Міністерство освіти і науки, молоді та спорту України

Національний технічний університет України

«Київський політехнічний інститут»

Факультет інформатики та обчислювальної техніки

Кафедра обчислювальної техніки

# Лабораторна робота №2

# з курсу «Паралельні та розподілені обчислення-2»

**Тема:** Паралельне програмування за допомогою бібліотеки Win32

Виконав: студент групи ІО-93

Глухенько Костянтин Анатолійович

2012 р.

**Завдання**

Математичний вираз: a = max(MB\*MC-α\*MX)

Засоби програмування: Win32 (Семафори, мютекси, події, критичні секції)

Структура ПКС:

ЗП

Т3

Т4

Т1

Т1

MC

MX

a α MB

**Виконання**

**Етап 1.** Побудова паралельного алгоритму.

a=max(a, ai) де ai=max(MB\*MCH-α\*MXH); i=1,P; Н=N/2 СР: МВ, α, a

**Етап 2.** Розроблення алгоритмів роботи кожного процесу.

|  |  |  |
| --- | --- | --- |
|  | Т1 | Точки синхронізації |
| 1) | Введення МВ і α | КД |
| 2) | Сигнал задачі Т2, Т3, Т4 про введення МВ | S2,3,4-1 |
| 3) | Копіювання МВ1 = МВ | КД |
| 4) | Копіювання α1 = α | КД |
| 5) | Чекати введення МС в задачі Т2 | W2-1 |
| 6) | Чекати введення МХ в задачі Т3 | W3-1 |
| 7) | Обчислення ai=max(MB1\*MCH-α1\*MXH) |  |
| 8) | Обчислення a=max(a, ai) | КД |
| 9) | Чекати завершення обчислень в задачі Т2 | W2-2 |
| 10) | Чекати завершення обчислень в задачі Т3 | W3-2 |
| 11) | Чекати завершення обчислень в задачі Т4 | W4-1 |
| 12) | Виведення а | КД |

|  |  |  |
| --- | --- | --- |
|  | Т2 | Точки синхронізації |
| 1) | Введення МС | КД |
| 2) | Сигнал задачі Т1, Т3, Т4 про введення МС | S1,3,4-1 |
| 3) | Чекати введення МВ і α в задачі Т1 | W1-1 |
| 4) | Копіювання МВ2 = МВ | КД |
| 5) | Копіювання α2 = α | КД |
| 6) | Чекати введення МХ в задачі Т3 | W3-1 |
| 7) | Обчислення ai=max(MB2\*MCH-α2\*MXH) |  |
| 8) | Обчислення a=max(a, ai) | КД |
| 9) | Сигнал задачі Т1 про завершення обчислень | S1-2 |
|  | Т3 | Точки синхронізації |
| 1) | Введення МХ | КД |
| 2) | Сигнал задачі Т1, Т2, Т4 про введення МХ | S1,2,4-1 |
| 3) | Чекати введення МВ і α в задачі Т1 | W1-1 |
| 4) | Копіювання МВ3 = МВ | КД |
| 5) | Копіювання α3 = α | КД |
| 6) | Чекати введення МС в задачі Т2 | W2-1 |
| 7) | Обчислення ai=max(MB3\*MCH-α3\*MXH) |  |
| 8) | Обчислення a=max(a, ai) | КД |
| 9) | Сигнал задачі Т1 про завершення обчислень | S1-2 |

|  |  |  |
| --- | --- | --- |
|  | Т4 | Точки синхронізації |
| 1) | Чекати введення МВ і α в задачі Т1 | W1-1 |
| 2) | Копіювання МВ4 = МВ | КД |
| 3) | Копіювання α4 = α | КД |
| 4) | Чекати введення МС в задачі Т2 | W2-1 |
| 5) | Чекати введення МХ в задачі Т3 | W3-1 |
| 6) | Обчислення ai=max(MB4\*MCH-α4\*MXH) |  |
| 7) | Обчислення a=max(a, ai) | КД |
| 8) | Сигнал задачі Т1 про завершення обчислень | S1-1 |

**Етап 3.** Розроблення структурної схеми взаємодії задач.

E2

E1

T2

T1

E2

E2

E1

E1

S1

T3

T4

E3

E3

E3

S1

S1

CP

M1

M2

M3

CS1

На структурній схемі введені такі елементи:

S1 для синхронізації завершення обчислень

Е1 для синхронізації із завершенням введення МВ і α

Е2 для синхронізації із завершення введення МС

Е3 для синхронізації із завершення введення МХ

CS1 для керування доступом до спільного ресурсу (а)

М1 для керування доступом до спільного ресурсу (ПВВ)

М2 для керування доступом до спільного ресурсу (МВ)

М3 для керування доступом до спільного ресурсу (α)

**Етап 4.** Розроблення програми**.**

1. {Глухенько Костянтин Анатолійович, група ІО-93
2. лаб№ 2. Паралельне програмування за допомогою бібліотеки Win32
3. a = max(MB\*MC-alfa\*MX)
4. }
5. program Project1;
6. {$APPTYPE CONSOLE}
7. uses
8. SysUtils,
9. Windows;
10. var
11. M1, M2, M3, S1, E1, E2, E3, T1, T2, T3, T4 : THandle;
12. CS1: TRTLCriticalSection;
13. MB, MC, MX: array of array of integer;
14. a, alfa, t, H, N: integer;
15. //=================Thread T1
16. procedure T1\_function();
17. var
18. valueMB, alfa1, i, j, k, ai, t: integer;
19. MB1: array of array of integer;
20. begin
21. SetLength(MB1, N, N);
22. ai := -2147483648;
23. //Введення МВ і альфа КД
24. WaitForSingleObject(M1, INFINITE);
25. write('Enter value for MB: ');
26. readln(valueMB);
27. for i := 0 to N - 1 do
28. for j := 0 to N - 1 do
29. MB[i][j] := valueMB;
30. write('Enter alfa: ');
31. readln(alfa);
32. ReleaseMutex(M1);
33. //Сигнал про введення МВ і альфа
34. setEvent(E1);
35. //Копіювання МВ
36. WaitForSingleObject(M2, INFINITE);
37. for i := 0 to N - 1 do
38. for j := 0 to N - 1 do
39. MB1[i][j] := MB[i][j];
40. ReleaseMutex(M2);
41. //Копіювання альфа
42. WaitForSingleObject(M3, INFINITE);
43. alfa1 := alfa;
44. ReleaseMutex(M3);
45. //Чекати МС
46. WaitForSingleObject(E2, INFINITE);
47. //Чекати МХ
48. WaitForSingleObject(E3, INFINITE);
49. //ai = max(MB\*MCh-alfa\*MXh)
50. for i := 0 to H - 1 do
51. for j := 0 to N - 1 do
52. begin
53. t := 0;
54. for k := 0 to N - 1 do
55. t := t + MC[i][k] \* MB1[k][j];
56. t := t - (alfa\*MX[i][j]);
57. if t > ai then
58. ai := t;
59. end;
60. //a= max(a, ai)
61. EnterCriticalSection(CS1);
62. if ai > a then
63. a := ai;
64. LeaveCriticalSection(CS1);
65. //Чекати завершення обчислень
66. for i := 1 to H - 1 do
67. WaitForSingleObject(S1, INFINITE);
68. //Виведення а
69. WaitForSingleObject(M1, INFINITE);
70. write('a= ', a);
71. ReleaseMutex(M1);
72. end;
73. //=================Thread T2
74. procedure T2\_function();
75. var
76. valueMC, alfa2, i, j, k, ai, t: integer;
77. MB2: array of array of integer;
78. begin
79. SetLength(MB2, N, N);
80. ai := -2147483648;
81. //Введення МC
82. WaitForSingleObject(M1, INFINITE);
83. write('Enter value for MC: ');
84. readln(valueMC);
85. for i := 0 to N - 1 do
86. for j := 0 to N - 1 do
87. MC[i][j] := valueMC;
88. ReleaseMutex(M1);
89. //Сигнал про введення МC
90. setEvent(E2);
91. //Чекати введення МВ і альфа
92. WaitForSingleObject(E1, INFINITE);
93. //Копіювання МВ
94. WaitForSingleObject(M2, INFINITE);
95. for i := 0 to N - 1 do
96. for j := 0 to N - 1 do
97. MB2[i][j] := MB[i][j];
98. ReleaseMutex(M2);
99. //Копіювання альфа
100. WaitForSingleObject(M3, INFINITE);
101. alfa2 := alfa;
102. ReleaseMutex(M3);
103. //Чекати МХ
104. WaitForSingleObject(E3, INFINITE);
105. //ai = max(MB\*MCh-alfa\*MXh)
106. for i := H to (H shl 1) - 1 do
107. for j := 0 to N - 1 do
108. begin
109. t := 0;
110. for k := 0 to N - 1 do
111. t := t + MC[i][k] \* MB2[k][j];
112. t := t - (alfa2\*MX[i][j]);
113. if t > ai then
114. ai := t;
115. end;
116. //a= max(a, ai)
117. EnterCriticalSection(CS1);
118. if ai > a then
119. a := ai;
120. LeaveCriticalSection(CS1);
121. //Сигнал про завершення обчислень
122. ReleaseSemaphore(S1, 1, nil);
123. end;
124. //=================Thread T3
125. procedure T3\_function();
126. var
127. valueMX, alfa3, i, j, k, ai, t: integer;
128. MB3: array of array of integer;
129. begin
130. SetLength(MB3, N, N);
131. ai := -2147483648;
132. //Введення МX
133. WaitForSingleObject(M1, INFINITE);
134. write('Enter value for MX: ');
135. readln(valueMX);
136. for i := 0 to N - 1 do
137. for j := 0 to N - 1 do
138. MX[i][j] := valueMX;
139. ReleaseMutex(M1);
140. //Сигнал про введення МX
141. setEvent(E3);
142. //Чекати введення МВ і альфа
143. WaitForSingleObject(E1, INFINITE);
144. //Копіювання МВ
145. WaitForSingleObject(M2, INFINITE);
146. for i := 0 to N - 1 do
147. for j := 0 to N - 1 do
148. MB3[i][j] := MB[i][j];
149. ReleaseMutex(M2);
150. //Копіювання альфа
151. WaitForSingleObject(M3, INFINITE);
152. alfa3 := alfa;
153. ReleaseMutex(M3);
154. //Чекати МC
155. WaitForSingleObject(E2, INFINITE);
156. //ai = max(MB\*MCh-alfa\*MXh)
157. for i := H shl 1 to (H \* 3) - 1 do
158. for j := 0 to N - 1 do
159. begin
160. t := 0;
161. for k := 0 to N - 1 do
162. t := t + MC[i][k] \* MB3[k][j];
163. t := t - (alfa3\*MX[i][j]);
164. if t > ai then
165. ai := t;
166. end;
167. //a= max(a, ai)
168. EnterCriticalSection(CS1);
169. if ai > a then
170. a := ai;
171. LeaveCriticalSection(CS1);
172. //Сигнал про завершення обчислень
173. ReleaseSemaphore(S1, 1, nil);
174. end;
175. //=================Thread T4
176. procedure T4\_function();
177. var
178. alfa4, i, j, k, ai, t: integer;
179. MB4: array of array of integer;
180. begin
181. SetLength(MB4, N, N);
182. ai := -2147483648;
183. //Чекати введення МВ і альфа
184. WaitForSingleObject(E1, INFINITE);
185. //Копіювання МВ
186. WaitForSingleObject(M2, INFINITE);
187. for i := 0 to N - 1 do
188. for j := 0 to N - 1 do
189. MB4[i][j] := MB[i][j];
190. ReleaseMutex(M2);
191. //Копіювання альфа
192. WaitForSingleObject(M3, INFINITE);
193. alfa4 := alfa;
194. ReleaseMutex(M3);
195. //Чекати МC
196. WaitForSingleObject(E2, INFINITE);
197. //Чекати МX
198. WaitForSingleObject(E3, INFINITE);
199. //ai = max(MB\*MCh-alfa\*MXh)
200. for i := H\*3 to N - 1 do
201. for j := 0 to N - 1 do
202. begin
203. t := 0;
204. for k := 0 to N - 1 do
205. t := t + MC[i][k] \* MB4[k][j];
206. t := t - (alfa4\*MX[i][j]);
207. if t > ai then
208. ai := t;
209. end;
210. //a= max(a, ai)
211. EnterCriticalSection(CS1);
212. if ai > a then
213. a := ai;
214. LeaveCriticalSection(CS1);
215. //Сигнал про завершення обчислень
216. ReleaseSemaphore(S1, 1, nil);
217. end;
218. begin
219. write('Enter N: ');
220. readln(N);
221. H := N div 4;
222. SetLength(MB,N,N);
223. SetLength(MC,N,N);
224. SetLength(MX,N,N);
225. a := -2147483648;
226. M1 := CreateMutex(nil, false, 'M1');
227. M2 := CreateMutex(nil, false, 'M2');
228. M3 := CreateMutex(nil, false, 'M3');
229. InitializeCriticalSection(CS1);
230. E1 := CreateEvent(nil, true, false, 'E1');
231. E2 := CreateEvent(nil, true, false, 'E2');
232. E3 := CreateEvent(nil, true, false, 'E3');
233. S1 := CreateSemaphore(nil, 0, 3, 'S1');
234. T1 := CreateThread(nil, 0, @T1\_function, nil, 0, T1);
235. T2 := CreateThread(nil, 0, @T2\_function, nil, 0, T2);
236. T3 := CreateThread(nil, 0, @T3\_function, nil, 0, T3);
237. T4 := CreateThread(nil, 0, @T4\_function, nil, 0, T4);
238. WaitForSingleObject(T1, INFINITE);
239. readln(t);
240. end.