Лабораторная работа №4

по курсу:

«Паралельные и распределённые вычисления»

Тема: «Ада. Защищенные модули»

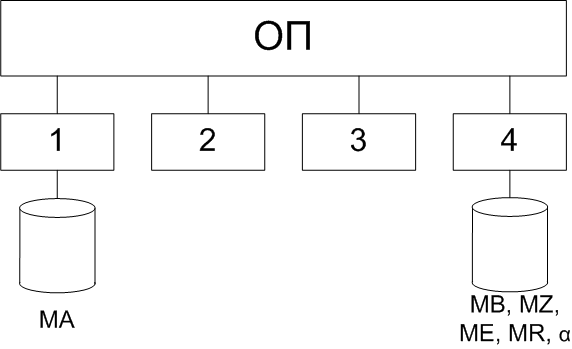
Выполнил: студент группы ИВ-83

НТУУ «КПИ» ФИВТ

Воробйов Виталий

**Техническое задание**

MA=(MB \* MZ \* α) + ME \* MR



**Этап 1. Построение параллельного алгоритма**

Вычисление данного матричного уравнения можно разбить на шаги:

1. MAH = (MBH \* MZ \* α) + MEH \* MR

**Общие ресурсы:** α, MZ, MR

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

**№ Т1**

1. Ожидание ввода в задаче Т4
2. Копирование α1:=α, MZ1:=MZ, MR1:= MR
3. Счёт MAH =(MBH\*MZ1\*α1) + MEH\*MR1
4. Ожидание окончания счёта от Т2, Т3, Т4
5. Вывод МА

**№ Т2**

1. Ожидание ввода в задаче Т4
2. Копирование α2:=α, MZ2:=MZ, MR2:=MR
3. Счёт MAH = (MBH\*MZ2\*α2) + MEH\*MR2
4. Сигнал задаче Т1 о конце счёта

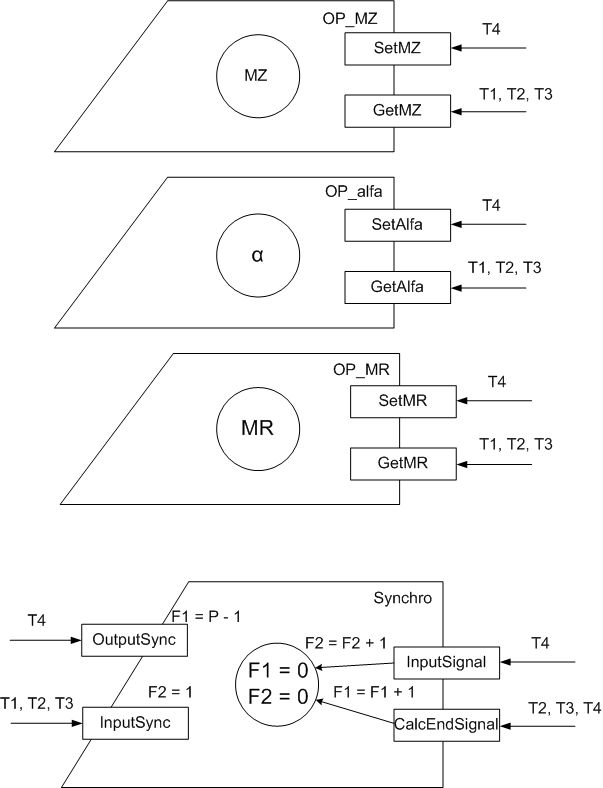
**№ Т3**

1. Ожидание ввода в задаче Т4
2. Копирование α3:=α, MZ3:=MZ, MR3:=MR
3. Счёт MAH = (MBH\*MZ3\*α3) + MEH\*MR3
4. Сигнал задаче Т1 о конце счёта

**№ Т4**

1. Ввод MB, MZ, ME, MR, α
2. Копирование α4:=α, MZ4:=MZ, MR4:=MR
3. Cигнал задачам Т1, Т2, Т3 о завершении ввода
4. Счёт MAH = (MBH\*MZ4\*α4) + MEH\*MR4
5. Сигнал задаче Т1 о конце счёта

**Этап 3. Разработка схемы взаимодействия задач**

****

**Этап 4. Разработка программы**

Source file: ..\lab1 Mon Mar 28 19:53:03 2011

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

2 -- Paralel and distributed computing --

3 -- Laboratory work #1. Ada. Semaphores --

4 -- Func: MA = (MB\*MZ\*Alfa) + ME\*MR --

5 -- IO-83 Vorobyev Vitaliy --

6 -- 28.03.2011 --

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

13 procedure Lab4 is

14 MA, MB, MZ, ME, MR: Matr;

15 Alfa: Integer;

16

17 -----------------------------------------------------------

18 ---------------ОПИСАТЕЛЬНАЯ ЧАСТЬ ЗАЩИЩЕННЫХ МОДУЛЕЙ-------

19 -----------------------------------------------------------

20

21 ----ЗАЩИЩЕННЫЙ МОДУЛЬ Synchro

22 protected Synchro is

23 procedure InputSignal;

24 procedure CalcEndSignal;

25 entry OutputSync;

26 entry InputSync;

27 --приватная часть защищенного модуля

28 private

29 F1: integer := 0;

30 F2: integer := 0;

31 end Synchro;

32

33 --ЗАЩИЩЕННЫЙ МОДУЛЬ OP\_MZ

34 protected OP\_MZ is

35 procedure SetMZ(M: in Matr);

36 function GetMZ return Matr;

37 --приватная часть защищенного модуля

38 private

39 MZ: Matr;

40 end OP\_MZ;

41

42 --ЗАЩИЩЕННЫЙ МОДУЛЬ OP\_alfa

43 protected OP\_alfa is

44 procedure SetAlfa(A: in integer);

45 function GetAlfa return Integer;

46 --приватная часть защищенного модуля

47 private

48 Alf: Integer;

49 end OP\_alfa;

50

51 --ЗАЩИЩЕННЫЙ МОДУЛЬ OP\_MZ

52 protected OP\_MR is

53 procedure SetMR(M: in Matr);

54 function GetMR return Matr;

55 --приватная часть защищенного модуля

56 private

57 MR: Matr;

58 end OP\_MR;

59

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

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

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

63

64 --ЗАЩИЩЕННЫЙ МОДУЛЬ Synchro

65 protected body Synchro is

66 procedure InputSignal is

67 begin

68 F2 := F2 + 1;

69 end InputSignal;

70 procedure CalcEndSignal is

71 begin

72 F1 := F1 + 1;

73 end CalcEndSignal;

74 entry OutputSync when F1 = P - 1 is

75 begin

76 null;

77 end OutputSync;

78 entry InputSync when F2 = 1 is

79 begin

80 null;

81 end InputSync;

82 end Synchro;

83

84 --ЗАЩИЩЕННЫЙ МОДУЛЬ OP\_MZ

85 protected body OP\_MZ is

86 procedure SetMZ(M: in Matr) is

87 begin

88 for i in 1 .. N loop

89 for j in 1 .. N loop

90 MZ(i)(j) := M(i)(j);

91 end loop;

92 end loop;

93 end SetMZ;

94

95 function GetMZ return Matr is

96 begin

97 return MZ;

98 end;

99 end OP\_MZ;

100

101 --ЗАЩИЩЕННЫЙ МОДУЛЬ OP\_Alfa

102 protected body OP\_Alfa is

103 procedure SetAlfa(A: in integer) is

104 begin

105 Alfa := A;

106 end SetAlfa;

107

108 function GetAlfa return Integer is

109 begin

110 return Alfa;

111 end;

112 end OP\_Alfa;

113

114 --ЗАЩИЩЕННЫЙ МОДУЛЬ OP\_MR

115 protected body OP\_MR is

116 procedure SetMR(M: in Matr) is

117 begin

118 for i in 1 .. N loop

119 for j in 1 .. N loop

120 MR(i)(j) := M(i)(j);

121 end loop;

122 end loop;

123 end SetMR;

124

125 function GetMR return Matr is

126 begin

127 return MR;

128 end;

129 end OP\_MR;

130

131 -----------------------------------------------------------

132 ------------------------З А Д А Ч И------------------------

133 -----------------------------------------------------------

134 task T1;

135 task T2;

136 task T3;

137 task T4;

138

139 -----------------------ЗАДАЧА Т1 --------------------------

140 task body T1 is

141 alfa1, Sum1, Sum2: Integer;

142 MZ1, MR1 : Matr;

143 begin

144 Put\_Line("T1 start");

145 --1. Ожидание ввода в задаче Т4

146 Synchro.InputSync;

147 --2. Копирование Alfa1:=Alfa, MZ1:=MZ, MR1:= MR

148 Alfa1 := OP\_Alfa.GetAlfa;

149 MZ1 := OP\_MZ.GetMZ;

150 MR1 := OP\_MR.GetMR;

151 --3. Счёт MAH = + MEH\*MR1

152 for i in 1.. N loop

153 for j in 1 .. H loop

154 sum1 := 0;

155 sum2 := 0;

156 for z in 1 .. N loop

157 sum1 := sum1 + MB(Z)(J) \* MZ1(I)(Z);

158 sum2 := sum2 + ME(Z)(J) \* MR1(I)(Z);

159 end loop;

160 MA(I)(J) := Sum1 \* Alfa1 + Sum2;

161 sum1 := 0;

162 sum2 := 0;

163 end loop;

164 end loop;

165 --4. Ожидание окончания счёта от Т2, Т3, Т4

166 Synchro.OutputSync;

167 --5. Вывод МА

168 Put\_Line(" ");

169 Matr\_Print(MA);

170 Put\_Line("T1 stop");

171 end T1;

172

173 -----------------------ЗАДАЧА Т2 --------------------------

174 task body T2 is

175 alfa2, Sum1, Sum2: Integer;

176 MZ2, MR2 : Matr;

177 begin

178 Put\_Line("T2 start");

179 --1. Ожидание ввода в задаче Т4

180 Synchro.InputSync;

181 --2. Копирование Alfa2:=Alfa, MZ2:=MZ, MR2:=MR

182 Alfa2 := OP\_Alfa.GetAlfa;

183 MZ2 := OP\_MZ.GetMZ;

184 MR2 := OP\_MR.GetMR;

185 --3. Счёт MAH = (MBH\*MZ2\*Alfa2) + MEH\*MR2

186 for i in 1.. N loop

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

188 sum1 := 0;

189 sum2 := 0;

190 for z in 1 .. N loop

191 sum1 := sum1 + MB(Z)(J) \* MZ2(I)(Z);

192 sum2 := sum2 + ME(Z)(J) \* MR2(I)(Z);

193 end loop;

194 MA(I)(J) := Sum1 \* Alfa2 + Sum2;

195 sum1 := 0;

196 sum2 := 0;

197 end loop;

198 end loop;

199 --4. Сигнал задаче Т1 о конце счёта

200 Synchro.CalcEndSignal;

201

202 Put\_Line("T2 stop");

203 end T2;

204

205 -----------------------ЗАДАЧА Т3 ----------------------------

206 task body T3 is

207 alfa3, Sum1, Sum2: Integer;

208 MZ3, MR3 : Matr;

209 begin

210 Put\_Line("T3 start");

211 --1. Ожидание ввода в задаче Т4

212 Synchro.InputSync;

213 --2. Копирование Alfa3:=Alfa, MZ3:=MZ, MR3:=MR

214 Alfa3 := OP\_Alfa.GetAlfa;

215 MZ3 := OP\_MZ.GetMZ;

216 MR3 := OP\_MR.GetMR;

217 --3. Счёт MAH = (MBH\*MZ3\*Alfa3) + MEH\*MR3

218 for i in 1.. N loop

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

220 sum1 := 0;

221 sum2 := 0;

222 for z in 1 .. N loop

223 sum1 := sum1 + MB(Z)(J) \* MZ3(I)(Z);

224 sum2 := sum2 + ME(Z)(J) \* MR3(I)(Z);

225 end loop;

226 MA(I)(J) := Sum1 \* Alfa3 + Sum2;

227 sum1 := 0;

228 sum2 := 0;

229 end loop;

230 end loop;

231 --4. Сигнал задаче Т1 о конце счёта

232 Synchro.CalcEndSignal;

233

234 Put\_Line("T3 stop");

235 end T3;

236

237 -----------------------ЗАДАЧА Т4 --------------------------

238 task body T4 is

239 alfa4, Sum1, Sum2: Integer;

240 MZ4, MR4 : Matr;

241 begin

242 Put\_Line("T4 start");

243 --1. Ввод MB, MZ, ME, MR, Alfa

244 --2. Копирование Alfa4:=Alfa, MZ4:=MZ, MR4:=MR

245 Matr\_Input(MB);

246 Matr\_Input(MZ4);

247 Matr\_Input(ME);

248 Matr\_Input(MR4);

249 Alfa4 := 2;

250

251 OP\_Alfa.SetAlfa(Alfa4);

252 OP\_MR.SetMR(MR4);

253 OP\_MZ.SetMZ(MZ4);

254 --3. Cигнал задачам Т1, Т2, Т3 о завершении ввода

255 Synchro.InputSignal;

256 --4. Счёт MAH = (MBH\*MZ4\*Alfa4) + MEH\*MR4

257 for i in 1.. N loop

258 for j in 3\*H+1 .. N loop

259 sum1 := 0;

260 sum2 := 0;

261 for z in 1 .. N loop

262 sum1 := sum1 + MB(Z)(J) \* MZ4(I)(Z);

263 sum2 := sum2 + ME(Z)(J) \* MR4(I)(Z);

264 end loop;

265 MA(I)(J) := Sum1 \* Alfa4 + Sum2;

266 sum1 := 0;

267 sum2 := 0;

268 end loop;

269 end loop;

270 --5. Сигнал задаче Т1 о конце счёта

271 Synchro.CalcEndSignal;

272

273 Put\_Line("T4 stop");

274 end T4;

275

276 begin

277 null;

278 end Lab4;