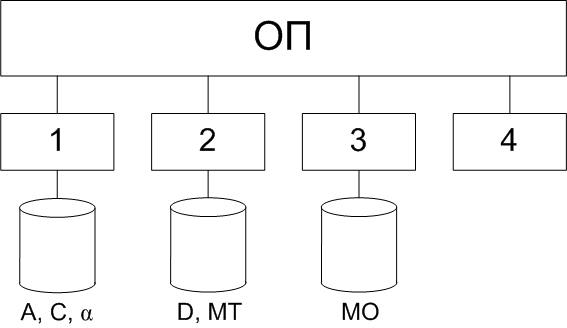
**Техническое задание**

А=sort(C \* α + D \* (MO \* MT))



**Этап 1. Построение параллельного алгоритма**

1. AH = sort(СH \* α – D \* (MO \* MTH)

**Общие ресурсы:** D, MO, α

1. A2H = sort\*(AH, AH)
2. A = sort\* (A2H, A2H)

\* - сортировка слиянием

**Этап 2. Разработка алгоритмов процессов (задач)**

**№ Т1 ТС/КУ**

1. Ввод C, α
2. Сигнал задачам Т2, Т3, Т4 о завершении ввода C, α S2.1, S3.1, S4.1
3. Ждать введения в T2, T3 W2.1, W3.1
4. Копирование MO1 := MO, α1 := α, D1 := D КУ
5. Cчёт AH = sort(СH \* α1 – D1 \* (MO1 \* MTH)
6. Ждать завершение счёта AH в задаче Т2 W1.2
7. Слияние A2H = sort(AH, AH)
8. Ждать завершение счёта A2Н в задаче Т4 W4.3
9. Слияние A = sort(A2Н, A2Н)
10. Вывод А.

**№ Т2 ТС/КУ**

1. Ввод MT, D
2. Сигнал задачам Т1, Т3, Т4 о завершении ввода МT, D S1.1, S3.1, S4.1
3. Ждать введения в T1, T3 W1.1, W3.1
4. Копирование MO2 := MO, α2 := α, D2 := D КУ
5. Cчёт AH = sort(СH \* α2 – D2 \* (MO2 \* MTH)
6. Сигнал задаче Т1 о завершении счёта AH

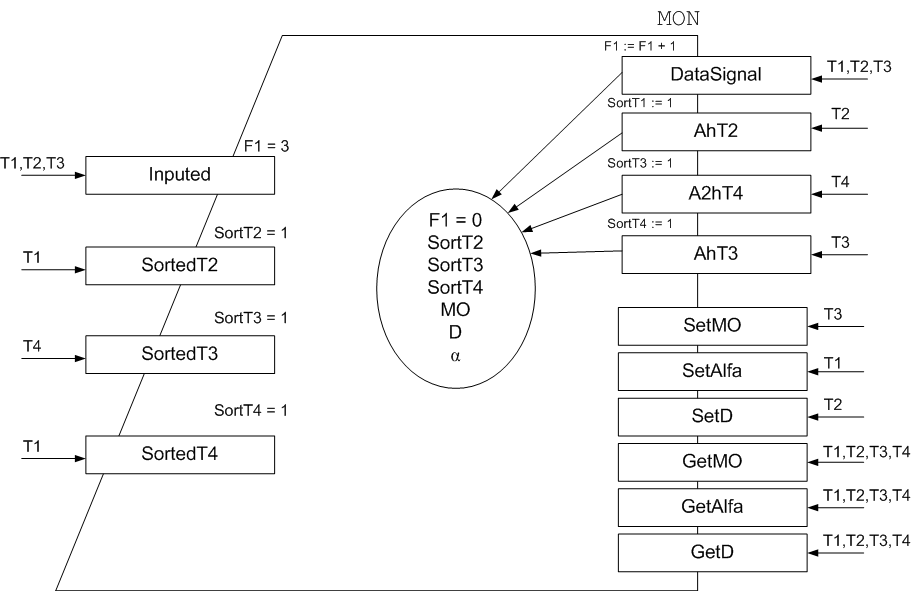
**№ Т3 ТС/КУ**

1. Ввод МО
2. Сигнал задачам Т1, Т2, Т4 о завершении ввода МО S1.1, S2.1, S4.1
3. Ждать введения в Т1, T2 W1.1, W2.1
4. Копирование MO4 := MO, α4 := α, D4 := D КУ
5. Cчёт AH = sort(СH \* α2 – D2 \* (MO2 \* MTH)
6. Сигнал задаче Т4 о завершении счёта AH S4.2

**№ Т4 ТС/КУ**

1. Ждать введения в Т1, T2, T3 W1.1, W2.1, W3.1
2. Копирование MO4 := MO, α4 := α, D4 := D КУ
3. Cчёт AH = sort(СH \* α4 – D4 \* (MO4 \* MTH)
4. Ждать завершение счёта AH в задаче Т3 W3.2
5. Слияние A2H = sort(AH, AH)
6. Сигнал задаче Т1 о завершении счёта A2H S1.1

**Этап 3. Разработка схемы взаимодействия задач**

****

**Этап 4. Разработка программы**

Source file: ..\lab4.adb Mon May 16 19:40:36 2011

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

2 -- Paralel and distributed computing --

3 -- Laboratory work #4. Ada. Semaphores --

4 -- Func: MA = MB + a \* (MC \* MO) --

5 -- IO-83 Maks Ryabko --

6 -- 27.04.2011 --

7 ----------------------------------------------------------------

8

9 with Package1; use Package1;

10 with Ada.Text\_IO; use Ada.Text\_IO;

11 with Ada.Integer\_Text\_IO; use Ada.Integer\_Text\_IO;

12 with Ada.Calendar; use Ada.Calendar;

13 with ada.float\_text\_io; use ada.float\_text\_io;

14

15 procedure Lab4 is

16 MT: Matr;

17 A: vec;

18 C: Vec;

19 Time\_all: duration;

20 time1,time2: time;

21 T: Integer;

22

23 -----------------------------------------------------------

24 ---------------ОПИСАТЕЛЬНАЯ ЧАСТЬ ЗАЩИЩЕННЫХ МОДУЛЕЙ-------

25 -----------------------------------------------------------

26

27 ----ЗАЩИЩЕННЫЙ МОДУЛЬ MON

28 protected MON is

29 --Синхронизация

30 procedure DataSignal;

31 procedure AhT2;

32 procedure A2hT4;

33 procedure AhT3;

34 --Общие ресурсы

35 procedure SetMO(M: in Matr);

36 procedure SetAlfa(M: in Integer);

37 procedure SetD(M: in Vec);

38 function GetMO return Matr;

39 function GetAlfa return Integer;

40 function GetD return Vec;

41 --Защищённые входы (синхронизация)

42 entry Inputed;

43 entry SortedT2;

44 entry SortedT3;

45 entry SortedT4;

46 --приватная часть защищенного модуля

47 private

48 --Синхронизация

49 F1: integer := 0;

50 F2: integer := 0;

51 SortT2: integer := 0;

52 SortT3: integer := 0;

53 SortT4: integer := 0;

54 --Общие ресурсы

55 MO: Matr;

56 D: Vec;

57 alfa: Integer;

58 end MON;

59

60 -----------------------------------------------------------

61 ------------------ТЕЛА ЗАЩИЩЕННЫХ МОДУЛЕЙ------------------

62 -----------------------------------------------------------

63

64 --ЗАЩИЩЕННЫЙ МОДУЛЬ Synchro

65 protected body MON is

66 -- REALIZE Синхронизация

67 procedure DataSignal is

68 begin

69 F1 := F1 + 1;

70 end DataSignal;

71

72 procedure AhT2 is

73 begin

74 SortT2 := 1;

75 end AhT2;

76 procedure A2hT4 is

77 begin

78 SortT4 := 1;

79 end A2hT4;

80 procedure AhT3 is

81 begin

82 SortT3 := 1;

83 end AhT3;

84 -- REALIZE Общие ресурсы

85 procedure SetMO(M: in Matr) is

86 begin

87 for i in 1 .. N loop

88 for j in 1 .. N loop

89 MO(i)(j) := M(i)(j);

90 end loop;

91 end loop;

92 end SetMO;

93 procedure SetAlfa(M: in Integer) is

94 begin

95 alfa := M;

96 end SetAlfa;

97 procedure SetD(M: in Vec) is

98 begin

99 for i in 1 .. N loop

100 D(i) := M(i);

101 end loop;

102 end SetD;

103 function GetMO return Matr is

104 begin

105 return MO;

106 end;

107 function GetAlfa return Integer is

108 begin

109 return alfa;

110 end;

111 function GetD return Vec is

112 begin

113 return D;

114 end;

115 --Защищённые входы (синхронизация)

116 entry Inputed when F1 = 3 is --1

117 begin

118 null;

119 end Inputed;

120

121 entry SortedT2 when SortT2 = 1 is --2

122 begin

123 null;

124 end SortedT2;

125 entry SortedT3 when SortT3 = 1 is --2

126 begin

127 null;

128 end SortedT3;

129 entry SortedT4 when SortT4 = 1 is --2

130 begin

131 null;

132 end SortedT4;

133 end MON;

134

135 procedure START is

136 -----------------------------------------------------------

137 ------------------------З А Д А Ч И------------------------

138 -----------------------------------------------------------

139 task T1;

140 task T2;

141 task T3;

142 task T4;

143

144 -----------------------ЗАДАЧА Т1 --------------------------

145 task body T1 is

146 Sum1, Temp: Integer;

147 MO1: Matr; alfa1: Integer; D1: Vec;

148 MTF: Matr;

149 Uk1, Uk2: Integer;

150 Tv: Vec;

151 begin

152 Put\_Line("T1 start");

153

154 --1. Ввод C, ?

155 Alfa1 := 1;

156 MON.SetAlfa(Alfa1);

157 Vec\_Input(C);

158 --2. Сигнал задачам Т2, Т3, Т4 о завершении ввода C, ?

159 MON.DataSignal;

160 --3. Ждать введения в T2, T3

161 MON.Inputed;

162 --4. Копирование MO1 := MO, ?1 := ?, D1 := D

163 MO1 := MON.GetMO;

164 D1 := MON.GetD;

165 --5. Cчёт AH = sort(СH \* ?1 - D1 \* (MO1 \* MTH)

166 for i in 1.. N loop

167 for j in 1 .. H loop

168 sum1 := 0;

169 for z in 1 .. N loop

170 sum1 := sum1 + MT(Z)(J) \* MO1(I)(Z);

171 end loop;

172 MTF(I)(J) := Sum1;

173 end loop;

174 end loop;

175 Sum1 := 0;

176 for i in 1.. H loop

177 for j in 1 .. N loop

178 sum1 := sum1 + D1(J) \* MTF(j)(I);

179 end loop;

180 A(I) := C(I) \* Alfa1 + Sum1;

181 Sum1 := 0;

182 end loop;

183 for Count1 in 1 .. H loop

184 for Count2 in 1 .. H loop

185 if (A(Count1) > A(Count2)) then

186 begin

187 Temp := A(Count1);

188 A(Count1) := A(Count2);

189 A(Count2) := Temp;

190 end;

191 end if;

192 end loop;

193 end loop;

194 --6. Ждать завершение счёта AH в задаче Т2 W1.2

195 MON.SortedT2;

196 --7. Слияние A2H = sort(AH, AH)

197 uk1 := 1;

198 uk2 := H+1;

199

200 for Count1 in 1 .. 2 \* H loop

201 if ((uk2 >= 2 \* H) or ((A(uk1) <= A(uk2)) and (uk1 < H))) then

202 begin

203 Tv(Count1) := A(Uk1);

204 Uk1 := Uk1 + 1;

205 end;

206 else

207 begin

208 Tv(Count1) := A(Uk2);

209 Uk2 := Uk2 + 1;

210 end;

211 end if;

212 end loop;

213 for Count1 in 1 .. 2 \* H loop

214 A(Count1) := Tv(Count1);

215 end loop;

216 --8. Ждать завершение счёта A2Н в задаче Т4 W4.3

217 MON.SortedT4;

218 --9. Слияние A = sort(A2Н, A2Н)

219 uk1 := 1;

220 uk2 := 2 \* H + 1;

221

222 for Count1 in 1 .. N loop

223 if ((uk2 >= N) or ((A(uk1) <= A(uk2)) and (uk1 < 2 \* H))) then

224 begin

225 Tv(Count1) := A(Uk1);

226 Uk1 := Uk1 + 1;

227 end;

228 else

229 begin

230 Tv(Count1) := A(Uk2);

231 Uk2 := Uk2 + 1;

232 end;

233 end if;

234 end loop;

235

236 for Count1 in 1 .. N loop

237 A(Count1) := Tv(Count1);

238 end loop;

239 --10. Вывод А.

240 if N < 24 then

241 Vec\_Print(A);

242 end if;

243

244 time2:=clock;

245 time\_all:=time2-time1;

246 Put(Integer(time\_all));

247

248 Put\_Line("T1 stop");

249 end T1;

250

251 -----------------------ЗАДАЧА Т2 --------------------------

252 task body T2 is

253 Sum1, Temp, Uk1, Uk2: Integer;

254 MO2: Matr; alfa2: Integer; D2: Vec;

255 MTF: Matr;

256 Tv: Vec;

257 begin

258 Put\_Line("T2 start");

259

260 --1. Ввод MT, D

261 Matr\_Input(MT);

262 Vec\_Input(D2);

263 MON.SetD(D2);

264 --2. Сигнал задачам Т1, Т3, Т4 о завершении ввода МT, D

265 MON.DataSignal;

266 --3. Ждать введения в T1, T3

267 MON.Inputed;

268 --4. Копирование MO2 := MO, ?2 := ?, D2 := D

269 MO2 := MON.GetMO;

270 Alfa2 := MON.GetAlfa;

271 --5. Cчёт AH = sort(СH \* ?2 - D2 \* (MO2 \* MTH)

272 for i in 1.. N loop

273 for j in H + 1 .. 2\*H loop

274 sum1 := 0;

275 for z in 1 .. N loop

276 sum1 := sum1 + MT(Z)(J) \* MO2(I)(Z);

277 end loop;

278 MTF(I)(J) := Sum1;

279 end loop;

280 end loop;

281 Sum1 := 0;

282 for i in H + 1 .. 2\*H loop

283 for j in 1 .. N loop

284 sum1 := sum1 + D2(J) \* MTF(j)(I);

285 end loop;

286 A(I) := C(I) \* Alfa2 + Sum1;

287 Sum1 := 0;

288 end loop;

289 for Count1 in H + 1 .. 2\*H loop

290 for Count2 in H + 1 .. 2\*H loop

291 if (A(Count1) > A(Count2)) then

292 begin

293 Temp := A(Count1);

294 A(Count1) := A(Count2);

295 A(Count2) := Temp;

296 end;

297 end if;

298 end loop;

299 end loop;

300 --6. Сигнал задаче Т1 о завершении счёта A2H

301 MON.AhT2;

302

303 Put\_Line("T2 stop");

304 end T2;

305

306 -----------------------ЗАДАЧА Т3 ----------------------------

307 task body T3 is

308 Sum1, Temp: Integer;

309 MO3: Matr; alfa3, Uk1, Uk2, Count1: Integer; D3: Vec;

310 MTF: Matr; Tv:Vec;

311 begin

312 Put\_Line("T3 start");

313

314 --1. Ввод МО

315 Matr\_Input(MO3);

316 MON.SetMO(MO3);

317 --2. Сигнал задачам Т1, Т2, Т4 о завершении ввода МО

318 MON.DataSignal;

319 --3. Ждать введения в T2, T3

320 MON.Inputed;

321 --4. Копирование MO4 := MO, ?4 := ?, D4 := D

322 Alfa3 := MON.GetAlfa;

323 D3 := MON.GetD;

324 --5. Cчёт AH = sort(СH \* ?2 - D2 \* (MO2 \* MTH)

325 for i in 1.. N loop

326 for j in 2\*H + 1 .. 3\*H loop

327 sum1 := 0;

328 for z in 1 .. N loop

329 sum1 := sum1 + MT(Z)(J) \* MO3(I)(Z);

330 end loop;

331 MTF(I)(J) := Sum1;

332 end loop;

333 end loop;

334 Sum1 := 0;

335 for i in 2\*H + 1 .. 3\*H loop

336 for j in 1 .. N loop

337 sum1 := sum1 + D3(J) \* MTF(j)(I);

338 end loop;

339 A(I) := C(I) \* Alfa3 + Sum1;

340 Sum1 := 0;

341 end loop;

342 for Count1 in 2\*H + 1 .. 3\*H loop

343 for Count2 in 2\*H + 1 .. 3\*H loop

344 if (A(Count1) > A(Count2)) then

345 begin

346 Temp := A(Count1);

347 A(Count1) := A(Count2);

348 A(Count2) := Temp;

349 end;

350 end if;

351 end loop;

352 end loop;

353 --6. Сигнал задаче Т4 о завершении счёта AH S4.2

354 MON.AhT3;

355

356 Put\_Line("T3 stop");

357 end T3;

358

359 -----------------------ЗАДАЧА Т4 --------------------------

360 task body T4 is

361 Sum1, Temp: Integer;

362 MO4: Matr; alfa4: Integer; D4: Vec;

363 MTF: Matr;

364 Uk1, Uk2: Integer;

365 Tv: Vec;

366 begin

367 Put\_Line("T4 start");

368

369 --1. Ждать введения в Т1, T2, T3

370 MON.Inputed;

371 --2. Копирование MO4 := MO, ?4 := ?, D4 := D

372 Alfa4 := MON.GetAlfa;

373 D4 := MON.GetD;

374 MO4 := MON.GetMO;

375 --3. Cчёт AH = sort(СH \* ?4 - D4 \* (MO4 \* MTH)

376 for i in 1.. N loop

377 for j in 3\*H + 1 .. n loop

378 sum1 := 0;

379 for z in 1 .. N loop

380 sum1 := sum1 + MT(Z)(J) \* MO4(I)(Z);

381 end loop;

382 MTF(I)(J) := Sum1;

383 end loop;

384 end loop;

385 Sum1 := 0;

386 for i in 3\*H + 1 .. n loop

387 for j in 1 .. N loop

388 sum1 := sum1 + D4(J) \* MTF(j)(I);

389 end loop;

390 A(I) := C(I) \* Alfa4 + Sum1;

391 Sum1 := 0;

392 end loop;

393 for Count1 in 3\*H + 1 .. n loop

394 for Count2 in 3\*H + 1 .. n loop

395 if (A(Count1) > A(Count2)) then

396 begin

397 Temp := A(Count1);

398 A(Count1) := A(Count2);

399 A(Count2) := Temp;

400 end;

401 end if;

402 end loop;

403 end loop;

404 --4. Ждать завершение счёта AH в задаче Т3 W3.2

405 MON.SortedT3;

406 --5. Слияние A2H = sort(AH, AH)

407 uk1 := 2 \* H + 1;

408 uk2 := 3 \* H + 1;

409

410 for Count1 in 2 \* H + 1 .. n loop

411 if ((uk2 >= N) or ((A(uk1) <= A(uk2)) and (uk1 < 3 \* H))) then

412 begin

413 Tv(Count1) := A(Uk1);

414 Uk1 := Uk1 + 1;

415 end;

416 else

417 begin

418 Tv(Count1) := A(Uk2);

419 Uk2 := Uk2 + 1;

420 end;

421 end if;

422 end loop;

423

424 for Count1 in 2 \* H + 1 .. N loop

425 A(Count1) := Tv(Count1);

426 end loop;

427 --6. Сигнал задаче Т1 о завершении счёта A2H S1.1

428 MON.A2hT4;

429

430 Put\_Line("T4 stop");

431 end T4;

432

433 begin

434 null;

435 end START;

436

437

438 begin

439 --Put\_Line("N == ");

440 --Put(N);

441 --Get(T);

442 time1:=clock;

443 START;

444 end Lab4;