Міністерство освіти і науки України

Національний технічний університет України

«Київський політехнічний інститут»

Факультет інформатики та обчислювальної техніки

Кафедра обчислювальної техніки

**Лабораторна робота № 5**

**з дисципліни**

**«Паралельні та розподілені обчислення»**

Виконав: студент групи ІО-93

Глухенько Костянтин Анатолійович

2011 р.

**Ціль роботи:** вивчення засобів бібліотеки Win32 для роботи з потоками.

**Завдання:**  розробити програму, яка вміщує паралельні потоки, кожен з яких реалізує функцію F1, F2, F3. Дослідити вплив пріоритетів потоків на чергу запуску та виконання задач.

**Лістинг**

lab5.cpp

1. /\*Глухенько Костянтин Анатолійович, група ІО-93
2. лаб№ 3. Потоки в мові JAVA
3. F1: 1 16 d = ((A + B)\* C)
4. F2: 2 24 MG = SORT(MA - MB \* MC)
5. F3: 3 22 S = SORT(MA\*T)\*/
6. #include "stdafx h"
7. #include "funcs h"
8. #include <iostream>
9. #include <fstream>
10. #include <Windows h>
11. using std::cout;
12. using std::endl;
13. DWORD WINAPI Thread\_F1(PVOID pParam){
14. std:: cout << "T1 started" << std::endl;
15. DWORD dwResult = 0;
16. int val = 1;
17. int A[N], B[N], C[N];
18. for(int i = 0; i < N; i++){
19. A[i] = val;
20. B[i] = val;
21. C[i] = val;
22. }
23. F1(A, B, C);
24. Sleep(1000);
25. std:: cout << "T1 finished" << std::endl;
26. return dwResult;
27. }
28. DWORD WINAPI Thread\_F2(PVOID pParam){
29. std:: cout << "T2 started" << std::endl;
30. DWORD dwResult = 0;
31. int val = 1;
32. Matrix MA, MB, MC;
33. for(int i = 0; i < N; i++)
34. for(int j = 0; j < N; j++){
35. MA mas[i][j]=1;
36. MB mas[i][j]=1;
37. MC mas[i][j]=1;
38. }
39. F2(MA, MB, MC);
40. Sleep(1000);
41. std:: cout << "T2 finished" << std::endl;
42. return dwResult;
43. }
44. DWORD WINAPI Thread\_F3(PVOID pParam){
45. std:: cout << "T3 started" << std::endl;
46. DWORD dwResult = 0;
47. int val = 1;
48. int T[N];
49. Matrix MA;
50. for(int i = 0; i < N; i++){
51. for(int j = 0; j < N; j++)
52. MA mas[i][j]=1;
53. T[i] = val;
54. }
55. F3(MA, T);
56. Sleep(1000);
57. std:: cout << "T3 finished" << std::endl;
58. return dwResult;
59. }
60. int \_tmain(int argc, \_TCHAR\* argv[])
61. {
62. cout << "lab5 started"<< endl;
63. DWORD t1, t2, t3;
64. HANDLE T1 = CreateThread(NULL, 0, Thread\_F1, 0, 0, &t1);
65. HANDLE T2 = CreateThread(NULL, 0, Thread\_F2, 0, 0, &t2);
66. HANDLE T3 = CreateThread(NULL, 0, Thread\_F3, 0, 0, &t3);
67. SetThreadPriority(T1, THREAD\_PRIORITY\_BELOW\_NORMAL);
68. SetThreadPriority(T2, THREAD\_PRIORITY\_TIME\_CRITICAL);
69. SetThreadPriority(T3, -3);
70. Sleep(1000);
71. WaitForSingleObject(T1, INFINITE);
72. WaitForSingleObject(T2, INFINITE);
73. WaitForSingleObject(T3, INFINITE);
74. cout << "lab5 finished"<< endl;
75. return 0;
76. }

funcs.h

1 /\*Глухенько Костянтин Анатолійович, група ІО-93

2 лаб№ 3. Потоки в мові JAVA

3 F1: 1.16 d = ((A + B)\* C)

4 F2: 2.24 MG = SORT(MA - MB \* MC)

5 F3: 3.22 S = SORT(MA\*T)\*/

6 #pragma once

7 int const N = 3;

8 struct Matrix{

9 int mas[N][N];

10 };

11 void F1(int A[], int B[], int C[]);

12 void F2(Matrix MA, Matrix MB, Matrix MC);

13 void F3(Matrix MA, int T[]);

14

funcs.cpp

1 /\*Глухенько Костянтин Анатолійович, група ІО-93

2 лаб№ 3. Потоки в мові JAVA

3 F1: 1.16 d = ((A + B)\* C)

4 F2: 2.24 MG = SORT(MA - MB \* MC)

5 F3: 3.22 S = SORT(MA\*T)\*/

6 #include "StdAfx.h"

7 #include "funcs.h"

8 #include <iostream>

9 #include <fstream>

10

11 int\* PLUS(int ARG1[], int ARG2[]){

12 static int RESULT[N];

13 for(int i=0; i < N; i++)

14 RESULT[i] = ARG1[i] + ARG2[i];

15 return RESULT;

16 }

17 int MUL(int ARG1[], int ARG2[]){

18 int RESULT = 0;

19 for(int i=0; i < N; i++)

20 RESULT += ARG1[i] \* ARG2[i];

21 return RESULT;

22 }

23

24 Matrix MINUS(Matrix ARG1, Matrix ARG2){

25 static Matrix RESULT;

26 for(int i=0; i < N; i++)

27 for(int j=0; j < N; j++)

28 RESULT.mas[i][j] = ARG1.mas[i][j] - ARG2.mas[i][j];

29 return RESULT;

30 }

31 Matrix MUL(Matrix ARG1, Matrix ARG2){

32 static Matrix RESULT;

33 for(int i=0; i<N; i++)

34 for(int j=0; j<N; j++)

35 for(int k=0; k<N; k++)

36 RESULT.mas[i][j] += ARG1.mas[i][k] \* ARG2.mas[k][j];

37 return RESULT;

38 }

39 Matrix SORT(Matrix ARG){

40 for(int i=0; i<N; i++)

41 for(int j=0; j<N; j++){

42 for(int k=j+1; k<N; k++)

43 i (ARG.mas[i][j] < ARG.mas[i][k]){

44 int t = ARG.mas[i][j];

45 ARG.mas[i][j] = ARG.mas[i][k];

46 ARG.mas[i][k] = t;

47 }

48 for(int k=i+1; k<N; k++)

49 for(int m=0; m<N; m++)

50 i (ARG.mas[i][j] < ARG.mas[k][m]){

51 int t = ARG.mas[i][j];

52 ARG.mas[i][j] = ARG.mas[k][m];

53 ARG.mas[k][m] = t;

54 }

55 }

56 return ARG;

57 }

58 int\* MUL(Matrix ARG1, int ARG2[]){

59 static int RESULT[N];

60 for(int i=0; i<N; i++)

61 for(int j=0; j<N; j++)

62 RESULT[i] += ARG1.mas[i][j] \* ARG2[j];

63 return RESULT;

64 }

65 int\* SORT(int\* ARG){

66 for(int i=0; i<N-1; i++)

67 for(int j=i+1; j<N; j++)

68 i (ARG[i] > ARG[j]){

69 int t = ARG[j];

70 ARG[j] = ARG[i];

71 ARG[i] = t;

72 }

73 return ARG;

74 }

75 void F1(int A[], int B[], int C[]){

76 int d = MUL(PLUS(A, B), C);

77 std:: ofstream out("F1.txt");

78 out << "d = ((A + B)\* C)= " << d;

79 out.close();

80 }

81 void F2(Matrix MA, Matrix MB, Matrix MC){

82 Matrix MG = SORT(MINUS(MA, MUL(MA, MB)));

83 std:: ofstream out("F2.txt");

84 out << "MG = SORT(MA - MB \* MC)" << std::endl;

85 for(int i = 0; i < N; i++){

86 for(int j = 0; j < N; j++)

87 out << MG.mas[i][j] << ' ';

88 out << std::endl;

89 }

90 out.close();

91 }

92 void F3(Matrix MA, int T[]){

93 int\* S = SORT(MUL(MA, T));

94 std:: ofstream out("F3.txt");

95 out << "S = SORT(MA\*T)" << std::endl;

96 for(int i = 0; i < N; i++)

97 out << S[i] << ' ';

98 out << std::endl;

99 out.close();

100 }

101