***Этап 4)* разработка программы**

Source file: ..\..\..\..\..\desktop\lab5\lab4.adb Thu Apr 26 05:30:01 201

1 ----------------------------------

2 ----------------------------------

3 -- Matviychyk B. IO - 91.

4 -- Lab5.

5 -- A=B\*(MO\*MX)\*l+Z(MR\*MT)

6 -- OR: MX,B,l,Z,MT

7 -- 25,04,2012

8 ----------------------------------

9 ----------------------------------

10

11 with Ada.Text\_IO, Ada.Integer\_Text\_IO, Ada.Synchronous\_Task\_Control, Ada.Calendar;

12 use Ada.Text\_IO, Ada.Integer\_Text\_IO, Ada.Synchronous\_Task\_Control, Ada.Calendar;

13

14 package Lab5 is

15 N:integer:=6;

16 P:integer:=6;

17 h:Integer:=N/p;

18 type Vector is array (1..n) of integer;

19 type Matrix is array (1..n) of Vector;

20 Lo:Integer;

21 MO,MR,MZS,MZB:Matrix;

22 a,bS,CS:Vector;

23 --Procedures

24 procedure VectorInput (v: out Vector);

25 procedure MatrixInput (M: out Matrix);

26 procedure VectorOutput (V: in Vector);

27 procedure MatrixOutput (M: in Matrix);

28 procedure NumbVectorMnozh (One,Two,Q: in Integer; V: in Vector; VR: out Vector);

29 procedure MatrixMnozh (One,Two: in Integer; M1,M2: in Matrix; MR2: out Matrix);

30 procedure VectorMatrixMnozh (One,Two: in Integer; v: in Vector; M: in Matrix; VR: out Vector);

31 procedure VecSlozh (One,Two: in Integer; v1,v2: in Vector; VR: out Vector);

32 end Lab5;

33

34 package body Lab5 is

35 procedure VectorInput (v: out Vector) is

36 i: integer;

37 begin

38 for I in 1..n loop

39 V(i):=1;

40 end loop;

41 end VectorInput;

42

43 procedure MatrixInput (M: out Matrix) is

44 i,j:integer;

45 begin

46 for I in 1..n loop

47 for J in 1..n loop

48 M(I)(J):=1;

49 end loop;

50 end loop;

51 end MatrixInput;

52

53 procedure VectorOutput (V: in Vector) is

54 I: integer;

55 begin

56 if n<=12 then

57 for I in 1..n loop

58 put(v(i));

59 end loop;

60 end if;

61 new\_line;

62 end VectorOutput;

63

64 procedure MatrixOutput (M: in Matrix) is

65 I,J: integer;

66 begin

67 if n<=12 then

68 for I in 1..n loop

69 for J in 1..n loop

70 put(M(i)(j));

71 Put (" ");

72 end loop;

73 new\_line;

74 end loop;

75 end if;

76 end MatrixOutput;

77

78 function FindMaxMatrix (One,Two,MIN: in Integer; M: in Matrix) return integer is

79 q: Integer;

80 begin

81 Q:= M(1)(1);

82 for i in One..Two loop

83 for J in 1..n loop

84 if Q < M(i)(J) then

85 Q:= M(I)(J);

86 end if;

87 end loop;

88 end loop;

89 return Q;

90 end FindMaxMatrix;

91

92 procedure NumbVectorMnozh (One,Two,Q: in Integer; V: in Vector; VR: out Vector) is

93 begin

94 for i in One..Two loop

95 vR(I):= Q \* V(I);--OP\*H

96 end loop;

97 end NumbVectorMnozh;

98

99 procedure MatrixMnozh (One,Two: in Integer; M1,M2: in Matrix; MR2: out Matrix) is

100 I,j,k: integer;

101 begin

102 for i in One..Two loop

103 for j in 1..n loop

104 MR2(i)(j):=0;

105 for k in 1..n loop

106 MR2(i)(j):=MR2(i)(j) + M1(K)(J) \* M2(i)(k);--OP\*H

107 end loop;

108 end loop;

109 end loop;

110 end MatrixMnozh;

111

112 procedure VectorMatrixMnozh (One,Two: in Integer; v: in Vector; M: in Matrix; VR: out Vector) is

113 I,j: integer;

114 begin

115 for i in One..Two loop

116 VR(I):=0;

117 for j in 1..n loop

118 VR(I):=VR(I) + V(J) \* M(I)(J);--OP\*H

119 end loop;

120 end loop;

121 end VectorMatrixMnozh;

122

123 procedure VecSlozh (One,Two: in Integer; v1,v2: in Vector; VR: out Vector) is

124 begin

125 for i in One..Two loop

126 VR(I):=V1(I) + V2(I);--H+H

127 end loop;

128 end VecSlozh;

129

130 end Lab5;

131

132 with Ada.Text\_IO, Ada.Integer\_Text\_IO, Ada.Synchronous\_Task\_Control, Lab5, Ada.Calendar;

133 use Ada.Text\_IO, Ada.Integer\_Text\_IO, Ada.Synchronous\_Task\_Control, Lab5, Ada.Calendar;

134

135 procedure Lab05 is

136 --monitor spec

137 protected Monitor is

138 procedure SetL(QI: in Integer);

139 procedure SetB (CI: in Vector);

140 procedure SetZ (CI: in Vector);

141 procedure SetMX(MI: in Matrix);

142 procedure SetMT(MI: in Matrix);

143 function CopyL return Integer;

144 function CopyB return Vector;

145 function CopyMX return Matrix;

146 function CopyZ return Vector;

147 function CopyMT return Matrix;

148 procedure SignalInput;

149 procedure SignalCalc;

150 entry WaitInput;

151 entry WaitCalc;

152 private

153 MX:Matrix;

154 B:Vector;

155 MT:Matrix;

156 Z:Vector;

157 L:Integer;

158 F1:integer:=0;

159 F2:integer:=0;

160 end Monitor;

161 --monitor body

162 protected body Monitor is

163 procedure SetL (QI: in Integer) is

164 begin

165 L:=QI;

166 end SetL;

167 procedure SetB (CI: in Vector) is

168 begin

169 for I in 1..N loop

170 B(I):=CI(I);

171 end loop;

172 end SetB;

173 procedure SetZ (CI: in Vector) is

174 begin

175 for I in 1..N loop

176 Z(I):=CI(I);

177 end loop;

178 end SetZ;

179 procedure SetMX(MI: in Matrix) is

180 begin

181 for I in 1..N loop

182 for J in 1..N loop

183 MX(I)(J):=MI(I)(J);

184 end loop;

185 end loop;

186 end SetMX;

187 procedure SetMT(MI: in Matrix) is

188 begin

189 for I in 1..N loop

190 for J in 1..N loop

191 MT(I)(J):=MI(I)(J);

192 end loop;

193 end loop;

194 end SetMT;

195 function CopyL return Integer is

196 begin

197 return L;

198 end CopyL;

199 function CopyB return Vector is

200 begin

201 return B;

202 end CopyB;

203 function CopyZ return Vector is

204 begin

205 return Z;

206 end CopyZ;

207 function CopyMX return Matrix is

208 begin

209 return MX;

210 end CopyMX;

211 function CopyMT return Matrix is

212 begin

213 return MT;

214 end CopyMT;

215 procedure SignalInput is

216 begin

217 F1:=F1+1;

218 end SignalInput;

219 procedure SignalCalc is

220 begin

221 F2:=F2+1;

222 end SignalCalc;

223 entry WaitInput when F1=3 is

224 begin

225 null;

226 end WaitInput;

227 entry WaitCalc when F2=6 is

228 begin

229 null;

230 end WaitCalc;

231 end Monitor;

232

233 --1 TASK

234 task t1;

235 task body t1 is

236 L1,Lo:Integer;

237 B1:Vector;

238 Z1:Vector;

239 MX1,MXo:Matrix;

240 MT1,MTo:Matrix;

241 S1:Vector;

242 S2:Vector;

243 S3:Vector;

244 MB1:Matrix;

245 MB2:Matrix;

246 One:Integer:=1;

247 Two:Integer:=H;

248 begin

249 put\_Line("T1 started");

250 MatrixInput(MTo);

251 MatrixInput(MXo);

252 Monitor.SetMX(MXo);

253 Monitor.SetMT(MTo);

254 Monitor.SignalInput;

255 Monitor.WaitInput;

256 l1:=Monitor.CopyL;

257 B1:=Monitor.CopyB;

258 Z1:=Monitor.CopyZ;

259 MX1:=Monitor.CopyMX;

260 MT1:=Monitor.CopyMT;

261 MatrixMnozh(One,Two,MT1,MR,MB1);

262 VectorMatrixMnozh(One,Two,Z1,MB1,S1);

263 MatrixMnozh(One,Two,MX1,MO,MB2);

264 VectorMatrixMnozh(One,Two,B1,MB2,S2);

265 NumbVectorMnozh(One,Two,L1,S2,S3);

266 VecSlozh(One,Two,S1,S3,A);

267 Monitor.SignalCalc;

268 put\_Line("T1 finished.");

269 end T1;

270 --2 TASK

271 task t2;

272 task body t2 is

273 L2:Integer;

274 B2:Vector;

275 Z2:Vector;

276 MX2:Matrix;

277 MT2:Matrix;

278 S1:Vector;

279 S2:Vector;

280 S3:Vector;

281 MB1:Matrix;

282 MB2:Matrix;

283 One:Integer:=H+1;

284 Two:Integer:=2\*H;

285 begin

286 put\_Line("T2 started");

287 MatrixInput(MR);

288 MatrixInput(MO);

289 Monitor.SignalInput;

290 Monitor.WaitInput;

291 l2:=Monitor.CopyL;

292 B2:=Monitor.CopyB;

293 Z2:=Monitor.CopyZ;

294 MX2:=Monitor.CopyMX;

295 MT2:=Monitor.CopyMT;

296 MatrixMnozh(One,Two,MT2,MR,MB1);

297 VectorMatrixMnozh(One,Two,Z2,MB1,S1);

298 MatrixMnozh(One,Two,MX2,MO,MB2);

299 VectorMatrixMnozh(One,Two,B2,MB2,S2);

300 NumbVectorMnozh(One,Two,L2,S2,S3);

301 VecSlozh(One,Two,S1,S3,A);

302 Monitor.SignalCalc;

303 put\_Line("T2 finished.");

304 end T2;

305 --3 TASK

306 task t3;

307 task body t3 is

308 L3:Integer;

309 B3:Vector;

310 Z3:Vector;

311 MX3:Matrix;

312 MT3:Matrix;

313 S1:Vector;

314 S2:Vector;

315 S3:Vector;

316 MB1:Matrix;

317 MB2:Matrix;

318 One:Integer:=2\*H+1;

319 Two:Integer:=3\*H;

320 begin

321 put\_Line("T3 started");

322 Monitor.WaitInput;

323 l3:=Monitor.CopyL;

324 B3:=Monitor.CopyB;

325 Z3:=Monitor.CopyZ;

326 MX3:=Monitor.CopyMX;

327 MT3:=Monitor.CopyMT;

328 MatrixMnozh(One,Two,MT3,MR,MB1);

329 VectorMatrixMnozh(One,Two,Z3,MB1,S1);

330 MatrixMnozh(One,Two,MX3,MO,MB2);

331 VectorMatrixMnozh(One,Two,B3,MB2,S2);

332 NumbVectorMnozh(One,Two,L3,S2,S3);

333 VecSlozh(One,Two,S1,S3,A);

334 Monitor.SignalCalc;

335 put\_Line("T3 finished.");

336 end T3;

337 --4 TASK

338 task t4;

339 task body t4 is

340 L4:Integer;

341 B4:Vector;

342 Z4:Vector;

343 MX4:Matrix;

344 MT4:Matrix;

345 S1:Vector;

346 S2:Vector;

347 S3:Vector;

348 MB1:Matrix;

349 MB2:Matrix;

350 One:Integer:=3\*H+1;

351 Two:Integer:=4\*H;

352 begin

353 put\_Line("T4 started");

354 Monitor.WaitInput;

355 l4:=Monitor.CopyL;

356 B4:=Monitor.CopyB;

357 Z4:=Monitor.CopyZ;

358 MX4:=Monitor.CopyMX;

359 MT4:=Monitor.CopyMT;

360 MatrixMnozh(One,Two,MT4,MR,MB1);

361 VectorMatrixMnozh(One,Two,Z4,MB1,S1);

362 MatrixMnozh(One,Two,MX4,MO,MB2);

363 VectorMatrixMnozh(One,Two,B4,MB2,S2);

364 NumbVectorMnozh(One,Two,L4,S2,S3);

365 VecSlozh(One,Two,S1,S3,A);

366 Monitor.SignalCalc;

367 put\_Line("T4 finished.");

368 end T4;

369 --5 TASK

370 task t5;

371 task body t5 is

372 L5:Integer;

373 B5:Vector;

374 Z5:Vector;

375 MX5:Matrix;

376 MT5:Matrix;

377 S1:Vector;

378 S2:Vector;

379 S3:Vector;

380 MB1:Matrix;

381 MB2:Matrix;

382 One:Integer:=4\*H+1;

383 Two:Integer:=5\*H;

384 begin

385 put\_Line("T5 started");

386 Monitor.WaitInput;

387 l5:=Monitor.CopyL;

388 B5:=Monitor.CopyB;

389 Z5:=Monitor.CopyZ;

390 MX5:=Monitor.CopyMX;

391 MT5:=Monitor.CopyMT;

392 MatrixMnozh(One,Two,MT5,MR,MB1);

393 VectorMatrixMnozh(One,Two,Z5,MB1,S1);

394 MatrixMnozh(One,Two,MX5,MO,MB2);

395 VectorMatrixMnozh(One,Two,B5,MB2,S2);

396 NumbVectorMnozh(One,Two,L5,S2,S3);

397 VecSlozh(One,Two,S1,S3,A);

398 Monitor.SignalCalc;

399 put\_Line("T5 finished.");

400 end T5;

401 --6 TASK

402 task t6;

403 task body t6 is

404 L6,Lo:Integer;

405 B6,Bo:Vector;

406 Z6,Zo:Vector;

407 MX6:Matrix;

408 MT6:Matrix;

409 S1:Vector;

410 S2:Vector;

411 S3:Vector;

412 MB1:Matrix;

413 MB2:Matrix;

414 One:Integer:=5\*H+1;

415 Two:Integer:=6\*H;

416 start, stop : time;

417 begin

418 start := Clock;

419 put\_Line("T6 started");

420 VectorInput(Bo);

421 VectorInput(Zo);

422 Monitor.SetB(Bo);

423 Monitor.SetZ(Zo);

424 lo:=1;

425 Monitor.SetL(Lo);

426 Monitor.SignalInput;

427 Monitor.WaitInput;

428 l6:=Monitor.CopyL;

429 B6:=Monitor.CopyB;

430 Z6:=Monitor.CopyZ;

431 MX6:=Monitor.CopyMX;

432 MT6:=Monitor.CopyMT;

433 MatrixMnozh(One,Two,MT6,MR,MB1);

434 VectorMatrixMnozh(One,Two,Z6,MB1,S1);

435 MatrixMnozh(One,Two,MX6,MO,MB2);

436 VectorMatrixMnozh(One,Two,B6,MB2,S2);

437 NumbVectorMnozh(One,Two,L6,S2,S3);

438 VecSlozh(One,Two,S1,S3,A);

439 Monitor.SignalCalc;

440 Monitor.WaitCalc;

441 VectorOutput(a);

442 put\_Line("T6 finished.");

443 stop := Clock;

444 Put\_Line ("finished " & Duration'Image (stop-start));

445 end T6;

446 begin

447 null;

448 end Lab05;