|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | | | | |
| **并行计算课程报告** | | | | |
|  | | | | |
|  | | | | |
|  | | | | |
|  | 姓名 | ： | 牟鑫一 |  |
|  | 学号 | ： | 20161001764 |  |
|  | 班级 | ： | 191174 |  |
|  | 指导老师 | ： | 李程俊 |  |

1. **题目说明**

查找给定范围内的所有素数，不设下界，即最小素数为2，上界为10000的n（整数）倍，n为运行程序时传入的参数，为了体现并行的特性，所以取较大的运算量。

1. **程序设计**

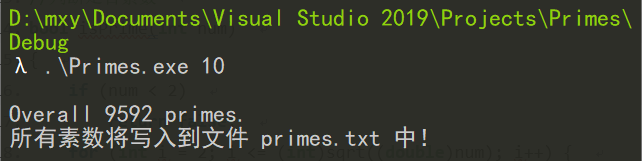
程序运行时传入线程数以及参数n，保存到main()函数默认参数argv[1]，argv[0]为程序本身，只接受一个参数，所以argc的值只允许为2，若参数个数不为2则报错并结束程序。程序查找1到10000 \* argv[1]之间的所有素数，通过多进程查找，各尽程记录找到的素数和个数，最终汇总各进程找到的素数以及总的素数个数，输出找到的素数个数并将所有素数写入到文件中保存。

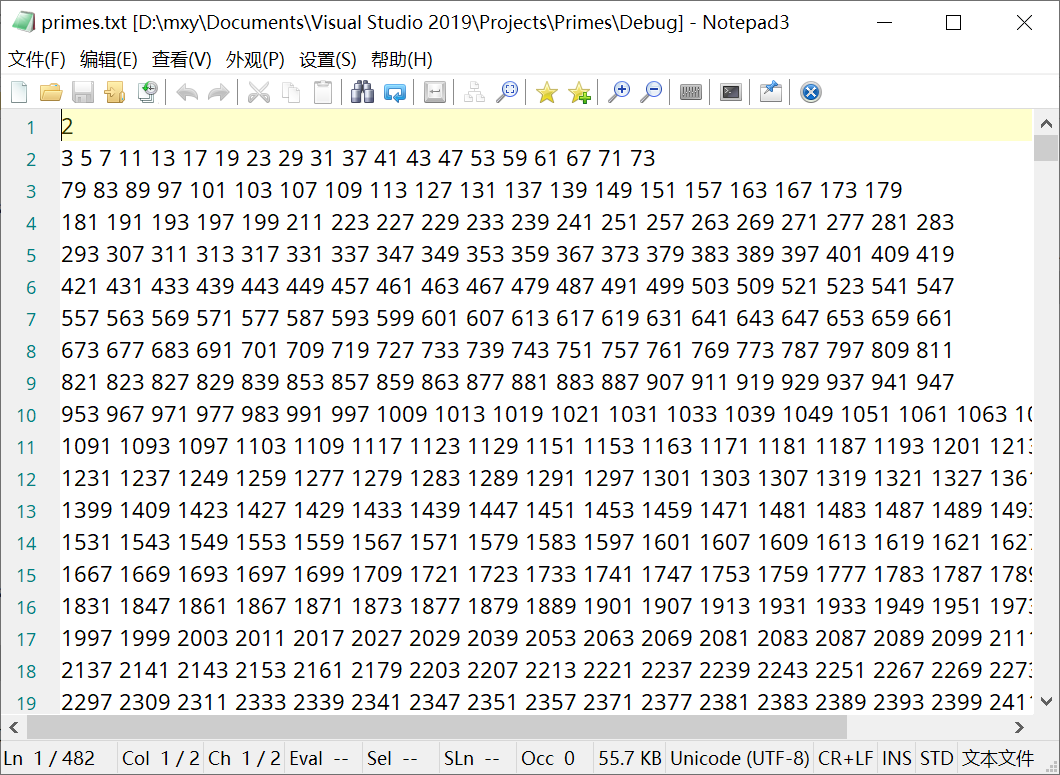
1. **串行源代码**
2. #include <stdio.h>
3. #include <iostream>
4. #include <cstdlib>
5. #include <cmath>
6. #include <vector>
8. **using** **namespace** std;
10. //checks wheather or not a number is prime with in O(n) time
11. //判断是否素数
12. **bool** isPrime(**int** num)
13. {
14. **if** (num < 2)
15. **return** **false**;
16. **for** (**int** i = 2; i <= (**int**)sqrt((**double**)num); i++) {
17. **if** (num % i == 0)
18. **return** **false**;
19. }
20. **return** **true**;
21. }
23. **int** n;//上界
24. **int**\* curPrimes;//临时保存找到的素数
25. **int** k;//素数个数
26. vector <**int**> allPrimes;//保存所有素数
28. **int** main(**int** argc, **char**\*\* argv)
29. {
30. /\*
31. //打印参数信息
32. int x = argc;//传入的参数个数
33. printf("传入参数个数: %d\n", x);//格式化输出
34. for (int i = 0; i < x; i++)
35. cout << "参数" << i + 1 << ": " << argv[i] << endl;
36. \*/
38. //check if we have the right number of arguments
39. //只接受除运行程序命令argv[0]外的一个参数argv[1]
40. **if** (argc != 2)
41. {
42. cout << "Invalid number of arguments" << endl;
43. **return** 0;
44. }
45. //calculate n as per requrenment
46. n = 10000 \* atoi(argv[1]);
47. //we cant have more than n/2 primes becuase even numbers for sure cant be prime
48. //素数不可能超过n/2,因为偶数不可能是素数
49. curPrimes = **new** **int**[n / 2];//分配n/2空间的数组
51. //calculate prime numbers each process will calculate its own portion
52. **for** (**int** i = 1; i < n; i++)
53. **if** (isPrime(i))
54. curPrimes[k++] = i;
55. //add the primes calculated here to our large prime vector
56. **for** (**int** i = 0; i < k; i++)
57. allPrimes.push\_back(curPrimes[i]);
58. //clear the memory of our primes
59. **delete**[] curPrimes;
61. string file\_name = "primes.txt";
62. cout << "\nOverall " << allPrimes.size() << " primes. " << endl;
63. cout << "所有素数将写入到文件 " << file\_name << " 中！" << endl;
64. //output all primes to a file
65. **FILE**\* stream;
66. freopen\_s(&stream, "primes.txt", "w", stdout);
67. **for** (**int** i = 0; i < allPrimes.size(); i++)
68. {
69. cout << allPrimes[i] << " ";
70. **if** (i % 20 == 0)
71. cout << "\n";
72. }
73. printf("\n");
74. **return** 0;
75. }
76. **串行测试**

**测试方法：**

在程序所在文件夹路径下，命令行模式输入命令.\Primes.exe <input>，其中<input>为整数n，即程序所求10000 \* n以内的素数

**测试：**100000（10000 \* 10）以内的素数



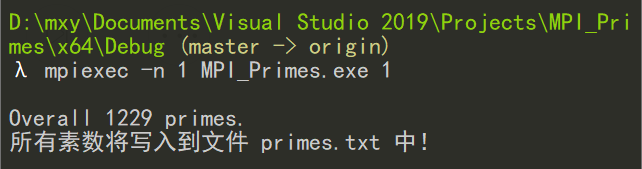


1. **并行源代码**
2. #include <mpi.h>
3. #include <stdio.h>
4. #include <iostream>
5. #include <cstdlib>
6. #include <cmath>
7. #include <vector>
9. #define \_CRT\_SECURE\_NO\_WARNINGS
11. **using** **namespace** std;
13. //checks wheather or not a number is prime with in O(n) time
14. //判断是否素数
15. **bool** isPrime(**int** num)
16. {
17. **if** (num < 2)
18. **return** **false**;
19. **for** (**int** i = 2; i <= (**int**)sqrt((**double**)num); i++) {
20. **if** (num % i == 0)
21. **return** **false**;
22. }
23. **return** **true**;
24. }
26. **int** n;//上界
27. **int** world\_size; //number of processes involved in our calculations
28. **int**\* curPrimes;//the primes that we will calculate in THIS process
29. **int** k;//素数个数
30. **int**\* recvPrimes;//temperary variable for primes the main process will recieve from secondary ones
31. vector <**int**> allPrimes;//保存所有素数
33. **int** main(**int** argc, **char**\*\* argv)
34. {
35. /\*
36. //打印参数信息
37. int x = argc;//传入的参数个数
38. printf("传入参数个数: %d\n", x);//格式化输出
39. for (int i = 0; i < x; i++)
40. cout << "参数" << i + 1 << ": " << argv[i] << endl;
41. \*/
43. //check if we have the right number of arguments
44. //只接受除运行程序命令argv[0]外的一个参数argv[1]
45. **if** (argc != 2)
46. {
47. cout << "invalid number of arguments" << endl;
48. **return** 0;
49. }
50. //calculate n as per requrenment
51. n = 10000 \* atoi(argv[1]);
52. //we cant have more than n/2 primes becuase even numbers for sure cant be prime
53. //素数不可能超过n/2,因为偶数不可能是素数
54. curPrimes = **new** **int**[n / 2];//分配n/2空间的数组
56. //init mpi
57. MPI\_Init(NULL, NULL);
59. //find out how many processes are there
60. //当前有多少个进程
61. MPI\_Comm\_size(MPI\_COMM\_WORLD, &world\_size);
62. //find out which process are we in
63. //找出当前所在进程
64. **int** world\_rank;
65. MPI\_Comm\_rank(MPI\_COMM\_WORLD, &world\_rank);
67. //calculate prime numbers each process will calculate its own portion
68. **for** (**int** i = n \* world\_rank / world\_size; i < n \* (world\_rank + 1) / world\_size; i++)
69. **if** (isPrime(i))
70. curPrimes[k++] = i;
71. //if we are not in the main process(process with rank 0) send our prime numbers to main process
72. **if** (world\_rank != 0)
73. {
74. MPI\_Send(&k, 1, MPI\_INT, 0, 0, MPI\_COMM\_WORLD);
75. MPI\_Send(curPrimes, k, MPI\_INT, 0, 0, MPI\_COMM\_WORLD);
76. }
77. **int** recvSize;
78. **if** (world\_rank == 0)
79. {
80. //this runs only on main process
81. //add the primes calculated here to our large prime vector
82. **for** (**int** i = 0; i < k; i++)
83. allPrimes.push\_back(curPrimes[i]);
84. //clear the memory of our primes
85. **delete**[] curPrimes;
86. //go through all the recieved primes and add them to our vector
87. **for** (**int** i = 1; i < world\_size; i++)
88. {
89. MPI\_Recv(&recvSize, 1, MPI\_INT, i, 0, MPI\_COMM\_WORLD, MPI\_STATUS\_IGNORE);
90. cout << "Recieved " << recvSize << " primes\nretrieving" << endl;
91. recvPrimes = **new** **int**[recvSize];
92. MPI\_Recv(recvPrimes, recvSize, MPI\_INT, i, 0, MPI\_COMM\_WORLD, MPI\_STATUS\_IGNORE);
93. **for** (**int** j = 0; j < recvSize; j++)
94. allPrimes.push\_back(recvPrimes[j]);
95. **delete**[] recvPrimes;
96. }
98. string file\_name = "primes.txt";
99. cout << "\nOverall " << allPrimes.size() << " primes. " << endl;
100. cout << "所有素数将写入到文件 " << file\_name << " 中！" << endl;
101. //output all primes to a file
102. **FILE**\* stream;
103. freopen\_s(&stream, "primes.txt", "w", stdout);
104. **for** (**int** i = 0; i < allPrimes.size(); i++)
105. {
106. cout << allPrimes[i] << " ";
107. **if** (i % 20 == 0)
108. cout << "\n";
109. }
110. printf("\n");
111. }
112. MPI\_Finalize();
113. **return** 0;
114. }
115. **并行测试**

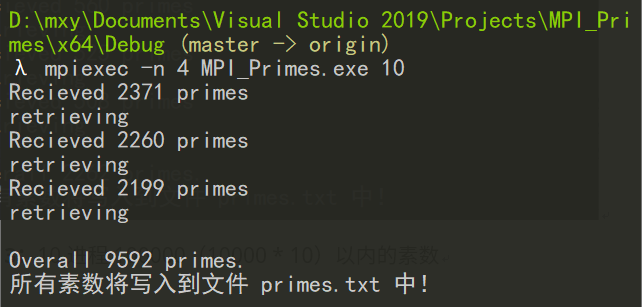
**测试方法：**

在程序所在文件夹路径下，命令行模式输入命令mpiexec -n <number of processes> MPI\_Primes.exe <input>，其中<number of processes>即进程数，<input>为整数n，即程序所求10000 \* n以内的素数

**测试1：**单进程10000（10000 \* 1）以内的素数



**测试2：**4进程100000（10000 \* 10）以内的素数



**测试3：**10进程100000（10000 \* 10）以内的素数

