Грызлова Т. П. Образец отчета по лабораторной работе «Введение в параллельное программирование»

Оглавление

[Цель 2](#_Toc769698)

[Простейшие взаимодействия 2](#_Toc769699)

[Архитектуры 2](#_Toc769700)

[Кодирование 2](#_Toc769701)

[Анализ работы программ 3](#_Toc769702)

[Читатели – писатели 3](#_Toc769703)

[Приложение. Интерпретация временных диаграмм 5](#_Toc769704)

# Цель

1. Научиться программировать простейшие задачи на ADA в среде GNAT.ADA в соответствии с заданными архитектурами программ
2. Научиться переводить автоматную модель в управляющую программу ADA
3. Запрограммировать решение задачи «Читатели – писатели» в соответствии с заданной архитектурой параллельной программы и заданного графа состояний управляющего автомата.

# Простейшие взаимодействия

## Архитектуры

Рис. 1. Параллельные взаимодействующие процессы: а) безусловный вызов; б) условный вызов; в) временной вызов

## Кодирование

Тело задачи A, реализующий безусловный вызов:

**Task** **body** A **is**

**Begin**

**Loop**

B.E;

**End** loop;

**End** A;

Тела задач C и D, реализующих условный и временной вызов, соответственно:

**Task** **body** D **is**

nd **: integer :=** 0;

TD : **integer** := 0.1;

**Begin**

**Loop**

**select**

B.E;

**or**

nd := nd + 1;

**delay** TD;

**end** **select**;

**End** loop;

**End** C;

**Task** **body** C **is**

nf **: integer :=** 0;

**Begin**

**Loop**

**select**

B.E;

**else**

nf := nf + 1;

**end** **select**;

**End** loop;

**End** C;

## Анализ работы программ

Ниже представлен фрагмент временных диаграмм работы программы с безусловным вызовом (рис. 2) и фрагменты временных диаграмм работы программы с условным вызовом (рис. 3, рис. 4).



Рис. 2. Временные диаграммы работы программы с безусловным вызовом



Рис. 3. Временные диаграммы начала работы программы с условным вызовом

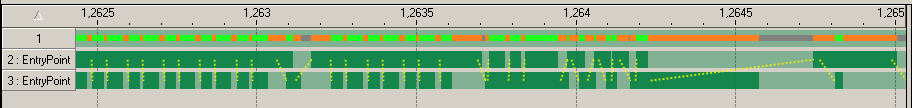


Рис. 4. Временные диаграммы середины работы программы с условным вызовом

# Читатели – писатели

На рис.5 представлена архитектура решения задачи «Читатели – писатели», на рис. 6 – управление этой системой.

Рис. 5. Архитектура программы решения задачи «Читатели – писатели»

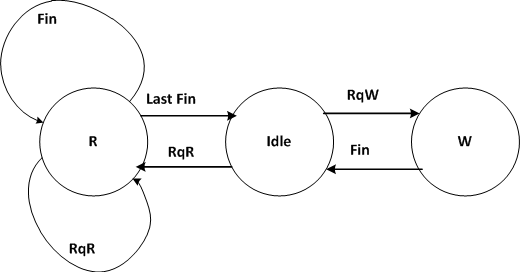


Рис. 6. Управляющий автомат, реализованный в задаче Sheduler

Ниже показано тело задачи Sheduler:

# Приложение. Интерпретация временных диаграмм

