



嵌入式系统设计与开发系列丛书

μ C/OS-II 原理与ARM 应用程序设计

μ C/OS-II Theory & ARM Application
Program Design

张勇 方勤 编著
蔡鹏 许波



西安电子科技大学出版社

<http://www.xduph.com>

嵌入式系统设计与开发系列丛书

μ C/OS- II 原理与 ARM 应用程序设计

μ C/OS- II Theory & ARM Application Program Design

张勇 方勤

编著

蔡鹏 许波

贾晓天

主审

西安电子科技大学出版社

2010

内 容 简 介

$\mu\text{C}/\text{OS}-\text{II}$ 是 J. J. Labrosse 编写的多任务实时操作系统内核, 是目前应用最广的公开源代码嵌入式实时内核。本书讲述 $\mu\text{C}/\text{OS}-\text{II}$ 原理与移植以及基于 ARM 嵌入式系统的程序设计, 偏重于 ARM 应用程序设计。全书共分九章, 主要内容包括认识 $\mu\text{C}/\text{OS}-\text{II}$ 、 $\mu\text{C}/\text{OS}-\text{II}$ 移植、 $\mu\text{C}/\text{OS}-\text{II}$ 内核结构、面向任务程序设计、任务协作与通信、中断与时钟、存储配置与 Bootloader、博创 UP-CUP2410 移植与实例以及天嵌 TQ2440 移植与实例。本书的特色在于实例丰富且完整。

本书是作者近几年来从事 $\mu\text{C}/\text{OS}-\text{II}$ 教学与研究的结晶, 使用 $\mu\text{C}/\text{OS}-\text{II}$ 最新版本 V2.86, 对学习 $\mu\text{C}/\text{OS}-\text{II}$ 原理和基于 $\mu\text{C}/\text{OS}-\text{II}$ 的 ARM 应用程序设计具有一定的帮助和指导作用。

本书可作为电子技术类、通信类、软件工程类研究生和高年级本科生学习 $\mu\text{C}/\text{OS}-\text{II}$ 及其 ARM 程序设计的教材, 也可作为从事嵌入式应用或嵌入式操作系统开发的电子工程师和软件工程师以及嵌入式爱好者的参考书。

★本书配有电子教案, 需要者可登录出版社网站, 免费下载。

图书在版编目(CIP)数据

$\mu\text{C}/\text{OS}-\text{II}$ 原理与 ARM 应用程序设计/张勇等编著. —西安: 西安电子科技大学出版社, 2010.2
(嵌入式系统设计与开发系列丛书)

ISBN 978-7-5606-2395-5

I. μ … II. 张… III. ① 实时操作系统—程序设计 ② 微处理器, ARM—程序设计
IV. TP316.2 TP332

中国版本图书馆 CIP 数据核字(2010)第 011991 号

策 划 李惠萍

责任编辑 雷鸿俊 李惠萍

出版发行 西安电子科技大学出版社(西安市太白南路 2 号)

电 话 (029)88242885 88201467 邮 编 710071

网 址 www.xduph.com 电子邮箱 xdupfb001@163.com

经 销 新华书店

印刷单位 陕西华沐印刷科技有限责任公司

版 次 2010 年 2 月第 1 版 2010 年 2 月第 1 次印刷

开 本 787 毫米 $\times$ 1092 毫米 1/16 印 张 22.25

字 数 526 千字

印 数 1~2000 册

定 价 36.00 元

ISBN 978-7-5606-2395-5/TP·1201

XDUP 2687001 - 1

如有印装问题可调换

本社图书封面为激光防伪覆膜, 谨防盗版。

前 言

➤ μ C/OS-II 的由来与特点

μ C/OS-II 是美国 J. J. Labrosse 编写的实时嵌入式操作系统内核, 该内核的雏形 μ C/OS 最早见于 1992 年 5 月他在《Embedded System Programming》杂志上发表的实时操作系统 (RTOS), 后来他升级了该内核。同时, 为了推动和普及该内核的应用, Labrosse 于 1999 年注册了 Micrium 公司, 并于同年和 2002 年编写了介绍 μ C/OS-II 的书, 即《Micro μ C/OS-II The Real-Time Kernel》, 邵贝贝于 2003 年将其译成了中文版《嵌入式实时操作系统 μ C/OS-II》(北京航空航天大学出版社出版), 书中介绍的 μ C/OS-II 版本号为 V2.52。

μ C/OS-II 和著名的 Windows CE 都属于多任务实时操作系统。一般认为, Windows CE 定位于基于 ARM、MIPS、SH4 和 X86 的高端嵌入式应用, 强调美观的用户界面和强大的系统管理功能; 而 μ C/OS-II 定位于基于单片机、ARM、DSP 和 SOC 等微控制器或微处理器的嵌入式应用系统, 注重系统的安全性、可靠性和实时性。因此, μ C/OS-II 的应用范围和领域更加广阔, 已经成功应用的领域有军事、航天、工业、医疗、汽车和消费电子等。目前, μ C/OS-II 是以源代码的形式提供, 但 μ C/OS-II 不是自由软件。

从程序设计方法来讲, 在 C++ 语言诞生之前, 基本的程序设计方法是面向过程的程序设计方法, 而在出现了 C++ 以后, 面向对象的程序设计思想和方法开始流行起来, 再后来, 随着 Microsoft Windows 视窗操作系统的普及以及集成开发环境 Visual Studio 与 Borland Developer Studio 的广泛应用, 又出现了面向事件(或消息)的程序设计方法, 大师 C. Petzold 在这方面做出了卓越的贡献。笔者在《Windows CE 应用程序设计》(西安电子科技大学出版社 2008 年出版)中介绍了采用面向消息的程序设计方法设计基于 Windows CE 的应用程序。而在出现了 μ C/OS-II 之后, 基于 μ C/OS-II 的程序设计方法可以被称为面向任务的程序设计方法。与面向对象的程序设计方法的不同点在于, 面向任务的程序设计方法的基本单位不再是类及其变量——对象, 而是任务。同时, 任务也是程序的组织核心, 任务之间也是通过“消息”进行通信, 虽然任务也包含数据和方法(或称函数), 但是与类或普通函数有实质性的区别。

➤ 本书的结构与简介

μ C/OS-II 于 2007 年 9 月 12 日升级为 V2.86 版本, 该版本是本书写作时的最新版本。因此, 本书内容基于 V2.86 版本, 在这个版本中最多可以创建 255 个任务。最近的报道显示 Micrium 公司推出了 μ C/OS-III, 因此, V2.86 可能是 μ C/OS-II 的最终版本。本书内容的编排结构为: 第一章“认识 μ C/OS-II”介绍了两个完整 μ C/OS-II 工程程序的文件组织结构以及 Borland C++ 4.51 和 IAR Embedded Workbench 开发环境, 并说明了本书实例的硬件工作平台情况; 第二章“EWARM 使用方法与芯片级程序”介绍了基于 EWARM 平台开发 ARM 芯片级程序的方法, 列举了三个实例; 第三章“将 μ C/OS-II 移植到 UP-Star 实验板上”重点介绍了 μ C/OS-II 在 S3C2410A 上的移植; 第四章“ μ C/OS-II 内核与面向任务程序设计”

重点讲述了内核文件 OS_CORE.C 和任务处理文件 OS_TASK.C 的逻辑结构,并结合三个实例介绍了如何基于 $\mu\text{C}/\text{OS-II}$ 有效地设计、组织和管理任务;第五章“任务协作与通信”全面介绍了信号量、互斥信号量、消息邮箱、消息队列、事件标志、多事件请求处理的创建和管理方法;第六章“中断与 $\mu\text{C}/\text{OS-II}$ 定时器”介绍了 $\mu\text{C}/\text{OS-II}$ 下中断响应处理及时钟管理任务;第七章“动态内存配置与 Bootloader”介绍了 $\mu\text{C}/\text{OS-II}$ 内存管理和可执行代码下载与启动方法;第八章“博创 UP-CUP2410 移植与实例”介绍了基于博创 UP-CUP2410 实验箱的移植和程序设计实例;第九章“天嵌 TQ2440 移植与实例”介绍了基于天嵌 TQ2440 开发板的移植和实例。

本书内容讲述 $\mu\text{C}/\text{OS-II}$ 原理与移植以及基于 ARM 嵌入式系统的程序设计,偏重于 ARM 应用程序设计,后续即将出版的《 $\mu\text{C}/\text{OS-II}$ 原理与面向任务程序设计》,则偏重于 $\mu\text{C}/\text{OS-II}$ 原理与实时操作系统设计。作为从事嵌入式方面教学的大学教师,笔者将会一直跟踪 $\mu\text{C}/\text{OS-II}$ 的升级和发展,并会不断充实和修订本书。

► 本书的编写分工

本书第一至三章由张勇执笔,第四至六章由蔡鹏博士执笔,第七章由方勤教授执笔,第八、九章由许波高级工程师执笔。全书由方勤教授统稿。贾晓天教授担任本书主审。

► 本书的学习方法

如果用本书作为大学教材,理论课时宜在 30 学时左右,上机课时不低于 20 学时。授课内容宜选择第一至七章,适宜作为专业拓展课和课程设计用书。

本书以实例为主线,讲解为辅,适合工程技术人员。如果自学本书,需要有一台配置较高的计算机和一套 ARM920T 实验板,需要安装 PDF 阅读器和 EWARM 软件。在学习本书的过程中,必须结合书中提到的资料同步学习,同时要反复调试程序和理解程序的运行过程,边学边实践。如果单纯浏览本书内容则无法学好本书。

本书不是自成体系的,笔者编写的《ARM 原理与 C 程序设计》(西安电子科技大学出版社 2009 年出版)是本书的基础,同时学习过程中要参考学习 $\mu\text{C}/\text{OS-II}$ V2.86 的文档资料、邵贝贝译的《嵌入式实时操作系统 $\mu\text{C}/\text{OS-II}$ (第 2 版)》以及作者即将出版的《 $\mu\text{C}/\text{OS-II}$ 原理与面向任务程序设计》等。

本书的起点稍高,建议读者应具有一些 C 程序设计和单片机技术方面的基础。

► 本书的特色

本书的特色在于实例丰富且完整,而且读者可以通过 E-mail: zhnyong@21cn.com 或 QQ: 493815991 和笔者联系索取源代码。但是,笔者强烈建议读者自行输入代码并进行调试以提高学习效率。尽管书中使用的硬件平台为博创 UP-Star 实验板(嵌入式工程师认证考试指定平台,第一章至第七章)、博创 UP-CUP2410 实验箱(笔者上课用的实验设备,第八章)和天嵌 TQ2440 开发板(笔者做项目用的平台,第九章),但是书中的程序几乎不用修改或略做修改就可以应用在其他 ARM 平台上。除第一章实例借助集成开发环境 Borland C++ 4.51 外,其余全部实例工程均基于上海爱亚软件技术咨询有限公司提供的正版软件 IAR Embedded Workbench for ARM V5.30.1(简称 EWARM)。

本书与已经出版的《ARM 原理与 C 程序设计》、《Windows CE 应用程序设计》以及即将出版的《 $\mu\text{C}/\text{OS-II}$ 原理与面向任务程序设计》同属于嵌入式系统设计与开发系列丛书,作者还拟近期出版基于单片机和 DSP 方面的 $\mu\text{C}/\text{OS-II}$ 著作,以进一步完善该系列丛书。

► 本书的平台赞助商

笔者的每本书都有平台赞助商，事实上，没有平台赞助商，很难写出这类针对性和工程性都很强的书来，而大学里面关于专业拓展课的要求也是工程性要强，这正好也是这些赞助商大学计划的一部分。

本书的平台赞助商为北京博创科技、广州天嵌科技和上海爱亚软件，在此向他们表示诚挚的感谢。笔者还要感谢那些阅读了我们的书并反馈了宝贵意见的读者，这些意见对于我们编写新书和修订再版有实质性的帮助。由于我们水平有限，本书中仍难免有纰漏之处，敬请同行专家和读者批评指正。

最后，感谢西安电子科技大学出版社李惠萍编辑为本书出版所做的细致工作。

► 特别说明

本书写作时，笔者所编写的《ARM 原理与 C 程序设计》一书还没有再版，但原书中有两处错误，其一为第 5.4.1 节关于 EINT5 的描述是错误的，其二为第 6.2 节中关于重定位代码的写法有一些问题。这两处错误在本书程序中都予以了更正，在此特别声明一下。这两处错误是 2009 年 5 月博创科技工程师王举提出来的，在此表示衷心感谢。

► 关于读者来信

在笔者出版第一本书时，导师陈天麒先生曾帮笔者修改全书并建议笔者给读者留下联系方式。后来正如先生所料，书出版后有几千封读者来信；最近的两本书也导致了同样的问题。因此，笔者申请了一个网站 <http://a.ooc.cn/zhnyong/>，本书的代码都放在上面。如果读者单纯为了索取代码，请直接到该网站上下载；如果来信指正错误，请用表格，并精确到某页的某行。尽管笔者没有足够的时间回复每一封来信，但是无论如何，笔者都会认真阅读所有读者的来信，与读者分享读书的乐趣，并感谢读者购买笔者的书。

► 免责声明

本书内容仅用于教学研究用，书中引用的相关内容的知识产权归相关公司所有，作者保留其余内容的所有权利。禁止任何单位或个人摘录本书内容出版发行，严禁依据本书出版实验教程。

张 勇

庚寅年于江西财经大学麦庐园

目 录

第一章 认识 μ C/OS-II	1
1.1 准备工作.....	1
1.2 BC4.51 和实例一	3
1.3 实例二.....	10
1.4 μ C/OS-II 文件与函数.....	13
1.4.1 μ C/OS-II 文件关联.....	14
1.4.2 μ C/OS-II 函数.....	15
1.5 UP-Star 硬件原理图.....	26
1.6 本章小结.....	33
第二章 EWARM 使用方法与芯片级程序.....	34
2.1 EWARM 和 ARM 芯片级实例	34
2.1.1 EWARM 开发环境和实例一.....	34
2.1.2 实例一工程源码.....	51
2.2 定时器中断与实例二.....	61
2.2.1 S3C2410 中断源.....	61
2.2.2 定时器 4 中断.....	63
2.2.3 实例 ex2_2.....	69
2.3 串口通信与实例三.....	75
2.3.1 串口通信方法.....	75
2.3.2 实例 ex2_3.....	78
2.4 本章小结.....	85
第三章 将 μ C/OS-II 移植到 UP-Star 实验板上.....	86
3.1 μ C/OS-II LED 实例	86
3.1.1 实例 ex3_1.....	87
3.1.2 实例 ex3_1 注解.....	105
3.2 μ C/OS-II 移植.....	106
3.2.1 os_cpu.h 文件	108
3.2.2 os_cpu_c.c 文件.....	110
3.2.3 os_cpu_a.asm 文件.....	114
3.2.4 bsp.c 文件	115
3.3 μ C/OS-II 串口通信实例	116
3.3.1 实例 ex3_2.....	116

3.3.2 实例 ex3_2 调试与运行.....	124
3.4 本章小结.....	126
第四章 μC/OS-II 内核与面向任务程序设计.....	127
4.1 μ C/OS-II 内核 OS_CORE.C.....	127
4.1.1 任务控制块.....	129
4.1.2 事件控制块.....	132
4.1.3 就绪表.....	133
4.1.4 空闲任务和统计任务.....	136
4.1.5 时钟节拍.....	137
4.1.6 μ C/OS-II 初始化.....	137
4.2 任务 OS_TASK.C.....	140
4.2.1 工程 ex4_1.....	140
4.2.2 主程序 app.c.....	144
4.2.3 任务程序 appfun.c.....	145
4.2.4 工程 ex4_2.....	153
4.3 时间 OS_TIME.C.....	158
4.4 本章小结.....	159
第五章 任务协作与通信.....	161
5.1 信号量 OS_SEM.C.....	161
5.1.1 工程 ex5_1.....	161
5.1.2 工程 ex5_1 代码与注解.....	163
5.2 互斥型信号量 OS_MUTEX.C.....	166
5.2.1 工程 ex5_2.....	167
5.2.2 工程 ex5_2 代码与注解.....	169
5.3 消息邮箱 OS_MBOX.C.....	172
5.3.1 工程 ex5_3.....	173
5.3.2 工程 ex5_3 功能注解.....	174
5.4 消息队列 OS_Q.C.....	179
5.4.1 工程 ex5_4.....	179
5.4.2 工程 ex5_4 功能注解.....	181
5.5 事件标志 OS_FLAGC.....	185
5.5.1 工程 ex5_5.....	186
5.5.2 工程 ex5_5 功能注解.....	188
5.6 多事件请求处理.....	191
5.6.1 工程 ex5_6.....	192
5.6.2 工程 ex5_6 功能注解.....	193
5.7 μ C/OS-II 组件配置 OS_CFGH.....	197
5.8 本章小结.....	200

第六章 中断与 μC/OS-II 定时器	201
6.1 外部中断处理	201
6.1.1 外部中断实例	201
6.1.2 工程 ex6_1 注解	202
6.2 定时器 OS_TMR.C	219
6.2.1 定时器实例	220
6.2.2 工程 ex6_2 注解	222
6.3 本章小结	228
第七章 动态内存配置与 Bootloader	229
7.1 动态存储 OS_MEM.C	229
7.1.1 内存分区实例	229
7.1.2 工程 ex7_1 注解	231
7.2 Bootloader 设计	234
7.2.1 读写 FLASH 工程	234
7.2.2 用于自举的工程 ex7_3	242
7.2.3 目标代码转化为 C 头文件 flash.h	252
7.2.4 Bootloader 工程 ex7_4	257
7.2.5 H-JTAG 下载方式	259
7.3 本章小结	262
第八章 博创 UP-CUP2410 移植与实例	263
8.1 准备工作	263
8.2 UP-CUP2410 工程实例	264
8.2.1 工程 ex8_1	264
8.2.2 工程 ex8_1 代码与注解	265
8.3 本章小结	281
第九章 天嵌 TQ2440 移植与实例	282
9.1 准备工作	282
9.2 TQ2440 工程实例	282
9.3 Bootloader	311
9.4 本章小结	322
附录一 第 2.1 节实例一的 MAP 表文件 ex2_1.map 内容	323
附录二 TQ2440 实验板 K9F1208 代码下载工程	329



第一章 认识 μ C/OS-II

本章介绍本书使用的软件、硬件平台，以及基于 μ C/OS-II 的两个实例。

1.1 准备工作

所谓“工欲善其事，必先利其器”（《论语》）。在开始学习本书内容之前，读者需要花点时间做一些软件和硬件方面的准备工作。

首先，需要一台配置较高的电脑。笔者使用的是一台笔记本电脑，基本配置为：酷睿 2 双核处理器 T5450、2 GB 内存、160 GB 硬盘、12.1 英寸宽显示屏和 3 个 USB 接口，操作系统为 Windows XP SP3。读者也需要准备类似的或更高的配置。

其次，应从网上下载软件 BC 4.51(Borland C++ 4.51)和 TASM 5.0(Borland Turbo Assembler 5.0)，其大小分别约为 41 MB 和 4.2 MB。将这两个软件安装在默认工作目录下，即分别安装在 C:\BC45 和 C:\TASM 下，然后，将 C:\TASM\BIN\TASM.EXE 文件拷贝到 C:\BC45\BIN 目录下。

说明：不是必须按默认目录安装，但是目录名应该是不多于 8 个英文字母且首字符为字母的字符串，即按 DOS 命名规则命名的目录名，不要使用汉字和过长的目录名。

安装完这两个软件后，将在 Windows 的“开始|所有程序”菜单中创建菜单启动组“Borland C++ 4.5”和“Borland Tasm 5.0”，单击“开始|所有程序|Borland C++ 4.5|Borland C++”，即可运行 Borland C++ 4.5。

再次，从 Micrium 网站 <http://www.micrium.com> 下载 μ C/OS-II，任何人注册登录后都可以下载。本书使用的版本为 V2.86，如果 Micrium 网站更新 μ C/OS-II 为更高的版本，为了学习本书的方便，请联系作者或从相关网站下载 V2.86 版本，同时作者也会针对新的版本更新本书实例。如果 Micrium 网站上提供的是 V2.86 版本，其下载文件名为 Micrium-uCOS-II-V286.ZIP(注：由于软件的缘故，本书中所有 uCOS 和 ucos 皆指 μ COS)。V2.86 版本自 2007 年 9 月 12 日定型后，两年来仍没有升级，最近的报道显示 Micrium 公司推出了 μ C/OS-III，因此，V2.86 可能是 μ C/OS-II 的最终版本。读者可同时在 Micrium 网站下载一些移植程序和自己认为有用的应用文档。

还需到 IAR Systems 网站 <http://www.iar.com> 下载 IAR Embedded Workbench for ARM(简称 EWARM)，最新版本为 V5.40，其 V5.30 版本在网上十分盛行。本书使用 V5.30 版本，并且是上海爱亚软件提供的正式版本，但是建议读者使用 EWARM 的最新版本，实例完全兼容。IAR Systems 是全球领先的嵌入式系统开发工具和服务供应商，1983 年成立于瑞典乌普萨拉市，2004 年在上海成立办事处，并且，IAR Systems 公司是 Micrium 公司产品的全球



经销商, 所以学习基于 ARM 的 μ C/OS-II, EWARM 是首选的软件。要启动 EWARM, 只需点击菜单项“开始|所有程序|IAR Systems|IAR Embedded Workbench for ARM 5.30|IAR Embedded Workbench”即可。

最后, 如果读者使用不带串口的笔记本电脑, 还需要一条 USB 转串口的接口线。除第一章实例一和实例二不需外接实验设备外, 第二章至第七章的实例均需要借助 UP-Star 认证考试实验板(嵌入式工程师认证考试专用实验板), 第八章的实例在博创 UP-CUP2410 实验箱(笔者上课时用的设备)上实现, 第九章的实例在天嵌 TQ2440 开发板(笔者做项目用的开发板)上实现。读者要体验这些实例的运行情况, 需要具备这些实验板或者载有 S3C2410 或 S3C2440 芯片的同类实验板卡。在开发和调试实例时, 需要用到 J-LINK for ARM 仿真器, 如果读者有一块 TQ2440 开发板, 由于 TQ2440 开发板上的 JTAG 接口是 2.00 mm 间距 10 针, 而 J-LINK 接口一般是 2.54 mm 间距 20 针, 因此需要做一个 20 针至 10 针 JTAG 转接头和 10 芯排线。如果读者有一块 UP-Star 实验板, 由于 UP-Star 实验板上 JTAG 接口是 2.54 mm 间距 14 针, 故需要做一个 20 针至 14 针 JTAG 转接头和 14 芯排线。这两个转接头的原理图如图 1-1 和图 1-2 所示。而 UP-CUP2410 实验箱上集成了 20 针 2.54 mm 间距 JTAG 接口, 可以直接使用 J-LINK 仿真器。

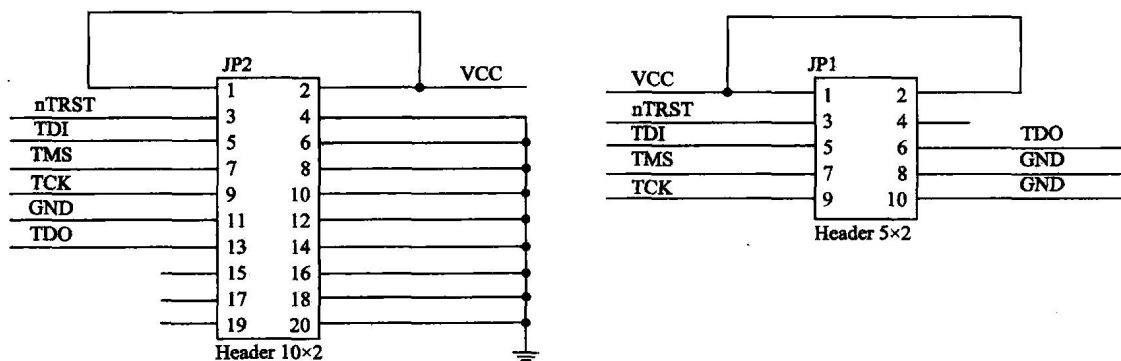


图 1-1 20 针至 10 针转接头原理图

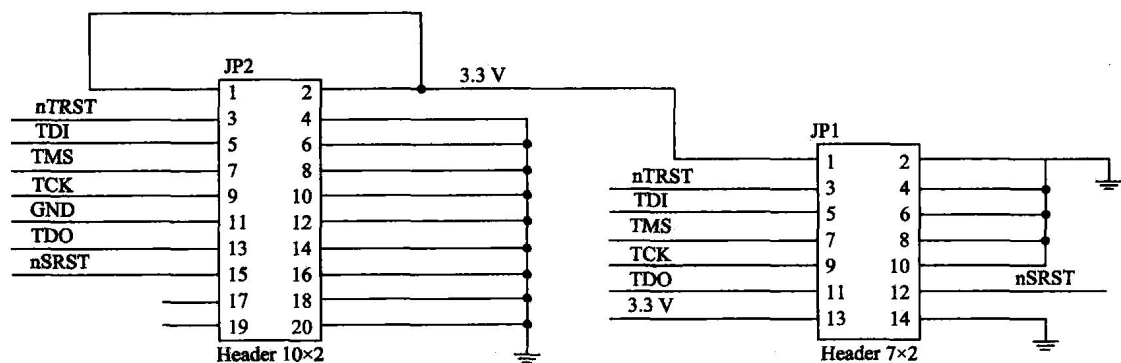


图 1-2 20 针至 14 针转接头原理图

将笔记本电脑的一个 USB 口通过 J-LINK(及其转接头)与 UP-Star 实验板(或 UP-CUP2410 实验箱、TQ2440 开发板或其他 ARM 平台)上的 JTAG 接头相连接, 笔记本电



脑的另一个 USB 口通过 USB 转串口线与实验板上的串口 0 相连接, 将实验板的电源线连接好, 打开实验板电源开关和笔记本电脑, 则实验环境与学习平台就建立好了。

在阅读本书时, 读者可能同步阅读 μ C/OS-II 源代码或书中一些实例代码, 作者建议使用 Source Insight 软件阅读, 用过的读者一定知道, 只有这个软件才能把阅读代码变成一种乐趣和享受。

1.2 BC4.51 和实例一

本节的实例一和第 1.3 节的实例二就是《嵌入式实时操作系统 μ C/OS-II(第 2 版)》(北京航空航天大学出版社出版)一书中的实例一和实例二, 这里用 μ C/OS-II V2.86 版本, 而原书中使用 V2.52 版本。这两个实例可用来加强读者对 μ C/OS-II 的感性认识。

在 D 盘上新建目录 ZYUCOSII, 本书的所有工程均位于这个目录下; 然后, 在 ZYUCOSII 下新建子目录 EX1_1, 用于存放第一章的第一个实例; 在目录 EX1_1 下新建 5 个子目录, 依次为 config、user、object、ports 和 ucosii, 依次存放配置文件、用户编写的程序文件、编译后的目标文件、移植文件和 μ C/OS-II 系统文件, 如图 1-3 所示。

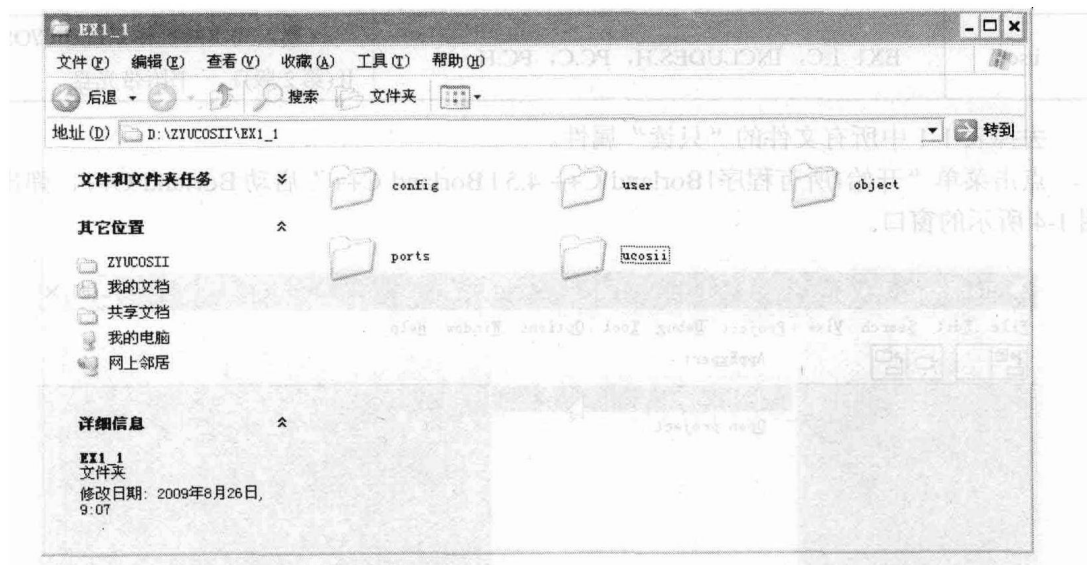


图 1-3 实例一工作目录

把从网络 www.micrium.com 上下载的文件 Micrium-uCOS-II-V286.ZIP 解压后, 将其中的目录 “Micrium-uCOS-II-V286\Micrium\SOFTWARE\uCOS-II\Source” 下的 14 个文件拷贝到图 1-3 中的 ucosii 目录下, 并将其中的文件 os_cfg_r.h 和 os_dbg_r.c 拷贝到图 1-3 中的 config 目录下, 将其更名为 os_cfg.h 和 os_dbg.c, 即去掉文件名中的_r。

从网上下载《嵌入式实时操作系统 μ C/OS-II(第 2 版)》一书附带光盘中的内容, 将光盘目录 “\SOFTWARE\BLOCKS\PC\BC45\” 下的两个文件 PC.C 和 PC.H 拷贝到图 1-3 中的 user 目录下; 将光盘目录 “\SOFTWARE\uCOS-II\EX1_x86\ABC45\SOURCE\” 下的 INCLUDES.H 和 TEST.C 文件拷贝到图 1-3 中的 user 目录下, 并将 TEST.C 更名为 EX1_1.C;

将光盘目录“\SOFTWARE\uCOS-II\Vx86\ABC45”下的3个文件 OS_CPU.H、OS_CPU_A.ASM 和 OS_CPU_C.C 拷贝到图 1-3 中的 ports 目录下。

在图 1-3 中的 config 目录下新建空的文本文件 APP_CFG.H。该文件一般用于定义任务优先级和堆栈大小。

至此,图 1-3 中各子目录下的文件如表 1-1 所示。

表 1-1 目录 EX1_1 下子目录中的文件

子目录名	包括的文件名	文件来源
config	os_cfg.h, os_dbg.c	Micrium-uCOS-II-V286.ZIP
	APP_CFG.H	新建文本文件
object	—	—
ports	OS_CPU.H, OS_CPU_A.ASM, OS_CPU_C.C	《嵌入式实时操作系统 μ C/OS-II(第2版)》一书附带光盘
ucosii	os_cfg_r.h, os_core.c, os_dbg_r.c, os_flag.c, os_mbox.c, os_mem.c, os_mutex.c, os_q.c, os_sem.c, os_task.c, os_time.c, os_tmr.c, ucos_ii.c, ucos_ii.h。其中, os_cfg_r.h 和 os_dbg_r.c 可有可无	Micrium-uCOS-II-V286.ZIP
user	EX1_1.C, INCLUDES.H, PC.C, PC.H	《嵌入式实时操作系统 μ C/OS-II(第2版)》一书附带光盘

去掉表 1-1 中所有文件的“只读”属性。

点击菜单“开始|所有程序|Borland C++ 4.5|Borland C++”启动 Borland C++, 弹出如图 1-4 所示的窗口。

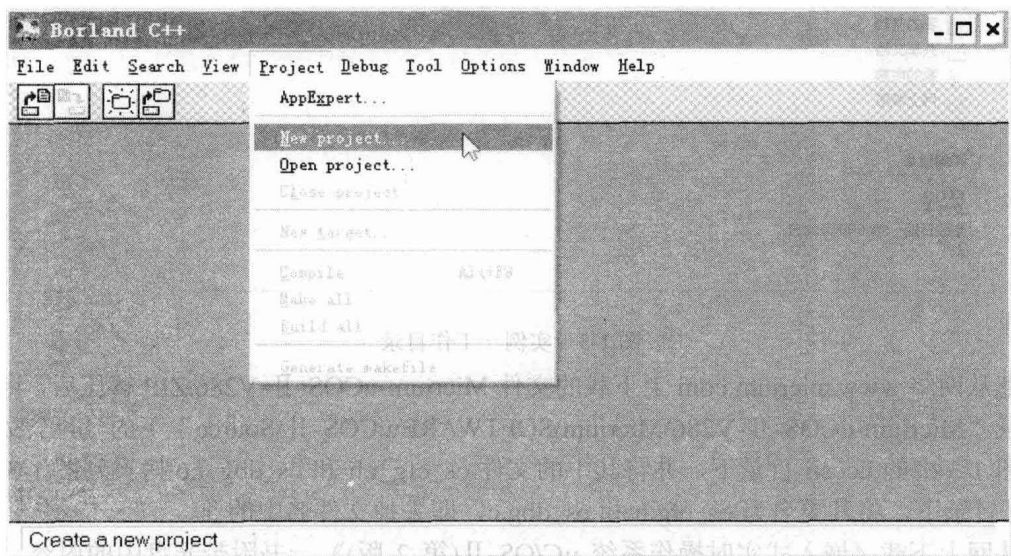


图 1-4 Borland C++ 4.5 工作主窗口

在图 1-4 中点击菜单“Project|New project...”, 弹出如图 1-5 所示的窗口。

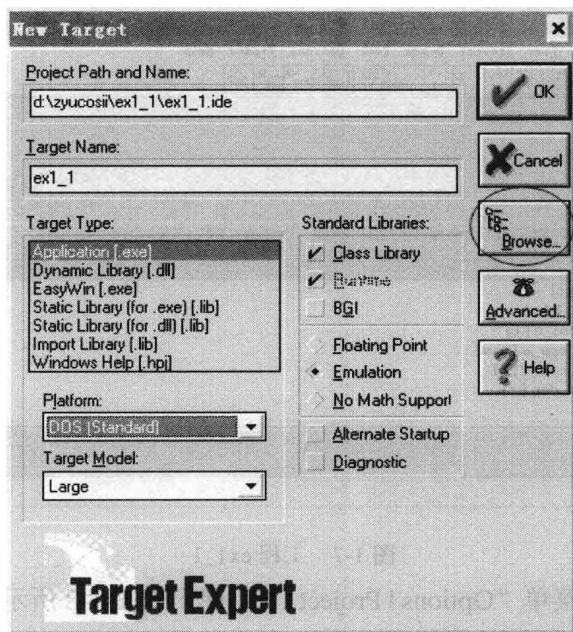


图 1-5 新工程命名窗口

如图 1-5 所示进行设置，其中，“Project Path and Name:”和“Target Name:”两项可通过图中的“Browse...”按钮进行设置。然后，点击图中的“OK”按钮，保存设置并关闭“New Target”对话框，得到如图 1-6 所示的窗口。

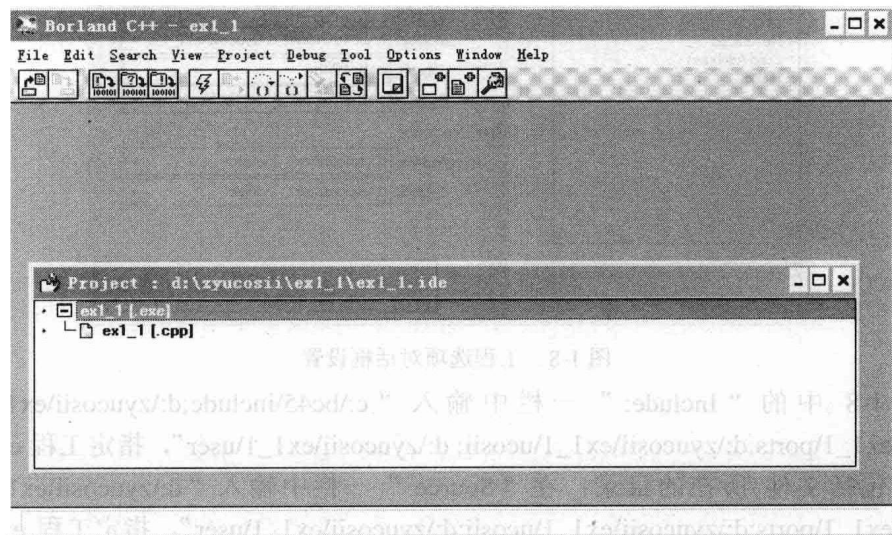


图 1-6 新建工程 ex1_1

在图 1-6 中删除文件节点“ex1_1.cpp”，添加文件节点“user\pc.c”、“user\ex1_1.c”、“ucosii\ucos_ii.c”、“ports\os_cpu_a.asm”、“ports\os_cpu_c.c”和“config\os_dbg.c”，如图 1-7 所示。注意，需要右键点击图 1-6 中的“ex1_1[.exe]”，在其右键弹出菜单中选择“Add node”向其添加文件节点。

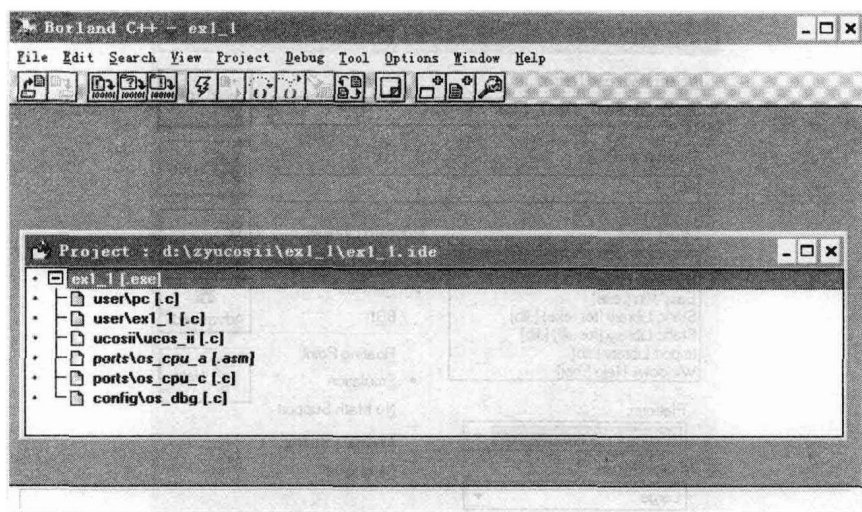


图 1-7 工程 ex1_1

在图 1-7 中，点击菜单“Options | Project...”，弹出如图 1-8 所示的对话框。

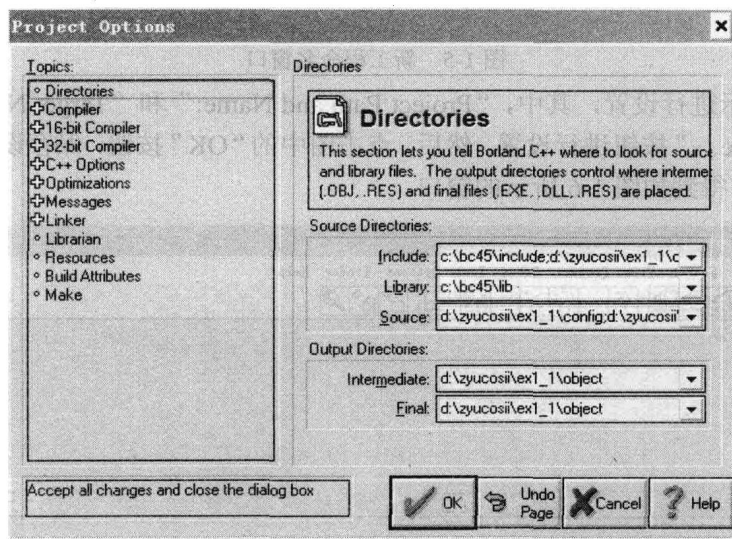


图 1-8 工程选项对话框设置

在图 1-8 中的“Include:”一栏中输入“c:\bc45\include;d:\zyucosii\ex1_1\config;d:\zyucosii\ex1_1\ports;d:\zyucosii\ex1_1\ucosii;d:\zyucosii\ex1_1\user”，指定工程 ex1_1 的头文件(或称包括文件)所在的目录；在“Source:”一栏中输入“d:\zyucosii\ex1_1\config;d:\zyucosii\ex1_1\ports;d:\zyucosii\ex1_1\ucosii;d:\zyucosii\ex1_1\user”，指定工程 ex1_1 的源文件所在的目录；在“Intermediate:”和“Final:”栏中均输入“d:\zyucosii\ex1_1\object”，表示编译产生的中间文件和目标文件存放在目录“d:\zyucosii\ex1_1\object”中。在图 1-8 中点击“OK”按钮保存设置，并关闭对话框。

在图 1-7 中点击菜单“File | Open...”打开文件“d:\zyucosii\ex1_1\user\includes.h”，将其第 21、23 和 24 行中的包括文件路径去掉后存盘，改为如图 1-9 所示。

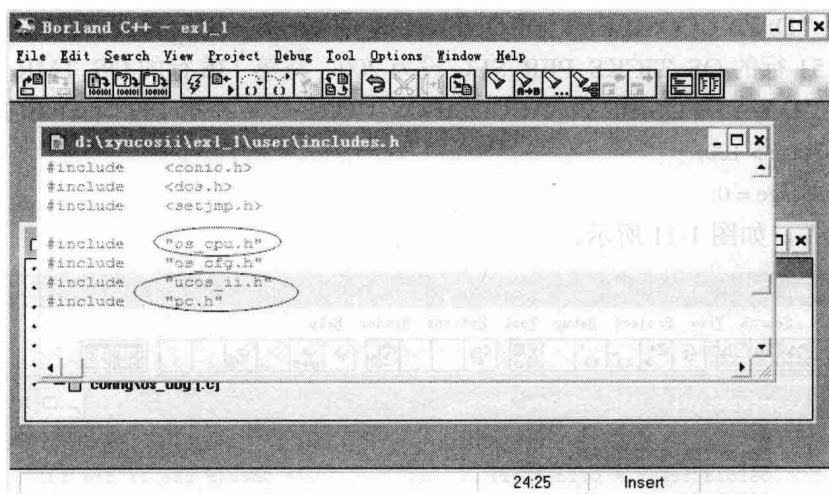


图 1-9 修改 includes.h 文件

在图 1-7 中双击文件节点“user/pc.c”弹出文件 pc.c 的内容，将该文件中的 TRUE 改为 OS_TRUE，FALSE 改为 OS_FALSE，其中，TRUE 位于第 233、269 和 420 行，FALSE 位于第 262 和 423 行。

同样，修改文件 ex1_1.c 中第 101 行的 TRUE 为 OS_TRUE。在 μ C/OS-II V2.86 中真为 OS_TRUE，假为 OS_FALSE。

在文件 OS_CPU.H 的末尾添加如下函数声明：

```
void OSCtxSw(void);  
void OSTickISR(void);  
void OSStartHighRdy(void);  
void OSIntCtxSw(void);
```

添加完后保存文件，如图 1-10 所示。

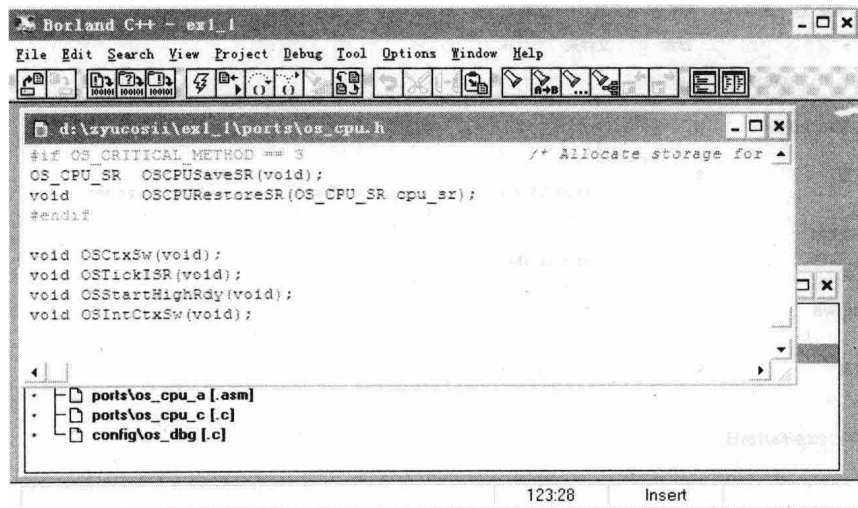


图 1-10 修改 OS_CPU.H 文件

打开文件“D:\ZYUCOSI\EX1_1\CONFIG\OS_CFG.H”，将第 34 行的 OS_DEBUG_EN 设为 0，将第 51 行的 OS_TICKS_PER_SEC 设为 200，将第 136 行的 OS_TMR_EN 设为 0。

打开文件“D:\ZYUCOSI\EX1_1\UCOSII\OS_CORE.C”，在第 1827 行下添加以下语句：
if (OSCPUUsage > 100)

OSCPUUsage = 0;

添加后保存文件，如图 1-11 所示。

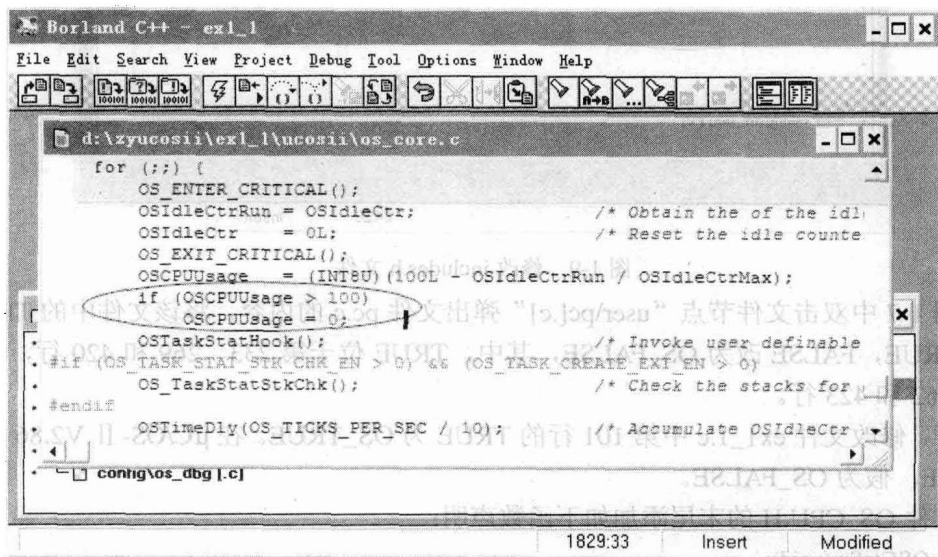


图 1-11 修改 os_core.c 文件

至此，在图 1-11 存盘后，点击菜单“Project | Build all”，可以成功编译链接并生成可执行文件 ex1_1.exe，位于目录“D:\ZYUCOSI\EX1_1\object”下，如图 1-12 所示。

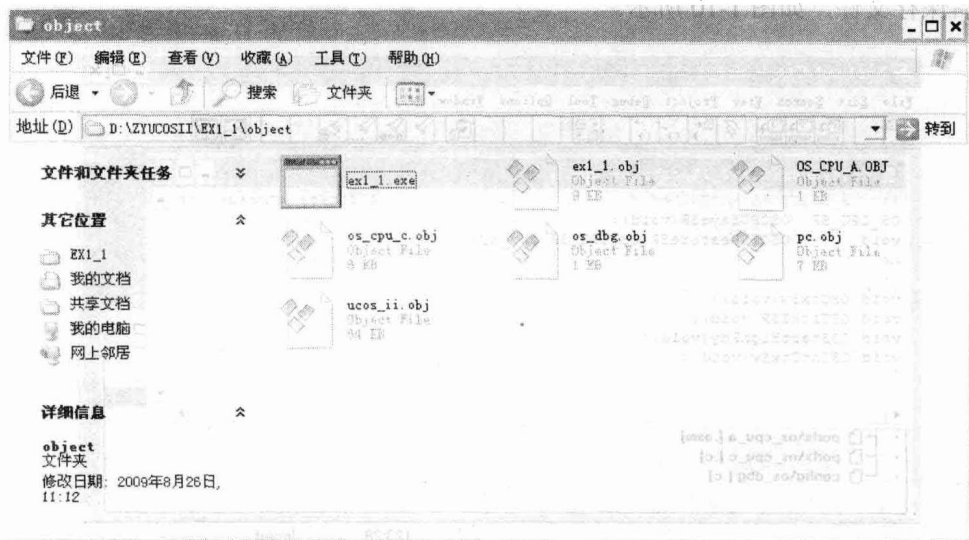


图 1-12 目录 D:\ZYUCOSI\EX1_1\object