

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное бюджетное
образовательное учреждение высшего образования
«ПЕНЗЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»**

ПРОГРАММИРОВАНИЕ СИСТЕМ РЕАЛЬНОГО ВРЕМЕНИ

**АНГЛО-РУССКИЙ, РУССКО-АНГЛИЙСКИЙ
СЛОВАРЬ-СПРАВОЧНИК**

**Составитель
В. Н. Князев**

PROGRAMMING OF REAL-TIME SYSTEMS

**ENGLISH-RUSSIAN, RUSSIAN-ENGLISH
REFERENCE DICTIONARY**

**Compiler
V. N. Knyazev**

ПЕНЗА 2024

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное
образовательное учреждение высшего образования
«Пензенский государственный университет» (ПГУ)

ПРОГРАММИРОВАНИЕ СИСТЕМ РЕАЛЬНОГО ВРЕМЕНИ

**АНГЛО-РУССКИЙ, РУССКО-АНГЛИЙСКИЙ
СЛОВАРЬ-СПРАВОЧНИК**

Составитель
В. Н. Князев

PROGRAMMING OF REAL-TIME SYSTEMS

**ENGLISH-RUSSIAN, RUSSIAN-ENGLISH
REFERENCE DICTIONARY**

Compiler
V. N. Knyazev

Пенза
Издательство ПГУ
2024

Р е ц е н з е н т

кандидат технических наук, доцент,
доцент кафедры «Вычислительная техника»
Пензенского государственного университета

А. С. Бычков

П78 Программирование систем реального времени : англо-русский, русско-английский словарь-справочник / сост. В. Н. Князев = Programming of real-time systems : English-Russian, Russian-English reference dictionary / compiler V. N. Knyazev. – Пенза : Изд-во ПГУ, 2024. – 28 с.

Содержатся слова, термины, понятия, определения и аббревиатуры по дисциплине «Программирование систем реального времени» с их толкованием и переводом на английский язык, а также обратным переводом на русский язык. Термины приведены в порядке их изучения в данной дисциплине.

Издание подготовлено на кафедре «Математическое обеспечение и применение ЭВМ» ПГУ и предназначено для обучающихся по направлению подготовки 09.03.04 «Программная инженерия». Кроме того, представленные материалы могут применяться и при освоении других дисциплин обучающимися по направлениям подготовки 09.03.02 «Информационные системы и технологии», 09.04.04 «Программная инженерия», 09.04.02 «Информационные системы и технологии».

The dictionary contains words, terms, concepts, definitions and abbreviations in the discipline «Programming real-time systems» with their interpretation and translation into English, as well as a reverse translation of the above from English into Russian.

The main purpose of this dictionary is to provide the opportunity to work with English-language literature and technical documentation for undergraduate students in the direction of 09.03.04 «Software Engineering» while studying the discipline «Programming of real-time systems». The terms are given in the order of their study in this discipline. In addition, the dictionary can be used in the study of other disciplines by students of the bachelor's degree 09.03.04 «Software Engineering», 09.03.02 «Information systems and Technologies», master's degree courses 09.04.04 «Software Engineering», 09.04.02 «Information systems and Technologies».

The dictionary can also be used in the implementation of teaching foreign undergraduate students 09.03.04 «Software Engineering» and 09.03.02 «Information Systems and Technologies» using an intermediary language.

УДК 004

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	4
1. РУССКО-АНГЛИЙСКИЙ СЛОВАРЬ	5
1.1. Понятие о системах реального времени	5
1.2. Детерминированные мультипроцессорные вычислительные системы жесткого реального времени	6
1.3. Основные требования к языку программирования реального времени	7
1.4. Методы программирования реального времени.....	9
1.5. Операционные системы реального времени.....	11
1.6. Системы реального времени на базе однокристальных микроконтроллеров	12
1.7. Системы реального времени на базе программируемых логических контроллеров	13
2. ENGLISH-RUSSIAN DICTIONARY	14
2.1. The concept of real-time systems	14
2.2. Deterministic multiprocessor computing systems of hard real time.....	15
2.3. Basic requirements for a real-time programming language.....	16
2.4. Real-time programming methods	19
2.5. Real-time operating systems.....	21
2.6. Real-time systems based on single-chip microcontrollers.....	21
2.7. Real-time systems based on programmable logic controllers	22
СПИСОК ЛИТЕРАТУРЫ	24

ВВЕДЕНИЕ

В рамках наблюдаемого интенсивного развития научно-технического прогресса нужно заметить необходимость знания английских технических терминов в области программной инженерии и информационных систем и технологий, в частности в области программирования систем реального времени, для студентов, аспирантов, преподавателей и научных работников при подготовке научных статей и докладов. Помимо этого, существенное значение имеет тот фактор, что для иностранных студентов при преподавании технических дисциплин должна применяться терминология и на английском, и на русском языке.

Этот словарь был разработан на кафедре «Математическое обеспечение и применение ЭВМ» Пензенского государственного университета.

INTRODUCTION

Within the framework of the observed intensive development of scientific and technological progress, it is necessary to note the need for knowledge of English technical terms in the field of software engineering and information systems and technologies, in particular in the field of programming real-time systems, for students, graduate students, teachers and researchers in the preparation of scientific articles and reports. In addition, the fact that terminology in both English and Russian should be used for foreign students when teaching technical disciplines is of significant importance.

This educational dictionary was developed at the Department of «Mathematical Support and Computer Application» of Penza State University.

1. РУССКО-АНГЛИЙСКИЙ СЛОВАРЬ

1.1. Понятие о системах реального времени

Система реального времени

Система реального времени (вычислительная система, работающая в режиме реального масштаба времени) – это система (аппаратно-программный комплекс), в которой на время реакции на запрос наложены ограничения извне и которая функционирует в темпе, соизмеримом со скоростью протекания внешних процессов.

A real-time system

A real-time system (a computing system operating in real-time mode) is a system (hardware and software complex) in which restrictions are imposed on the response time to a request from the outside and which operates at a pace commensurate with the speed of external processes.

Системы мягкого реального времени

Диалоговые и информационные системы реального времени (системы мягкого реального времени) работают на достаточно большое количество абонентов (пользователей), запросы от которых поступают в случайные моменты времени и обслуживаются, как правило, одним и тем же набором программ.

Soft real -time systems

Real-time dialog and information systems (soft real -time systems) work for a fairly large number of subscribers (users), whose requests are received at random times and are served, as a rule, by the same set of programs.

Системы жесткого реального времени

К управляющим вычислительным системам (системам жесткого реального времени) относятся системы управления физическими (техническими) объектами, в том числе технологическими процессами и научным экспериментом. В этих системах превышение критического времени реакции не допускается, поскольку может привести к аварийным ситуациям.

Hard real-time systems

Control computing systems (hard real-time systems) include control systems for physical (technical) objects, including technological processes and scientific experiments. In these systems, exceeding the critical reaction time is not allowed, as it can lead to emergency situations.

1.2. Детерминированные мультипроцессорные вычислительные системы жесткого реального времени

Детерминированные мультипроцессорные вычислительные системы жесткого реального времени

Управление детерминированными процессами или их детерминированными участками может проводиться с помощью особого вида систем реального времени – детерминированных мультипроцессорных вычислительных систем жесткого реального времени. Основной целью при применении такого рода систем реального времени является составление расписания, указывающего, когда и какому компоненту программного обеспечения должны выделяться ресурсы системы реального времени при выполнении конкретной программы реального времени.

Deterministic multiprocessor hard real-time computing systems

The control of deterministic processes or their deterministic sections can be carried out using a special type of real-time systems – deterministic multiprocessor computing systems of hard real time. The main purpose of using this kind of real-time systems is to create a schedule indicating when and to which software component real-time system resources should be allocated when executing a specific real-time program.

Точные методы планирования

В основе точных методов планирования лежит принцип полного перебора. Сокращение вариантов перебора достигается разделением сложной комбинаторной задачи на ряд более простых, реализуемых независимо, но без потери оптимального решения. При построении оптимальных расписаний могут использоваться, например, методы динамического программирования, методы целочисленного линейного программирования, в частности метод ветвей и границ и некоторые другие методы.

Precise planning methods

Accurate planning methods are based on the principle of a complete search. The reduction of iteration options is achieved by dividing a complex combinatorial problem into a number of simpler ones that can be solved independently, but without losing the optimal solution. When building optimal schedules, for example, dynamic programming methods, integer linear programming methods, in particular the branch and boundary method, and some other methods can be used.

Эвристические методы планирования

В основе эвристических методов планирования лежат различные интуитивно-рациональные правила и дисциплины планирования, которые в среднем дают решения, достаточно близкие к оптимальным. В этих методах используются различные правила назначения приоритетов, приближенные оценки характеристик планируемых работ, введение пошагового принципа планирования, ограничение числа анализируемых работ на каждом шаге и т.д. К эвристическим методам планирования относятся, например, эвристические методы составления списочных расписаний, эвристический алгоритм Шварца и др.

Heuristic planning methods

Heuristic planning methods are based on various intuitively rational rules and disciplines of planning, which on average give solutions that are quite close to optimal. These methods use various rules for assigning priorities, approximate estimates of the characteristics of planned works, the introduction of a step-by-step planning principle, limiting the number of analyzed works at each step, etc.

Heuristic planning methods include, for example, heuristic methods for drawing up scheduled schedules, the Schwartz heuristic algorithm, and others.

1.3. Основные требования к языку программирования реального времени

Надежность

Надежность языка – мера степени автоматического обнаружения ошибок, которое может осуществляться либо компилятором, либо системой, поддерживающей выполнение откомпилированной программы.

Reliability

The reliability of a language is a measure of the degree of automatic error detection that can be performed either by a compiler or by a system that supports the execution of a compiled program.

Типизация данных

Язык программирования реального времени должен иметь четкую, строгую систему типов данных, чтобы можно было эффективно обнаружить ошибки, проявляющиеся во время манипуляций с данными,

а также однозначно специфицировать предполагаемый вид использования данных.

Data typing

A real-time programming language should have a clear, strict data type system, so that errors that occur during data manipulation can be effectively detected, and also so that the intended use of data can be unambiguously specified.

Гибкость

Язык должен предоставлять программисту достаточное количество различных возможностей для выражения операций, которые требуются в программе, не заставляя его прибегать к вставкам ассемблерного кода или различным искусственным приемам, особенно в тех случаях, когда требуется работа с широким спектром нестандартного периферийного оборудования в режиме реального времени.

Flexibility

The language should provide the programmer with a sufficient number of different possibilities to express the operations that are required in the program, without forcing him to resort to inserts of assembly code or various artificial techniques, especially in cases where it is necessary to work with a wide range of non-standard peripheral equipment in real time.

Простота

Простота языка уменьшает затраты на обучение программистов и вероятность совершения программистских ошибок, возникающих в результате неправильной интерпретации спецификаций языка.

Simplicity

The simplicity of the language reduces the cost of training programmers and the likelihood of making programming mistakes resulting from misinterpretation of language specifications.

Эффективность

Системы реального времени должны обеспечивать высокую вычислительную пропускную способность, чтобы не нарушить ограничений, налагаемых управляемым или внешним оборудованием.

Effectiveness

Real-time systems must provide high computational throughput in order not to violate the limitations imposed by managed or external equipment.

Мобильность

Язык, не зависимый от аппаратной платформы, дает возможность относительно легко переносить программное обеспечение с одной системы на другую.

Mobility

The language, independent of the hardware platform, makes it relatively easy to transfer software from one system to another.

Обработка ошибок

Из требований высокой надежности и непрерывного функционирования систем реального времени следует, что язык программирования реального времени должен обеспечивать механизмы реакции на условия возникновения ошибок и программирование действий по восстановлению после ошибок.

Error handling

From the requirements of high reliability and continuous operation of real-time systems, it follows that the real-time programming language should provide mechanisms for responding to error conditions and programming actions to recover from errors.

Мультипрограммирование (многозадачность)

Поскольку системы реального времени, по существу, параллельны по своей природе и описываются как множество одновременно выполняемых задач, язык программирования реального времени должен предоставлять возможности для мультипрограммирования.

Multi-programming (multitasking)

Since real-time systems are essentially parallel in nature and are described as a multitude of simultaneous tasks, a real-time programming language should provide opportunities for multi-programming.

1.4. Методы программирования реального времени

Среда программирования

Среда разработки прикладного программного обеспечения систем реального времени может варьироваться от кросс-систем разработки для однокристальных микроконтроллеров до CASE-систем и систем разработки операционных систем реального времени.

Programming environment

The real-time application software development environment can range from cross-development systems for single-chip microcontrollers to CASE systems and real-time operating system development systems.

Структура программы реального времени

Принципиальной особенностью программ реального времени является постоянная готовность и отсутствие условий нормального, а не аварийного завершения. Если программа не исполняется и не обрабатывает данные, она остается в режиме ожидания прерывания, события или истечения некоторого интервала времени.

The structure of the real-time program

A fundamental feature of real-time programs is their constant readiness and the absence of conditions for normal, rather than emergency termination. If the program is not executed and does not process data, it remains in the interrupt, event, or expiration mode for a certain time interval.

Обработка прерываний и исключений

Получив сигнал прерывания, процессор приостанавливает исполнение текущего процесса, сохраняет в стеке его контекст, считывает из таблицы адрес программы обработки прерывания и передает ей управление. Исключениями являются нештатные ситуации, когда процессор не может правильно выполнить команду, например деление на ноль или обращение по несуществующему адресу.

Handling interrupts and exceptions

Upon receiving an interrupt signal, the processor suspends the execution of the current process, stores its context on the stack, reads the address of the interrupt processing program from the table and transfers control to it. Exceptions are called abnormal situations when the processor cannot execute a command correctly. An example of an exception is division by zero or addressing a non-existent address.

Программирование операций ожидания

Процесс реального времени может явным образом ждать истечения некоторого интервала (относительное время) или наступления заданного момента (абсолютное время).

Programming of waiting operations

A real-time process can explicitly wait for the expiration of a certain interval (relative time) or the onset of a given moment (absolute time).

Внутренние подпрограммы операционной системы

Типичной ситуацией при программировании систем реального времени является непосредственное обращение к подпрограммам операционной системы из-за того, что в используемом языке программирования отсутствует эквивалентное средство.

Internal operating system routines

A typical situation in real-time programming is to directly access the operating system routines due to the fact that there is no equivalent tool in the programming language used.

Приоритеты процессов и производительность системы

Операционная система реального времени должна допускать назначение приоритетов исполняемым процессам. Обычно приоритеты являются динамическими: это означает, что во время исполнения они могут изменяться как самими процессами, так и операционной системой.

Process priorities and system performance

A real-time operating system should allow prioritization of executable processes. Priorities are usually dynamic, which means that they can be changed by both the processes themselves and the operating system during execution.

Тестирование и отладка

При тестировании программ реального времени существует дополнительная сложность из-за большого количества возможных связей между задачами. Только после того как все модули проверены по отдельности и все обнаруженные ошибки исправлены, можно приступить к параллельному исполнению для отладки взаимодействия.

Testing and debugging

When testing real-time programs, there is additional complexity due to the large number of possible relationships between tasks. Only after all modules are checked individually and all detected errors are fixed, you can start parallel execution to debug the interaction.

1.5. Операционные системы реального времени

Операционные системы реального времени

Операционной системой реального времени называется такая операционная система, которая может быть использована для построения систем жесткого реального времени.

Real-time operating systems

A real-time operating system is an operating system that can be used to build hard real-time systems.

1.6. Системы реального времени на базе однокристальных микроконтроллеров

Однокристальные микроконтроллеры

Однокристальные (однокорпусные) микроконтроллеры представляют собой устройства, конструктивно выполненные в виде большой интегральной схемы и включающие в себя все составные части микро-ЭВМ: микропроцессор, память программ и память данных, а также программируемые интерфейсные схемы для связи с внешней средой. Однокристальные микроконтроллеры являются устройствами реального времени.

Single-chip microcontrollers

Single-chip (single-body) microcontrollers are devices structurally made in the form of an large integrated circuit and include all the components of a microcomputer: a microprocessor, program memory and data memory, as well as programmable interface circuits for communication with the external environment. Single-chip microcontrollers are real-time devices.

Программирование систем реального времени на базе однокристальных микроконтроллеров

Архитектура однокристальных микроконтроллеров не позволяет обеспечить разработку собственных прикладных программ, поскольку микроконтроллеры представляют собой не системы общего назначения, а специализированные системы для работы в реальном времени. Поэтому для них используются соответствующие кросс-системы разработки прикладных программ с применением в основном языка Си, языка Оберон и языка ассемблера вследствие ограниченного объема памяти микроконтроллера.

SRV programming based on single-chip microcontrollers

The architecture of single-chip microcontrollers does not allow for the development of proprietary application programs, since microcontrollers are not general-purpose systems, but specialized systems designed to work in real time. Therefore, they use appropriate cross-systems for developing application programs using, mainly, the C language, the Oberon language and the assembly language, due to the limited memory capacity of the microcontroller.

1.7. Системы реального времени на базе программируемых логических контроллеров

Программируемый логический контроллер

Программируемый логический контроллер – это специализированный цифровой компьютер, используемый для автоматизации управления технологическими процессами и различными объектами в режиме жесткого реального времени.

Programmable logic controller

A programmable logic controller (PLC) is a specialized digital computer used to automate the control of technological processes and various objects in hard real time.

Стандартная технология программирования программируемого логического контроллера

В промышленной автоматизации стандарты, особенно международные, являются главными правилами построения систем. Пример такого стандарта – международный стандарт Международной электротехнической комиссии 61131 «Программируемые контроллеры». Он широко используется разработчиками, производителями и пользователями во всем мире.

Standard PLC programming technology

In industrial automation, standards, especially international ones, are the main rules for building systems. An example of this standard is the international standard of the International Electrotechnical Commission 61131 «Programmable controllers». It is widely used by developers, manufacturers and users all over the world.

2. ENGLISH-RUSSIAN DICTIONARY

2.1. The concept of real-time systems

A real-time system

A real-time system (a computing system operating in real-time mode) is a system (hardware and software complex) in which restrictions are imposed on the response time to a request from the outside and which operates at a pace commensurate with the speed of external processes.

Система реального времени

Система реального времени (вычислительная система, работающая в режиме реального масштаба времени) – это система (аппаратно-программный комплекс), в которой на время реакции на запрос наложены ограничения извне и которая функционирует в темпе, соизмеримом со скоростью протекания внешних процессов.

Soft real -time systems

Real-time dialog and information systems (soft real -time systems) work for a fairly large number of subscribers (users), whose requests are received at random times and are served, as a rule, by the same set of programs.

Системы мягкого реального времени

Диалоговые и информационные системы реального времени (системы мягкого реального времени) работают на достаточно большое количество абонентов (пользователей), запросы от которых поступают в случайные моменты времени и обслуживаются, как правило, одним и тем же набором программ.

Hard real-time systems

Control computing systems (hard real-time systems) include control systems for physical (technical) objects, including technological processes and scientific experiments. In these systems, exceeding the critical reaction time is not allowed, as it can lead to emergency situations.

Системы жесткого реального времени

К управляющим вычислительным системам (системам жесткого реального времени) относятся системы управления физическими (техническими) объектами, в том числе технологическими процессами и научным экспериментом. В этих системах превышение критического

времени реакции не допускается, поскольку может привести к аварийным ситуациям.

2.2. Deterministic multiprocessor computing systems of hard real time

Deterministic multiprocessor hard real-time computing systems

The control of deterministic processes or their deterministic sections can be carried out using a special type of real-time systems - deterministic multiprocessor computing systems of hard real time. The main purpose of using this kind of real-time systems is to create a schedule indicating when and to which software component real-time system resources should be allocated when executing a specific real-time program.

Детерминированные мультипроцессорные вычислительные системы жесткого реального времени

Управление детерминированными процессами или их детерминированными участками может проводиться с помощью особого вида систем реального времени – детерминированных мультипроцессорных вычислительных систем жесткого реального времени. Основной целью при применении такого рода систем реального времени является составление расписания, указывающего, когда и какому компоненту программного обеспечения должны выделяться ресурсы системы реального времени при выполнении конкретной программы реального времени.

Precise planning methods

Accurate planning methods are based on the principle of a complete search. The reduction of iteration options is achieved by dividing a complex combinatorial problem into a number of simpler ones that can be solved independently, but without losing the optimal solution. When building optimal schedules, for example, dynamic programming methods, integer linear programming methods, in particular the branch and boundary method, and some other methods can be used.

Точные методы планирования

В основе точных методов планирования лежит принцип полного перебора. Сокращение вариантов перебора достигается разделением сложной комбинаторной задачи на ряд более простых, реализуемых независимо, но без потери оптимального решения. При построении

оптимальных расписаний могут использоваться, например, методы динамического программирования, методы целочисленного линейного программирования, в частности метод ветвей и границ и некоторые другие методы.

Heuristic planning methods

Heuristic planning methods are based on various intuitively rational rules and disciplines of planning, which on average give solutions that are quite close to optimal. These methods use various rules for assigning priorities, approximate estimates of the characteristics of planned works, the introduction of a step-by-step planning principle, limiting the number of analyzed works at each step, etc.

Heuristic planning methods include, for example, heuristic methods for drawing up scheduled schedules, the Schwartz heuristic algorithm, and others.

Эвристические методы планирования

В основе эвристических методов планирования лежат различные интуитивно-рациональные правила и дисциплины планирования, которые в среднем дают решения, достаточно близкие к оптимальным. В этих методах используются различные правила назначения приоритетов, приближенные оценки характеристик планируемых работ, введение пошагового принципа планирования, ограничение числа анализируемых работ на каждом шаге и т.д. К эвристическим методам планирования относятся, например, эвристические методы составления списочных расписаний, эвристический алгоритм Шварца и др.

2.3. Basic requirements for a real-time programming language

Reliability

The reliability of a language is a measure of the degree of automatic error detection that can be performed either by a compiler or by a system that supports the execution of a compiled program.

Надежность

Надежность языка – мера степени автоматического обнаружения ошибок, которое может осуществляться либо компилятором, либо системой, поддерживающей выполнение откомпилированной программы.

Data typing

A real-time programming language should have a clear, strict data type system, so that errors that occur during data manipulation can be effectively detected, and also so that the intended use of data can be unambiguously specified.

Типизация данных

Язык программирования реального времени должен иметь четкую, строгую систему типов данных, чтобы можно было эффективно обнаружить ошибки, проявляющиеся во время манипуляций с данными, а также однозначно специфицировать предполагаемый вид использования данных.

Flexibility

The language should provide the programmer with a sufficient number of different possibilities to express the operations that are required in the program, without forcing him to resort to inserts of assembly code or various artificial techniques, especially in cases where it is necessary to work with a wide range of non-standard peripheral equipment in real time.

Гибкость

Язык должен предоставлять программисту достаточное количество различных возможностей для выражения операций, которые требуются в программе, не заставляя его прибегать к вставкам ассемблерного кода или различным искусственным приемам, особенно в тех случаях, когда требуется работа с широким спектром нестандартного периферийного оборудования в режиме реального времени.

Simplicity

The simplicity of the language reduces the cost of training programmers and the likelihood of making programming mistakes resulting from misinterpretation of language specifications.

Простота

Простота языка уменьшает затраты на обучение программистов и вероятность совершения программистских ошибок, возникающих в результате неправильной интерпретации спецификаций языка.

Effectiveness

Real-time systems must provide high computational throughput in order not to violate the limitations imposed by managed or external equipment.

Эффективность

Системы реального времени должны обеспечивать высокую вычислительную пропускную способность, чтобы не нарушить ограничений, налагаемых управляемым или внешним оборудованием.

Mobility

The language, independent of the hardware platform, makes it relatively easy to transfer software from one system to another.

Мобильность

Язык, не зависящий от аппаратной платформы, дает возможность относительно легко переносить программное обеспечение с одной системы на другую.

Error handling

From the requirements of high reliability and continuous operation of real-time systems, it follows that the real-time programming language should provide mechanisms for responding to error conditions and programming actions to recover from errors.

Обработка ошибок

Из требований высокой надежности и непрерывного функционирования систем реального времени следует, что язык программирования реального времени должен обеспечивать механизмы реакции на условия возникновения ошибок и программирование действий по восстановлению после ошибок.

Multi-programming (multitasking)

Since real-time systems are essentially parallel in nature and are described as a multitude of simultaneous tasks, a real-time programming language should provide opportunities for multi-programming.

Мультипрограммирование (многозадачность)

Поскольку системы реального времени, по существу, параллельны по своей природе и описываются как множество одновременно выполняемых задач, язык программирования реального времени должен предоставлять возможности для мультипрограммирования.

2.4. Real-time programming methods

Programming environment

The real-time application software development environment can range from cross-development systems for single-chip microcontrollers to CASE systems and real-time operating system development systems.

Среда программирования

Среда разработки прикладного программного обеспечения систем реального времени может варьироваться от кросс-систем разработки для однокристальных микроконтроллеров до CASE-систем и систем разработки операционных систем реального времени.

The structure of the real-time program

A fundamental feature of real-time programs is their constant readiness and the absence of conditions for normal, rather than emergency termination. If the program is not executed and does not process data, it remains in the interrupt, event, or expiration mode for a certain time interval.

Структура программы реального времени

Принципиальной особенностью программ реального времени является постоянная готовность и отсутствие условий нормального, а не аварийного завершения. Если программа не исполняется и не обрабатывает данные, она остается в режиме ожидания прерывания, события или истечения некоторого интервала времени.

Handling interrupts and exceptions

Upon receiving an interrupt signal, the processor suspends the execution of the current process, stores its context on the stack, reads the address of the interrupt processing program from the table and transfers control to it. Exceptions are called abnormal situations when the processor cannot execute a command correctly. An example of an exception is division by zero or addressing a non-existent address.

Обработка прерываний и исключений

Получив сигнал прерывания, процессор приостанавливает исполнение текущего процесса, сохраняет в стеке его контекст, считывает из таблицы адрес программы обработки прерывания и передает ей управление. Исключениями являются нештатные ситуации, когда процессор не может правильно выполнить команду, например деление на ноль или обращение по несуществующему адресу.

Programming of waiting operations

A real-time process can explicitly wait for the expiration of a certain interval (relative time) or the onset of a given moment (absolute time).

Программирование операций ожидания

Процесс реального времени может явным образом ждать истечения некоторого интервала (относительное время) или наступления заданного момента (абсолютное время).

Internal operating system routines

A typical situation in real-time programming is to directly access the operating system routines due to the fact that there is no equivalent tool in the programming language used.

Внутренние подпрограммы операционной системы

Типичной ситуацией при программировании систем реального времени является непосредственное обращение к подпрограммам операционной системы из-за того, что в используемом языке программирования отсутствует эквивалентное средство.

Process priorities and system performance

A real-time operating system should allow prioritization of executable processes. Priorities are usually dynamic, which means that they can be changed by both the processes themselves and the operating system during execution.

Приоритеты процессов и производительность системы

Операционная система реального времени должна допускать назначение приоритетов исполняемым процессам. Обычно приоритеты являются динамическими: это означает, что во время исполнения они могут изменяться как самими процессами, так и операционной системой.

Testing and debugging

When testing real-time programs, there is additional complexity due to the large number of possible relationships between tasks. Only after all modules are checked individually and all detected errors are fixed, you can start parallel execution to debug the interaction.

Тестирование и отладка

При тестировании программ реального времени существует дополнительная сложность из-за большого количества возможных связей

между задачами. Только после того как все модули проверены по отдельности и все обнаруженные ошибки исправлены, можно приступать к параллельному исполнению для отладки взаимодействия.

2.5. Real-time operating systems

Real-time operating systems

A real-time operating system is an operating system that can be used to build hard real-time systems.

Операционные системы реального времени

Операционной системой реального времени называется такая операционная система, которая может быть использована для построения систем жесткого реального времени.

2.6. Real-time systems based on single-chip microcontrollers

Single-chip microcontrollers

Single-chip (single-body) microcontrollers are devices structurally made in the form of an BIS (large integrated circuit) and include all the components of a microcomputer: a microprocessor, program memory and data memory, as well as programmable interface circuits for communication with the external environment. Single-chip microcontrollers are real-time devices.

Однокристалльные микроконтроллеры

Однокристалльные (однокорпусные) микроконтроллеры представляют собой устройства, конструктивно выполненные в виде большой интегральной схемы и включающие в себя все составные части микро-ЭВМ: микропроцессор, память программ и память данных, а также программируемые интерфейсные схемы для связи с внешней средой. Однокристалльные микроконтроллеры являются устройствами реального времени.

Real-time programming based on single-chip microcontrollers

The architecture of single-chip microcontrollers does not allow for the development of proprietary application programs, since microcontrollers are not general-purpose systems, but specialized systems designed to work in real

time. Therefore, they use appropriate cross-systems for developing application programs using, mainly, the C language, the Oberon language and the assembly language, due to the limited memory capacity of the microcontroller.

Программирование систем реального времени на базе однокристалльных микроконтроллеров

Архитектура однокристалльных микроконтроллеров не позволяет обеспечить разработку собственных прикладных программ, поскольку микроконтроллеры представляют собой не системы общего назначения, а специализированные системы для работы в реальном времени. Поэтому для них используются соответствующие кросс-системы разработки прикладных программ с применением в основном языка Си, языка Оберон и языка ассемблера вследствие ограниченного объема памяти микроконтроллера.

2.7. Real-time systems based on programmable logic controllers

Programmable logic controller

A programmable logic controller (PLC) is a specialized digital computer used to automate the control of technological processes and various objects in hard real time.

Программируемый логический контроллер

Программируемый логический контроллер – это специализированный цифровой компьютер, используемый для автоматизации управления технологическими процессами и различными объектами в режиме жесткого реального времени.

Standard PLC programming technology

In industrial automation, standards, especially international ones, are the main rules for building systems. An example of such a standard is the international standard «Programmable Controllers» IEC 61131 (International Electrotechnical Commission). This standard is widely used by developers, manufacturers and users all over the world.

Стандартная технология программирования программируемого логического контроллера

В промышленной автоматизации стандарты, особенно международные, являются главными правилами построения систем. Пример

такого стандарта – международный стандарт Международной электротехнической комиссии 61131 «Программируемые контроллеры». Он широко используется разработчиками, производителями и пользователями во всем мире.

СПИСОК ЛИТЕРАТУРЫ

1. Семенченко И. Г. Системы реального времени. Новочеркасск : Южно-Российский государственный политехнический университет, 2017. 60 с.
2. Пьявченко Т. А. Автоматизированные информационно-управляющие системы с применением SCADA-системы TRACE MODE : учеб. пособие. СПб. : Лань, 2015. 336 с.
3. Денисенко В. В. Компьютерное управление технологическими процессами, экспериментом, оборудованием. М. : Горячая линия-Телеком, 2013. 606 с.
4. Антипин М. Е. Основы автоматизированных систем управления технологическими процессами : учеб.-метод. пособие. Томск : Томский государственный университет систем управления и радиоэлектроники, 2012. 37 с.
5. Балюбаш В. А., Добряков В. А., Назарова В. А. Автоматизированные системы управления технологическими процессами : учеб.-метод. пособие. СПб. : Санкт-Петербургский национальный исследовательский университет информационных технологий, механики и оптики, 2012. 26 с.
6. Князев В. Н., Коннов Н. Н. Системы жесткого реального времени : метод. указания к выполнению лабораторных работ // Электронная библиотека системы федеральных образовательных порталов. М. : Информика, 2007. Рег. № 58-01/0081. URL: <http://window.edu.ru/window/library>
7. Бернер Л. И., Николаев А. Б., Хадеев А. С. Системы реального времени. Красноярск : Научно-инновационный центр, 2015. 102 с.
8. Древис Ю. Г. Технические и программные средства систем реального времени : учебник для высшей школы. М. : БИНОМ. Лаборатория знаний, 2015. 336 с.
9. Котельникова С. В. Системы реального времени. Красноярск : Сибирский государственный аэрокосмический университет, 2011. 36 с.
10. Ляпунцова Е. В., Дутова Е. А., Заманов Е. А. Системы реального времени : метод. указания. М. : МИИТ, 2011. 29 с.
11. Гриценко Ю. Б. Системы реального времени : учеб. пособие. Томск : Томский государственный университет систем управления и радиоэлектроники, 2017. 253 с.
12. Деменков Н. П. Языки программирования промышленных контроллеров. М. : МГТУ им. Н. Э. Баумана, 2004. 172 с.

13. Парр Э. Программируемые контроллеры. М. : БИНОМ. Лаборатория знаний, 2007. 516 с.
14. Лившиц Ю. Е., Лакин В. И., Монич Ю. И. Программируемые логические контроллеры для управления технологическими процессами : учеб.-метод. пособие : в 2 ч. Минск : БНТУ, 2014. Ч. 1. 206 с.
15. Петров И. В. Программируемые контроллеры. Стандартные языки и приемы прикладного проектирования. М. : СОЛОН-Пресс, 2008. 256 с.
16. Нестеров К. Е., Зюзев А. М. Программирование промышленных контроллеров. Екатеринбург : Уральский университет, 2019. 96 с.
17. ISaGRAF : руководство пользователя : пер. с англ. URL: https://aveon.ru/technical_support/icpdas/ISaGRAF.pdf (дата обращения: 15.03.2021).
18. Шехтман М. Б., Угреватов А. Ю., Гурьянов Л. В. Современный универсальный промышленный контроллер DevLink-C1000 // Автоматизация в промышленности. М., 2014. С. 52–54.
19. Слета В. А., Гурьянов Л. В. От измерения и обработки тегов к объектам и быстрой разработке автоматизированных систем // Control Engineering Россия. 2015. № 6 (60). С. 20–23.
20. ГОСТ Р МЭК61131-3 – 2016. Контроллеры программируемые. Ч. 3. Языки программирования. М. : Стандартинформ, 2016. URL: <https://www.meganorm.ru/Data2/1/4293755/4293755016.pdf> (дата обращения: 02.04.2024).
21. Липаев В. В. Надежность и функциональная безопасность комплексов программ реального времени. М. : ИСП РАН, 2013. 176 с.
22. Втюрин В. А. Компьютерные технологии в области автоматизации и управления : учеб. пособие. СПб. : Изд-во СПбГЛУ, 2011. 103 с.
23. Луканов А. С. Системы реального времени : учеб. пособие. Самара : Издательство Самарского университета, 2020. 156 с.
24. Эрджиес К. Распределенные системы реального времени. Теория и практика. М. : ДМК Пресс, 2020. 382 с.
25. Древс Ю. Г. Технические и программные средства систем реального времени. М. : Лаборатория знаний, 2020. 337 с.
26. Голубев А. С. Системы реального времени : конспект лекций. Владимир : Изд-во Владимирского государственного университета, 2010. 127 с.
27. Назаров С. В. Программное обеспечение систем реального времени : монография. М. : Русайнс, 2022. 211 с.

28. Подольный В. П. Архитектура высоконагруженных систем. Системы сбора информации, распределенные системы управления, системы реального времени. М. : Сам Полиграфист, 2022. 160 с.

Учебное издание

ПРОГРАММИРОВАНИЕ СИСТЕМ РЕАЛЬНОГО ВРЕМЕНИ =
PROGRAMMING OF REAL-TIME SYSTEMS

Составитель
Князев Виктор Николаевич

Редактор *А. А. Есавкина*
Технический редактор *Д. А. Сатыров*
Компьютерная верстка *Д. А. Сатырова*

Подписано в печать 21.10.2024.
Формат 60×84¹/₁₆. Усл. печ. л. 1,62.
Заказ № 551. Тираж 40.

Издательство ПГУ
440026, г. Пенза, ул. Красная, 40.
Тел./факс: (8412) 66-60-49, 66-67-77; e-mail: iic@pnzgu.ru

Вниманию авторов!

Издательство ПГУ выпускает учебную, научную и художественную литературу, презентационную и акцидентную продукцию, а также полноцветные юбилейные и мемориальные издания в соответствии с ГОСТ 7.60–2003.

Издательство ПГУ принимает к изданию рукописи, подготовленные с использованием текстового редактора Microsoft Word for Windows версий **2003 и выше**. Формат – А4, основной шрифт – Times New Roman, 14–16 pt через одинарный интервал (минимальный размер шрифта в таблицах и сносках – 12,5 pt). Тип файла в электронном виде – doc, docx.

Работа должна содержать индекс УДК, аннотацию.

Аннотация (ГОСТ 7.86–2003, ГОСТ 7.9–1995) включает характеристику основной темы, проблемы объекта, цели работы и ее результаты. В аннотации указывают, что нового несет в себе данный документ в сравнении с другими, родственными по тематике и целевому назначению. Аннотация может включать сведения о достоинствах произведения. Текст аннотации начинают фразой, в которой сформулирована главная тема документа. Заканчивается аннотация читательским адресом.

Рисунки и таблицы должны быть размещены в тексте после ссылки на них (растровые рисунки представляются в виде отдельных файлов в формате jpg, BMP с разрешением 300 dpi, векторные рисунки в формате Corel Draw с минимальной толщиной линии 0,75 pt. (Рисунки должны быть доступны для правки!). Рисунки должны сопровождаться подрисовочными подписями, на все рисунки и таблицы в тексте должны быть ссылки.

Формулы в тексте выполняются только в редакторе формул **MathType версия 5.0** и выше. Символы греческого и русского алфавита должны быть набраны прямо, нежирно; латинского – курсивом, нежирно; обозначения векторов и матриц – прямо, жирно; цифры – прямо, нежирно. Наименования химических элементов набираются прямо, нежирно. Эти же требования необходимо соблюдать и в рисунках.

В списке литературы **нумерация источников** должна соответствовать очередности ссылок на них в тексте ([1], [2], ...). Номер источника указывается в квадратных скобках. Требования к оформлению списка литературы на русские и иностранные источники (ГОСТ 7.1–2003): для книг – фамилия и инициалы автора, название, город, издательство, год издания, том, количество страниц; для журнальных статей, сборников трудов – фамилия и инициалы автора, название статьи, полное название журнала или сборника, серия, год, том, номер, страницы; для материалов конференций – фамилия и инициалы автора, название статьи, название конференции, город, издательство, год, страницы.

К материалам **должна** прилагаться следующая информация: фамилия, имя, отчество, контактные телефоны.

Контакты Издательства ПГУ: (8412) 56-47-33, 36-84-91. E-mail: iic@pnzgu.ru



