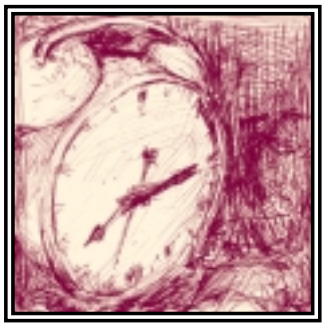

Программное обеспечение реального времени фирмы On Time INFORMATIK GmbH

Александр Локотков

RTKernel 4.5

Многозадачное ядро реального времени, профессиональное инструментальное средство разработки 16-разрядных приложений реального времени и реализации многозадачности в среде MS-DOS

Общие сведения

RTKernel представляет собой мощную многозадачную систему реального времени, предназначенную для разработки программного обеспечения, исполняющегося на IBM PC совместимых контроллерах с открытой архитектурой в среде MS-DOS.

RTKernel является библиотекой или модулем, который может быть скомпонован с прикладной программой. В состав RTKernel входят многочисленные функции и процедуры управления задачами, семафорами и прерываниями, а также средства обмена данными между задачами. Запуск на исполнение задач RTKernel производится из единственной программы, которая содержит ядро, необходимые драйверы и все задачи. Данная программа может выполняться на любой вычислительной системе, содержащей операционную систему MS-DOS. Хотя программа, в которой используется RTKernel, и обладает свойствами, характерными для мультизадачных систем реального времени, она по-прежнему остается приложением DOS.

Основные характеристики

- Количество задач, выполняемых под управлением RTKernel, ограничивается общим объемом оперативной памяти. Для каждой задачи RTKernel дополнительно требуется около 1 кбайт памяти.
- Время переключения задачи не зависит от количества задач и составляет около 6 мкс (80486/33 МГц).
- Количество приоритетов задач – 64.
- Виды планирования: коллективное (Cooperative), с вытеснением (Preemptive), с выделением квантов времени (Time-Slicing).
- Переключение задач по событию или прерыванию.
- Возможность активизации задачи при возникновении аппаратного прерывания.
- Возможность изменения периода поступления прерываний от таймера в диапазоне 0,1...55,0 мс.
- Возможность измерения временных интервалов с разрешением 1 мкс.
- Поддержка арифметического сопроцессора и его программной эмуляции.
- Семафоры: двоичные, счетные, ресурсов.
- Обмен данными между задачами с использованием очередей сообщений.
- Непосредственный обмен данными между задачами с использованием механизма передачи сообщений.
- Коммуникационный драйвер обслуживания последовательных портов в количестве до 36 с использованием прерываний.
- Поддержка аппаратного буфера универсальных асинхронных приемопередатчиков (УАПП) семейства 16C550.
- Драйверы для работы с таймером, видеоподсистемой, клавиатурой, принтером и локальными вычислительными сетями (ЛВС) Novell с протоколом IPX.
- Использование простоев клавиатуры и дисковых накопителей для предоставления процессора другим задачам.
- Отсутствие проблем повторной входимости, свойственных MS-DOS.

- Возможность создания приложений RTKernel в виде резидентных программ.
- Возможность запуска других DOS-программ, в том числе Windows 3.0/3.1, из приложения RTKernel.
- Удобство отладки приложений путем использования встроенной возможности компиляции с добавлением отладочной информации, необходимой для Turbo Debugger или CodeView.
- Возможность создания приложений, загружаемых из ПЗУ.
- Возможность поставки версии с полным комплектом исходных текстов ядра.
- Отсутствие ограничений на количество разрабатываемых приложений.

Структура и принцип функционирования

Задачи RTKernel

Задачи RTKernel являются обычными Pascal-процедурами или Си-функциями без параметров, методы создания которых ничем не отличаются от традиционных. При старте программы на исполнение запускается только одна функция main. Несмотря на то что в состав программы входит мультизадачная система реального времени, ее выполнение происходит последовательно, как в традиционных приложениях DOS. После обращения к ядру путем вызова функции RTKernel(RTKernelInit(MainPriority)) на основе функции main формируется главная задача приложения. Далее главная задача с помощью функции RTKCreateTask преобразует функции (процедуры) программы, которые должны исполняться одновременно, в задачи RTKernel. Простейшая структура приложения RTKernel и результат его исполнения приведены на рис. 1.

Каждой задаче присваивается приоритет, значение которого может быть установлено в диапазоне от 1 до 64, а также выделяется собственная область стека. Самый высокий приоритет соответствует значению 64. Самый низкий – 1. Все локальные переменные (за исключением статических), объявленные в пределах преобразованных в задачи функций (процедур), помещаются в собственные области стека каждой задачи. Это же относится и к функциям, вызываемым из задач RTKernel. Поскольку для каждой задачи формируется собственный стек, имеется возможность одновременного выполнения

```
#include "RTKernel.h"
#define MainPriority        3       //Приоритет главной задачи

TaskHandle      HandleA;            //Определение дескриптора задачи A
Semaphore       S;                  //Определение семафора

void TaskA(void)
{
        printf("Задача A:ожидание семафора S\n");
        RTKWait(S);
        printf("\n3адача A:возобновление исполнения\n");
}

void main(void)
{
        printf("\n");
        //Инициализация RTKernel, создание главной задачи и "пустой" задачи
        RTKernelInit(MainPriority);
        //Создание счетного семафора
        S = RTKCreateSemaphore(Counting, 0, "Semaphore S");
        //Преобразование функции A в задачу
        printf("Main : создание задачи A\n");
        HandleA = RTKCreateTask(TaskA, MainPriority + 1, 1024, "Task-A");
        //Освобождение семафора
        printf("Main : освобождение семафора S\n");
        RTKSignal(S);
        printf("Main : завершение приложения\n");
}
```

Результат:

```
Main : создание задачи A
Задача A:ожидание семафора S
Main : освобождение семафора S

Задача A:возобновление исполнения
Main : завершение приложения
```

Рис. 1. Законченная программа RTKernel с двумя задачами. Применен способ активизации с использованием счетного семафора

разными задачами одних и тех же участков кода. Структура приложения RTKernel показана на рис. 2.

Взаимодействие задач

Термин «взаимодействие задач» относится ко всем способам обмена информацией между задачами и методам синхронизации. RTKernel представляет три различных механизма взаимодействия.

Семафоры

Семафорные примитивы являются одним из наиболее популярных средств синхронизации параллельно исполняющихся процессов и могут рассматриваться в качестве счетчиков событий, значение которых никогда не может быть отрицательным. Основным свойством семафора считается неделимость (непрерываемость) выполняемых по отношению к нему операций. Семафоры позволяют задачам обмениваться сигналами для активизации и приостановки исполнения. RTKernel поддерживает двоичные и счетные семафоры, а также семафоры ресурсов. Семафоры ресурсов предназначены для эффективного управления доступом к ресурсам. Семафор ресурса гарантирует, что приоритет задачи, занимающей ресурс, будет всегда наивысшим среди приоритетов других задач, ожидающих данный ресурс. Данное свойство ресурсных семафоров, известное под названием «наследование приоритета», делает невозможным блокирование текущей задачи менее приоритетной задачей.

Очереди сообщений

Очередь сообщений представляет собой буфер данных, в который может быть помещено фиксированное количество записей (сообщений). Очереди сообщений позволяют осуществлять обмен данными между задачами. Количество записей, помещаемых в очередь сообщений, устанавливается при ее создании. Очереди сообщений особенно удобны для буферизации данных при обмене между задачами или процедурами обслуживания прерываний.

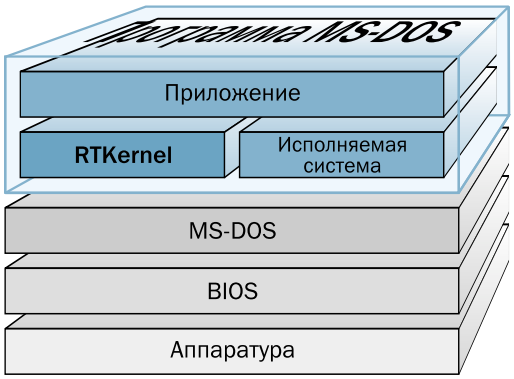

Рис. 2. Структура приложения RTKernel

Передача сообщений

Механизм передачи сообщений обеспечивает возможность обмена данными между двумя задачами без использования сигналов или очередей сообщений. Копирование данных производится непосредственно из одной задачи в другую. Данный механизм обмена может рассматриваться как передача информации через «почтовый ящик» нулевой длины. Передача сообщений позволяет осуществить наиболее тесное взаимодействие вовлекаемых в нее задач, поскольку их синхронизация осуществляется ядром RTKernel.

Планировщик

Планировщик RTKernel предназначен для управления состояниями задач. Основной функцией планировщика является выявление задачи, которая должна исполняться в текущий момент времени.

Любая задача RTKernel всегда пребывает в одном из следующих состояний (рис. 3).

Current (состояние исполнения)

Активная в настоящий момент времени задача характеризуется состоянием Current. По крайней мере одна задача под управлением RTKernel должна пребывать в указанном состоянии.

Ready (состояние готовности)

Все задачи, готовые к исполнению, находятся в состоянии Ready. В текущий момент времени одна из данных задач переходит в состояние Current (состояние исполнения). Задачи в состоянии Ready имеют одинаковый или более низкий приоритет по сравнению с текущей активной задачей.

Suspended (состояние приостановки)

Задачи, исполнение которых явно приостановлено обращением к ядру посредством вызова RTKSuspend. Данные задачи могут быть переведены в состояние Ready только с помощью вызова функции RTKResume.

Blocked (состояние блокировки)

Задачи в указанном состоянии не могут исполняться, поскольку ожидают какого-либо события, например, освобождения семафора (ресурса) или прихода сообщения в очередь. Перевод данных задач в состояние Ready возможен только по инициативе другой задачи или обработчика прерывания.

Delaying (состояние задержки)

Задачи в данном состоянии блокируют себя на определенный интервал времени. По истечении заданного интервала обработчиком прерывания RTKernel по системному таймеру производится перевод указанных задач в состояние Ready.

Timed (состояние ожидания события с синхронизацией)

Состояние ожидания задачей события в течение заданного интервала времени. Перевод задачи в состояние Ready производится по истечении установленного интервала либо при наступлении ожидаемого события.

При инициализации RTKernel создаются две задачи: основная (Main Task) и «пустая» (Idle Task). Пустая задача, имеющая нулевой (самый низкий) приоритет, необходима для функционирования планировщика, поскольку условием его работы является наличие хотя бы одной задачи, находящейся в состоянии Ready.

Планирование в RTKernel происходит согласно следующим правилам.

1.Из всех задач, находящихся в состоянии Ready, в активное состояние переводится задача с наивысшим приоритетом.

2. Если в состоянии Ready пребывают несколько задач, имеющих одинаковый приоритет, в активное состояние переводится задача, не исполнявшаяся в течение наиболее длительного интервала времени.

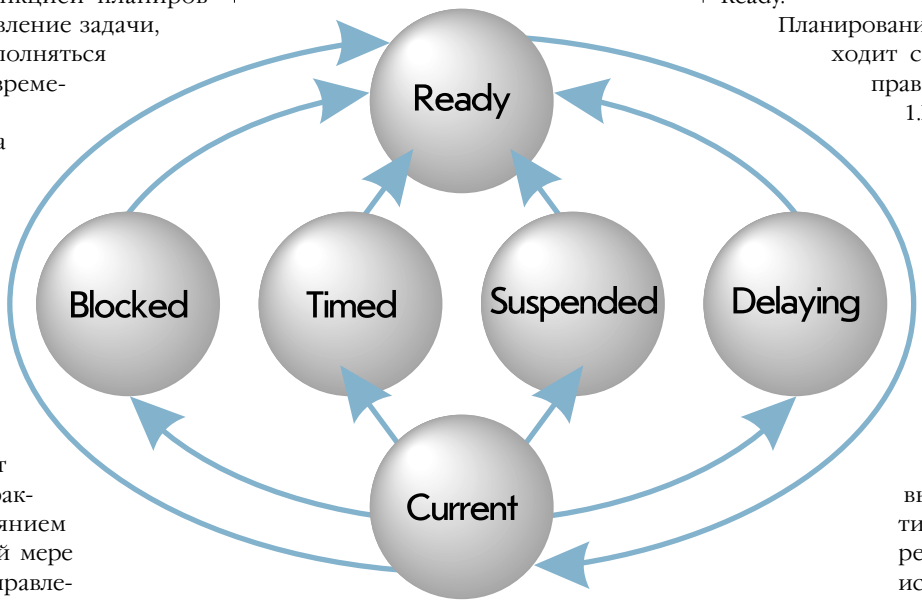

Рис. 3. Состояния задачи RTKernel

3. Если несколько задач находятся в состоянии ожидания события, порядок их активизации при наступлении события осуществляется в соответствии с порядком убывания их приоритетов.
4. За исключением планирования исполнения задач с выделением каждой заданного кванта времени (Time-Slicing Task Switches), переключение задачи производится только в том случае, когда возникают предпосылки нарушения правила 1, что позволяет минимизировать количество переключений задач.

Поскольку планировщик RTKernel реализован в строгом соответствии с данными правилами, исполнение задач, составляющих приложение RTKernel, всегда предсказуемо и абсолютно управляемо.

При создании приложений может применяться как метод коллективного (Cooperative) планирования, так и планирования с вытеснением (Preemptive).

Коллективное планирование (без вытеснения) предполагает возможность переключения задачи только после ее обращения к ядру. Таким образом обеспечивается «джентльменское» поведение параллельно исполняемых задач по отношению друг к другу, а принятие решения о предоставлении процессора какой-либо из них возлагается на ядро. При планировании с вытеснением переключение задачи может быть произведено непосредственно обработчиком прерывания по какому-либо событию без обращения к ядру.

В случае, если в приложении необходимо использовать различные алгоритмы планирования, имеется возможность временного блокирования встроенного планировщика RTKernel.

Кроме того, возможно применение метода планирования с выделением параллельно выполняющимся задачам фиксированных интервалов времени (Time-Slicing).

Прерывания

Процедуры обслуживания прерываний могут приостанавливать и активизировать исполнение задач. Данный факт позволяет квалифицировать RTKernel как систему «реального времени». Создание обработчиков прерываний производится обычным образом, принятым в средах программирования Си и Pascal, но с использованием специальных функций RTKernel получения вектора, установки вектора и т. п. Обработчики прерываний могут совершать обмен данными или сигналами с задачами путем использования семафоров или очередей сообщений. Операции по отношению к семафорам или очередям сообщений, в свою очередь, при необходимости позволяют осуществлять переключение задач.

RTKernel, DOS и Windows

MS-DOS является однозадачной однопользовательской операционной системой. Таким образом, мультизадачные системы, подобные RTKernel, должны обеспечивать решение проблемы повторной входимости, присущей DOS, а также проблемы совместного доступа параллельно выполняющихся задач к файлам. В состав RTKernel входят средства защиты DOS, обеспечивающие возможность параллельно выполняющимся процессам использовать системные вызовы INT 20H, 21H, 25H, 26H и 27H.

Программа RTKernel может быть загружена в качестве резидентной (TSR – terminate and stay resident), после чего имеется возможность запуска на исполнение Windows 3.0/3.1 или приложения под управлением DOS-расширителя. Запуск Windows в реальном (Real), стандартном (Standard) или защищенном (386-Enhanced) режимах осуществляется путем использования функции RTKExec, назначение которой аналогично функции DOS Exec. В дальнейшем Windows и все приложения, запущенные под управлением Windows, рассматриваются RTKernel как одна задача. Другие задачи, входящие в состав RTKernel-приложения, продолжают выполняться в соответствии с требованиями, предъявляемыми к системам реального времени, и с возможностью неограниченного использования функций дискового ввода-вывода, а также взаимодействия с приложениями Windows посредством программных прерываний или специального модуля IPC (Inter-Process Communication – модуль межпроцессного взаимодействия). Таким образом, имеется способ создания сложных приложений реального времени, использующих интерфейс Windows (рис. 4). Вся работа в режиме реального времени выполняется под управлением RTKernel, тогда как интерфейс пользователя реализуется средствами Windows.

Средства отладки приложений

Поскольку RTKernel полностью интегрирована с исполнительной средой поддерживаемых компиляторов, для отладки приложений RTKernel могут быть использованы широко распространенные отладчики типа CodeView или Turbo Debugger.

Для предоставления наибольших удобств разработчикам программного обеспечения при выполнении отладки в комплект поставки RTKernel, помимо стандартной версии библиотеки, оптимизированной для обеспечения максимальной производительности и минимального размера кода, входит так называемая отладочная версия (Debug

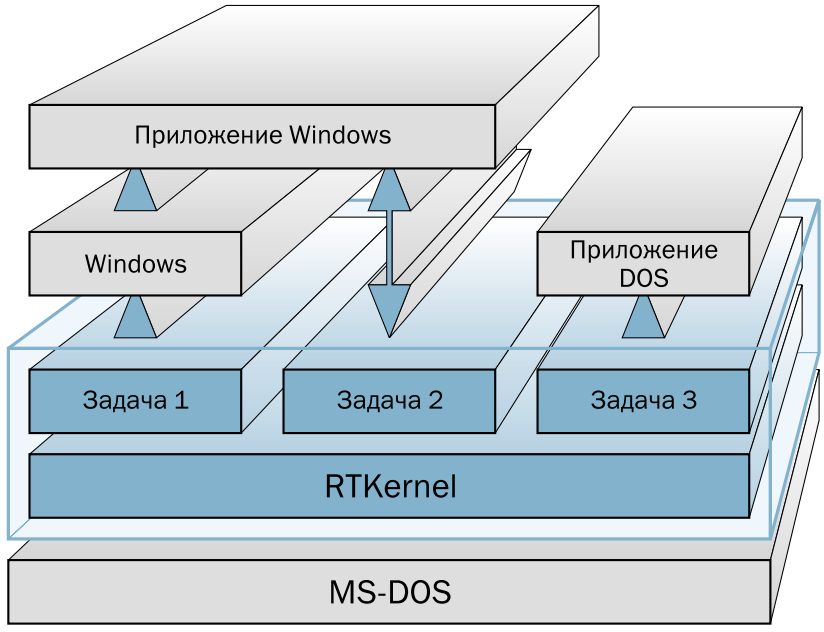

Рис. 4. Взаимодействие Windows и RTKernel

Динамические характеристики

Время, мкс			Выполняемая операция RTKernel
486-33 МГц	*386-20 МГц*	*286-12 МГц*	
6	27	58	Циклическое переключение задачи
10	42	89	Переключение задачи по сигналу
3	13	22	Освобождение семафора (операция V(S))
2	8	13	Занятие семафора (операция P(S))
14	65	115	Активизация задачи при освобождении семафора другой задачей
7	20	35	Помещение данных в очередь сообщений
7	21	39	Извлечение данных из очереди
18	63	114	Непосредственная передача данных между задачами (Message-Passing)
24	77	150	Передача данных между задачами с использованием очереди сообщений (Task-to-Mailbox-to-Task)

Version), которая содержит дополнительный код для проверки корректности используемых параметров и функций, а также для вывода предупреждающих сообщений в процессе исполнения приложения. При использовании отладочной версии под управлением упомянутых отладчиков программисту предоставляется обширный перечень средств отладки приложения. Например, в любой точке исходного кода в процессе пошаговой отладки возможно вывести на экран монитора список всех занятых ресурсов. Особенно мощным отладочным средством является трассировщик ядра (Kernel Tracer), который позволяет вести протокол всех происшедших событий, связанных с ядром или приложением, для последующего анализа.

Драйверы устройств

RTKernel является аппаратнонезависимой системой, осуществляющей только управление выполнением задач. Однако в приложениях реального времени, как правило, интенсивно используется обмен данными с периферийными устройствами. Хотя приложения RTKernel могут без каких-либо ограничений осуществлять указанный обмен через стандартные драйверы устройств, созданные для DOS, наличие специально оптимизированных для функционирования в многозадачной среде драйверов аппаратуры позволит существенно улучшить производительность системы. В комплект поставки RTKernel входит приведенный далее перечень драйверов аппаратуры с полными исходными текстами.

Timer

Позволяет осуществлять измерение любого количества независимых временных интервалов с разрешением около 0,8 мкс. Длительность интервала может составлять до 3,7 лет. Кроме того, данный модуль предоставляет возможность изменения частоты системного таймера, которая обычно равна 18,2 Гц.

RTKeybrd

Устанавливает собственную процедуру обслуживания прерывания от клавиатуры, позволяя тем самым задачам, ожидающим ввода с клавиатуры, не занимать для этой цели процессор.

KillKey

Дает возможность избежать нежелательных последствий, к которым может привести нажатие клавиш <Pause>, <PrintScreen> и т. п. во время исполнения приложения реального времени.

RTTextIO

Содержит средства, обеспечивающие совместное использование видеоподсистемы различными задачами путем предоставления каждой собственного окна ввода/вывода.

Spooler

Обеспечивает возможность вывода файлов на печатающее устройство (параллельный принтер) в фоновом режиме.

CPUMoni

Позволяет критичным по времени исполнения приложениям реального времени контролировать степень загруженности процессора.

RTCom

Содержит полную реализацию взаимодействия с последовательными портами. Прием и передача данных через последовательные порты COM1–COM4 может осуществляться как в режиме опроса, так и по прерыванию. Автоматически распознает УАПП семейства 16C550 и использует его встроенный буфер FIFO.

RTIPX

Содержит полную реализацию протокола IPX фирмы Novell. Может использоваться на любой рабочей станции сети, в программной среде которой установлен стандартный драйвер IPX. Длина пакета передаваемых данных может составлять до 64 кбайт. В отличие от протокола Novell IPX обеспечивает гарантированную выдачу ответа на принятое сообщение. Обработка тайм-аутов и формирование запросов повторной передачи при обнаружении коммуникационных ошибок осуществляются непосредственно из приложения.

IPC

Облегчает реализацию взаимодействия между разными DOS-приложениями путем использования мультиплексного прерывания DOS. В основном это относится к резидентным программам, передающим данные основному исполняемому приложению. Кроме того, посредством IPC может быть организован обмен данными между приложением Windows и резидентной программой DOS.

RTKernel-32

Многозадачное ядро реального времени для 32-разрядных встраиваемых систем

RTKernel-32 представляет собой мощную мультизадачную систему реального времени, предназначенную для разработки встраиваемых приложений, использующих 32-разрядное линейное адресное пространство памяти.

Время, мкс			Выполняемая операция RTKernel-32
386EX-20 МГц	*486-33 МГц*	*Pentium-120 МГц*	
43	5	0,73	Циклическое переключение задачи
79	10	1,61	Переключение задачи по сигналу
37	6	1,18	Освобождение семафора (операция V(S))
25	4	1,24	Занятие семафора (операция P(S))
100	13	3,13	Активизация задачи при освобождении семафора другой задачей
31	12	3,95	Помещение данных в очередь сообщений
30	10	2,77	Извлечение данных из очереди
96	12	2,50	Непосредственная передача данных между задачами (Message-Passing)
107	18	4,03	Передача данных между задачами с использованием очереди сообщений (Task-to-Mailbox-to-Task)

RTKernel-32 является библиотекой или модулем, который может быть скомпонован с прикладной программой. В состав RTKernel-32 входят многочисленные функции и процедуры управления задачами, семафорами и прерываниями, а также средства обмена между задачами. Запуск на исполнение задач RTKernel-32 производится из единственной программы, которая содержит ядро, необходимые драйверы и все задачи. Исполняемый модуль приложения для запуска в составе встраиваемой системы, оснащенной процессором любого типа, поддерживающим Intel совместимый 32-разрядный защищенный режим, может быть подготовлен с помощью кросс-системы, подобной RTTarget-32, поставляемой фирмой On Time.

- Количество задач, выполняемых под управлением RTKernel-32, ограничивается общим объемом оперативной памяти. Для каждой задачи RTKernel дополнительно требуется около 1 кбайт памяти.
- Время переключения задачи не зависит от количества задач и составляет около 5 мкс (80486 33 МГц).
- Количество приоритетов задач 16.
- Виды планирования: коллективное (Cooperative), с вытеснением (Preemptive), с выделением квантов времени (Time-Slicing).
- Переключение задач по событию или прерыванию.
- Возможность активизации задачи при возникновении аппаратного прерывания.
- Возможность измерения временных интервалов с высоким разрешением.
- Поддержка арифметического сопроцессора и его программной эмуляции.
- Пять типов семафорных примитивов.
- Обмен данными между задачами с использованием очередей сообщений.

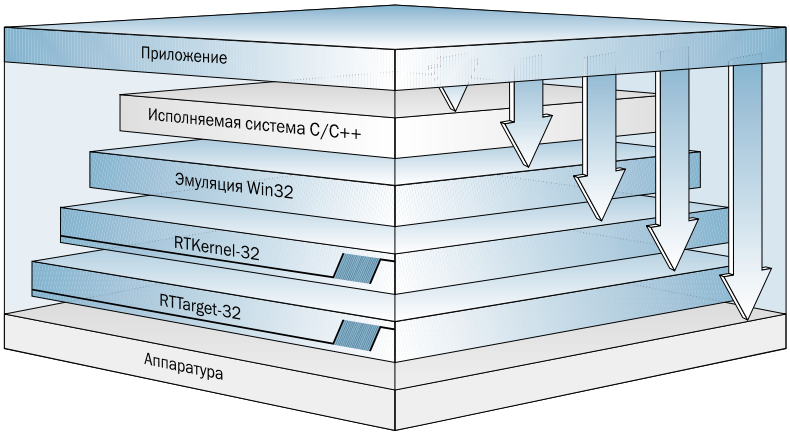

Рис. 5. Структура приложения RTKernel-32

- Непосредственный обмен данными между задачами с использованием механизма передачи сообщений.
- Управление памятью в режиме реального времени (пулы памяти).
- Коммуникационный драйвер обслуживания последовательных портов в количестве до 36 с использованием прерываний.
- Совместимость с RTKernel-C для DOS.
- Совместимость с Win32 API
- Поддержка RTTarget-32 и других кросс-средств.
- Возможность запуска в составе целевых систем на базе процессоров, совместимых с i386.
- Возможность создания приложений, загружаемых из ПЗУ.
- Возможность поставки версии с полным комплектом исходных текстов ядра.
- Отсутствие ограничений на количество разрабатываемых приложений.

Структура приложения RTKernel-32 показана на рис. 5.

Для обеспечения возможности переноса в среду RTKernel-32 существующих многозадачных приложений, созданных для функционирования под управлением Windows NT или Windows 95, в ее состав включен практически полный набор функций, образующих интерфейс Win32.

RTTarget-32

Инструментальное кросс-средство для переноса 32-разрядных приложений во встраиваемые системы

RTTarget-32 является кросс-средством, которое позволяет переносить и запускать на исполнение без участия операционной системы стандартные приложения Windows NT, функционирующие в режиме консоли, на аппаратных средствах, имеющих в своем составе Intel-совместимый 32-разрядный процессор. Кросс-отладка при этом осуществляется с помощью стандартного отладчика Windows NT.

Разработка программ, подлежащих запуску совместно с RTTarget-32, произ-

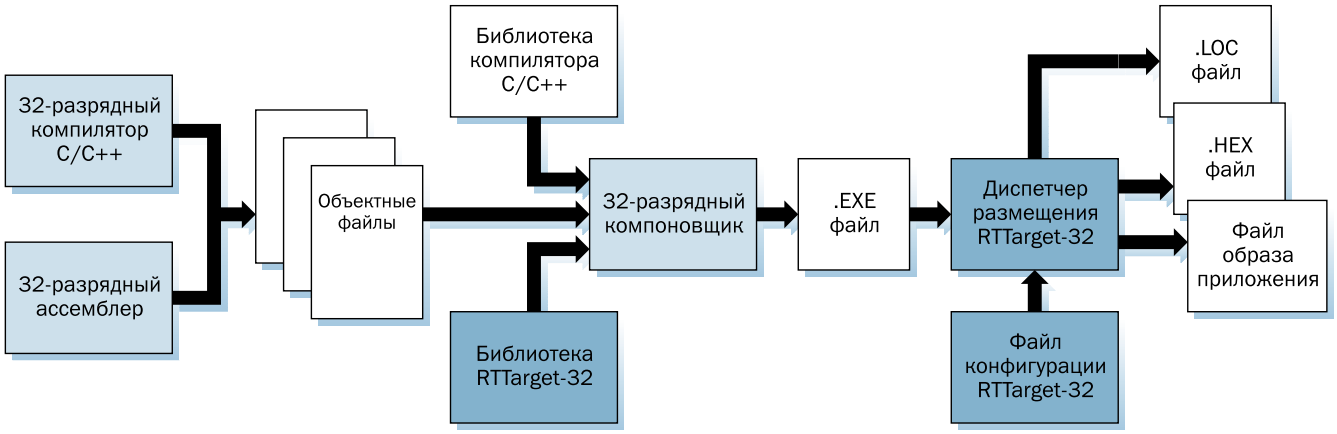

Рис. 6. Последовательность переноса приложения с помощью RTTarget-32

водится путем использования 32-разрядных компиляторов, таких как Borland C++, Microsoft Visual C++, Watcom C/C++, которые позволяют генерировать NT-приложения, функционирующие в режиме консоли. Для построения исполняемых файлов на основе выходного объектного кода, формируемого указанными компиляторами, выполняется компоновка и дополнительная обработка с использованием средств, предоставляемых RTTarget-32.

Процедура построения исполняемого файла для переноса во встраиваемую систему показана на рис. 6.

Состав RTTarget-32

Загрузочный код целевой системы (Target Boot Code)

Загрузочный код выполняет инициализацию процессора и по возможности других аппаратных средств. Далее производится размещение в оперативной памяти и инициализация приложения. Загрузка приложений может выполняться с НГМД, НЖМД, твердотельного диска на основе РПЗУ или флэш-диска, а также непосредственно из ПЗУ.

Утилита BootDisk

Данная сервисная программа предназначена для формирования дисков, с которых может производиться загрузка приложений, имеющих в своем составе компоненты RTTarget-32. Загрузочный диск может быть сформирован на основе любого дискового накопителя или его эмуляции (дискеты 5,25", 3,5", НЖМД, твердотельного диска), в котором реализована поддержка файловой структуры, базирующейся на FAT.

Менеджер размещения (Locator)

Менеджер размещения RTLoc предназначен для присвоения приложению фиксированных адресов в оперативной памяти. RTLoc позволяет выполнять загрузку программы в любую заданную область памяти целевой системы. Части кода приложения, доступные для чтения/записи и только для чтения, могут быть разделены путем помещения в различные области памяти.

Монитор удаленной отладки (Debug Monitor)

Монитор отладки, входящий в состав RTTarget-32, является приложением, которое разработано с использованием RTTarget-32. Монитор отладки может быть установлен на целевую вычислительную систему для осуществления управления загрузкой тестируемого приложения и связи с отладчиком уровня исходного текста, запускаемым на основной вычислительной системе.

Утилита запуска на исполнение (Run Utility)

В качестве альтернативного способа запуска программы, который производится из ПЗУ или с дискового накопителя, RTRun предоставляет возможность загрузки и запуска приложения с использованием последовательного канала связи между основной и целевой вычислительными системами.

Библиотека эмуляции интерфейса Win32

Данная библиотека представляет собой подмножество программного интерфейса Win32 API. Указанное подмножество обеспечивает поддержку стандартных исполнительных систем перечисленных ранее компиляторов, что позволяет использовать стандартные функции типа malloc(), printf() и других в приложении, которое подлежит запуску на целевой системе.

Библиотека ввода/вывода

В комплект поставки RTTarget-32 входит мощная библиотека ввода/вывода, в которой реализована поддержка связи одновременно через четыре последовательных порта с использованием прерываний.

Основные возможности RTTarget-32

Загрузка целевой системы

RTTarget-32 позволяет выполнять загрузку целевой системы с НГМД, НЖМД, твердотельного и флэш-диска или непосредственно из ПЗУ. Для загрузочного кода требуется около 6 кбайт памяти. Кроме того, загрузочный код формирует глобальную дескрипторную таблицу (GDT), дескрипторную таблицу прерываний (IDT) и таблицу страничного преобразования памяти целевой системы.

Поддержка Borland C/C++, Microsoft C/C++, Watcom C/C++

Для разработки мощных 32-разрядных встраиваемых приложений могут быть применены указанные популярные 32-разрядные компиляторы.

Поддержка удаленной кросс-отладки

Для RTTarget-32 обеспечена возможность отладки на уровне исходных текстов с помощью отладчика TD32 фирмы Borland.

Поддержка исполнительных систем C/C++

Реализована возможность использования в приложениях стандартных процедур, входящих в состав исполнительных систем, таких как printf(), malloc() и т. п.

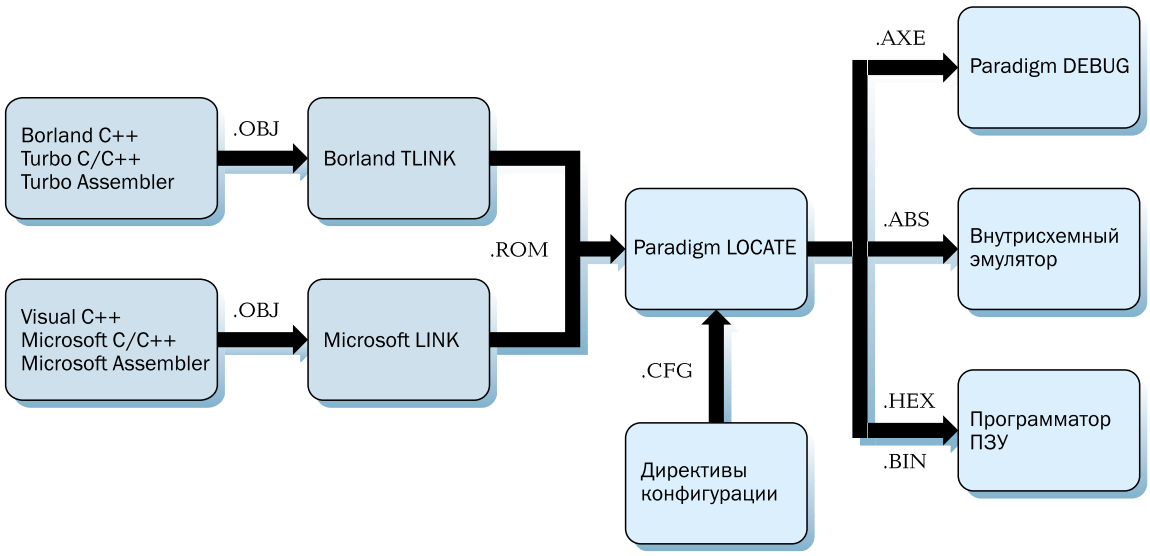

Рис. 7. Структура создания встраиваемого приложения

Невысокие требования к системным ресурсам

Для исполнения 32-разрядных приложений, созданных с использованием RTTarget-32, достаточно как минимум 16 кбайт оперативной памяти.

Поддержка уровней привилегий

Приложения могут запускаться на исполнение с различными уровнями привилегий для оптимизации либо с точки зрения максимума защиты, либо с точки зрения обеспечения максимальной производительности.

Поддержка страничной организации памяти

Обеспечена полная поддержка страничного преобразования адресного пространства памяти. Возможность защиты памяти с помощью страничного преобразования на процессорах, начиная с i386, используется для гарантированного отсутствия возможности разрушения приложениями защищаемых областей данных, кода и особо важных системных таблиц.

Перераспределение оперативной памяти

Помимо поддержки страничного преобразования в целях защиты данных, кода и системных таблиц, RTTarget-32 позволяет выполнять перераспределение страниц оперативной памяти для создания последовательно располагаемых блоков памяти большого размера. Кроме того, в RTTarget-32 реализована возможность создания областей виртуальной памяти путем комбинирования различных участков физической памяти. Однако значения физических адресов в виртуальном адресном пространстве остаются фиксированными, что обеспечивает простоту абсолютной адресации памяти.

Виртуальные файлы в оперативной памяти

Хотя в RTTarget-32 отсутствует собственная файловая система, возможна эмуляция файлового ввода-вывода путем создания виртуальных файлов в оперативной памяти целевой системы.

Возможность запуска под управлением DOS, Windows 3.1, Windows 95 и Windows NT

RTTarget-32 может применяться совместно с указанными популярными операционными системами.

Paradigm C/C++ PowerPack

Инструментальное кросс-средство разработки 16-разрядных встраиваемых приложений, функционирующих в реальном режиме

В состав PowerPack включены следующие продукты фирмы Paradigm Systems.

LOCATE

Менеджер размещения, предназначенный для присвоения 16-разрядному приложению фиксированных адресов для размещения в памяти. На рис. 7 показана последовательность действий, которые выполняются при создании файлов приложения для удаленной отладки на целевой системе, внутрисхемном эмуляторе и для сохранения в ПЗУ целевой системы.

DEBUG/RT

Средство удаленной отладки встраиваемых приложений на уровне исходного текста, представляющее собой адаптированную реализацию популярного отладчика Turbo Debugger. DEBUG/RT обеспечивает поддержку компиляторов фирм Borland, Microsoft и Intel; интегрированных операционных систем реального времени; использования сложных точек останова по множественным условиям; различных способов просмотра данных; сохранения и восстановления состояния отладчика.

Помимо перечисленных инструментальных кросс-средств, фирма Paradigm Systems предлагает отладчик DEBUG/RTOS с расширениями для работы совместно с приложениями RTKernel. Отладчик позволяет осуществлять просмотр состояния задач, семафоров и очередей сообщений, использовать точки останова и т. д. Данный отладчик совместим с компиляторами Turbo/Borland C/C++ версии 1.0 и выше, Microsoft C/C++ версии 5.1 или выше. В качестве основной операционной системы может использоваться DOS, Windows и Windows NT.