

# Оглавление

<b>Предисловие</b> .....	15
<b>Предисловие от издательства</b> .....	18
<b>ЧАСТЬ I. ВВЕДЕНИЕ</b> .....	19
<b>Глава 1. Введение в системы реального времени</b> .....	21
1.1. Введение .....	21
1.2. Что такое система реального времени .....	22
1.3. Базовая архитектура .....	23
1.4. Характеристики систем реального времени .....	25
1.5. Классификация систем реального времени .....	25
1.6. Пример системы: конвейер бутылок .....	27
1.7. Обзор книги .....	29
1.8. Контрольные вопросы .....	29
1.9. Примечания к главе .....	30
Справочные материалы .....	30
<b>Глава 2. Аппаратура</b> .....	31
2.1. Введение .....	31
2.2. Архитектура процессора .....	32
2.2.1. Шина данных с одним циклом .....	33
2.2.2. Канал передачи данных многими циклами .....	38
2.2.3. Конвейер .....	40
2.2.3.1. Риски .....	43
2.2.4. Микроконтроллеры .....	46
2.3. Память .....	46
2.3.1. Интерфейс процессора .....	47
2.3.2. Кеш .....	48
2.4. Доступ к вводу/выводу .....	50
2.4.1. Интерфейс устройства ввода .....	51
2.4.2. Интерфейс устройства вывода .....	52
2.4.3. Отображение в память и изолированный ввод/вывод .....	53
2.4.4. Программный интерфейс ввода/вывода .....	54
2.4.4.1. Опрос .....	54
2.4.4.2. Ввод/вывод с прерыванием .....	55
2.4.4.3. Прямой доступ к памяти .....	56
2.4.4.4. Исключения .....	57
2.4.4.5. Таймеры .....	57
2.5. Многоядерные процессоры .....	58
2.6. Мультипроцессоры .....	59
2.7. Контрольные вопросы .....	60
2.8. Примечания к главе .....	61
2.9. Упражнения .....	62
Справочные материалы .....	62

<b>Глава 3. Распределенные системы реального времени</b> .....	63
3.1. Введение .....	63
3.2. Модели .....	64
3.2.1. Распределение по времени и событию .....	64
3.2.2. Конечные автоматы .....	65
3.3. Распределенные операционные системы реального времени и промежуточное программное обеспечение .....	68
3.3.1. Промежуточное программное обеспечение .....	69
3.3.2. Распределенное планирование .....	69
3.3.3. Динамическая балансировка нагрузки .....	71
3.4. Связь в реальном времени .....	72
3.4.1. Трафик в реальном времени .....	72
3.4.2. Модель взаимосвязи открытых систем .....	73
3.4.3. Топология .....	74
3.4.4. Уровень канала передачи данных .....	76
3.4.4.1. Протоколы доступа к среде .....	77
3.4.5. Протокол контроллерной сети .....	77
3.4.6. Протокол запуска по времени .....	79
3.4.7. Сеть реального времени Ethernet .....	80
3.4.8. Стандарт реального времени IEEE 802.11 .....	80
3.5. Проблемы в распределенных системах реального времени с встроенными элементами .....	81
3.6. Примеры распределенных систем реального времени .....	82
3.6.1. Современный автомобиль .....	82
3.6.2. Беспроводная мобильная сенсорная сеть .....	83
3.7. Контрольные вопросы .....	84
3.8. Примечания к главе .....	85
3.9. Упражнения .....	86
Справочные материалы .....	87
<b>Часть II. СИСТЕМНОЕ ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ</b> .....	89
<b>Глава 4. Операционные системы реального времени</b> .....	91
4.1. Введение .....	91
4.2. Общие операционные системы и операционные системы реального времени .....	92
4.3. Управление задачей .....	93
4.3.1. Задача управления в UNIX .....	95
4.3.2. Синхронизация задач .....	97
4.3.3. Межзадачные коммуникации .....	98
4.3.4. Межпроцессорное взаимодействие в UNIX .....	101
4.4. Потoki .....	102
4.4.1. Управление потоками .....	102
4.4.2. Потoki POSIX .....	103
4.4.2.1. Взаимное исключение .....	104
4.4.2.2. Синхронизация .....	105
4.4.2.3. Связь .....	106

4.5. Управление памятью .....	108
4.5.1. Статическое распределение памяти .....	108
4.5.2. Динамическое распределение памяти .....	108
4.5.3. Виртуальная память .....	108
4.5.4. Управление памятью в реальном времени.....	109
4.6. Управление вводом/выводом.....	110
4.6.1. Управляемый прерываниями ввод/вывод.....	110
4.6.2. Драйверы устройств .....	111
4.7. Обзор операционных систем реального времени .....	112
4.7.1. Операционная система с открытым кодом RTOS.....	113
4.7.2. Операционная система VxWorks.....	113
4.7.3. Система реального времени Linux .....	113
4.8. Контрольные вопросы .....	114
4.9. Примечания к главе .....	115
4.10. Упражнения по программированию .....	115
Справочные материалы .....	116

## **Глава 5. Проектирование экспериментального**

<b>распределенного ядра реального времени .....</b>	<b>117</b>
5.1. Введение .....	117
5.2. Стратегия дизайна .....	118
5.3. Функции ядра нижнего уровня .....	119
5.3.1. Структуры данных и операции с очередями .....	119
5.3.1.1. Тип блока данных .....	120
5.3.1.2. Тип данных блока управления задачами .....	121
5.3.2. Планировщик с несколькими очередями .....	123
5.3.3. Обработка прерываний и управление временем.....	126
5.3.3.1. Дельта-очередь .....	127
5.3.4. Управление состоянием задачи.....	128
5.3.4.1. Программа обработки прерываний по времени.....	129
5.3.5. Управление вводом/выводом .....	130
5.4. Функции ядра верхнего уровня .....	132
5.4.1. Синхронизация задач.....	132
5.4.2. Коммуникация задач.....	136
5.4.3. Управление верхней памятью с использованием пулов .....	139
5.4.4. Управление задачами.....	140
5.5. Инициализация.....	142
5.6. Тестирование DRTK.....	144
5.7. Контрольные вопросы.....	145
5.8. Примечания к главе .....	145
5.9. Проекты программирования .....	146
Справочные материалы .....	147

## **Глава 6. Операционные системы реального времени**

<b>и промежуточное программное обеспечение.....</b>	<b>148</b>
6.1. Введение .....	148

6.2. Распределенные операционные системы реального времени .....	149
6.2.1. Интерфейс транспортного уровня .....	150
6.2.2. Интерфейс уровня канала передачи данных.....	151
6.3. Промежуточное программное обеспечение реального времени.....	153
6.3.1. Группы задач в реальном времени .....	154
6.3.2. Синхронизация часов.....	155
6.3.2.1. Логические часы .....	156
6.3.2.2. Векторные часы.....	157
6.3.2.3. Сетевой протокол времени.....	158
6.3.2.4. Алгоритм Беркли.....	160
6.3.2.5. Синхронизация часов в беспроводных сенсорных сетях .....	160
6.3.3. Алгоритмы выбора .....	162
6.3.3.1. Выбор в однонаправленном кольце.....	162
6.3.3.2. Выборы в беспроводных сенсорных и мобильных специальных сетях.....	163
6.4. Реализация DRTK .....	165
6.4.1. Инициализация сети .....	165
6.4.2. Интерфейс транспортного уровня .....	167
6.4.3. Задачи интерфейса канального уровня передачи данных.....	171
6.4.4. Групповое управление .....	173
6.4.5. Алгоритм синхронизации часов.....	175
6.4.6. Выбор лидера в кольце .....	176
6.5. Контрольные вопросы .....	178
6.6. Примечания к главе .....	178
6.7. Проекты программирования.....	179
Справочные материалы .....	179

## **ЧАСТЬ III. ПЛАНИРОВАНИЕ И РАСПРЕДЕЛЕНИЕ РЕСУРСОВ.....**

181

<b>Глава 7. Задача планирования однопроцессорной независимой задачи.....</b>	<b>183</b>
7.1. Введение .....	183
7.2. Предпосылки .....	184
7.2.1. Тест планируемости.....	185
7.2.2. Применение.....	186
7.3. Политики планирования .....	186
7.3.1. Приоритетное или неприоритетное планирование .....	186
7.3.2. Статичное или динамичное планирование .....	188
7.3.3. Независимые или зависимые задачи .....	189
7.4. Таксономия алгоритмов планирования в реальном времени.....	190
7.5. Потактовое планирование.....	191
7.5.1. Планирование на основе таблиц .....	191
7.5.2. Циклическое выполнение .....	193
7.6. Приоритетное планирование.....	195
7.6.1. Монотонное планирование.....	195

7.6.2. Планирование с первым самым крайним сроком .....	197
7.6.2.1. Аperiodический алгоритм EDF .....	198
7.6.2.2. Периодический алгоритм EDF .....	199
7.6.3. Алгоритм наименьшего времени незанятости .....	201
7.6.4. Анализ времени отклика .....	202
7.7. Аperiodическое планирование задач .....	203
7.7.1. Основные методы .....	204
7.7.2. Периодические серверы .....	206
7.7.2.1. Сервер опроса .....	206
7.7.2.2. Отложенный сервер .....	207
7.7.2.3. Спорадический сервер .....	207
7.7.2.4. Серверы с динамическим приоритетом .....	207
7.8. Планирование спорадических задач .....	208
7.9. Реализация в DTRK .....	210
7.9.1. Планировщик монотонного рейтинга .....	210
7.9.2. Планировщик самого раннего первого срока .....	212
7.9.3. Планировщик первой наименее занятой задачи .....	212
7.9.4. Сервер опроса .....	214
7.10. Контрольные вопросы .....	214
7.11. Примечания к главе .....	215
7.12. Упражнения .....	217
Справочные материалы .....	218

## **Глава 8. Планирование однопроцессорной**

<b>зависимой задачи .....</b>	<b>219</b>
8.1. Введение .....	219
8.2. Планирование зависимых задач .....	220
8.2.1. Алгоритм первым последний конечный срок .....	220
8.2.2. Модифицированный алгоритм первым ранний конечный срок .....	221
8.3. Планирование задач при совместном использовании ресурсов .....	223
8.3.1. Случай марсианского зонда .....	226
8.3.2. Основной протокол наследования приоритетов .....	227
8.3.3. Протокол приоритетного потолка .....	231
8.4. Реализация DRTK .....	233
8.4.1. Зависимое планирование задач LDF .....	234
8.4.2. Протокол приоритетного наследования .....	235
8.5. Контрольные вопросы .....	237
8.6. Примечания к главе .....	237
8.7. Упражнения .....	238
Справочные материалы .....	239

## **Глава 9. Планирование многопроцессорных**

<b>распределенных задач реального времени .....</b>	<b>240</b>
9.1. Введение .....	240
9.2. Многопроцессорное планирование .....	241
9.2.1. Раздельное планирование .....	242

9.2.1.1. Распределение задач.....	242
9.2.1.2. Алгоритм балансировки использования с EDF .....	243
9.2.1.3. Алгоритмы .....	244
9.2.1.4. Алгоритм EDF с упаковкой первое соответствие .....	246
9.2.1.5. Монотонный алгоритм первое соответствие (RM-FF).....	246
9.2.2. Глобальное планирование .....	247
9.2.2.1. Глобальный монотонный алгоритм.....	248
9.2.2.2. Аномалии .....	249
9.2.2.3. Пропорциональный алгоритм справедливого планирования.....	249
9.3. Распределенное планирование.....	249
9.3.1. Балансировка нагрузки .....	250
9.3.1.1. Центральная балансировка нагрузки .....	250
9.3.1.2. Распределенная балансировка нагрузки .....	251
9.3.2. Метод целенаправленной адресации и назначения ставок.....	252
9.3.3. Алгоритм Buddy .....	253
9.3.4. Планирование сообщений .....	254
9.4. Реализация DRTK .....	255
9.4.1. Центральные задачи балансировки нагрузки .....	255
9.4.2. Задача балансировки распределенной нагрузки .....	257
9.5. Контрольные вопросы .....	259
9.6. Примечания к главе .....	259
9.7. Упражнения .....	260
Справочные материалы .....	260

## **ЧАСТЬ IV. ПРИКЛАДНОЕ ПРОЕКТИРОВАНИЕ.....263**

<b>Глава 10. Разработка программного обеспечения систем реального времени .....</b>	<b>265</b>
10.1. Введение .....	265
10.2. Жизненный цикл разработки программного обеспечения.....	266
10.2.1. Пошаговая модель Waterfall.....	267
10.2.2. V-модель.....	268
10.2.3. Спиральная модель Spiral Model.....	268
10.3. Разработка программного обеспечения систем реального времени.....	269
10.4. Анализ требований и спецификация .....	270
10.5. Временной анализ.....	271
10.6. Структурное проектирование с диаграммами потоков данных .....	272
10.7. Объектно-ориентированное проектирование .....	274
10.8. Методы реализации в реальном времени.....	275
10.8.1. Еще о конечных автоматах .....	275
10.8.1.1. Параллельные иерархические конечные автоматы .....	277
10.8.2. Временные автоматы.....	278
10.8.3. Сети Петри .....	279
10.8.3.1. Временные сети Петри .....	282
10.9. Унифицированный язык моделирования в реальном времени.....	283
10.9.1. UML-диаграммы .....	283
10.9.2. Функции реального времени.....	285
10.10. Метод практического проектирования и реализации .....	286

10.11. Контрольные вопросы .....	287
10.12. Примечания к главе .....	288
10.13. Программирование проектов .....	289
Справочные материалы .....	290

## **Глава 11. Языки программирования в реальном времени** ..... 291

11.1. Введение .....	291
11.2. Требования .....	292
11.3. Приложение в реальном времени.....	293
11.4. Операционная система C/POSIX в реальном времени.....	293
11.4.1. Инкапсуляция данных и управление модулями .....	294
11.4.2. Управление потоком POSIX .....	295
11.4.2.1. Управление временем .....	296
11.4.2.2. Синхронизация потоков и связь .....	297
11.4.2.3. Сигналы.....	297
11.4.2.4. Взаимное исключение.....	298
11.4.2.5. Условная синхронизация .....	299
11.4.2.6. Семафоры.....	299
11.4.3. Обработка исключений и низкоуровневое программирование .....	300
11.4.4. Реализация управления процессом C/POSIX в реальном времени.....	300
11.5. Ада .....	302
11.5.1. Параллелизм .....	303
11.5.1.1. Управление временем .....	304
11.5.1.2. Periodic Tasks.....	304
11.5.1.3. Приоритеты задач .....	305
11.5.1.4. Синхронизация задач и связь.....	305
11.5.2. Обработка исключений .....	306
11.5.3. Реализация управления процессами на языке ADA.....	308
11.6. Язык программирования Java .....	309
11.6.1. Потоки Java.....	310
11.6.2. Синхронизация потоков .....	311
11.6.2.1. Управление временем и расписание .....	311
11.6.3. Обработка исключений .....	312
11.7. Контрольные вопросы.....	312
11.8. Примечания к главе .....	313
11.9. Упражнения по программированию .....	314
Справочные материалы .....	314

## **Глава 12. Отказоустойчивость** ..... 315

12.1. Введение .....	315
12.2. Понятия и терминология.....	316
12.3. Классификация неисправностей .....	317
12.4. Резервирование.....	318
12.4.1. Аппаратное резервирование .....	318
12.4.2. Избыточность информации.....	320
12.4.2.1. Кодирование .....	320
12.4.3. Резервирование времени.....	322

12.4.4. Резервирование программного обеспечения.....	323
12.4.4.1. Методы с одной версией.....	323
12.4.4.2. Многократное резервирование.....	324
12.5. Отказоустойчивые системы реального времени.....	325
12.5.1. Статичное планирование.....	326
12.5.2. Динамическое планирование.....	326
12.6. Отказоустойчивость в распределенных системах реального времени.....	327
12.6.1. Классификация отказов.....	327
12.6.2. Пересмотр состава целевых групп.....	328
12.6.2.1. Надежная многоадресная связь.....	328
12.7. Реализация DRTK.....	331
12.8. Контрольные вопросы.....	334
12.9. Примечания к главе.....	335
12.10. Упражнения.....	335
Справочные материалы.....	336

## **Глава 13. Тематический пример: мониторинг окружающей среды по беспроводной сенсорной сети.....**

13.1. Введение.....	338
13.2. Вопросы проектирования.....	339
13.3. Требования к спецификации.....	339
13.4. Временной анализ и функциональные характеристики.....	340
13.5. Связующее дерево и кластеризация.....	341
13.6. Вопросы проектирования.....	345
13.7. Листовой узел.....	346
13.7.1. Дизайн высокого уровня.....	347
13.7.2. Детальная разработка и реализация.....	349
13.8. Промежуточный узел.....	355
13.8.1. Дизайн верхнего уровня.....	355
13.8.2. Детальное проектирование и реализация.....	356
13.9. Узел управления кластером.....	360
13.9.1. Дизайн высокого уровня.....	361
13.9.1. Дизайн верхнего уровня.....	361
13.10. Приемник.....	364
13.10.1. Дизайн высокого уровня.....	364
13.11. Тестирование.....	365
13.12. Альтернативная реализация с потоками POSIX.....	367
13.13. Примечания к главе.....	367
13.14. Упражнения по программированию.....	368
Справочные материалы.....	368

## **Приложение А. Соглашение о псевдокоде.....**

## **Приложение В. Функции нижнего ядра.....**

## **Предметный указатель.....**



# Предисловие

Распределенные системы реального времени с встроенными элементами присутствуют повсюду – от производственных площадок заводов до автомобилей и авионики. Распределенные системы реального времени характеризуются числом вычислительных элементов, объединенных в сеть и выполняющих задачи в реальном времени. Задачи реального времени имеют конечные сроки, и многие применения требуют получения решения до их истечения. Современные технологические достижения привели к значительному увеличению числа элементов в распределенных системах реального времени, что, в свою очередь, вызвало необходимость разработки соответствующего программного обеспечения. Элементы распределенных систем реального времени имеют вычислительные мощности и обычно связаны с внешним миром при помощи датчиков и исполнительных механизмов. Не все системы с встроенными элементами работают в режиме реального времени, и мы будем использовать термин *распределенная система реального времени* только для таких распределенных систем, которые характеризуются работой в реальном времени.

Эта книга о разработке и внедрении программного обеспечения для распределенных систем реального времени с использованием метода восходящего проектирования. На протяжении нескольких десятилетий я читал соответствующие курсы бакалаврам и магистрам и принимал участие в крупных программных проектах систем реального времени, что дало мне возможность наблюдать основные узкие места, которые встречаются при разработке и проектировании таких систем. Прежде всего разработчик или проектировщик часто сталкивается с проблемой сопряжения приложения с некоторой коммерческой операционной системой или промежуточным программным обеспечением реального времени и иногда вынужден писать патчи к ним. Это требует глубокого понимания концепций как аппаратного обеспечения, так и операционной системы, обеспечивающих обработку в реальном времени, и поэтому часть книги посвящена системному программному обеспечению. Первая часть книги состоит из трех глав. В первой главе мы рассматриваем основные концепции операционных систем реального времени. Затем, во второй главе, мы строим с нуля экспериментальное ядро распределенной операционной системы реального времени (*experimental distributed real-time operating system kernel, DRTK*), раскрывая все необходимые детали.

Далее, в последней главе этой части книги, рассматриваются концепции связи распределенной операционной системы реального времени и промежуточного программного обеспечения и рассказывается, как спроектировать

сетевые коммуникации, чтобы в режиме реального времени ядра могли взаимодействовать и иметь распределенный системный программный фрейм. Постепенно мы превращаем экспериментальное ядро в ядро распределенной операционной системы реального времени со связанным промежуточным программным обеспечением, показывая в последующих разделах книги все детали реализации.

Вторая проблема – это, конечно же, планирование задач, которое обеспечивает выполнение всех конечных сроков. Задачи в системе реального времени можно в широком смысле классифицировать как жесткие, мягкие и твердые задачи, которые могут быть также периодическими и аperiodическими и требующими различных стратегий планирования. Более того, они могут быть независимыми или зависеть от каждого из необходимых средств синхронизации задач. Вместе с тем нам нужно обеспечить такое сквозное планирование задач, чтобы сроки были соблюдены, а нагрузка распределялась между элементами распределенной системы реального времени равномерно. Еще одной связанной с этим проблемой является управление ресурсами в сети. Все эти проблемы исследуются во второй части, и всегда имеются в виду методы их реализации.

Наконец, перед разработчиком стоит задача выполнения всех этапов разработки программного обеспечения, начиная с определения требований и разработки общего плана и до последующей детальной разработки и кодирования. Именно здесь встречается большинство затруднений. Мы предоставляем новый метод для простого и эффективного выполнения всех этапов. В книге есть глава о проектировании верхнего уровня и детальной разработке с использованием конечных автоматов, которые могут быть реализованы с применением потоков в операционной системе. В системе реального времени для предотвращения катастрофических событий обязательна отказоустойчивость, и в книге есть глава на эту тему. Также рассматриваются различные языки программирования в реальном времени, включая C/POSIX, Ada и Java. Наконец, в последней главе представлено тематическое исследование режима реального времени, включающее все методы, которые были рассмотрены и разработаны, начиная с проектирования верхнего уровня с последующей подробной разработкой и кодированием.

В каждой главе мы сначала рассматриваем концепции, а затем описываем методы разработки и реализации необходимого программного обеспечения, давая краткое описание коммерчески доступного программного обеспечения. В главах второй части книги, где это целесообразно, мы показываем, как реализовать описанные концепции в экспериментальном образце ядра, отображая исполняемый код. Обычно эти части в каждой главе называются «Реализация DRTK», и их можно пропустить вместе с главой 5, которая описывает DRTK, если книга используется для курса с ограниченным охватом систем реального времени. Наконец, мы включили обзорные вопросы, а затем и краткое примечание к главе, подчеркивая основные моменты, давая современный обзор соответствующей литературы и возможные открытые для исследований области.

## **О ЯДРЕ РАСПРЕДЕЛЕННОЙ ОПЕРАЦИОННОЙ СИСТЕМЫ РЕАЛЬНОГО ВРЕМЕНИ**

Основные модули DRTK, рассмотренные в деталях в главе 5, протестированы при обучении студентов на различных курсах по обработке в реальном времени. Однако материал глав этой части книги, следующих за этой главой и относящихся к распределенной обработке, не полностью протестирован и с некоторой вероятностью может иметь синтаксические или другие ошибки при реализации. Я буду сохранять домашнюю страницу книги по адресу [http://akademik.ube.eg.edu.tr/\\*erciyes/DRTS](http://akademik.ube.eg.edu.tr/*erciyes/DRTS) для кода DRTK, учебных слайдов и ошибок. Приветствуются любые изменения в коде DRTK в надежде сделать его практически экспериментально распределенным ядром реального времени, которое может затем быть исследовано на соответствующих учебных курсах.

Целевой аудиторией книги являются студенты и выпускники старших курсов, проектировщики и исследователи электротехники и вычислительной техники, информатики и машиностроения в целом, все, имеющие базовый опыт работы в области компьютерной архитектуры и операционных систем. Текст содержит значительное количество кода Си в примерах реализации DRTK и других различных примерах. Я хотел бы поблагодарить студентов и аспирантов различных университетов, включая (в хронологическом порядке) Университет Эге, Университет штата Орегон, Калифорнийский государственный университет Дэвис, Калифорнийский государственный университет Сан-Маркос, Измирский Технологический институт, Измирский университет и Ускюдарский университет, в которых я преподавал курсы «Системы реального времени», «Встроенные системы», «Современные операционные системы» или другие аналогичные курсы, за их ценную обратную связь, когда различные части включенных в книгу материалов излагались во время лекций, а образец ядра тестировался в лабораторных работах. Я также хотел бы поблагодарить главного редактора издательства Springer Уэйна Уилера (Wayne Wheeler) и помощника редактора Саймона Риса (Simon Rees) за их постоянную поддержку в процессе написания книги.

*Стамбул, Турция К. Эрджиес (K. Erciyes)*

# Предисловие от издательства

## ОТЗЫВЫ И ПОЖЕЛАНИЯ

Мы всегда рады отзывам наших читателей. Расскажите нам, что вы думаете об этой книге – что понравилось или, может быть, не понравилось. Отзывы важны для нас, чтобы выпускать книги, которые будут для вас максимально полезны.

Вы можете написать отзыв прямо на нашем сайте [www.dmkpress.com](http://www.dmkpress.com), зайдя на страницу книги, и оставить комментарий в разделе «Отзывы и рецензии». Также можно послать письмо главному редактору по адресу [dmkpress@gmail.com](mailto:dmkpress@gmail.com), при этом напишите название книги в теме письма.

Если есть тема, в которой вы квалифицированы, и вы заинтересованы в написании новой книги, заполните форму на нашем сайте по адресу [http://dmkpress.com/authors/publish\\_book/](http://dmkpress.com/authors/publish_book/) или напишите в издательство по адресу [dmkpress@gmail.com](mailto:dmkpress@gmail.com).

## СПИСОК ОПЕЧАТОК

Хотя мы приняли все возможные меры для того, чтобы удостовериться в качестве наших текстов, ошибки все равно случаются. Если вы найдете ошибку в одной из наших книг – возможно, ошибку в тексте или в коде, – мы будем очень благодарны, если вы сообщите нам о ней. Сделав это, вы избавите других читателей от расстройств и поможете нам улучшить последующие версии этой книги.

Если вы найдете какие-либо ошибки в коде, пожалуйста, сообщите о них главному редактору по адресу [dmkpress@gmail.com](mailto:dmkpress@gmail.com), и мы исправим это в следующих тиражах.

## НАРУШЕНИЕ АВТОРСКИХ ПРАВ

Пиратство в интернете по-прежнему остается насущной проблемой. Издательства «ДМК Пресс» и Packt очень серьезно относятся к вопросам защиты авторских прав и лицензирования. Если вы столкнетесь в интернете с незаконно выполненной копией любой нашей книги, пожалуйста, сообщите нам адрес копии или веб-сайта, чтобы мы могли применить санкции.

Пожалуйста, свяжитесь с нами по адресу электронной почты [dmkpress@gmail.com](mailto:dmkpress@gmail.com) со ссылкой на подозрительные материалы.

Мы высоко ценим любую помощь по защите наших авторов, помогающую нам предоставлять вам качественные материалы.

**ЧАСТЬ I**



**ВВЕДЕНИЕ**



# Глава 1

## Введение в системы реального времени

### 1.1. ВВЕДЕНИЕ

Системы реального времени присутствуют повсюду, от автомобилей и мобильных телефонов до авиационного оборудования и систем управления ядерными установками. Они характеризуются своевременностью реагирования на некоторый вход. Ошибка в реагировании в течение определенного времени может иногда привести к катастрофе. Правильность работы системы реального времени зависит как от правильности результатов, так и от времени получения этих результатов, которые соответственно называются *логической корректностью* (logical correctness) и *временной корректностью* (temporal correctness). В *системе реального времени с встроенными элементами* (embedded real-time system) вычислительный элемент реального времени встроен в контролируемую систему реального времени, и в эту категорию попадает большая часть элементов такой системы.

Существует много типов систем реального времени; система управления процессом в реальном времени получает входные данные от датчиков, выполняет некоторые операции с этими данными и создает выход для управления различными функциями системы, такими как включение и выключение, при необходимости активирует аварийные сигналы и отображает системные данные. Самолет в режиме реального времени является системой со строгими временными ограничениями, которые должны быть соблюдены, в то время как несоблюдение временных ограничений мультимедийной системы, как правило, приведет лишь к некачественному воспроизведению. Другие примеры систем реального времени включают роботизированные системы, атомные электростанции и мобильные телефоны. В этой главе мы рассмотрим основные понятия, связанные с системами реального времени, и, начиная с определения, разберем типы систем реального времени и свойства этих систем.

Затем опишем работу систем реального времени на примерах и в заключение дадим краткий обзор этой книги.

## 1.2. ЧТО ТАКОЕ СИСТЕМА РЕАЛЬНОГО ВРЕМЕНИ

Система реального времени работает с временным ограничением, то есть значимым временем вывода результата. Другими словами, получать результат в другое время, чем требуемое, не имеет смысла. Альтернативный вариант системы реального времени – система обработки данных, которая реагирует на внешние входы в течение заданного времени, называемого крайним сроком. Неспособность дать реакцию на ввод за определенный промежуток времени может нанести существенный вред жизни или имуществу. Система реального времени разрабатывается в соответствии с динамикой физических процессов, которые она должна контролировать. Вычислительная система реального времени состоит из аппаратного и программного обеспечения, работающих с временным ограничением. Программное обеспечение реального времени в целом включает в себя следующие компоненты:

- *операционная система реального времени*: операционная система имеет две основные функции: предоставление удобного доступа к аппаратному обеспечению и эффективное управление ресурсами. Для этого она обеспечивает управление процессом (задачей), памятью, а также управление вводом/выводом в целом. Операционная система реального времени должна выполнять эти задачи со строгими временными ограничениями. Операционная система реального времени, как правило, небольшого размера и легко размещается в системах со встроенными элементами, а также должна быть быстрой при минимальных потерях;
- *язык программирования реального времени*: язык программирования реального времени обеспечивает основные схемы, такие как межзадачное общение и синхронизация, обработка ошибок и планирование задач в реальном времени. Заметим, что эти функции обеспечиваются также и операционной системой реального времени. Следовательно, язык программирования реального времени может быть использован для обработки в реальном времени альтернативно операционной системе. На самом низком уровне для доступа к регистрам и другому аппаратному обеспечению процессора может использоваться ассемблер. Однако язык ассемблера подвержен ошибкам и не переносим на другой процессор. Язык Си широко используется в режиме реального времени системы, поскольку обеспечивает доступ к оборудованию и имеет простой интерфейс ввода/вывода. Вместе с тем для систем реального времени, с целью упрощения программирования, разработаны другие языки реального времени, такие как Ada и Java;
- *сеть реального времени*: компьютерная сеть обеспечивает передачу данных между различными вычислительными устройствами. Сеть реального времени должна обеспечивать надежную доставку сообщений в заданные



сроки. Ключ к работе сети в реальном времени – это надежная и своевременная передача сообщений. Протокол общения в реальном времени обеспечивает своевременную и гарантированную доставку сообщений по сети.

Следует отличать систему реального времени от системы, которая выглядит похожей на систему реального времени, но таковой на самом деле не является. Например, *онлайновая система* взаимодействия с пользователем не считается системой реального времени, поскольку решение ею задач не имеет определенных сроков. Кроме того, если система должна просто реагировать быстро, это не обязательно означает, что она является системой реального времени.

*Система со встроенными элементами* характеризуется тем, что в состав системы включен вычислительный элемент. Такие системы не являются общими вычислительными системами, в которых компьютер может быть динамически запрограммирован для решения различных задач. Такие системы, например, как микроволновая печь, имеют специальное назначение. Система со встроенными элементами тесно связана с внешним физическим миром. Примерами систем со встроенными элементами могут служить домашние приборы бытового назначения, используемые для улучшения качества жизни. Системы со встроенными элементами фактически составляют большинство компьютерных систем. Система со встроенными элементами может быть или не быть системой с режимом реального времени в зависимости от назначения приложения. Система со встроенными элементами не является системой реального времени, если не имеет временных границ. Такой системой, например, является MP3-плеер. Вместе с тем значительный процент приложений с реальным временем имеет встроенные элементы. Система реального времени со встроенными элементами разрабатывается для реализации определенной функции и работает с ограничениями по реальному времени. Наша терминология в этой книге относится к системам реального времени в целом, включая ее встроенные элементы.

Системы реального времени можно классифицировать, используя различные подходы; базовым условием является наличие синхронизации и взаимодействия между различными компонентами системы. Соответственно, система реального времени может быть управляема временем, в которой каждому действию определено специфическое время; внешние входы определяют поведение системы и ее взаимодействие, и пользователь может изменять режимы операций. Во многих случаях система реального времени работает как комбинация всех режимов подобного рода.

### 1.3. БАЗОВАЯ АРХИТЕКТУРА

Система реального времени состоит из компонентов аппаратного обеспечения, показанных на рис. 1.1. Их можно детализировать следующим образом.

- *Датчики*: датчик преобразует физический параметр в обработанный электрический сигнал. Например, термопара – это датчик температуры, который преобразует тепло в электрический сигнал.

- *Элемент обработки (Processing Element, PE)*: элемент обработки в реальном времени или компьютер реального времени, который принимает цифровые данные, предоставляемые интерфейсным узлом ввода, обрабатывает их и выбирает некоторую функцию вывода для выполнения.
- *Приводы*: привод вводит электрический сигнал выходного интерфейса обрабатываемого элемента и активирует физическое действие. Примером привода может являться реле, которое при активации сигналом замыкает/размыкает контакты, осуществляя переключение.
- *Интерфейс ввода*: интерфейс ввода в компьютер реального времени преобразовывает и обрабатывает электрический сигнал в цифровую форму для обработки. Датчики могут иметь необходимые встроенные в них преобразователи.
- *Интерфейс вывода*: двоичный вывод из компьютера реального времени, обработанный и преобразованный в форму сигнала, необходимую для привода или других устройств вывода в выходной интерфейс.

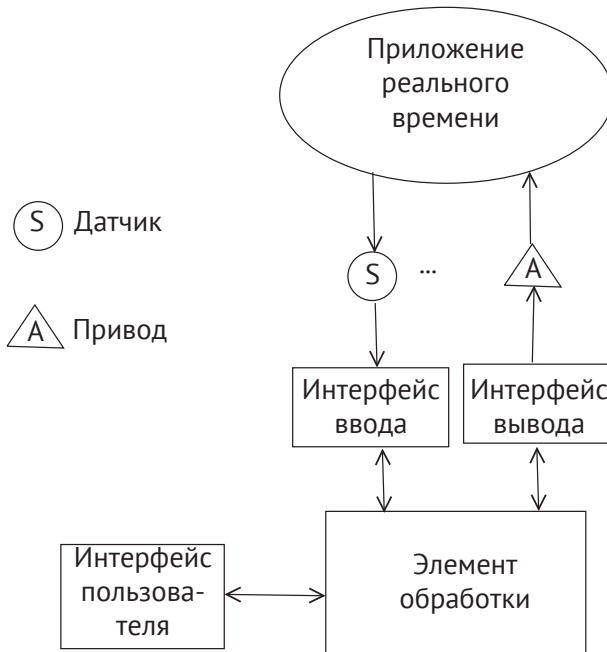


Рис. 1.1. Архитектура типичной системы реального времени

## 1.4. ХАРАКТЕРИСТИКИ СИСТЕМ РЕАЛЬНОГО ВРЕМЕНИ

Общие характеристики систем реального времени:

- *соблюдение сроков*: для предотвращения гибели людей и имущества должны быть выдержаны в реальном времени сроки всех важных задач;
- *большой размер*: системы реального времени обычно большие и сложные в части аппаратного обеспечения. Однако особенно это относится к величине используемого программного обеспечения. Даже небольшая система реального времени может потребовать для реализации сотни тысяч строк кода;
- *предсказуемость*: мы должны быть в состоянии предсказать время ответа для наихудшего варианта реализации и знать, будут ли соблюдены системой реального времени заданные сроки выполнения всех предписанных задач. Предсказуемость предполагает обычно теоретическое доказательство того, что все сроки в реализуемой системе реального времени будут соблюдаться;
- *безопасность и надежность*: системы реального времени работают и управляют процессами в таких средах, как атомные станции, где безопасности и надежности уделяется предельное внимание. Небезопасные и ненадежные системы подвержены ошибкам, которые могут привести ущерб или к потере здоровья людей и потере имущества;
- *отказоустойчивость*: считается, что *сбой* происходит, когда компонент аппаратного или программного обеспечения компьютерной системы выходит из строя, и система перестает вести себя в соответствии с ее спецификацией. *Отказ* системы реального времени является результатом сбоя, что может привести к потере жизни людей и собственности. Сбой может быть постоянным, вызывая либо временный выход системы из строя, который через некоторое время может исчезнуть, либо постоянный. Отказоустойчивость – это способность системы противостоять сбоям и продолжать правильно функционировать при наличии сбоев;
- *параллелизм*: физическая среда, управляемая компьютером реального времени, в которой обычно реализуются параллельные события. Компьютер реального времени должен быть в состоянии справиться с параллельной операцией, используя возможности параллельного системного программного обеспечения или распределенного оборудования.

## 1.5. КЛАССИФИКАЦИЯ СИСТЕМ РЕАЛЬНОГО ВРЕМЕНИ

Мы можем указать типы систем реального времени по временным ограничениям и типу процессов в системе в зависимости от полученного ввода. Можно дать следующую классификацию при рассмотрении конечных сроков:

- *жесткое реальное время*: несоблюдение сроков может привести к катастрофическим последствиям – потере имущества или жизни. Авиаци-

онные системы маневрирования, химические станции и атомные электростанции являются примерами таких систем. Все сроки выполнения задачи в жесткой системе реального времени устанавливаются однозначно. Полезность системы, если заданный срок пропущен, равна нулю, как показано на рис. 1.2а;

- *мягкое реальное время*: пропуск мягких сроков не критичен, однако когда это происходит, качество сервиса системы ухудшается. Мягкая система реального времени выполняет задачи с мягким, не жестко установленным сроком. Примерами мягких систем реального времени являются мультимедийные системы и системы бронирования авиабилетов. Полезность системы, когда крайний срок пропущен, ухудшается и обнуляется, однако, не сразу, как показано на рис. 1.2б;
- *твердое реальное время*: эти системы в основном представляют собой мягкий тип системы реального времени, полезность которых при пропуске срока отсутствует, но он допускается. В этих системах полезность сразу сводится к нулю, как показано на рис. 1.2а, но система допускает такие пропуски сроков.

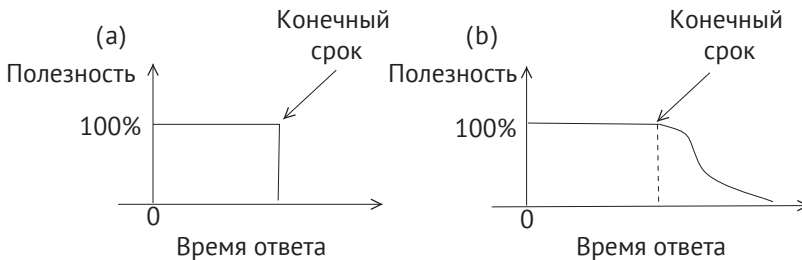


Рис. 1.2. Полезность (а) в жестких, твердых и (б) мягких системах реального времени

Система реального времени может сочетать задачи, выполняемые жестко, мягко или твердо в реальном времени. Процессы в системе могут быть *статическими* с заранее известными детерминированными характеристиками, тогда как характеристики в динамической системе реального времени меняются. Другое отличие заключается в том, как выполняется обработка:

- *системы реального времени, управляемые событиями*: когда и как выполняется обработка в реальном времени в этих системах, определяют внешние события. Система, управляемая событиями, обрабатывает асинхронные входы, активируемые событиями;
- *системы реального времени, управляемые временем*: обработка в этих системах выполняется в четко определенной временной базе, а действия, которые необходимо предпринять, выполняются в строго определенные моменты времени. Управляемая временем система реального времени имеет синхронные входы.

Например, мониторинг влажности воздуха с помощью датчика, измеряющего влажность каждые  $t$  единиц времени, является системой, управляемой временем, тогда как система, которая активируется асинхронным внешним событием, – это система, управляемая событиями. Обычно в типичной системе реального времени мы имеем сочетание этих режимов. Например, контроль температуры печи может быть выполнен путем измерения температуры через равные промежутки времени (управление событиями) и выключением обогревателя при открытии дверцы печи (по событию).

Распределенная система состоит из автономных компьютеров, каждый из которых способен функционировать в одиночку. Эти компьютеры связаны сетью и взаимодействуют, выполняя глобальную задачу и делясь ресурсами. Распределенная система реального времени является распределенной вычислительной системой, которая работает с ограничениями по времени.

Распределенные системы обеспечивают эффективное совместное использование ресурсов и избыточность. Предоставляет удобство также выполнение вычислений на нескольких компонентах, поскольку при этом возможно проведение быстрой локальной обработки. Еще одной причиной, по которой использование распределенной системы реального времени является целесообразным, является то, что различные подсистемы разных производителей могут быть связаны между собой с использованием стандартных протоколов связи. Автомобиль – это распределенная система реального времени, так как данные считываются из многих датчиков и передаются по сети на различные узлы для обработки в реальном времени. Еще одним преимуществом использования распределенных систем реального времени является повышенная надежность, поскольку сбой узла не может значительно повлиять на работу всей системы, если заранее приняты меры предосторожности.

## 1.6. ПРИМЕР СИСТЕМЫ: КОНВЕЙЕР БУТЫЛОК

Давайте рассмотрим очень простую систему управления процессом реального времени, наполняющую молоком бутылки. Молочные бутылки движутся на конвейерной ленте, и есть три машины: моечная машина, моющая бутылки, наполнитель, который наполняет бутылку молоком, и упаковщик, который устанавливает крышку на заполненную молоком бутылку, как показано на рис. 1.3.



Рис. 1.3. Архитектура типичной системы реального времени

Каждая машина подключена к компьютерной системе реального времени с датчиками и исполнительными механизмами (приводами). Компьютерная система должна выполнять три задачи в следующей последовательности:

- 1) *мойка*: о появлении грязной бутылки на конвейерной ленте сигнализирует датчик. Лента останавливается, и мойщик активируется на время  $t_w$  секунд, затем активируется лента. Мы видим здесь операцию, основанную на событии (поступление бутылки);
- 2) *наполнение*: прибытие бутылки фиксируется, и активированный наполнитель осуществляет наполнение. Уровень молока в бутылке контролируется датчиком, и наполнитель останавливается, когда желаемый уровень достигнут;
- 3) *упаковка*: бутылка прибывает к этому механизму, крышка помещается на горлышко бутылки, и она упаковывается.

Здесь мы видим обработку, преимущественно управляемую событиями. Тем не менее при определенном времени весь процесс может быть выполнен за счет периодического выполнения процессов мойки, розлива и укупорки, при условии что всегда присутствуют пустые бутылки. Обратите внимание, что весь процесс идет медленно, и простой компьютер оставляет значительное время для выполнения другой обработки.

## 1.7. ОБЗОР КНИГИ

Мы имеем в книге следующие четыре части.

1. *Введение*: эта часть книги служит введением в системы реального времени с главами по архитектуре реального времени и распределенным системам реального времени.
2. *Системное программное обеспечение*: операционная система реального времени находится в центре внимания данной части, где мы опишем основные понятия, такие как задачи, память и управление вводом/выводом. Мы также предоставляем подробное пошаговое построение ядра операционной системы в режиме реального времени, которое используется для тестирования различных реализаций более высокого уровня в последующих главах.
3. *Планирование и управление ресурсами*: планирование задач таким образом, чтобы были выполнены конечные сроки, является фундаментальной функцией любой системы реального времени. Сначала в этой части книги мы опишем независимое периодическое и аperiodическое планирование, а затем зависимое планирование с управлением ресурсами и распределенным планированием.
4. *Проектирование приложения*: эта часть является фундаментальной с точки зрения помощи разработчикам программного обеспечения систем реального времени. Мы описываем процедуру проектирования от методов проектирования высокого уровня до низкоуровневого проектирования и реализации. Мы также представляем обзор языков программирования и методов отказоустойчивости в режиме реального времени. Эта часть завершается подробным примером, в котором мы рассматриваем реализацию методов на основе реального приложения.

## 1.8. КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Какова связь между системой реального времени и системой со встроенными элементами? Все ли системы со встроенными элементами работают в режиме реального времени? Обсудите, приводя примеры.
2. Каковы основные компоненты системы реального времени?
3. Что такое датчик и исполнительный механизм в системе реального времени? Приведите пример каждого из них.
4. Какой тип оборудования вы ожидаете встретить во входном интерфейсе компьютерной системы реального времени?
5. Что такое жесткий режим реального времени, мягкий и твердый режимы реального времени в системах реального времени?
6. Приведите пример каждой из управляемых событиями и управляемых временем систем реального времени.
7. Каковы основные характеристики распределенной системы реального времени?

## 1.9. ПРИМЕЧАНИЯ К ГЛАВЕ

Мы рассмотрели основные компоненты компьютерной системы реального времени, ее основную архитектуру и различные классификации. Системы реального времени могут иметь жесткие, мягкие и твердые сроки выполнения заданий. С другой стороны, обработка может быть выполнена как управляемая событиями, так и управляемая временем. В режиме реального времени компьютер может быть как встроен в среду, которой он должен управлять в системе со встроенными элементами, так и быть распределен по сети реального времени. Как правило, в режиме реального времени система имеет комбинацию этих свойств. Общие понятия о реальном времени системы описаны в [2], а языки программирования в реальном времени содержатся в [1].

## СПРАВОЧНЫЕ МАТЕРИАЛЫ

1. Burns A., Wellings A. (2001) Real time systems and programming languages: Ada 95, Real-time Java and real-time C/POSIX, 3rd edn. Addison-Wesley.
2. Laplante P. A., Ovaska S. J. (2011) Real-time systems design and analysis: tools for the practitioner, 4th edn. Wiley.