

嵌入式 应用程序设计综合教程 微课版

华清远见嵌入式学院 刘洪涛 熊家 主编
王建秋 安东 副主编



扫一扫 立即观看配套视频



图书学习 + 在线学习

27个视频配合图书同步讲解，
帮助读者轻松掌握每一个知识点



学习拓展

华清创客学院资源丰富，更多学
习资料帮助读者提高开发能力



在线答疑

如遇到学习问题，可在线提问，
辅导老师第一时间答疑



中国工信出版集团



人民邮电出版社
POSTS & TELECOM PRESS

嵌入式 应用程序设计综合教程 微课版

华清远见嵌入式学院 刘洪涛 熊家 主编
王建秋 安东 副主编

人民邮电出版社
北京

图书在版编目 (CIP) 数据

嵌入式应用程序设计综合教程：微课版 / 刘洪涛，熊家主编. -- 3版. -- 北京：人民邮电出版社，2017.2
高等院校嵌入式人才培养规划教材
ISBN 978-7-115-44448-6

I. ①嵌… II. ①刘… ②熊… III. ①Linux操作系统—高等学校—教材 IV. ①TP316.89

中国版本图书馆CIP数据核字(2016)第316311号

内 容 提 要

本书结合大量实例，讲解了嵌入式应用程序设计的核心概念和基本方法。主要内容包括标准 I/O 编程、文件 I/O 编程、多任务编程、进程间通信、多线程编程、Linux 网络编程等。重视应用是贯穿全书的最大特点，本书在各章多处设置了微课，帮助读者攻克重点与难点。

本书可以作为院校嵌入式相关专业和计算机相关专业的教材，也可以作为计算机软硬件培训班教材，还可供嵌入式研究方向的专业人员和广大计算机爱好者自学使用。

-
- ◆ 主 编 华清远见嵌入式学院 刘洪涛 熊 家
 - 副 主 编 王建秋 安 东
 - 责任编辑 桑 珊
 - 执行编辑 左仲海
 - 责任印制 焦志炜
 - ◆ 人民邮电出版社出版发行 北京市丰台区成寿寺路 11 号
 - 邮编 100164 电子邮件 315@ptpress.com.cn
 - 网址 <http://www.ptpress.com.cn>
 - 三河市海波印务有限公司印刷
 - ◆ 开本：787×1092 1/16
 - 印张：13.5 2017 年 2 月第 3 版
 - 字数：240 千字 2017 年 2 月河北第 1 次印刷
-

定价：42.00 元

读者服务热线：(010)81055256 印装质量热线：(010)81055316
反盗版热线：(010)81055315

前言

Foreword

随着消费群体对产品要求的日益提高,嵌入式技术在机械器具制造、电子产品制造,通信、信息服务等领域得到了大显身手的机会,应用日益广泛,相应地,企业对嵌入式人才的需求也越来越多。近几年来,很多院校纷纷开设了嵌入式专业或方向。虽然目前市场上的嵌入式开发相关书籍比较多,但很多是针对有一定基础的行业内研发人员而编写的,并不完全符合学校的教学要求。学校教学需要一套充分考虑学生现有知识基础和接受程度、明确各门课程教学目标的、便于学校安排课时的嵌入式专业教材。

针对教材缺乏的问题,我们以多年来在嵌入式工程技术领域内人才培养、项目研发的经验为基础,汇总了近几年积累的数百家企业对嵌入式研发相关岗位的真实需求,调研了数十所开设嵌入式专业的院校的课程设置情况、学生特点和教学用书现状。经过细致的整理和分析,对专业技能和基本知识进行合理划分,我们于2013年编写了这套高等院校嵌入式人才培养规划教材,包括以下4本。

《嵌入式操作系统(Linux篇)(微课版)》

《嵌入式Linux C语言程序设计基础教程(微课版)》

《ARM嵌入式体系结构与接口技术(Cortex-A9版)(微课版)》

《嵌入式应用程序设计综合教程(微课版)》

经过了3年,嵌入式行业发生了巨大变化,产品也得到了升级换代,同时,高等院校嵌入式专业日臻成熟,首批教材有些已无法满足新的需要,所以本次编写对原有教材进行修订。

全书共7章,内容涵盖嵌入式Linux应用开发的主要方面。

第1章介绍Linux标准I/O编程,让读者了解用户编程接口(API)和系统调用之间的关系并掌握基本的文件访问方法。

第2章介绍Linux文件I/O编程,分析了标准I/O和文件I/O的区别,重点讲解文件描述符的含义和具体的文件I/O编程接口。

第3章介绍Linux多任务编程,主要讲解Linux中进程和线程的区别和联系、如

何创建多进程以及守护进程。

第4章介绍Linux进程间通信,主要讲解几种常用的进程通信方法,包括管道通信、信号通信、共享内存、消息队列等。

第5章介绍Linux多线程编程,主要讲解Linux环境下的多线程编程方法及注意事项。

第6章介绍Linux网络编程基础,主要讲解Linux环境下的网络编程方法,涉及网络体系结构、TCP编程、UDP编程和服务模型等。

第7章介绍Linux高级网络编程,主要讲解网络超时检测、广播、组播和UNIX域套接字的基本编程方法。

本书由刘洪涛、熊家、王建秋、安东合作编写。本书的完成需要感谢华清远见嵌入式学院,教材内容参考了学院与嵌入式企业需求无缝对接的、科学的专业人才培养体系。同时,嵌入式学院从业或执教多年的行业专家团队也对教材的编写工作做出了贡献,季久峰、贾燕枫、关晓强、刘金格等在书稿的编写过程中认真阅读了所有章节,提供了大量在实际教学中积累的重要素材,对教材结构、内容提出了中肯的建议,并在后期审校工作中提供了很多帮助,在此表示衷心的感谢。

本书所有源代码、PPT课件、教学素材等辅助教学资料,请到人民邮电出版社教育社区(www.ryjiaoyu.com)免费下载。

由于作者水平所限,书中不妥之处在所难免,恳请读者批评指正。对于本书的批评和建议,可以发到www.embedu.org技术论坛。

编 者

2016年11月

平台支撑 Platform

华清创客学院 (www.makeru.com.cn) 是一家创客 O2O 在线教育平台, 由国内高端 IT 培训领导品牌华清远见教育集团鼎力打造。学院依托于华清远见教育集团在高端 IT 培训行业积累的十多年教学及研发经验, 以及上百位优秀讲师资源, 专注为用户提供高端、前沿的 IT 开发技术培训课程。以就业为导向, 以提高开发能力为目标, 努力让每一位用户在这里学到真本领, 为用户成为嵌入式、物联网、智能硬件时代的技术专家助力!

一、我们致力于这样的发展理念

我们有一种情怀: 为中国、为世界智能化变革的发展培养更多的优秀人才。

我们有一种坚持: 坚持做专业教育、做良心教育、做受人尊敬的职业教育。

我们有一种变革: 在互联网高速发展的时代, 打造“互联网+教育”模式下的 IT 人才终身学习教学体系。

华清创客学院
makeru.com.cn

课程 进入你感兴趣的课程 搜索

学习记录 查看购物车

首页 课程库 精品直播 直播课 学习路线图 品牌课 技术问答 会员福利

VIP会员低至28元啦

一顿午餐钱就能学习2000元的课程, 巨划算!

金牌讲师授课, 会员课程周周更新!

会员权限: 抢先参加直播课程、辅导老师在线答疑、优先获得课程优惠...

咨询客服
官方微博
官方微信
帮助中心

产品拆解课堂



华清智能WiFi小车开发技术大...
198人学习 | 1人评论



小米手环软硬件开发技术解析
198人学习 | 1人评论



WiFi音箱产品拆解
918人学习 | 1人评论



电子血压计拆解解析
918人学习 | 2人评论

二、我们致力于提供这样的学习方式

1. 多元化的课程学习体系

(1) 学习模式的多元化。您可以根据自身的实际情况选择 3 种学习模式，在线学习、线下报班学习、线上线下结合式学习。每一种模式都有专业的学习路线指导，并有辅导老师悉心答疑，对于学完整套课程的同学有高薪就业职位推荐。

(2) 学习内容的多元化。我们提供基础知识课程、会员提升课程、流行技术精品套餐课程、就业直通车课程、职业成长课程等丰富的课程体系。不管您是职场“小白”还是 IT 从业人员，都可以在这里找到您的学习路线。

(3) 直播课程的多元化。包括基础类、技术问答类、IT 人的职业素养类、IT 企业的面试技巧类、IT 人的职业发展规划类、智能硬件产品解析类。

2. 大数据支撑下的过程化学习模式

(1) 自主学习课程。我们提供习题练习模式支持您的学习，每章学习完成后都有配套的练习题助您检验学习成果，整个课程学习完成后，系统会自动根据您的答题情况，分析出您对课程的整体掌握程度，帮助您随时掌握自身学习情况。

(2) 报班模式下的学习课程。系统会根据您选取的班级，为您制定详细的阶段化学习路线，学习路线采用游戏通关模式，课程章节有考核测验、课程有综合检验、每阶段有项目开发任务。学习过程全程通过大数据进行数据分析，帮助您与班主任随时了解您的课程学习掌握程度，班主任会定期根据您的学习情况开放直播课程，为您的薄弱环节进行细致讲解。考核不合格则无法通过关卡进入下一个环节。

三、我们致力于提供这样的服务保障

1. 与企业岗位的无缝对接

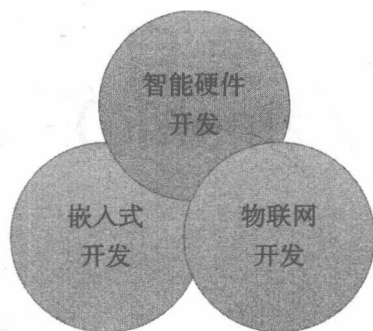
(1) 在线课程经过企业实体培训检验。华清远见是国内最早的高端 IT 定制培训服务机构，在业界享有盛誉。每年我们都会为不同的企业“量身订制”满足企业需求的高端企业内训课程，曾先后为 Intel、松下、通用电器、摩托罗拉、ST 意法半导体、三星、华为、大唐电信等众多知名企业进行员工内训。

(2) 拥有独立的自主研发中心。为开发和培训提供技术和产品支持，已经研发多款智能硬件产品、实验平台、实验箱等设备，并与中南大学、中国科学技术大学等高校共建嵌入式、物联网实验室。目前已经公开出版 80 多本教材，深受读者的欢迎。

(3) 平台提供企业招聘通道。学员可在线将自己的学习成果全部展现给企业 HR，增加您进入大型企业的机会。众多合作企业定期发布人才需求，还有企业上门招聘，全国 11 大城市就业推荐。

2. 丰富的课程资源

华清创客学院紧跟市场需求，全新录制高质量课程，深入讲解当下热的开发技术，包括嵌入式、Android、物联网、智能硬件课程（VR/AR、智能手表、智能小车、无人机等），希望我们的课程能帮您抓住智能硬件时代的发展机遇，打开更广阔的职业发展空间。



3. 强大的师资队伍

由华清远见金牌讲师团队+技术开发“大牛”组成的上百人讲师团队，有着丰富的开发与培训经验，其中不乏行业专家和企业项目核心开发者。

4. 便捷的学习方式

下载学院 APP 学习，不论您是在学校、家里还是外面，都可以随时随地学习。与教材配套使用，利用碎片时间学习，提升求职就业竞争力！

5. 超值的会员福利

会员可免费观看学院 70% 的课程，还可优先参加直播课程、新课程上线抢先试学、学习积分翻倍等活动，并有机会免费参加线下体验课。

四、我们期待您的加入

欢迎关注华清创客学院官网 www.makeru.com.cn，见证我们的成长。期待您的加入，愿与您一起打造未来 IT 人的终身化学习体系。

本书配套课程视频观看方法：手机上注册华清创客学院，扫描二维码即可观看课程视频；或在计算机上搜索书名，查找配套课程视频。

目录

Contents

第1章 Linux 标准 I/O 编程	1
1.1 Linux 系统调用和用户程序编程接口	2
1.1.1 系统调用	2
1.1.2 用户程序编程接口	2
1.2 Linux 标准 I/O 概述	3
1.2.1 标准 I/O 的由来	3
1.2.2 流的含义	3
1.3 标准 I/O 编程	4
1.3.1 流的打开	4
1.3.2 流的关闭	5
1.3.3 错误处理	6
1.3.4 流的读写	7
1.3.5 流的定位	11
1.3.6 格式化输入/输出	12
1.4 实验内容	13
1.4.1 文件的复制	13
1.4.2 循环记录系统时间	14
小结	16
思考与练习	16
第2章 Linux 文件 I/O 编程	17
2.1 Linux 文件 I/O 概述	18
2.1.1 POSIX 规范	18
2.1.2 虚拟文件系统	18
2.1.3 文件和文件描述符	19
2.1.4 文件 I/O 和标准 I/O 的区别	20
2.2 文件 I/O 操作	20
2.2.1 文件打开和关闭	21
2.2.2 文件读写	23
2.2.3 文件定位	24
2.2.4 文件锁	27
2.3 实验内容——生产者 and 消费者	33
小结	41

思考与练习	41
第3章 Linux 多任务编程	42
3.1 Linux 下多任务机制的介绍	43
3.1.1 任务	43
3.1.2 进程	43
3.1.3 线程	49
3.2 进程编程	50
3.2.1 进程编程基础	50
3.2.2 Linux 守护进程	62
3.3 实验内容——编写多进程程序	71
小结	76
思考与练习	76
第4章 Linux 进程间通信	77
4.1 Linux 下进程间通信概述	78
4.2 管道通信	79
4.2.1 管道简介	79
4.2.2 无名管道系统调用	80
4.2.3 有名管道	83
4.3 信号通信	87
4.3.1 信号概述	87
4.3.2 信号发送与设置	89
4.4 信号量	96
4.4.1 信号量概述	96
4.4.2 信号量编程	97
4.5 共享内存	102
4.6 消息队列	109
4.7 实验内容	115
4.7.1 有名管道通信实验	115
4.7.2 共享内存实验	121
小结	127
思考与练习	127

第5章 Linux 多线程编程 128

5.1 线程基本编程	129
5.2 线程之间的同步与互斥	133
5.2.1 互斥锁线程控制	133
5.2.2 信号量线程控制	134
5.3 线程属性	139
5.4 实验内容——多线程编程	143
小结	150
思考与练习	150

第6章 Linux 网络编程基础 151

6.1 网络体系结构	152
6.1.1 OSI 模型和 TCP/IP 模型	152
6.1.2 TCP/IP 模型特点	153
6.1.3 TCP 和 UDP	154
6.2 网络基础编程	157
6.2.1 套接字概述	157
6.2.2 IP 地址	158
6.2.3 端口	160
6.2.4 字节序	161
6.2.5 TCP 编程	161
6.2.6 UDP 编程	170

6.3 服务器模型	174
6.3.1 循环服务器 (TCP)	174
6.3.2 并发服务器 (TCP)	180
小结	183
思考与练习	183

第7章 Linux 高级网络编程 184

7.1 网络超时检测	185
7.1.1 套接字接收超时检测	185
7.1.2 定时器超时检测	188
7.2 广播	190
7.2.1 广播地址	191
7.2.2 广播包的发送和接收	191
7.3 组播	195
7.3.1 组播地址	195
7.3.2 组播包的发送和接收	195
7.4 UNIX 域套接字	199
7.4.1 本地地址	199
7.4.2 UNIX 域流式套接字	200
7.4.3 UNIX 域用户数据报套接字	203
小结	206
思考与练习	206

第1章

Linux标准I/O编程

本章主要内容

Linux系统调用和用户程序
编程接口 (API)。■
Linux标准I/O概述。■
标准I/O编程。■

■ 用户在应用开发过程中会经常需要访问文件。Linux 下读写文件的方式有两大类：标准 I/O 和文件 I/O。其中，标准 I/O 是最常用也是最基本的内容，希望读者好好掌握。

1.1 Linux 系统调用和用户程序编程接口

1.1.1 系统调用

操作系统负责管理和分配所有的计算机资源。为了更好地服务于应用程序，操作系统提供了一组特殊接口——系统调用。通过这组接口，用户程序可以使用操作系统内核提供的各种功能，如分配内存、创建进程、实现进程之间的通信等。

为什么不允许程序直接访问计算机资源？答案是不安全。单片机开发中，由于不需要操作系统，所以开发人员可以编写代码直接访问硬件。而在嵌入式系统中通常都要运行操作系统，程序访问资源的方式就发生了改变。操作系统基本上都支持多任务，即同时可以运行多个程序。如果允许程序直接访问系统资源，肯定会带来很多问题。因此，所有软硬件资源的管理和分配都由操作系统负责。程序要获取资源（如分配内存、读写串口）必须通过操作系统来完成，即用户程序向操作系统发出服务请求，操作系统收到请求后执行相关的代码来处理。

用户程序向操作系统提出请求的接口就是系统调用。所有的操作系统都会提供系统调用接口，只不过不同的操作系统提供的系统调用接口各不相同。Linux 系统调用接口非常精简，它继承了 UNIX 系统调用中最基本和最有用的部分。这些系统调用按照功能大致可分为进程控制、进程间通信、文件系统控制、存储管理、网络管理、套接字控制、用户管理等。

1.1.2 用户程序编程接口

前面提到利用系统调用接口程序可以访问各种资源，但在实际开发中程序并不直接使用系统调用接口，而是使用用户程序编程接口（API）。为什么不直接使用系统调用接口呢？原因如下。

- （1）系统调用接口功能非常简单，无法满足程序的需求。
- （2）不同操作系统的系统调用接口不兼容，程序移植时工作量大。

用户程序编程接口通俗的解释就是各种库（最重要的就是 C 库）中的函数。为了提高开发效率，C 库中实现了很多函数。这些函数实现了常用的功能，供程序员调用。这样一来，程序员不需要自己编写这些代码，直接调用库函数就可以实现基本功能，提高了代码的复用率。使用用户程序编程接口还有一个好处：程序具有良好的可移植性。几乎所有的操作系统上都实现了

C 库，所以程序通常只需要重新编译一下就可以在其他操作系统下运行。

用户程序编程接口（API）在实现时，通常都要依赖系统调用接口。例如，创建进程的 API 函数 `fork()` 对应于内核空间的 `sys_fork()` 系统调用。很多 API 函数需要通过多个系统调用来完成其功能。还有一些 API 函数不需要调用任何系统调用。

在 Linux 中，用户程序编程接口（API）遵循了在 UNIX 中最流行的应用编程界面标准——POSIX 标准。POSIX 标准是由 IEEE 和 ISO/IEC 共同开发的标准系统，该标准基于当时的 UNIX 实践和经验，描述了操作系统的系统调用编程接口（实际上就是 API），用于保证应用程序可以在源代码一级上在多种操作系统上移植运行。这些系统调用编程接口主要是通过 C 库（`libc`）实现的。

1.2 Linux 标准 I/O 概述

1.2.1 标准 I/O 的由来

标准 I/O 指的是 ANSI C 中定义的用于 I/O 操作的一系列函数。

只要操作系统中安装了 C 库，标准 I/O 函数就可以调用。换句话说，如果程序中使用的是标准 I/O 函数，那么源代码不需要修改就可以在其他操作系统下编译运行，具有更好的可移植性。

除此之外，使用标准 I/O 可以减少系统调用的次数，提高系统效率。标准 I/O 函数在执行时也会用到系统调用。在执行系统调用时，Linux 必须从用户态切换到内核态，处理相应的请求，然后再返回到用户态。如果频繁地执行系统调用会增加系统的开销。为了避免这种情况，标准 I/O 使用时在用户空间创建缓冲区，读写时先操作缓冲区，在合适的时机再通过系统调用访问实际的文件，从而减少了使用系统调用的次数。

1.2.2 流的含义

标准 I/O 的核心对象就是流。当用标准 I/O 打开一个文件时，就会创建一个 `FILE` 结构体描述该文件（或者理解为创建一个 `FILE` 结构体和实际打开的文件关联起来）。我们把这个 `FILE` 结构体形象地称为流。标准 I/O 函数都基于流进行各种操作。



标准 I/O 概述

标准 I/O 中的流的缓冲类型有以下 3 种。

(1) 全缓冲。在这种情况下,当填满标准 I/O 缓冲区后才进行实际 I/O 操作。对于存放在磁盘上的普通文件,用标准 I/O 打开时默认是全缓冲的。当缓冲区已满或执行 flush 操作时才会进行磁盘操作。

(2) 行缓冲。在这种情况下, 当在输入和输出中遇到换行符时执行 I/O 操作。标准输入流和标准输出流就是使用行缓冲的典型例子。

(3) 无缓冲。不对 I/O 操作进行缓冲, 即在对流的读写时会立刻操作实际的文件。标准出错流是不带缓冲的, 这就使得出错信息可以立刻显示在终端上, 而不管输出的内容是否包含换行符。

下面讨论具体函数时，请读者注意区分以上的 3 种不同情况。

1.3 标准 I/O 编程

本节所要讨论的 I/O 操作都是基于流的，它符合 ANSI C 的标准。有一些函数读者已经非常熟悉了（如 `printf()`、`scanf()` 函数等），因此本节中仅介绍最常用的函数。

1.3.1 流的打开

使用标准 I/O 打开文件的函数有 `fopen()`、`fdopen()` 和 `freopen()`。它们可以以不同的模式打开文件，都返回一个指向 `FILE` 的指针，该指针指向对应的 I/O 流。此后，对文件的读写都通过这个 `FILE` 指针来进行。其中，`fopen()` 可以指定打开文件的路径和模式，`fdopen()` 可以指定打开的文件描述符和模式，而 `freopen()` 除可指定打开的文件和模式外，还可指定特定的 I/O 流。

fopen()函数语法要点如表 1.1 所示。

表 1.1 fopen()函数语法要点

所需头文件	#include <stdio.h>
函数原型	FILE * fopen (const char * path, const char * mode) ;
函数参数	path: 包含要打开的文件路径及文件名
	mode: 文件打开方式, 详细信息参考表 1.2
函数返回值	成功: 指向 FILE 的指针
	失败: NULL

其中，mode 用于指定打开文件的方式。表 1.2 说明了 fopen() 中 mode 的各种取值。

表 1.2 mode 取值说明

r 或 rb	打开只读文件，该文件必须存在
r+或 r+b	打开可读写的文件，该文件必须存在
w 或 wb	打开只写文件，若文件存在则文件长度为 0，即会擦写文件以前的内容；若文件不存在则建立该文件
w+或 w+b	打开可读写文件，若文件存在则文件长度为 0，即会擦写文件以前的内容；若文件不存在则建立该文件
a 或 ab	以附加的方式打开只写文件。若文件不存在，则会建立该文件；如果文件存在，写入的数据会被加到文件尾，即文件原先的内容会被保留
a+或 a+b	以附加方式打开可读写的文件。若文件不存在，则会建立该文件；如果文件存在，写入的数据会被加到文件尾，即文件原先的内容会被保留



在每个选项中加入 b 字符用来告诉函数库打开的文件为二进制文件，而非纯文本文件。不过在 Linux 系统中会忽略该符号。

当用户程序运行时，系统自动打开了 3 个流：标准输入流 stdin、标准输出流 stdout 和标准错误流 stderr。stdin 用来从标准输入设备(默认是键盘)中读取输入内容；stdout 用来向标准输出设备(默认是当前终端)输出内容；stderr 用来向标准错误设备(默认是当前终端)输出错误信息。



流的打开与关闭

1.3.2 流的关闭

关闭流的函数为 fclose()，该函数将流的缓冲区内的数据全部写入文件中，并释放相关资源。

fclose()函数语法要点如表 1.3 所示。

表 1.3 fclose()函数语法要点

所需头文件	#include <stdio.h>
函数原型	int fclose (FILE * stream) ;
函数参数	stream: 已打开的流指针
函数返回值	成功: 0
	失败: EOF

程序结束时会自动关闭所有打开的流。

1.3.3 错误处理



标准 I/O 函数执行时如果出现错误，会把错误码保存在全局变量 `errno` 中。程序员可以通过相应的函数打印错误信息。

错误处理相关函数 `perror()` 语法要点如表 1.4 所示。

表 1.4 `perror()` 函数语法要点

所需头文件	<code>#include <stdio.h></code>
函数原型	<code>void perror (const char* s);</code>
函数参数	<code>s</code> : 在标准错误流上输出的信息
函数返回值	无

```
#include <stdio.h>

int main()
{
    FILE *fp; // 定义流指针
    if ( ( fp = fopen ( "1.txt", "r" ) ) == NULL ) // NULL是系统定义的宏，其值为0
    {
        perror ( "fail to fopen" ); // 输出错误信息
        return -1;
    }
    fclose ( fp );
    return 0;
}
```

如果文件 `1.txt` 不存在，程序执行时会打印如下信息。

`fail to fopen: No such file or directory`

错误处理相关函数 `strerror()` 语法要点如表 1.5 所示。

表 1.5 `strerror()` 函数语法要点

所需头文件	<code>#include <string.h></code> <code>#include <errno.h></code>
函数原型	<code>char *strerror (int errnum);</code>
函数参数	错误码
函数返回值	错误码对应的错误信息



```
#include <stdio.h>

int main()
{
    FILE *fp;
    if ( ( fp = fopen ( "1.txt" , "r" ) ) == NULL )
    {
        printf ( " fail to fopen: %s\n" , strerror ( errno ) ) ;
        return -1;
    }

    fclose ( fp ) ;
    return 0;
}
```

如果文件 1.txt 不存在，程序执行时会打印如下信息。
fail to fopen: No such file or directory

1.3.4 流的读写

1. 按字符（字节）输入/输出

字符输入/输出函数一次仅读写一个字符。字符输入/输出函数语法要点如表 1.6 和表 1.7 所示。

表 1.6 字符输入函数语法要点

所需头文件	#include <stdio.h>
函数原型	int getc (FILE * stream) ; int fgetc (FILE * stream) ; int getchar (void) ;
函数参数	stream: 要输入的文件流
函数返回值	成功: 读取的字符
	失败: EOF

getc()和 fgetc ()从指定的流中读取一个字符（节），getchar()从 stdin 中读取一个字符（节）。

表 1.7 字符输出函数语法要点

所需头文件	#include <stdio.h>
函数原型	int putc (int c, FILE * stream) ; int fputc (int c, FILE * stream) ; int putchar (int c) ;