数分割出来, 应用在其他程序里面, 可以帮助大型程序的开发。在本节中, 要说明如何分割一个 Linux C 程序, 将程序中的函数分割出来。

考虑如下的 Linux C 程序, 此程序会要求用户输入两个整数, 接着调用 min() 函数, 输出 2 个数的最小值。

```
#include <stdio.h>
int min(int x, int y);
```

```
int main()
{
    int n1,n2,n3;
    printf("输入第一个整数: ");
    scanf("%d",&n1);
    printf("输入第二个整数: ");
    scanf("%d",&n2);
    n3=min(n1,n2);
    printf("最小值为 %d\n",n3);
    return 0;
}
int min(int x, int y)
{
    if (x < y)
    return x;
    else
    return y;
}
```

上述程序，若要将 min()函数分割出来，可将该程序分割成下列 3 个 Linux C 程序。

mymin.h

此文件包含函数类型的说明，程序代码如下。

```
/* mymin.h */
int min(int x, int y);
```

mymin.c

此文件包含 min()函数的定义，程序代码如下。

```
/* mymin.c */
int min(int x, int y)
{
    if (x < y)
    return x;
    else
    return y;
}
```