**Kpi-best**

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ**

**“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ”**

**ФАКУЛЬТЕТ ІНФОРМАТИКИ І ОБЧИСЛЮВАЛЬНОЇ ТЕХНІКИ**

**КАФЕДРА ОБЧИСЛЮВАЛЬНОЇ ТЕХНІКИ**

**Паралельне програмування**

**Лабораторна робота №7**

Ada. Rendezvous

Виконав:

студент 3-го курсу

групи ІО-01

Куріло С.А.

Прийняв:

Корочкін О.В.

Київ – 2013 р.

Тема: Програмування для комп’ютерних систем із локальною пам’яттю. Ада. Рандеву

Технічне завдання

Розробити програму для розв’язання в ПКС з ЛП (структура на рис. Рисунок 1) математичної задачі: .

Мова програмування: Ада.

Механізм організації взаємодії: рандеву.



Рисунок 1 – Структура ПКС з ЛП

Виконання роботи

Етап 1. Побудова паралельного алгоритму



Етап 2. Розроблення алгоритмів роботи кожного процесу

Задача T1

1. **Прийняти**  від задачі T2
2. **Передати**  задачі T6
3. **Прийняти**  від задачі T6
4. Обчислення 
5. **Прийняти** результат  від задачі T6
6. **Передати** результат  задачі T2

Задача T2

1. **Введення** 
2. **Передати**  задачі T1
3. **Передати**  задачі T3
4. **Прийняти**  від задачі T3
5. Обчислення 
6. **Прийняти** результат  від задачі T1
7. **Прийняти** результат  від задачі T3
8. **Виведення** результату 

Задача T3

1. **Прийняти**  від задачі T2
2. **Передати**  задачі T4
3. **Прийняти**  від задачі T4
4. **Передати**  задачі T2
5. Обчислення 
6. **Прийняти** результат  від задачі T4
7. **Передати** результат  задачі T2

Задача T4

1. **Введення** 
2. **Прийняти**  від задачі T3
3. **Передати**  задачі T3
4. **Передати**  задачі T5
5. Обчислення 
6. **Передати** результат  задачі T3

Задача T5

1. **Прийняти**  від задачі T6
2. **Прийняти**  від задачі T4
3. **Передати**  задачі T6
4. Обчислення 
5. **Передати** результат  задачі T6

Задача T6

1. **Прийняти**  від задачі T1
2. **Передати**  задачі T5
3. **Прийняти**  від задачі T5
4. **Передати**  задачі T1
5. Обчислення 
6. **Прийняти** результат  від задачі T6
7. **Передати** результат  задачі T1

Етап 3. Розроблення структурної схеми взаємодії задач

Структурна схема взаємодії задач зображена на рис. Рисунок 2.



Рисунок 2 – Структурна схема взаємодії задач

Етап 4. Розроблення програми

1. with Ada.Text\_IO, Ada.Integer\_Text\_IO, Ada.Synchronous\_Task\_Control, Ada.Command\_Line;

2. use Ada.Text\_IO, Ada.Integer\_Text\_IO, Ada.Synchronous\_Task\_Control, Ada.Command\_Line;

3.

4. procedure Main is

5.

6. N: integer := 6; --Size of vectors and matrixes

7. P: integer := 6; --Number of tasks

8. H: integer := N/P;

9.

10. type Vector\_General is array(integer range <>) of Integer;

11. subtype Vector is Vector\_General(1..N);

12.

13. type Matrix\_General is array(integer range <>) of Vector;

14. subtype Matrix is Matrix\_General(1..N);

15.

16. procedure print(MA: in Matrix) is

17. begin

18. put\_line("[");

19. for i in MA'Range loop

20.

21. put(" [");

22. for j in MA'Range loop

23.

24. put(MA(i)(j), Width => 10);

25.

26. if j /= MA'Last then

27. put(", ");

28. end if;

29. end loop;

30. put("]");

31.

32. if i /= MA'Last then

33. put\_line(", ");

34. else

35. new\_line;

36. end if;

37. end loop;

38. put\_line("]");

39. end print;

40.

41. procedure print(A: in Vector) is

42. begin

43. put("[");

44. for i in A'Range loop

45. put(A(i), Width => 10);

46.

47. if i /= A'Last then

48. put(", ");

49. end if;

50. end loop;

51. put\_line("]");

52. end print;

53.

54. protected Operation is

55. procedure output(MA: Matrix);

56. procedure output(V: Vector);

57. end Operation;

58.

59. protected body Operation is

60. procedure output(MA: Matrix) is

61. begin

62. print(MA);

63. end output;

64.

65. procedure output(V: Vector) is

66. begin

67. print(V);

68. end output;

69. end Operation;

70.

71. procedure calculation(left, right: Integer; A: out Vector; B: out Vector; MB1, MC1, MO1, MT1, ME1: in Matrix; MA1: out Matrix) is

72. begin

73. for i in left..right loop

74. for j in 1..N loop

75. A(j) := 0;

76. B(j) := 0;

77. for k in 1..N loop

78. A(j) := A(j) + MB1(k)(j) \* MC1(i)(k);

79. B(j) := B(j) + MO1(k)(j) \* MT1(i)(k);

80. end loop;

81. MA1(i)(j) := A(j)+B(j)+ME1(i)(j);

82. end loop;

83. end loop;

84. end calculation;

85.

86. --Task specification T1-----------------------------------------------

87. task T1 is

88. entry Send\_fromT2(MB: in Matrix; MO: in Matrix; MEh: in Matrix\_General; ME2h: in Matrix\_General);

89. entry Send\_fromT6(MCh: in Matrix\_General; MTh: in Matrix\_General);

90. entry Send\_MAT6(MA2h: in Matrix\_General);

91. end T1;

92. ----------------------------------------------------------------------

93.

94. --Task specification T2-----------------------------------------------

95. task T2 is

96. entry Send\_fromT3(MCh: in Matrix\_General; MTh: in Matrix\_General);

97. entry Send\_MAT1(MAh: in Matrix\_General; MA2h: in Matrix\_General);

98. entry Send\_MAT3(MA2h: in Matrix\_General);

99. end T2;

100. ----------------------------------------------------------------------

101.

102. --Task specification T3-----------------------------------------------

103. task T3 is

104. entry Send\_fromT2(MB: in Matrix; MO: in Matrix; ME2h: in Matrix\_General);

105. entry Send\_fromT4(MC2h: in Matrix\_General; MT2h: in Matrix\_General);

106. entry Send\_MAT4(MAh: in Matrix\_General);

107. end T3;

108. ----------------------------------------------------------------------

109.

110. --Task specification T4-----------------------------------------------

111. task T4 is

112. entry Send\_fromT3(MB: in Matrix; MO: in Matrix; MEh: in Matrix\_General);

113. end T4;

114. ----------------------------------------------------------------------

115.

116. --Task specification T5-----------------------------------------------

117. task T5 is

118. entry Send\_fromT4(MCh: in Matrix\_General; MC2h: in Matrix\_General; MTh: in Matrix\_General; MT2h: in Matrix\_General);

119. entry Send\_fromT6(MB: in Matrix; MO: in Matrix; MEh: in Matrix\_General);

120. end T5;

121. ----------------------------------------------------------------------

122.

123. --Task specification T6-----------------------------------------------

124. task T6 is

125. entry Send\_fromT5(MC1h: in Matrix\_General; MCh: in Matrix\_General; MT1h: in Matrix\_General; MTh: in Matrix\_General);

126. entry Send\_fromT1(MB: in Matrix; MO: in Matrix; ME2h: in Matrix\_General);

127. entry Send\_MAT5(MAh: in Matrix\_General);

128. end T6;

129. ----------------------------------------------------------------------

130.

131. --Task body T1--------------------------------------------------------

132. task body T1 is

133. MA1, MB1, MO1: Matrix;

134. MC1, MT1, ME1: Matrix;

135. A, B: Vector;

136. begin

137. put\_line("Task T1 has started.");

138.

139. -- 1. Receive MB, MO, ME3h from T2

140. accept Send\_fromT2(MB: in Matrix; MO: in Matrix; MEh: in Matrix\_General; ME2h: in Matrix\_General) do

141. MB1 := MB;

142. MO1 := MO;

143. ME1(MEh'Range) := MEh;

144. ME1(ME2h'Range) := ME2h;

145. end Send\_fromT2;

146.

147. -- 2. Send MB, MO, ME2h to T6

148. T6.Send\_fromT1(MB1, MO1, ME1(4\*H+1..N));

149.

150. -- 3. Receive MCh, MTh from T6

151. accept Send\_fromT6(MCh: in Matrix\_General; MTh: in Matrix\_General) do

152. MC1(MCh'Range) := MCh;

153. MT1(MTh'Range) := MTh;

154. end Send\_fromT6;

155.

156. -- 4. Calculate

157. calculation(1, H, A => A,

158. B => B,

159. MB1 => MB1,

160. MC1 => MC1,

161. MO1 => MO1,

162. MT1 => MT1,

163. ME1 => ME1,

164. MA1 => MA1);

165.

166. -- 5. Receive MA2h from T6

167. accept Send\_MAT6(MA2h: in Matrix\_General) do

168. MA1(MA2h'Range) := MA2h;

169. end Send\_MAT6;

170.

171. -- 6. Send MA3h to T2

172. T2.Send\_MAT1(MA1(1..H), MA1(4\*H+1..N));

173.

174. put\_line("Task T1 has finished.");

175. end T1;

176. ----------------------------------------------------------------------

177.

178. --Task body T2--------------------------------------------------------

179. task body T2 is

180. MA, MB, MO: Matrix;

181. MC, MT, ME: Matrix;

182. A, B: Vector;

183.

184. begin

185. put\_line("Task T2 has started.");

186.

187. -- 1. Input MB, MO, ME

188. for i in 1..N loop

189. for j in 1..N loop

190. MB(i)(j) := 1;

191. MO(i)(j) := 1;

192. ME(i)(j) := 1;

193. end loop;

194. end loop;

195.

196. -- 2. Send MB, MO, ME3h to T1

197. T1.Send\_fromT2(MB, MO, ME(1..H), ME(4\*H+1..N));

198.

199. -- 3. Send MB, MO, ME2h to T3

200. T3.Send\_fromT2(MB, MO, ME(2\*H+1..4\*H));

201.

202. -- 4. Receive MCh, MTh from T3

203. accept Send\_fromT3(MCh: in Matrix\_General; MTh: in Matrix\_General) do

204. MC(MCh'Range) := MCh;

205. MT(MTh'Range) := MTh;

206. end Send\_fromT3;

207.

208. -- 5. Calculate

209. calculation(H+1, 2\*H, A => A,

210. B => B,

211. MB1 => MB,

212. MC1 => MC,

213. MO1 => MO,

214. MT1 => MT,

215. ME1 => ME,

216. MA1 => MA);

217.

218. -- 6. Receive MA3h from T1

219. accept Send\_MAT1(MAh: in Matrix\_General; MA2h: in Matrix\_General) do

220. MA(MAh'Range) := MAh;

221. MA(MA2h'Range) := MA2h;

222. end Send\_MAT1;

223.

224. -- 7. Receive MA2h from T3

225. accept Send\_MAT3(MA2h: in Matrix\_General) do

226. MA(MA2h'Range) := MA2h;

227. end Send\_MAT3;

228.

229. -- 8. Output MA

230. Operation.Output(MA);

231.

232. put\_line("Task T2 has finished.");

233. end T2;

234. ----------------------------------------------------------------------

235.

236. --Task body T3--------------------------------------------------------

237. task body T3 is

238. MA3, MB3, MO3: Matrix;

239. MC3, MT3, ME3: Matrix;

240. A, B: Vector;

241. begin

242. put\_line("Task T3 has started.");

243.

244. -- 1. Receive MB, MO, ME2h from T2

245. accept Send\_fromT2(MB: in Matrix; MO: in Matrix; ME2h: in Matrix\_General) do

246. MB3 := MB;

247. MO3 := MO;

248. ME3(ME2h'Range) := ME2h;

249. end Send\_fromT2;

250.

251. -- 2. Send MB, MO, MEh to T4

252. T4.Send\_fromT3(MB3, MO3, ME3(3\*H+1..4\*H));

253.

254. -- 3. Receive MC2h, MT2h from T5

255. accept Send\_fromT4(MC2h: in Matrix\_General; MT2h: in Matrix\_General) do

256. MC3(MC2h'Range) := MC2h;

257. MT3(MT2h'Range) := MT2h;

258. end Send\_fromT4;

259.

260. -- 4. Send MCh, MTh to T2

261. T2.Send\_fromT3(MC3(H+1..2\*H), MT3(H+1..2\*H));

262.

263. -- 5. Calculate

264. calculation(2\*H+1, 3\*H, A => A,

265. B => B,

266. MB1 => MB3,

267. MC1 => MC3,

268. MO1 => MO3,

269. MT1 => MT3,

270. ME1 => ME3,

271. MA1 => MA3);

272.

273. -- 6. Receive MAh from T4

274. accept Send\_MAT4(MAh: in Matrix\_General) do

275. MA3(MAh'Range) := MAh;

276. end Send\_MAT4;

277.

278. -- 7. Send MA2h to T2

279. T2.Send\_MAT3(MA3(2\*H+1..4\*H));

280.

281. put\_line("Task T3 has finished.");

282. end T3;

283. ----------------------------------------------------------------------

284.

285. --Task body T4--------------------------------------------------------

286. task body T4 is

287. MA4, MB4, MO4: Matrix;

288. MC, MT, ME4: Matrix;

289. A, B: Vector;

290. begin

291. put\_line("Task T4 has started.");

292.

293. -- 1. Input MC, MT

294. for i in 1..N loop

295. for j in 1..N loop

296. MT(i)(j) := 1;

297. MC(i)(j) := 1;

298. end loop;

299. end loop;

300.

301. -- 2. Receive MB, MO, MEh from T3

302. accept Send\_fromT3(MB : Matrix; MO : Matrix; MEh : Matrix\_General) do

303. MB4 := MB;

304. MO4 := MO;

305. ME4(MEh'Range) := MEh;

306. end Send\_fromT3;

307.

308. -- 3. Send MC2h, MT2h to T3

309. T3.Send\_fromT4(MC(H+1..3\*H), MT(H+1..3\*H));

310.

311. -- 4. Send MC3h, MT3h to T5

312. T5.Send\_fromT4(MC(1..H), MC(4\*H+1..N), MT(1..H), MT(4\*H+1..N));

313.

314. -- 5. Calculate

315. calculation(3\*H+1, 4\*H, A => A,

316. B => B,

317. MB1 => MB4,

318. MC1 => MC,

319. MO1 => MO4,

320. MT1 => MT,

321. ME1 => ME4,

322. MA1 => MA4);

323.

324. -- 6. Send MAh to T3

325. T3.Send\_MAT4(MA4(3\*H+1..4\*H));

326.

327. put\_line("Task T4 has finished.");

328. end T4;

329. ----------------------------------------------------------------------

330.

331. --Task body T5--------------------------------------------------------

332. task body T5 is

333. MA5, MB5, MO5: Matrix;

334. MC5, MT5, ME5: Matrix;

335. A, B: Vector;

336. begin

337. put\_line("Task T5 has started.");

338.

339. -- 1. Receive MB, MO, MEh from T6

340. accept Send\_fromT6(MB : Matrix; MO : Matrix; MEh : Matrix\_General) do

341. MB5 := MB;

342. MO5 := MO;

343. ME5(MEh'Range) := MEh;

344. end Send\_fromT6;

345.

346. -- 2. Receive MC3h, MT3h from T4

347. accept Send\_fromT4(MCh : Matrix\_General; MC2h : Matrix\_General; MTh : Matrix\_General; MT2h : Matrix\_General) do

348. MC5(MCh'Range) := MCh;

349. MC5(MC2h'Range) := MC2h;

350. MT5(MTh'Range) := MTh;

351. MT5(MT2h'Range) := MT2h;

352. end Send\_fromT4;

353.

354. -- 3. Send MC2h, MT2h to T6

355. T6.Send\_fromT5(MC5(1..H), MC5(5\*H+1..N), MT5(1..H), MT5(5\*H+1..N));

356.

357. -- 4. Calculate

358. calculation(4\*H+1, 5\*H, A => A,

359. B => B,

360. MB1 => MB5,

361. MC1 => MC5,

362. MO1 => MO5,

363. MT1 => MT5,

364. ME1 => ME5,

365. MA1 => MA5);

366.

367. -- 5. Send MAh to T6

368. T6.Send\_MAT5(MA5(4\*H+1..5\*H));

369.

370. put\_line("Task T5 has finished.");

371. end T5;

372. ----------------------------------------------------------------------

373.

374. --Task body T6--------------------------------------------------------

375. task body T6 is

376. MA6, MB6, MO6: Matrix;

377. MC6, MT6, ME6: Matrix;

378. A, B: Vector;

379. begin

380. put\_line("Task T6 has started.");

381.

382. -- 1. Receive MB, MO, ME2h from T1

383. accept Send\_fromT1(MB : Matrix; MO : Matrix; ME2h : Matrix\_General) do

384. MB6 := MB;

385. MO6 := MO;

386. ME6(ME2h'Range) := ME2h;

387. end Send\_fromT1;

388.

389. -- 2. Send MB, MO, MEh to T5

390. T5.Send\_fromT6(MB6, MO6, ME6(4\*H+1..5\*H));

391.

392. -- 3. Receive MC2h, MT2h from T5

393. accept Send\_fromT5(MC1h : Matrix\_General; MCh: Matrix\_General; MT1h : Matrix\_General; MTh : Matrix\_General) do

394. MC6(MC1h'Range) :=MC1h;

395. MC6(MCh'Range) := MCh;

396. MT6(MT1h'Range) := MT1h;

397. MT6(MTh'Range) := MTh;

398. end Send\_fromT5;

399.

400. -- 4. Send MCh, MTh to T1

401. T1.Send\_fromT6(MC6(1..H), MT6(1..H));

402.

403. -- 5. Calculate

404. calculation(5\*H+1, N, A => A,

405. B => B,

406. MB1 => MB6,

407. MC1 => MC6,

408. MO1 => MO6,

409. MT1 => MT6,

410. ME1 => ME6,

411. MA1 => MA6);

412.

413. -- 6. Receive MAh from T5

414. accept Send\_MAT5(MAh: in Matrix\_General) do

415. MA6(MAh'Range) := MAh;

416. end Send\_MAT5;

417.

418. -- 7. Send MA2h to T1

419. T1.Send\_MAT6(MA6(4\*H+1..N));

420.

421. put\_line("Task T6 has finished.");

422. end T6;

423. ----------------------------------------------------------------------

424.

425. begin

426. null;

427. end Main;