## LAB2\_OpenMP

Yunzhe Li 20123101

2021.12.11

符合运行真实结果的是 <mark>4)</mark>,即#pragma omp parallel for default(none) private(i) firstprivate(a) lastprivate(b)。

## 运行结果及错误声明的解释如下所示:

1) #pragma omp parallel for default(shared)

```
a=50 b=924 (expected a=50 b=1049)
```

**shared** 标识符表示在计算机内存中建立一个共有的区域,所有线程均可访问其中的变量,因此其中的 **a**、**b**、**i**均是共有的。

并行线程物理上执行的最后一个线程得到的 **b** 的值将返回主线程中显示。而多次测试,错误结果均为 924,便采用 hello 程序观察线程的物理执行情况,**如图 1 所示。**发现:标号为 6 的线程总是物理最后一个执行,于是猜测错误结果的来源每一次都是这个最后执行的线程。

而在每一个并行线程中,a的值都不会改变,因此,a的值是 expected 的。

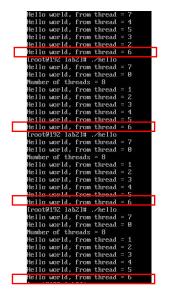


图 1 hello 程序观察线程执行物理顺序

2) #pragma omp parallel for default(none) private(i) private(a) private(b)

**Private** 标识符表示每一个线程在内存中分别建立一块独立的区域,每一块独立的区域中都存有独立的主线程中 **a**, **b**, **i** 的变量副本,各自独立,结束并行进入主线程时,都会被销毁(仅为 **private** 情况下)。有些类似函数中的局部变量。

结束并行后, a, b, i的值不会被带出。因此仍旧为 a=50, b=0.

3) #pragma omp parallel for default(none) private(i) private(a) lastprivate(b)

a=50 b=999 (expected a=50 b=1049)

**Lastprivate** 标识符表示结束并行线程后,逻辑上执行的最后一个线程的 b 的值会返回公有空间,即 999。由于 a 仍旧是 private,因此,每一个线程中的 a 是 0,如图 2 所示:

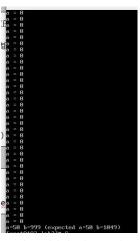


图 2 查看 a 在各线程中的运行结果

4) #pragma omp parallel for default(none) private(i) firstprivate(a) lastprivate(b)

a=50 b=1049 (expected a=50 b=1049)

Firstprivate 标识符表示在 private 内存区域基础上,会用公有区域的变量值 a,赋给并行线程中的同名变量进行初始化,因此每个线程中的 a 不再是 0,而是 50。因此结果正