Ch1 搭建基础实验环境

**背景知识**

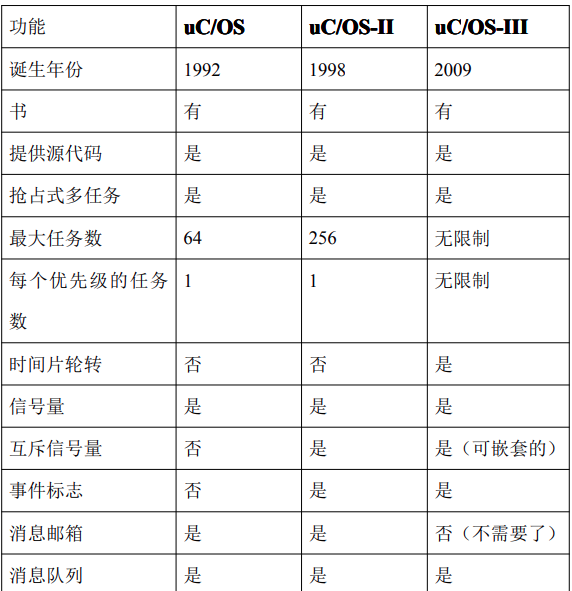
1.1 μC/OS-II操作系统简介

1.1.1 μC/OS项目由来

μC/OS最早出自美国嵌入式系统专家Jean J. Labrosse在《嵌入式系统编程》杂志1992年5月和6月刊上的文章连载。经过不断修改完善，Jean于1998年发布了μC/OS-II，它是一种基于优先级的抢占式多任务实时操作系统，包含了实时内核、任务管理、时间管理、任务间通信同步（信号量、邮箱、消息队列）和内存管理等功能。μC/OS-II已经在世界范围内得到广泛应用，包括在移动设备、路由器集线器、不间断电源、飞行器、医疗设备及工业控制等领域。

作为操作系统，μC/OS-II通过了非常严格的测试，其稳定性、安全性和可靠性得到了美国航空管理局认证。μC/OS-II的鲜明特点就是源码公开、精简，便于移植和维护。（为书写方便，下文统一用u替换希腊字母μ）

1.1.2 uC/OS-II与uC/OS-III版本区别

uC/OS-II采用ANSI C语言开发，可移植范围涵盖了8位到64位的多种主流处理器体系。第三代内核uC/OS-III发布于2009年，由uC/OS-II升级而来，是一个可裁剪、可固化、可剥夺的多任务系统，并且没有任务数目的限制。下表给出了不同版本间的主要区别。

**本实验课程基于uC/OS-II的源码进行**，它构思巧妙，结构精炼，可读性很强，同时具备了操作系统的大部分功能，非常适合初次接触操作系统源代码的学生学习并用于实践。

1.2 下载uC/OS-II的WIN32移植版本

虽然官方提供了WIN32的移植代码，但是这个版本的代码只能支持比较简单的范例，类似外部中断、动态优先级转换等功能没有进行移植，并且官方移植支持的ucos-ii的版本也比较低。

官网源码下载地址：

https://www.silabs.com/developers/micrium

本实验采用德国Hochschule Esslingen大学Zimmermann教授的WIN32移植版本。该版本显式支持到ucos-ii 2.84版本，经测试通过少量修改后可支持最新的ucos-ii 2.92版本。该移植使用软件中断仿真外部中断。

Zimmermann教授的个人网址：

http://www2.hs-esslingen.de/~zimmerma/software/index\_uk.html

以下对Zimmermann移植版本进行简要说明：

1. 移植主要指Port文件夹下的四个文件pc.h、pc.c、os\_cpu.h和os\_cpu.c，它们用来面向x86体系结构下的WIN32环境，即我们通常所用的主流桌面环境。
2. 移植分GCC、VC两个不同的编译器，考虑易用性，这里选择VC编译（本书使用VC2010，VC2010之后版本经测试可正常运行）。
3. 用户例程含三个头文件app\_cfg.h、INCLUDE.h、OS\_CFG.h，以及程序文件Test.c。ucos-ii操作系统的配置在OS\_CFG.h头文件中进行。多数情况下，上述头文件无需修改。用户在Test.c中编写多任务测试代码。

**特别说明：**以上所有初始源码均已提供，分别位于App、Port和Source三个文件夹下。其中Port文件夹中为处理器相关源码，Source文件夹中为内核源码，App文件夹中为用户例程。

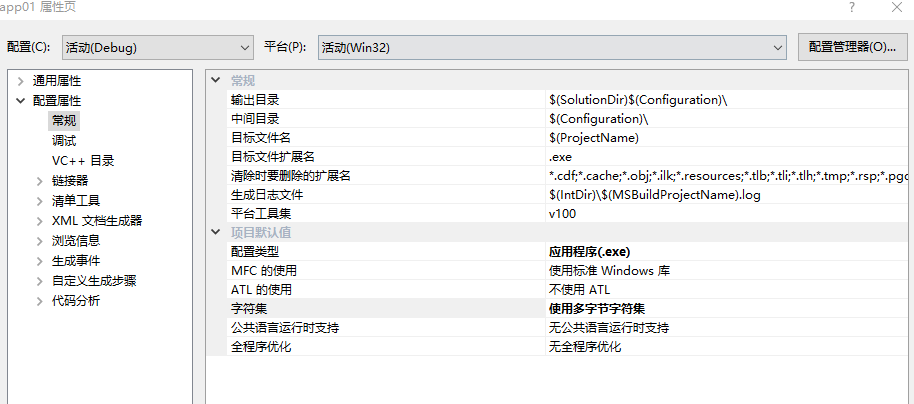
下文具体说明如何使用这三个文件夹在VC2010中建立ucos-ii工程项目。

**实验内容**

1.3 第一个uC/OS-II工程

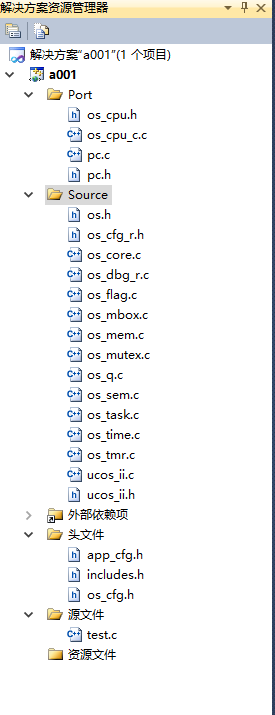
1.3.1 新建uC/OS-II工程

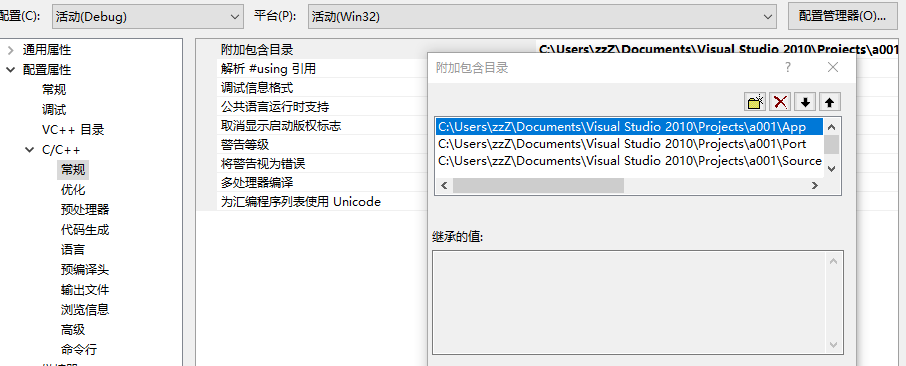
1. 在VC2010新建项目窗口选择“Win32控制台应用程序”，之后取消“预编译头”，再勾选“空项目”。如图示：

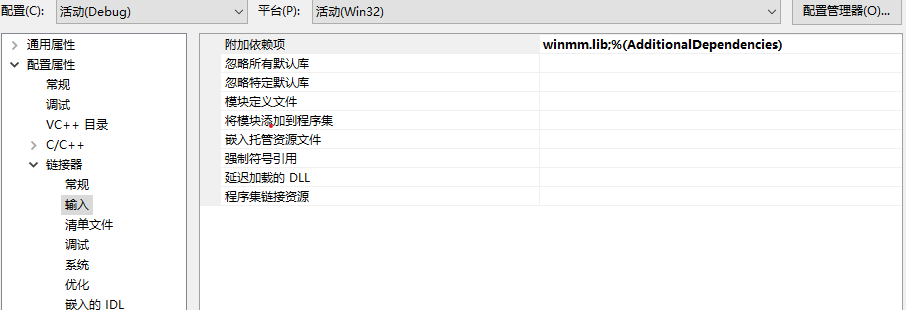
2. 工程项目属性配置“常规”-“字符集”下选择“使用多字节字符集”，如：

3. 将例程文件夹App拷入工程项目目录下，App中含有app\_cfg.h、INCLUDE.h、OS\_CFG.h以及程序文件Test.c。

在“源文件”下添加现有项Test.c，在“头文件”下添加现有项app\_cfg.h、INCLUDE.h、OS\_CFG.h。

将移植文件夹Port以及ucos-ii 2.92的内核源码文件夹Source完整拷入工程项目目录下。在项目中新建筛选器“Port”以及“Source”，对应分别将上述二个文件夹中的所有文件按现有项添加。下图给出了完整添加后的情形：

4. 在项目属性配置“C/C++”-“常规”-“附加包含目录”下添加App、Port和Source目录的完整路径：

5. 在“链接器”-“输入”-“附件依赖项”下添加winmm.lib，目的是使用Win32API来精确模拟时钟：

1.3.2 运行第一个工程

当上述配置完成后，第一个工程运行结果如下：

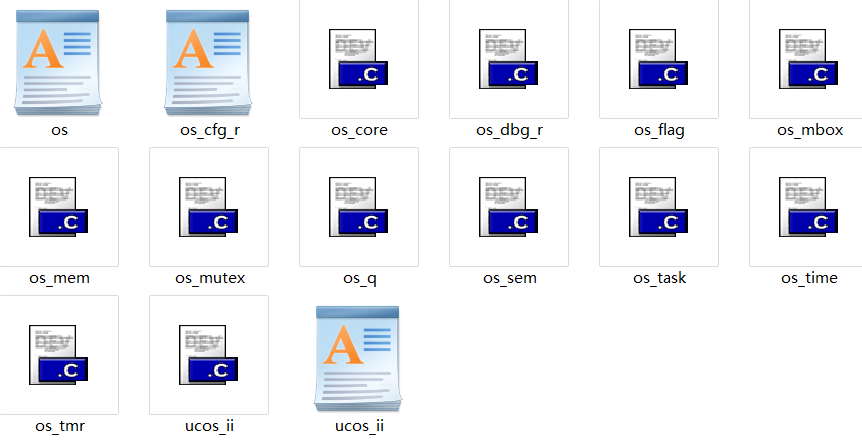
该例程使用ucos-ii内核函数OSTaskCreate创建了第一个任务（进程），此任务每隔100ticks发送给当前控制台一条提示信息。

思考：通过代码分析，你能否计算在当前环境下1个tick大致等于多少ms？（提示：从函数调用OSTimeDly(100)开始）

1.4 认识uC/OS-II内核文件

本实验所使用的ucos-ii内核为2.92版本，直接来自嵌入式开发软件Keil 5.0中配套的原始文件，没有经过任何修改。ucos内核与平台无关，当跨平台移植时仅考虑适配的Port目录下文件即可（在Keil中，Port目录对应了ARM平台架构，而本实验目标架构是x86下的Win32平台）。

ucos-ii内核结构简洁清晰，仅含有12个c文件与3个h文件：



下面简要说明这些文件：

os\_core.c：系统初始化，开启多任务环境等的代码

os\_flag .c：事件标志组管理

os\_mbox .c：消息邮箱管理

os\_mem.c：内存管理

os\_mutex.c：互斥型信号量管理

os\_q.c：消息队列管理

os\_sem.c：信号量管理

os\_task.c：任务管理

os\_time.c：系统时钟

os\_tmr.c：定时器

ucos\_ii.c：内核其它

本课程实验将主要围绕以上内核文件开展。在后续章节，将分别对它们进行详细说明与代码分析，并进行必要的改造以完成相应的实验设计。

**附：实验报告——实验1**



**计算机与信息 学院实验报告**

|  |  |  |
| --- | --- | --- |
| 实验课程： | 操作系统实验 | |
| 实验编号： | 1 | |
| 实验名称： | 搭建基础实验环境 | |
| 实验人员： | 学号 |  |
| 姓名 |  |
| 班级 | 20级XXX |
| 指导教师： |  | |
| 实验室： |  | |
| 实验日期： | 2022年2月21日 | |

【注】每次实验后，将此电子版实验报告与可执行文件（\*.exe）等打包整体上传。

一、实验目的

1．阅读实验指导书，熟悉μC/OS-II项目背景及其代码组成，理解操作系统在不同处理器架构下移植的概念。

2．通过建立第一个uC/OS-II工程项目，掌握使用VS2010新建uC/OS-II工程的具体过程与方法。

3．认识如何在uC/OS-II操作系统中启动用户程序。

二、实验内容与要求

1．对照实验指导书，根据已给出的三个源码文件夹App、Port和Source，建立第一个uC/OS-II控制台工程。

2．调试程序，并回答问题：

1）在用户程序Test.c中，内核函数OSTaskCreate主要做了什么事情？操作系统启动的第一个任务是什么？

启动一个任务，启动的是TaskStart

1. 将控制台每隔100ticks输出更改为每隔200ticks。

OSTimeDly(200); /\* Wait 100 ticks \*/

3）分析函数调用关系后回答：一个tick约等于实际时间的多少ms？

3．在控制台输出开发者信息（学号、姓名）。

4．仅提交最终生成的exe可执行文件。

三、设计步骤（学生填写）

1．概要设计（文字）

1）打开VS2010，新建工程a001，

……

2）将a001工程的xxx属性值调整为

……

3）新建筛选器，将目录App、Port和Source中源码文件加入工程

4）对文件xxx的xxx代码部分进行调试，测试需要实现的xxx功能：

2．详细设计（文字+关键代码）

为实现xxx功能，在xxx.c文件中作如下代码更改：

四、结果分析（学生填写）

1．实现基本功能（输出截图）

2．存在的问题（文字+部分代码 简要总结）