**高性能并行计算第3次作业**

姓名：代宏刚 学号：2020317110061

**代码地址：**

1. **.C语言条件编译**
2. #include <stdio.h>
3. #include <stdlib.h>
4. **int** main(){
5. #ifdef \_CN
6. printf("你好！\n");
7. #else
8. prinf("Hello!\n");
9. #endif
10. **return** 0;
11. **.OpenMP Hello World 结合条件编译**
12. #include <stdio.h>
13. #include <omp.h>
14. **int** main(**int** argc,**char**\* argv[]) {
15. #pragma omp parallel
16. {
17. **int** nthreads,tid;
18. **int** nprocs;
19. tid = omp\_get\_thread\_num();
20. #ifdef \_CN
21. printf("你好，世界 多线程%d\n",tid);
22. #else
23. printf("Hello World From OMP thread %d\n ",tid);
24. #endif
25. **if**(tid==0) {
26. nthreads = omp\_get\_num\_threads();
27. #ifdef \_CN
28. printf("总线程数%d\n",nthreads);
29. #else
30. printf("Num of threads %d\n",nthreads);
31. #endif
32. }
33. }
34. #ifdef \_CN
35. printf("你好，世界\n");
36. #else
37. printf("Hello, World! \n");
38. #endif
39. **return** 0;
40. }
41. **.输出全部OpenMP环境变量**
42. #include <stdio.h>
43. #include <omp.h>
44. **int** main(){
45. **int** num\_threads;
46. omp\_set\_dynamic(0);//禁用动态调整线程数
47. omp\_set\_num\_threads(omp\_get\_num\_procs());
48. #pragma omp parallel
49. {
50. **int** id = omp\_get\_thread\_num();
51. printf("id=%d\n",id);
52. }
53. printf("-----------------------------\n");
54. **int** id = omp\_get\_thread\_num();
55. **int** max = omp\_get\_max\_threads();
56. **int** dynamics = omp\_get\_dynamic();
57. **int** procs\_num = omp\_get\_num\_procs();
58. printf("id=%d\n",id);
59. printf("max=%d\n",max);
60. printf("dynamics=%d\n",dynamics);
61. printf("procs\_num=%d\n",procs\_num);
62. **return** 0;
63. }

**实验结果：**

**①.C语言条件编译**：

|  |  |  |
| --- | --- | --- |
| 编译指令 | gcc -D\_CN if.c -o if | gcc if.c -o if |
| 输出结果 | 你好！ | Hello! |

源程序中，若定义有\_CN 则编译中文的程序，否则编译英文的程序。编译指令中若加上 -D\_CN参数，则编译中文部分，缺省则编译引英文部分

**②.OpenMP HelloWorld 结合 条件编译**

编译指令：gcc -fopenmp hello.c -o hello

输出结果：

Hello World From OMP thread 49

Hello World From OMP thread 57

Hello World From OMP thread 39

……

Hello World From OMP thread 62

Hello World From OMP thread 0

Num of threads 64

Hello, World!

编译指令： gcc -fopenmp -D\_CN hello.c -o hello

输出结果：

你好，世界 多线程62

你好，世界 多线程9

你好，世界 多线程1

……

你好，世界 多线程47

你好，世界 多线程22

你好，世界

源程序中，引入OpenMP多线程，每个线程都打印一次Hello World，因服务器有64个核，程序打印的64个Hello World。在程序中加入条件编译，若编译时加上-D\_CN，则编译中文的程序，否则，默认编译英文的程序。

**③.输出OpenMP所有的环境变量**

输出结果：

id=20

id=7

id=37

……

id=57

-----------------------------

id=0

max=64

dynamics=0

procs\_num=64

结果分析：

在并行程序中，输出每个线程的线程号。并行结束后，输出主线程的线程号id=0，最大的线程数max=64,是否动态调整线程数dynamics=0,cpu核心数proc\_num = 64