CSC4005 High Performance Computing (2018/19)

**Assignment 1 - Sequential Searching and Parallel Searching using OpenMP REPORT**

Tianbai Peng

1.Abstract

2.introduction

3. Content

Part A :

The Worst case:

Time function:

High precision time function - gettimeofday() has been choose to be used on this code because the time it returns accurate to microseconds which meet the requirement perfectly. What’s more it’s also in the time.h.

Snippet of code:

[Length of Pattern and text choose] part:



Folder creation part, Generate text and pattern, and write them to each document:



Selection of pattern length:

Points selection:

Results(output):

Read test number 0

Text length = 50

Pattern length = 2

Pattern found at position 48

# comparisons = 98

Test 0 elapsed wall clock time = 0.000006

Test 0 elapsed CPU time = 0.000000

Read test number 1

Text length = 25

Pattern length = 4

Pattern found at position 21

# comparisons = 88

Test 1 elapsed wall clock time = 0.000005

Test 1 elapsed CPU time = 0.000000

Read test number 2

Text length = 20

Pattern length = 5

Pattern found at position 15

# comparisons = 80

Test 2 elapsed wall clock time = 0.000006

Test 2 elapsed CPU time = 0.000000

Read test number 3

Text length = 10

Pattern length = 10

Pattern found at position 0

# comparisons = 10

Test 3 elapsed wall clock time = 0.000004

Test 3 elapsed CPU time = 0.000000

Read test number 4

Text length = 500

Pattern length = 20

Pattern found at position 480

# comparisons = 9620

Test 4 elapsed wall clock time = 0.000032

Test 4 elapsed CPU time = 0.000000

Read test number 5

Text length = 250

Pattern length = 40

Pattern found at position 210

# comparisons = 8440

Test 5 elapsed wall clock time = 0.000028

Test 5 elapsed CPU time = 0.000000

Read test number 6

Text length = 200

Pattern length = 50

Pattern found at position 150

# comparisons = 7550

Test 6 elapsed wall clock time = 0.000025

Test 6 elapsed CPU time = 0.000000

Read test number 7

Text length = 100

Pattern length = 100

Pattern found at position 0

# comparisons = 100

Test 7 elapsed wall clock time = 0.000004

Test 7 elapsed CPU time = 0.000000

Read test number 8

Text length = 5000

Pattern length = 200

Pattern found at position 4800

# comparisons = 960200

Test 8 elapsed wall clock time = 0.002570

Test 8 elapsed CPU time = 0.000000

Read test number 9

Text length = 2500

Pattern length = 400

Pattern found at position 2100

# comparisons = 840400

Test 9 elapsed wall clock time = 0.002281

Test 9 elapsed CPU time = 0.000000

Read test number 10

Text length = 2000

Pattern length = 500

Pattern found at position 1500

# comparisons = 750500

Test 10 elapsed wall clock time = 0.002368

Test 10 elapsed CPU time = 0.000000

Read test number 11

Text length = 1000

Pattern length = 1000

Pattern found at position 0

# comparisons = 1000

Test 11 elapsed wall clock time = 0.000007

Test 11 elapsed CPU time = 0.000000

Read test number 12

Text length = 50000

Pattern length = 2000

Pattern found at position 48000

# comparisons = 96002000

Test 12 elapsed wall clock time = 0.264886

Test 12 elapsed CPU time = 0.260000

Read test number 13

Text length = 25000

Pattern length = 4000

Pattern found at position 21000

# comparisons = 84004000

Test 13 elapsed wall clock time = 0.234184

Test 13 elapsed CPU time = 0.220000

Read test number 14

Text length = 20000

Pattern length = 5000

Pattern found at position 15000

# comparisons = 75005000

Test 14 elapsed wall clock time = 0.209121

Test 14 elapsed CPU time = 0.210000

Read test number 15

Text length = 10000

Pattern length = 10000

Pattern found at position 0

# comparisons = 10000

Test 15 elapsed wall clock time = 0.000031

Test 15 elapsed CPU time = 0.000000

Read test number 16

Text length = 500000

Pattern length = 20000

Pattern found at position 480000

# comparisons = 9600020000

Test 16 elapsed wall clock time = 26.139411

Test 16 elapsed CPU time = 25.629999

Read test number 17

Text length = 250000

Pattern length = 40000

Pattern found at position 210000

# comparisons = 8400040000

Test 17 elapsed wall clock time = 22.517896

Test 17 elapsed CPU time = 22.389999

Read test number 18

Text length = 200000

Pattern length = 50000

Pattern found at position 150000

# comparisons = 7500050000

Test 18 elapsed wall clock time = 20.308212

Test 18 elapsed CPU time = 20.070000

Read test number 19

Text length = 100000

Pattern length = 100000

Pattern found at position 0

# comparisons = 100000

Test 19 elapsed wall clock time = 0.000357

Test 19 elapsed CPU time = 0.000000

Results(graphs):

Part B:

Result:

[40120405@login1(kelvin) assignment-1b]$ $time ./searching\_sequential

Read test number 0

Text length = 2000000

Pattern length = 2000

Pattern found at position 1998000

# comparisons = 3996002000

Test 0 elapsed wall clock time = 11

Test 0 elapsed CPU time = 10.680000

Read test number 1

Text length = 10000000

Pattern length = 1000

Pattern found at position 1001

# comparisons = 1002000

Test 1 elapsed wall clock time = 0

Test 1 elapsed CPU time = 0.000000

Read test number 2

Text length = 10000000

Pattern length = 1000

Pattern found at position 5001001

# comparisons = 5001002000

Test 2 elapsed wall clock time = 14

Test 2 elapsed CPU time = 13.380000

[40120405@login1(kelvin) assignment-1b]$ $time ./searching\_OMP\_0

Read test number 0

Text length = 2000000

Pattern length = 2000

Pattern found at position 1998000

# comparisons = 1264070266

Test 0 elapsed wall clock time = 16

Test 0 elapsed CPU time = 55.310001

Read test number 1

Text length = 10000000

Pattern length = 1000

Pattern found at position 1001

# comparisons = -398171989

Test 1 elapsed wall clock time = 41

Test 1 elapsed CPU time = 114.849998

Read test number 2

Text length = 10000000

Pattern length = 1000

Pattern found at position 5015996

# comparisons = 1512478177

Test 2 elapsed wall clock time = 22

Test 2 elapsed CPU time = 74.320000

conclusion