Міністерство освіти і науки України

Національний технічний університет України

«Київський політехнічний інститут ім. Ігоря Сікорського»

Факультет інформатики та обчислювальної техніки

Кафедра обчислювальної техніки

**Лабораторна робота №1**

«C#. Семафори, мютекси, події, критичні секції, атомік змінні»

ВИКОНАВ:

студент ІІІ курсу ФІОТ

групи

ПЕРЕВІРИВ:

Корочкін О. В.

Київ 2021

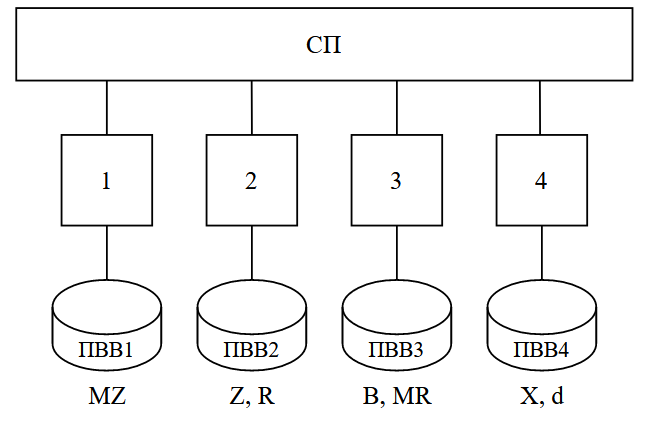
**Завдання**

Мова програмування: С#

Засоби організації взаємодії потоків: семафори, мютекси, події, критичні секції, атомік змінні

Математична задача: Z= (B\*X)\*(d\*X + R\*(MZ\*MR))

Структура ПКС:



**Розробка програми**

Етап 1. Побудова паралельного алгоритму

1. ai = (BH\*XH), i=1..4

2. a = a+ai

3. ZH=a\*(d\*X + R\*(MZH\*MRH))

Спільні ресурси: a, a, d, X, R

Етап 2. Розробка алгоритмів

Точки синхронізації,

Задача Т1 критичні ділянки

1. Сигнал Т2, T3, T4 про введення даних. S2,3,4-1

2. Очікування Т2, T3, T4 введення даних. W2,3,4-1

3. Копіювати а=а1, d=d1, X=X1, R=R1. КД

4. Обчислення a1 = (BH\*XH).

5. Обчислення a=a+a1.

6. Обчислення ZH=a1\*(d1\*X1 + R1\*(MZH\*MRH)).

7. Сигнал Т2 про завершення обчислень. S2-2

Задача Т2

1. Сигнал Т1, T3, T4 про введення даних. S1,3,4-1

2. Очікування Т1, T3, T4 введення даних. W1,3,4-1

3. Копіювати а=а2, d=d2, X=X2, R=R2. КД

4. Обчислення a2 = (BH\*XH).

5. Обчислення a=a+a2.

6. Обчислення ZH=a2\*(d2\*X2 + R2\*(MZH\*MRH)).

7. Чекати завершення обчислень Т1, Т3, Т4. W1,3,4-2

8. Вивід Z.

Задача Т3

1. Сигнал Т1, T2, T4 про введення даних. S1,2,4-1

2. Очікування Т1, T2, T4 введення даних. W1,2,4-1

3. Копіювати а=а3, d=d3, X=X3, R=R3. КД

4. Обчислення a3 = (BH\*XH).

5. Обчислення a=a+a3.

6. Обчислення ZH=a3\*(d3\*X3 + R3\*(MZH\*MRH)).

7. Сигнал Т2 про завершення обчислень. S2-2

Задача Т4

1. Сигнал Т1, T2, T3 про введення даних. S1,2,3-1

2. Очікування Т1, T2, T3 введення даних. W1,2,3-1

3. Копіювати а=а4, d=d4, X=X4, R=R4. КД

4. Обчислення a4 = (BH\*XH).

5. Обчислення a=a+a4.

6. Обчислення ZH=a4\*(d4\*X4 + R4\*(MZH\*MRH)).

7. Сигнал Т2 про завершення обчислень. S2-2

Етап 3. Розробка структурної схеми взаємодії потоків

E0 - подія для синхронізації із завершення вводу в Т1.

E1 - подія для синхронізації із завершення вводу в Т2.

E2 - подія для синхронізації із завершення вводу в Т3.

E3 - подія для синхронізації із завершення вводу в Т4.

S1 - семафор для синхронізаіції обчилень ZH у задачі Т1.

S2 - семафор для синхронізаіції обчилень ZH у задачі Т3.

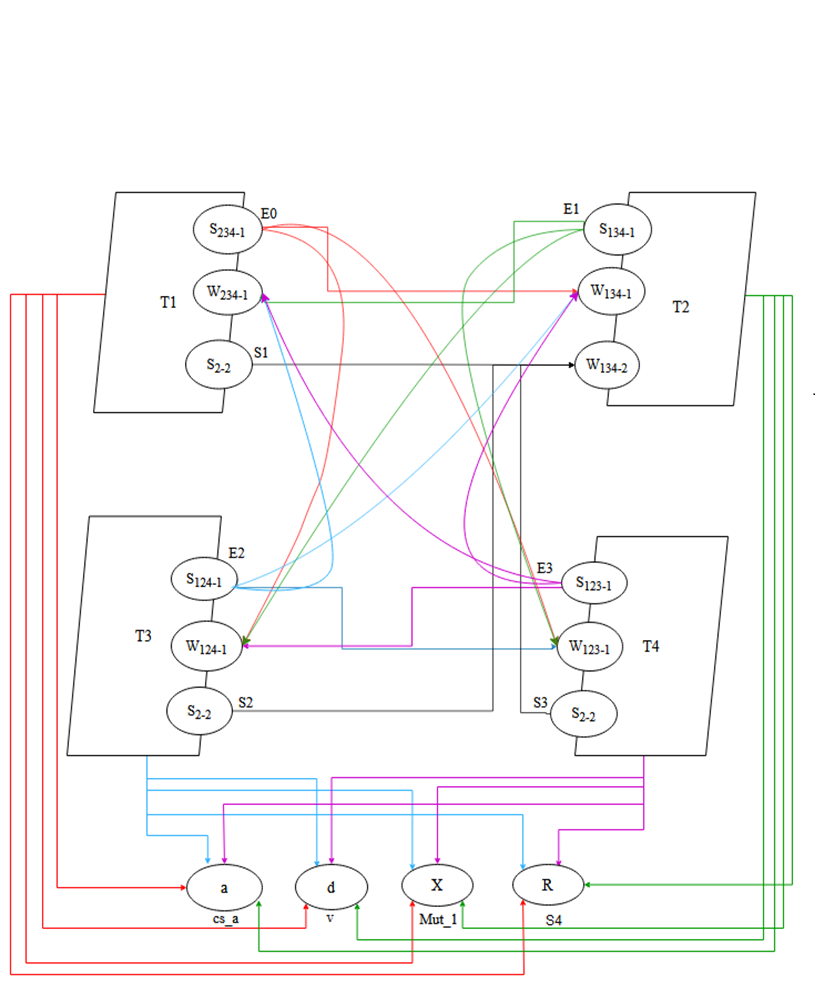
S3 - семафор для синхронізаіції обчилень ZH у задачі Т4.

cs\_a - замок для доступу до спільного ресурсу a.

volatile - для доступу до спільного ресурсу d.

S4 - семафор для керування доступом до спільного ресурсу R.

Mut\_1 - мютекс для керування доступом до спільного ресурсу X.

Етап 4. Лістинг програми

1. /\* -- Lab #1. C#

2. \* -- Z = (B\*X)\*(d\*X + R\*(MZ\*MR))

3. \* --

4. \* -- Date

5. \*/

6. using System;

7. using System.Threading;

8. namespace lab1

9. {

10. public class Lab1

11. {

12. public const int N = 8;

13. public const int P = 4;

14. public const int H = N/P;

15. public static volatile int d = 1;

16. public static int a = 0;

17. public static object cs\_a = new object();

18. public static Mutex M1 = new Mutex();

19. public static int[] B = new int[N];

20. public static int[] X = new int[N];

21. public static int[] R = new int[N];

22. public static int[,] MZ = new int[N, N];

23. public static int[,] MR = new int[N, N];

24. public static int[] Z = new int[N];

25. public static Semaphore S0;

26. public static Semaphore S1;

27. public static Semaphore S2;

28. public static Semaphore S3;

29. public static Semaphore S4;

30. public static EventWaitHandle E0;

31. public static EventWaitHandle E1;

32. public static EventWaitHandle E2;

33. public static EventWaitHandle E3;

34. /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Задача Т1 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

35. public static void T1()

36. {

37. Console.WriteLine("T1 started.");

38. // Введення MZ

39. FillMatrix(MZ);

40. // Сигнал задачам Т2, Т3, Т4 про завершення вводу MZ.

41. E0.Set();

42. // Чекати на завершення вводу в задачі Т2.

43. E1.WaitOne();

44. // Чекати на завершення вводу в задачі Т3.

45. E2.WaitOne();

46. // Чекати на завершення вводу в задачі Т4.

47. E3.WaitOne();

48. // копіювання а, d, X, R

49. int a1;

50. lock (cs\_a){

51. a1 = a;

52. }

53. int d1 = d;

54. int[] X1 = new int[N];

55. M1.WaitOne();

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

57. X1[i] = X[i];

58. }

59. M1.ReleaseMutex();

60. int[] R1 = new int[N];

61. S4.WaitOne();

62. for (int i = 0; i < N; i++)

63. {

64. R1[i] = R[i];

65. }

66. S4.Release();

77. // обчислення а1

68. for (int i = 0; i < H; i++){

69. a1 += B[i] \* X[i];

70. }

71. a = a + a1;

72. lock (cs\_a){

73. a1 = a;

74. }

75. // ZH=a1\*(d1\*X1 + R1\*(MZH\*MRH)).

76. int[] F1 = new int[H];

77. int[] Dx1 = new int[H];

78. for (int i = 0; i < H; i++)

79. {

80. F1[i] = 0;

81. Dx1[i] = 0;

82. for (int j = 0; j < N; j++)

83. {

84. Dx1[i] += d1 \* X1[i];

85. F1[i] += R1[i]\*(MZ[i, j]\*MR[i, j]);

86. F1[i] += Dx1[i];

87. }

88. }

89.

90.

91. for (int i = 0; i < H; i++)

92. {

93. Z[i] = a1 \* F1[i];

94.

95. }

96. // сигнал про завершення обчислень.

97. S1.Release();

98. Console.WriteLine("T1 Finished.");

99. }

100. /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Задача Т2 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

101. public static void T2()

102. {

103. Console.WriteLine("T2 started.");

104. // Введення R

105. FillVector(R);

106. // Сигнал задачам Т1, Т3, Т4 про завершення вводу R.

107. E1.Set();

108. // Чекати на завершення вводу в задачі Т1.

109. E0.WaitOne();

110. // Чекати на завершення вводу в задачі Т3.

111. E2.WaitOne();

112. // Чекати на завершення вводу в задачі Т4.

113. E3.WaitOne();

114. // копіювання а, d, X, R

115. int a2;

116. lock (cs\_a){

117. a2 = a;

118. }

119. int d2 = d;

120. int[] X2 = new int[N];

121. M1.WaitOne();

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

123. X2[i] = X[i];

124. }

125. M1.ReleaseMutex();

126. int[] R2 = new int[N];

127. S4.WaitOne();

128. for (var i = 0; i < N; i++)

129. {

130. R2[i] = R[i];

131.

132. }

133.

134.

135. S4.Release();

136. // обчислення а2

137. for (var i = 0; i < 2\*H; i++){

138. a2 += B[i] \* X[i];

139. }

140. a += a2;

141. lock (cs\_a){

142. a2 = a;

143. }

144. // ZH=a2\*(d2\*X2 + R2\*(MZH\*MRH)).

145. int[] F2 = new int[H];

146. int[] Dx2 = new int[H];

147. for (var i = 0; i < 2\*H; i++)

148. {

149. F2[i] = 0;

150. Dx2[i] = 0;

151. for (var j = 0; j < N; j++)

152. {

153. Dx2[i] += d2 \* X2[i];

154. F2[i] += R2[i]\*(MZ[i, j]\*MR[i, j]);

155. F2[i] += Dx2[i];

156. }

157. }

158. for (var i = 0; i < 2\*H; i++)

159. {

160. Z[i] = a2 \* F2[i];

161. }

162. // очікування сигналів від Т1, Т3, Т4 про завершення обчислень.

163. S1.WaitOne();

164. S2.WaitOne();

165. S3.WaitOne();

166. OutputVector(Z);

167. Console.WriteLine("T2 Finished.");

168. }

169.

170. /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Задача Т3 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

171. public static void T3()

172. {

173. Console.WriteLine("T3 started.");

174. // Введення B, MR

175. FillVector(B);

176. FillMatrix(MR);

177. // Сигнал задачам Т1, Т2, Т4 про завершення вводу B, MR.

178. E2.Set();

179. // Чекати на завершення вводу в задачі Т1.

180. E0.WaitOne();

181. // Чекати на завершення вводу в задачі Т2.

182. E1.WaitOne();

183. // Чекати на завершення вводу в задачі Т4.

184. E3.WaitOne();

185. // копіювання а, d, X, R

186. int a3;

187. lock (cs\_a)

188. {

189. a3 = a;

190. }

191. int d3 = d;

192. int[] X3 = new int[N];

193. M1.WaitOne();

194. for (int i = 0; i < N; i++)

195. {

196. X3[i] = X[i];

197. }

198. M1.ReleaseMutex();

199. int[] R3 = new int[N];

200. S4.WaitOne();

201. for (int i = 0; i < N; i++)

202. {

203.

204. R3[i] = R[i];

205.

206. }

207. S4.Release();

208. // обчислення а3

209. for (int i = 0; i < 3\*H; i++){

210. a3 += B[i] \* X[i];

211. }

212. a = a + a3;

213. lock (cs\_a){

214. a3 = a;

215. }

216. // ZH=a3\*(d3\*X3 + R3\*(MZH\*MRH)).

217. int[] F3 = new int[H];

218. int[] Dx3 = new int[H];

219. for (var i = 0; i < 3\*H; i++)

220. {

221. F3[i] = 0;

222. Dx3[i] = 0;

223. for (var j = 0; j < N; j++)

224. {

225. Dx3[i] += d3 \* X3[i];

226. F3[i] += R3[i]\*(MZ[i, j]\*MR[i, j]);

227. F3[i] += Dx3[i];

228. }

229. }

230. for (var i = 0; i < 3\*H; i++)

231. {

232. Z[i] = a3 \* F3[i];

233. }

234. // сигнал Т2 про завершення обчислень.

235. S2.Release();

236. Console.WriteLine("T3 Finished.");

237. }

238. /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Задача Т4 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

239. public static void T4()

240. {

241. Console.WriteLine("T4 started.");

242. // Введення X, d

243. FillVector(X);

244. d = 1;

245. // Сигнал задачам Т1, Т2, Т3 про завершення вводу X.

246. E3.Set();

247. // Чекати на завершення вводу в задачі Т1.

248. E0.WaitOne();

249. // Чекати на завершення вводу в задачі Т2.

250. E1.WaitOne();

251. // Чекати на завершення вводу в задачі Т3.

252. E2.WaitOne();

253. // копіювання а, d, X, R

254. int a4;

255. lock (cs\_a){

256. a4 = a;

257. }

258. var d4 = d;

259. var X4 = new int[N];

260. M1.WaitOne();

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

262. X4[i] = X[i];

263. }

264. M1.ReleaseMutex();

265. int[] R4 = new int[N];

266. S4.WaitOne();

267. for (var i = 0; i < N; i++)

268. {

269.

270. R4[i] = R[i];

271.

272. }

273.

274. S4.Release();

275. // обчислення а4

276. for (int i = 0; i < 4\*H; i++){

277. a4 += B[i] \* X[i];

278. }

279. a += a4;

280. lock (cs\_a){

281. a4 = a;

282. }

283. // ZH=a4\*(d4\*X4 + R4\*(MZH\*MRH)).

284. int[] F4 = new int[H];

285. int[] Dx4 = new int[H];

286. for (var i = 0; i < 4\*H; i++)

287. {

288. F4[i] = 0;

289. Dx4[i] = 0;

290. for (var j = 0; j < N; j++)

291. {

292. Dx4[i] += d4 \* X4[i];

293. F4[i] += R4[i]\*(MZ[i, j]\*MR[i, j]);

294. F4[i] += Dx4[i];

295. }

296. }

297. for (var i = 0; i < 4\*H; i++)

298. {

299. Z[i] = a4 \* F4[i];

300. }

301. // сигнал про завершення обчислень.

302. S3.Release();

303. Console.WriteLine("T4 Finished.");

304. }

305. static void FillMatrix(int[,] matrix){

306. for (var i = 0; i < N; i++){

307. for(var j = 0; j < N; j++){

308. matrix[i, j] = 1;

309. }

310. }

311. }

312. static void FillVector(int[] vector){

313. for (var i = 0; i < N; i++){

314. vector[i] = 1;

315. }

316. }

317. static void OutputVector(int[] vector){

318. for (var i = 0; i < N; i++){

319. Console.Write($"{vector[i]}");

320. }

321. }

322. public static void Main(string[] args)

323. {

324. Console.WriteLine("Lab1 started.");

325. S1 = new Semaphore(0, 1);

326. S2 = new Semaphore(0, 1);

327. S3 = new Semaphore(0, 1);

328. S4 = new Semaphore(0, 1);

329. E0 = new ManualResetEvent(false);

330. E1 = new ManualResetEvent(false);

331. E2 = new ManualResetEvent(false);

332. E3 = new ManualResetEvent(false);

333. Thread t1 = new Thread(T1);

334. Thread t2 = new Thread(T2);

335. Thread t3 = new Thread(T3);

336. Thread t4 = new Thread(T4);

337. t1.Start();

338. t2.Start();

339. t3.Start();

340. t4.Start();

341. t4.Join();

342. Console.WriteLine("Lab1 finished.");

343. }

344. }

345. }