

OpenMP 实验-参考教程

V0.3

华南理工大学 计算机学院

目录

一、OpenMP 运行环境配置.....	3
二、OpenMP 常用指令和库函数介绍.....	4

一、OpenMP 运行环境配置

OpenMP 程序可以在多种环境和编程语言下开发，许多编译器和工具实现了 OpenMP API ，
<https://www.openmp.org/resources/openmp-compilers-tools/>
本文档以 VS2010 为例。

1、系统环境

操作系统：Win7

开发 IDE：VS 2010

2、配置 OpenMP 开发和运行环境

现流行的 C/C++编译器基本上都支持 openmp 编程，只需在编译器上启动 openmp 选项即可。但是在开启 OpenMP 编译选项之后程序较难调试。建议在单线程环境下进行 debug，确定程序运行正确后在多线程模式下进行实验。

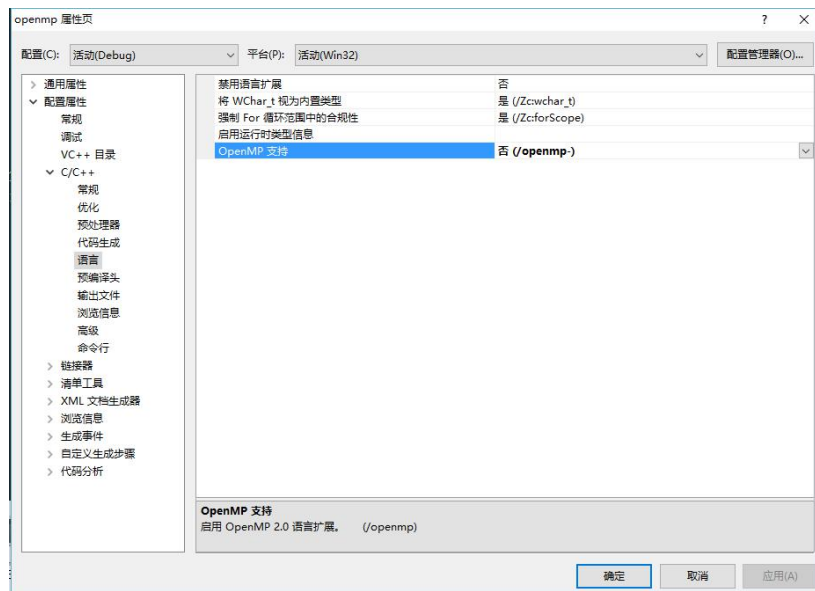
基于此，针对 Visual Studio 做如下配置：

1. 配置 Debug 环境（即开发环境）

在解决方案管理器选择“Debug”

右击项目，选择“Property Pages(属性页)”；

点击“Configuration Properties(配置属性)”，找到并点击“C/C++”，找到“Language(语言)”选项，里面有关于 openmp 的选项“OpenMP Support(OpenMP 支持)”，修改该选项为“否”



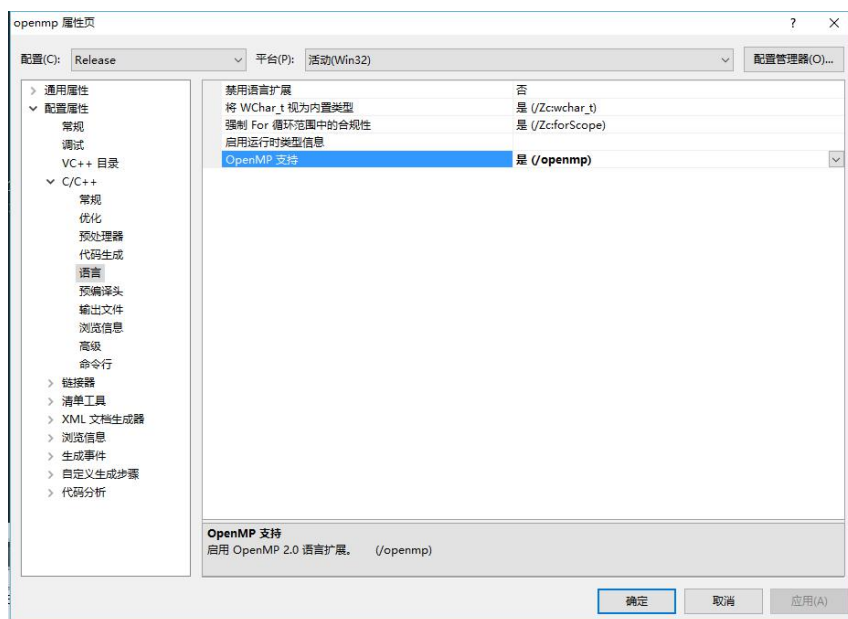
2. 配置 Release 环境（即运行环境）

在解决方案管理器选择“Release”

右击项目，选择“Property Pages(属性页)”；

点击“Configuration Properties(配置属性)”，找到并点击“C/C++”，找到“Language(语言)”

言)”选项，里面有关于 openmp 的选项“OpenMP Support(OpenMP 支持)”，修改该选项为 Yes。



注：如果是 Linux 环境，使用 `gcc -fopenmp` 编译程序

二、OpenMP 常用指令和库函数介绍

在 C/C++ 中，OpenMP 指令（编译指导语句）使用的格式为：

#pragma omp 指令 [子句[子句]...]

例如：`#pragma omp parallel private(i, j)`

`parallel` 就是指令，`private` 是子句。为叙述方便把包含 `#pragma` 和 OpenMP 指令的一行叫做语句，如上面那行叫 `parallel` 语句。

OpenMP 的常用指令：

parallel: 用在一个代码段之前，表示这段代码将被多个线程并行执行

for: 用于 `for` 循环之前，将循环分配到多个线程中并行执行，必须保证每次循环之间无相关性。

parallel for: `parallel` 和 `for` 语句的结合，也是用在一个 `for` 循环之前，表示 `for` 循环的代码将被多个线程并行执行。

critical: 用在一段代码临界区之前

single: 用在一段只被单个线程执行的代码段之前，表示后面的代码段将被单线程执行。

barrier: 用于并行区内代码的线程同步，所有线程执行到 **barrier** 时要停止，直到所有线程都执行到 **barrier** 时才继续往下执行。

atomic: 用于指定一块内存区域被制动更新

master: 用于指定一段代码块由主线程执行

ordered: 用于指定并行区域的循环按顺序执行

threadprivate: 用于指定一个变量是线程私有的。

下面列出几个常用的库函数：

omp_get_wtime,	返回当前时间（以秒为单位）
omp_get_num_procs,	返回运行本线程的多处理机的处理器个数。
omp_get_num_threads,	返回当前并行区域中的活动线程个数。
omp_get_thread_num,	返回线程号
omp_set_num_threads,	设置并行执行代码时的线程个数
omp_init_lock,	初始化一个简单锁
omp_set_lock,	上锁操作
omp_unset_lock,	解锁操作，要和 omp_set_lock 函数配对使用。
omp_destroy_lock,	omp_init_lock 函数的配对操作函数，关闭一个锁

OpenMP 的常用子句：

private, 指定每个线程都有它自己的变量私有副本。

firstprivate, 指定每个线程都有它自己的变量私有副本，并且变量要被继承主线程中的初值。

lastprivate, 主要是用来指定将线程中的私有变量的值在并行处理结束后复制回主线程中的对应变量。

reduce, 用来指定一个或多个变量是私有的，并且在并行处理结束后这些变量要执行指定的运算。

nowait, 忽略指定中暗含的等待

num_threads, 指定线程的个数

schedule, 指定如何调度 **for** 循环迭代

shared, 指定一个或多个变量为多个线程间的共享变量

default, 用来指定并行处理区域内的变量的使用方式，缺省是 **shared**。