Национальный технический университет Украины

«Киевский политехнический институт»

Лабораторная работа №7

По курсу

Параллельные и распределенные вычисления

Тема:

Ада. Рандеву

Работу выполнил

Студент 3-го курса ФИВТ

Группы ИВ-73

Грубый Павел

Киев-2010

Техническое задание



A = B×(MX×MZ - a×ME)

Этап 1. Построение параллельного алгоритма

AH = B × ( MX×MZH - a×MEH)

Этап 2. Разработка алгоритмов работы каждого процесса

|  |
| --- |
| **Задача Т1** |
| 1. Ввод B, ME. |
| 1. Передать B и ME4H  задачам T2 и Т6. |
| 1. Принять от задачи Т6 данные a, MX. |
| 1. Принять от задачи Т2 данные MZH. |
| 1. Вычисление AH = B × ( MX×MZH - a×MEH) |
| 1. Передать в Т2 результат AH |

|  |
| --- |
| **Задача Т2** |
| 1. Принять от задачи Т1 данные B, ME4H |
| 1. Передать B и ME2H  задаче Т5. |
| 1. Передать B и MEH  задаче Т3. |
| 1. Принять от задачи Т5 данные MZ2H. |
| 1. Передать MZH  задаче Т1. |
| 1. Принять от задачи Т1 данные a, MX. |
| 1. Передать a, MX задаче Т3. |
| 1. Вычисление AH = B × ( MX×MZH - a×MEH) |
| 1. Принять от задачи Т1 результат AH. |
| 1. Передать в Т3 результат A2H |

|  |
| --- |
| **Задача Т3** |
| 1. Принять от задачи Т2 данные B, MEH |
| 1. Принять от задачи Т4 данные MZH. |
| 1. Принять от задачи Т2 данные a, MX. |
| 1. Вычисление AH = B × ( MX×MZH - a×MEH) |
| 1. Принять от задачи Т2 результат A2H |
| 1. Передать в Т4 результат AH |

|  |
| --- |
| **Задача Т4** |
| 1. Принять от задачи Т5 данные B, MEH |
| 1. Принять от задачи Т5 данные a, MX |
| 1. Принять от задачи Т9 данные MZ4H |
| 1. Передать MZ2H  задаче Т5 |
| 1. Передать MZH  задаче Т3 |
| 1. Вычисление AH = B × ( MX×MZH - a×MEH) |
| 1. Принять от задачи Т3 результат A3H |
| 1. Передать в Т9 результат A4H |

|  |
| --- |
| **Задача Т5** |
| 1. Принять от задачи Т2 данные ME2H |
| 1. Принять от задачи Т6 данные B, ME2H |
| 1. Передать B и ME2H  задаче T8. |
| 1. Передать B и MEH  задаче T4. |
| 1. Принять от задачи Т6 данные a, MX |
| 1. Передать a, MX задаче Т4. |
| 1. Принять от задачи Т8 данные MZ2H |
| 1. Принять от задачи Т4 данные MZ2H |
| 1. Передать MZH  задаче T6 |
| 1. Передать MZ2H  задаче T2 |
| 1. Вычисление AH = B × ( MX×MZH - a×MEH) |
| 1. Принять от задачи Т6 результат AH. |
| 1. Передать в Т8 результат A2H |

|  |
| --- |
| **Задача Т6** |
| 1. Принять от задачи Т1 данные B, ME4H |
| 1. Передать B и MEH  задаче T7 |
| 1. Передать B и ME2H  задаче T5 |
| 1. Принять от задачи Т7 данные a, MX. |
| 1. Передать a, MX задаче Т1 |
| 1. Передать a, MX задаче Т5 |
| 1. Принять от задачи Т5 данные MZH. |
| 1. Вычисление AH = B × ( MX×MZH - a×MEH) |
| 1. Передать в Т5 результат AH |

|  |
| --- |
| **Задача Т7** |
| 1. Ввод a, MX. |
| 1. Передать задачам Т6, Т8 данные a, MX. |
| 1. Принять от задачи Т6 данные B, MEH |
| 1. Принять от задачи Т8 данные MZH |
| 1. Вычисление AH = B × ( MX×MZH - a×MEH) |
| 1. Передать в Т8 результат AH |

|  |
| --- |
| **Задача Т8** |
| 1. Принять от задачи Т9 данные MZ4H |
| 1. Передать MZH  задаче T7 |
| 1. Передать MZ2H  задаче T5 |
| 1. Принять от задачи Т5 данные B, ME2H |
| 1. Передать B и MEH  задаче T9 |
| 1. Принять от задачи Т7 данные a, MX |
| 1. Передать a, MX задаче Т9. |
| 1. Вычисление AH = B × ( MX×MZH - a×MEH) |
| 1. Принять от задачи Т7 результат AH. |
| 1. Принять от задачи Т5 результат A2H. |
| 1. Передать в Т9 результат A4H |

|  |
| --- |
| **Задача Т9** |
| 1. Ввод ME |
| 1. Передать MZ4H  задаче T4. |
| 1. Передать MZ4H  задаче T8. |
| 1. Принять от задачи Т8 данные B, MEH |
| 1. Принять от задачи Т8 данные a, MX |
| 1. Вычисление AH = B × ( MX×MZH - a×MEH) |
| 1. Принять от задачи Т4 результат A4H. |
| 1. Принять от задачи Т8 результат A4H. |
| 1. Вывод А. |

Этап 3. Разработка структурной схемы взаимодействия задач



рис. 3.1. Схема взаимодействия задач.

Этап 4. Разработка программы

GNAT GPL 2009 (20090519)

Copyright 1992-2009, Free Software Foundation, Inc.

Compiling: D:\Kache\AdaWorkspace\AdaRondevouz\source\main.adb (source file time stamp: 2010-04-18 19:09:20)

1. ----------------------------------------------------------------

2. -- Paralel and distributed computing --

3. -- Laboratory work #7. Ada. Rondevouz --

4. -- Func: A = B\*(MX\*MZ - a\*ME) --

5. -- IO-73 Grubyi Pavel --

6. -- 17.03.2010 --

7. ----------------------------------------------------------------

8. with Ada.Text\_Io, Matrix\_Pack;

9. use Ada.Text\_Io, Matrix\_Pack;

10.

11. procedure main is

12. task T1 is

13. entry input2 (in\_a : in integer; in\_MX : in Matrix);

14. entry input3 (in\_MZ : in Matrix\_H);

15. end T1;

16. ---------------------------------------------------

17. task T2 is

18. entry input1 (in\_B : in Vector; in\_ME : in Matrix\_4H);

19. entry input2 (in\_a : in integer; in\_MX : in Matrix);

20. entry input3 (in\_MZ : in Matrix\_2H);

21. entry result (in\_A : in Vector\_H);

22. end T2;

23. ---------------------------------------------------

24. Task T3 is

25. entry input1 (in\_B : in Vector; in\_ME : in Matrix\_H);

26. entry input2 (in\_a : in integer; in\_MX : in Matrix);

27. entry input3 (in\_MZ : in Matrix\_H);

28. entry result (in\_A : in Vector\_2H);

29. end T3;

30. ---------------------------------------------------

31. task T4 is

32. entry input1 (in\_B : in Vector; in\_ME : in Matrix\_H);

33. entry input2 (in\_a : in integer; in\_MX : in Matrix);

34. entry input3 (in\_MZ : in Matrix\_4H);

35. entry result (in\_A : in Vector\_3H);

36. end T4;

37. ---------------------------------------------------

38. task T5 is

39. entry input1T2 (in\_ME : in Matrix\_2H);

40. entry input1T6 (in\_B : in Vector; in\_ME : in Matrix\_2H);

41. entry input2 (in\_a : in integer; in\_MX : in Matrix);

42. entry input3T4 (in\_MZ : in Matrix\_2H);

43. entry input3T8 (in\_MZ : in Matrix\_2H);

44. entry result (in\_A : in Vector\_H);

45. end T5;

46. ---------------------------------------------------

47. Task T6 is

48. entry input1 (in\_B : in Vector; in\_ME : in Matrix\_4H);

49. entry input2 (in\_a : in integer; in\_MX : in Matrix);

50. entry input3 (in\_MZ : in Matrix\_H);

51. end T6;

52. ---------------------------------------------------

53. task T7 is

54. entry input1 (in\_B : in Vector; in\_ME : in Matrix\_H);

55. entry input3 (in\_MZ : in Matrix\_H);

56. end T7;

57. ---------------------------------------------------

58. task T8 is

59. entry input1 (in\_B : in Vector; in\_ME : in Matrix\_2H);

60. entry input2 (in\_a : in integer; in\_MX : in Matrix);

61. entry input3 (in\_MZ : in Matrix\_4H);

62. entry resultT5 (in\_A : in Vector\_2H);

63. entry resultT7 (in\_A : in Vector\_H);

64. end T8;

65. ---------------------------------------------------

66. task T9 is

67. entry input1 (in\_B : in Vector; in\_ME : in Matrix\_H);

68. entry input2 (in\_a : in integer; in\_MX : in Matrix);

69. entry resultT4 (in\_A : in Vector\_4H);

70. entry resultT8 (in\_A : in Vector\_4H);

71. end T9;

72.

73. task body T1 is

74. a : integer;

75. B : Vector;

76. MZ : Matrix\_H;

77. MX, ME : Matrix;

78. Ah : Vector\_H;

79. begin

80. Put\_line ("T1 started");

81. B := getVector(1);

82. ME := getMatrix (1);

83. --ME (1) (2) := 10;

84. Put\_line ("T1 input finished");

85.

86. T2.input1 (B, ME (H + 1 .. 5 \* H));

87. Put\_line ("T1 data sent T2");

88. T6.input1 (B, ME (5 \* H + 1 .. 9 \* H));

89. Put\_line ("T1 data sent T6");

90.

91. accept input2 (in\_a : in integer; in\_MX : in Matrix) do

92. a := in\_a;

93. MX := in\_MX;

94. end input2;

95.

96. Put\_line ("T1 input2 accepted");

97.

98. T2.input2 (a, MX);

99. Put\_line ("T1 input2 T2");

100.

101. accept input3 (in\_MZ : in Matrix\_H) do

102. MZ := in\_MZ;

103. end input3;

104.

105. Put\_line ("T1 input3");

106.

107. mainCalc (Ah, a, B, MX, MZ, ME (1 .. H));

108. Put\_line ("T1 calc finished");

109. T2.result (Ah);

110. Put\_line ("Process T1 complete");

111. end T1;

112. ---------------------------------------------

113. task body T2 is

114. a : integer;

115. B : Vector;

116. MX : Matrix;

117. MZ : Matrix\_2H;

118. ME : Matrix\_4H;

119. A2h : Vector\_2H;

120. begin

121. Put\_line ("Process T2 started");

122.

123. accept input1 (in\_B : in Vector; in\_ME : in Matrix\_4H) do

124. B := in\_B;

125. ME := in\_ME;

126. end input1;

127.

128. T3.input1 (B, ME (H + 1 .. 2 \* H));

129. Put\_line ("T2 input1 T3");

130. T5.input1T2 (ME (2 \* H + 1 .. 4 \* H));

131. Put\_line ("T2 input1 T5");

132.

133. accept input2 (in\_a : in integer; in\_MX : in Matrix) do

134. a := in\_a;

135. MX := in\_MX;

136. end input2;

137.

138. T3.input2 (a, MX);

139. Put\_line ("T2 input2 T3");

140.

141. accept input3 (in\_MZ : in Matrix\_2H) do

142. MZ := in\_MZ;

143. end input3;

144.

145. T1.input3(MZ(1..H));

146.

147. mainCalc (A2h (H+1..2\*H), a, B, MX, MZ(H+1..2\*H), ME (1 .. H));

148.

149. accept result (in\_A : in Vector\_H) do

150. A2h (1..H) := in\_A;

151. end result;

152.

153. T3.result(A2h);

154. Put\_line ("Process T2 complete");

155. end T2;

156. ---------------------------------------------

157. task body T3 is

158. a : integer;

159. B : Vector;

160. MX : Matrix;

161. MZ : Matrix\_H;

162. ME : Matrix\_H;

163. A3h : Vector\_3H;

164. begin

165. Put\_line ("Process T3 started");

166.

167. accept input1 (in\_B : in Vector; in\_ME : in Matrix\_H) do

168. B := in\_B;

169. ME := in\_ME;

170. end input1;

171. Put\_line ("T3 input1 accepted");

172.

173. accept input2 (in\_a : in integer; in\_MX : in Matrix) do

174. a := in\_a;

175. MX := in\_MX;

176. end input2;

177. Put\_line ("T3 input2 accepted");

178.

179. accept input3 (in\_MZ : in Matrix\_H) do

180. MZ := in\_MZ;

181. end input3;

182. Put\_line ("T3 input3 accepted");

183.

184. mainCalc (A3h (2\*H + 1 .. 3 \* H), a, B, MX, MZ, ME);

185.

186. accept result (in\_A : in Vector\_2H) do

187. A3h (1 .. 2\*H) := in\_A;

188. end result;

189.

190. T4.result (A3h);

191. Put\_line ("Process T3 complete");

192. end T3;

193. ---------------------------------------------

194. task body T4 is

195. a : integer;

196. B : Vector;

197. MX : Matrix;

198. MZ : Matrix\_4H;

199. ME : Matrix\_H;

200. A4h : Vector\_4H;

201. begin

202. Put\_line ("Process T4 started");

203.

204. accept input1 (in\_B : in Vector; in\_ME : in Matrix\_H) do

205. B := in\_B;

206. ME := in\_ME;

207. end input1;

208.

209. accept input2 (in\_a : in integer; in\_MX : in Matrix) do

210. a := in\_a;

211. MX := in\_MX;

212. end input2;

213.

214. accept input3 (in\_MZ : in Matrix\_4H) do

215. MZ := in\_MZ;

216. end input3;

217.

218. T5.input3T4 (MZ (1 .. 2 \* H));

219. T3.input3 (MZ (2 \* H + 1 .. 3 \* H));

220.

221. mainCalc (A4h (3\*H + 1 .. 4 \* H), a, B, MX, MZ(3\*H+1..4\*H), ME);

222.

223. accept result (in\_A : in Vector\_3H) do

224. A4h (1 .. 3\*H) := in\_A;

225. end result;

226.

227. T9.resultT4 (A4h);

228. Put\_line ("Process T4 complete");

229. end T4;

230. ---------------------------------------------

231. task body T5 is

232. a : integer;

233. B : Vector;

234. MX : Matrix;

235. MZ : Matrix\_4H;

236. ME : Matrix\_4H;

237. A2h : Vector\_2H;

238. begin

239. Put\_line ("Process T5 started");

240.

241. accept input1T2 (in\_ME : in Matrix\_2H) do

242. ME(1..2\*H) := in\_ME;

243. end input1T2;

244.

245. accept input1T6 (in\_B : in Vector; in\_ME : in Matrix\_2H) do

246. B := in\_B;

247. ME(2\*H+1..4\*H) := in\_ME;

248. end input1T6;

249.

250. T4.input1 (B, ME (1 .. H));

251. Put\_line ("T5 input1 T4");

252. T8.input1 (B, ME (2\*H + 1 .. 4 \* H));

253. Put\_line ("T5 input1 T8");

254.

255. accept input2 (in\_a : in integer; in\_MX : in Matrix) do

256. a := in\_a;

257. MX := in\_MX;

258. end input2;

259.

260. T4.input2 (a, MX);

261.

262. accept input3T4 (in\_MZ : in Matrix\_2H) do

263. MZ(1..2\*H) := in\_MZ;

264. end input3T4;

265.

266. accept input3T8 (in\_MZ : in Matrix\_2H) do

267. MZ(2\*H+1..4\*H) := in\_MZ;

268. end input3T8;

269.

270. T6.input3 (MZ (3 \* H + 1 .. 4 \* H));

271. T2.input3 (MZ (1..2\*H));

272.

273. mainCalc (A2h (1 .. H), a, B, MX, MZ (2\*H + 1 .. 3 \* H), ME (H+1 .. 2\*H));

274.

275. accept result (in\_A : in Vector\_H) do

276. A2h (H + 1 .. 2 \* H) := in\_A;

277. end result;

278.

279. T8.resultT5 (A2h);

280. Put\_line ("Process T5 complete");

281. end T5;

282. ---------------------------------------------

283. task body T6 is

284. a : integer;

285. B : Vector;

286. MX : Matrix;

287. MZ : Matrix\_H;

288. ME : Matrix\_4H;

289. Ah : Vector\_H;

290. begin

291. Put\_line ("Process T6 started");

292.

293. accept input1 (in\_B : in Vector; in\_ME : in Matrix\_4H) do

294. B := in\_B;

295. ME := in\_ME;

296. end input1;

297.

298. T5.input1T6 (B, ME (2 \* H + 1 .. 4 \* H));

299. Put\_line ("T6 input1 T5");

300. T7.input1 (B, ME (H + 1 .. 2 \* H));

301. Put\_line ("T6 input1 T7");

302.

303. accept input2 (in\_a : in integer; in\_MX : in Matrix) do

304. a := in\_a;

305. MX := in\_MX;

306. end input2;

307.

308. Put\_line ("T6 input2 accepted");

309.

310. T1.input2 (a, MX);

311. Put\_line ("T6 input2 T1");

312. T5.input2 (a, MX);

313. Put\_line ("T6 input2 T5");

314.

315.

316. accept input3 (in\_MZ : in Matrix\_H) do

317. MZ := in\_MZ;

318. end input3;

319.

320. mainCalc (Ah, a, B, MX, MZ, ME (1 .. H));

321. T5.result (Ah);

322. Put\_line ("Process T6 complete");

323. end T6;

324. ---------------------------------------------

325. task body T7 is

326. a : integer;

327. B : Vector;

328. MX : Matrix;

329. MZ : Matrix\_H;

330. ME : Matrix\_H;

331. Ah : Vector\_H;

332. begin

333. Put\_line ("Process T7 started");

334. a := 1;

335. MX := getMatrix (1);

336.

337. accept input1 (in\_B : in Vector; in\_ME : in Matrix\_H) do

338. B := in\_B;

339. ME := in\_ME;

340. end input1;

341.

342. Put\_line ("T7 input 1 accepted");

343.

344. T6.input2 (a, MX);

345. Put\_line ("T7 input2 T6");

346. T8.input2 (a, MX);

347. Put\_line ("T7 input2 T8");

348.

349. accept input3 (in\_MZ : in Matrix\_H) do

350. MZ := in\_MZ;

351. end input3;

352.

353. mainCalc (Ah, a, B, MX, MZ, ME);

354. T8.resultT7 (Ah);

355. Put\_line ("Process T7 complete");

356. end T7;

357. ---------------------------------------------

358. task body T8 is

359. a : integer;

360. B : Vector;

361. MX : Matrix;

362. MZ : Matrix\_4H;

363. ME : Matrix\_2H;

364. A4h : Vector\_4H;

365. begin

366. Put\_line ("Process T8 started");

367.

368. accept input2 (in\_a : in integer; in\_MX : in Matrix) do

369. a := in\_a;

370. MX := in\_MX;

371. end input2;

372.

373. T9.input2 (a, MX);

374. Put\_line ("T8 input2 T9");

375.

376. accept input1 (in\_B : in Vector; in\_ME : in Matrix\_2H) do

377. B := in\_B;

378. ME := in\_ME;

379. end input1;

380.

381. T9.input1 (B, ME (H + 1 .. 2 \* H));

382. Put\_line ("T8 input1 T9");

383.

384. accept input3 (in\_MZ : in Matrix\_4H) do

385. MZ := in\_MZ;

386. end input3;

387.

388. T7.input3 (MZ (3 \* H + 1 .. 4 \* H));

389. T5.input3T8 (MZ (1 .. 2 \* H));

390.

391. mainCalc (A4h (3\*H + 1 .. 4 \* H), a, B, MX, MZ (3 \* H + 1 .. 4 \* H), ME (1 .. H));

392.

393. accept resultT5 (in\_A : in Vector\_2H) do

394. A4h (1 .. 2\*H) := in\_A;

395. end resultT5;

396.

397. accept resultT7 (in\_A : in Vector\_H) do

398. A4h (2\*H+1 .. 3 \* H) := in\_A;

399. end resultT7;

400.

401. T9.resultT8 (A4h);

402. Put\_line ("Process T8 complete");

403. end T8;

404. ---------------------------------------------

405. task body T9 is

406. a : integer;

407. B : Vector;

408. MX : Matrix;

409. MZ : Matrix;

410. ME : Matrix\_H;

411. A9h : Vector;

412. begin

413. Put\_line ("Process T9 started");

414. MZ := getMatrix (1);

415.

416. accept input2 (in\_a : in integer; in\_MX : in Matrix) do

417. a := in\_a;

418. MX := in\_MX;

419. end input2;

420.

421. accept input1 (in\_B : in Vector; in\_ME : in Matrix\_H) do

422. B := in\_B;

423. ME := in\_ME;

424. end input1;

425.

426. T4.input3 (MZ (1 .. 4 \* H));

427. Put\_line ("T9 input3 T4");

428. T8.input3 (MZ (4 \* H + 1 .. 8 \* H));

429. Put\_line ("T9 input3 T8");

430.

431. mainCalc (A9h (8 \* H + 1 .. 9 \* H), a, B, MX, MZ(8 \* H + 1 .. 9 \* H), ME);

432.

433. accept resultT4 (in\_A : in Vector\_4H) do

434. A9h (1 .. 4 \* H) := in\_A;

435. end resultT4;

436.

437. accept resultT8 (in\_A : in Vector\_4H) do

438. A9h (4 \* H + 1 .. 8 \* H) := in\_A;

439. end resultT8;

440.

441. putVector(A9h);

442. Put\_line ("Process T9 complete");

443. end T9;

444.

445. begin

446. null;

447. end main;

447 lines: No errors

GNAT GPL 2009 (20090519)

Copyright 1992-2009, Free Software Foundation, Inc.

Compiling: D:\Kache\AdaWorkspace\AdaRondevouz\source\matrix\_pack.adb (source file time stamp: 2010-04-18 18:53:40)

1. with Ada.Text\_IO, Ada.Integer\_Text\_Io;

2. use Ada.Text\_IO, Ada.Integer\_Text\_Io;

3.

4. package body Matrix\_Pack is

5. function getVector (d : in integer) return Vector is

6. result : Vector;

7. begin

8. for i in 1 .. N loop

9. result (i) := d;

10. end loop;

11. return result;

12. end getVector;

13.

14. function getMatrix (d : in integer) return Matrix is

15. result : Matrix;

16. begin

17. for i in 1 .. N loop

18. for j in 1 .. N loop

19. result (i) (j) := d; end loop;

20. end loop;

21. return result;

22. end GetMatrix;

23. ----------------------------------------

24. procedure putVector (A : in Vector) is

25. begin

26. -- new\_Line;

27. for i in 1 .. N loop

28. put (A (i) , 5);

29. end loop;

30. new\_line;

31. end PutVector;

32.

33. procedure putMatrix (MA : in matrix) is

34. begin

35. -- new\_Line;

36. for i in 1 .. N loop

37. for j in 1 .. N loop

38. put (MA (i) (j), 3);

39. end loop;

40. new\_line;

41. end loop;

42. new\_line;

43. end PutMatrix;

44. -----------------------------------------

45. procedure mainCalc (Ah : out Vector\_H; a : in integer; B : in Vector; MX : in Matrix; MZ : in Matrix\_H; ME : in Matrix\_H) is

46. Mbuf : Matrix\_H;

47. begin

48. Put\_Line("MainCalc started");

49. for j in 1 .. H loop

50. for i in 1 .. N loop

51. Mbuf (j) (i) := 0;

52. for k in 1 .. N loop

53. Mbuf (j) (i) := Mbuf (j) (i) + MX (k) (i) + MZ (j) (k);

54. end loop;

55. Mbuf (j) (i) := Mbuf (j) (i) - a \* ME (j) (i);

56. end loop;

57. end loop;

58.

59. for j in 1 .. H loop

60. Ah (j) := 0;

61. for i in 1 .. N loop

62. Ah (j) := Ah (j) + B (i) \* Mbuf (j) (i);

63. end loop;

64. end loop;

65. end MainCalc;

66. end Matrix\_Pack;

Compiling: d:\kache\adaworkspace\adarondevouz\source\matrix\_pack.ads (source file time stamp: 2010-04-18 19:09:18)

1. Package Matrix\_Pack Is

2. n : integer := 100\*9;

3. p : integer := 9;

4. h : integer := n / p;

5.

6. type All\_Vector is array (positive range<>)of integer;

7. subtype Vector is All\_Vector (1 .. N);

8. subtype Vector\_H is All\_Vector (1 .. H);

9. subtype Vector\_2H is All\_Vector (1 .. 2 \* H);

10. subtype Vector\_3H is All\_Vector (1 .. 3 \* H);

11. subtype Vector\_4H is All\_Vector (1 .. 4 \* H);

12.

13. type All\_Matrix is array (positive range<>)of Vector;

14. subtype Matrix is All\_Matrix (1 .. N);

15. subtype Matrix\_H is All\_Matrix (1 .. H);

16. subtype Matrix\_2H is All\_Matrix (1 .. 2 \* H);

17. subtype Matrix\_4H is All\_Matrix (1 .. 4 \* H);

18.

19. function getVector (d : in integer) return Vector;

20. function getMatrix (d : in integer) return Matrix;

21. procedure putMatrix (MA : in Matrix);

22. procedure putVector (A : in Vector);

23. procedure mainCalc (Ah : out Vector\_H; a : in integer; B : in Vector; MX : in Matrix; MZ : in Matrix\_H; ME : in Matrix\_H);

24.

25. End Matrix\_Pack;

66 lines: No errors