

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«ДОНСКОЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ
УНИВЕРСИТЕТ»
(ДГТУ)**

УПКВК, Отдел магистратуры
Кафедра «Автоматизация производственных процессов»

«Операционные системы реального времени»

Вводная лекция
для студентов заочной формы обучения

Ростов-на-Дону
2023

УДК 62-192

Составитель: Быкадор В.С.

Вводная лекция. – Ростов-на-Дону : Донской гос. техн. ун-т,
2023.

Вводная лекция по дисциплине «Операционные системы реального времени» предназначены для студентов заочной формы обучения, обучающихся по направлению подготовки 15.04.04 «Автоматизация технологических процессов и производств» профиль «Интеллектуальные системы сбора и анализа больших данных».

УДК 62-192

Печатается по решению редакционно-издательского совета
Донского государственного технического университета

В печать _____.____.20__ г.
Формат 60х84/16. Объем _____ усл. п. л.
Тираж _____ экз. Заказ № _____.

Издательский центр ДГТУ
Адрес университета и полиграфического предприятия:
344000, г. Ростов-на-Дону, пл. Гагарина, 1

© Донской государственный
технический университет, 2023

Вводная лекция.

Общее понятие об операционных системах реального времени

Основы работы операционной системы реального времени

Прежде чем говорить об особенностях операционной системы реального времени (далее ОСРВ), следует остановиться на основных принципах работы любой ОСРВ [1].

Основой ОСРВ является ядро (Kernel) операционной системы. Ядро реализует основополагающие функции любой операционной системы (ОС). В ОС общего назначения, таких как Windows, Mac Os и Linux, ядро позволяет нескольким пользователям выполнять множество программ на одном компьютере одновременно. Каждая выполняющаяся программа представляет собой задачу (Task). Если ОС позволяет одновременно выполнять множество задач, она является мультизадачной (Multitasking). Большинство процессоров, за счёт нескольких ядер могут выполнять несколько задач в один момент времени. На рис. 1 показано истинно параллельное выполнение трех задач.

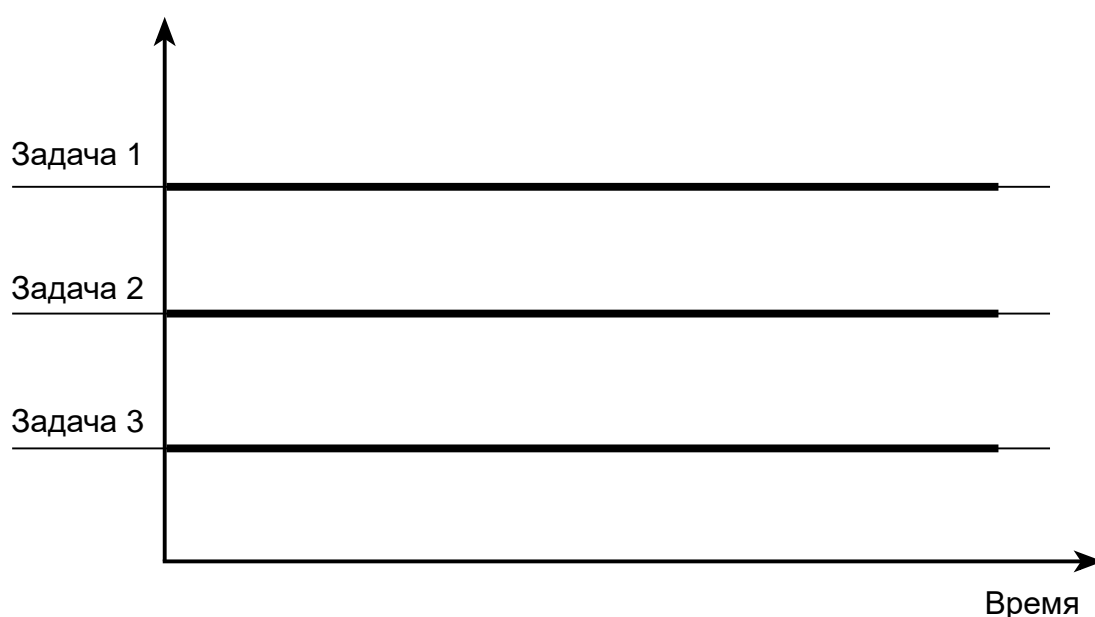


Рис. 1. Временная диаграмма реальной многозадачности

Микроконтроллеры и более простые процессоры – процессоры предыдущих поколений, за один момент времени могут выполнять только одну задачу (рис. 2).

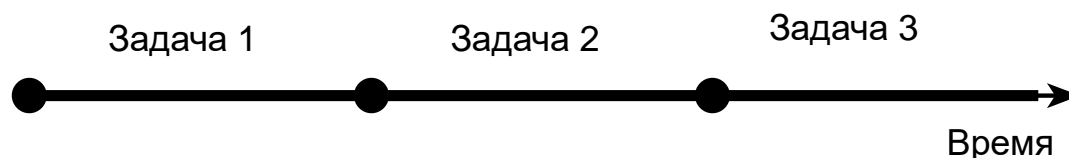


Рис. 2. Временная диаграмма последовательного выполнения задач

Однако при помощи быстрого переключения между задачами можно достичь эффекта параллельного выполнения всех задач – псевдопараллельное выполнение задач. В реальном одноядерном процессоре или микроконтроллере при работе ОСРВ выполнение задач носит периодический характер: каждая задача выполняется определенное время, после чего одноядерный процессор или микроконтроллер «переключается» на следующую задачу (рис. 3).

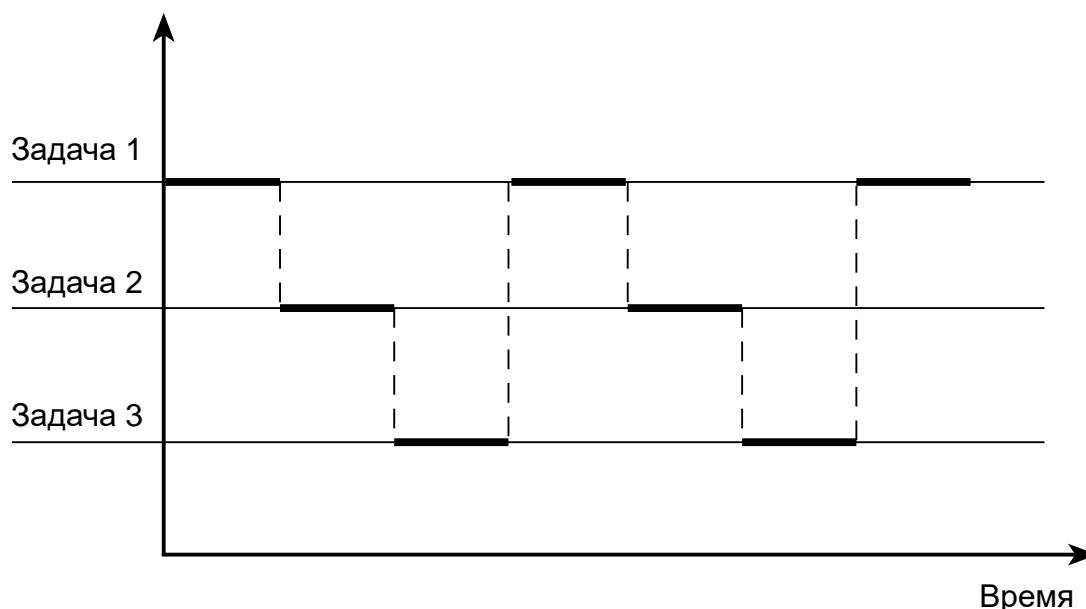


Рис. 3. Временная диаграмма псевдопараллельной многозадачности

В ОСРВ есть специальный программно реализованный элемент, который занимается переключением задач. Этот элемент называется планировщик (Scheduler).

Планировщик — это часть ядра ОСРВ, которая определяет, какая из задач, готовых к выполнению будет выполняться в данный конкретный момент времени. Планировщик может приостанавливать, а затем снова возобновлять выполнение задачи в течение всего её жизненного цикла (то есть с момента создания задачи до момента ее уничтожения). Алгоритм работы планировщика (Scheduling policy) — это алгоритм, по которому функционирует планировщик для принятия решения, какую задачу выполнять в данный момент времени. Алгоритм работы планировщика в ОС общего назначения заключается в предоставлении каждой задаче процессорного времени в равной пропорции. Алгоритм работы планировщика в ОСРВ имеет отличие и будет описан ниже.

Среди всех задач в системе в один момент времени может выполняться только одна задача. Говорят, что она находится в состоянии выполнения. Остальные задачи в этот момент не выполняются, ожидая, когда планировщик выделит каждой из них процессорное время. Таким образом, задача может находиться в двух основных состояниях: выполняться или не выполняться. Кроме того, что выполнение задачи может быть приостановлено планировщиком принудительно, задача может сама приостановить свое выполнение. Это происходит в двух случаях. Первый — это когда задаче необходимо задержать свое выполнение на определённый промежуток времени (в таком случае она переходит в состояние сна (sleep)). Второй — когда задача ожидает освобождения какого-либо аппаратного ресурса (например, последовательного порта) или наступления какого-то события (event), в этом случае говорят, что задача заблокирована (block). Блокированная или «спящая» задача не нуждается в процессорном времени до наступления соответствующего события или истечения определенного интервала времени.

Задачи и планировщик задач

Любая программа, которая выполняется под управлением ОСРВ, представляет собой множество отдельных независимых задач. Каждая задача выполняется в своем собственном контексте без случайных зависимостей от других задач и ядра ОСРВ. Только одна задача из множества может выполняться в один момент времени, и планировщик ответственен, какая именно. Планировщик останавливает и возобновляет выполнение всех задач по очереди, чтобы достичь эффекта одновременного выполнения нескольких задач на одном процессоре. Так как задача «не знает» об активности планировщика, то он отвечает за переключение контекста при смене выполняющейся задачи. Для достижения этого каждая задача имеет свой собственный стек. При смене задачи ее контекст сохраняется в ее собственном стеке, что позволяет восстановить контекст при возобновлении задачи.

Как было сказано выше, при грубом приближении задача может находиться в двух состояниях: выполняться и не выполняться. При подробном рассмотрении состояние «задача не выполняется» подразделяется на несколько различных состояний в зависимости от того, как она была остановлена (рис. 4).

Подробно рассмотрим состояния задачи в ОСРВ FreeRTOS. Говорят, что задача выполняется (*running*), если в данный момент времени процессор занят ее выполнением. Состояние готовности (*ready*) характеризует задачу, готовую к выполнению, но не выполняющуюся, так как в данный момент времени процессор занят выполнением другой задачи. Готовые к выполнению задачи (с одинаковым приоритетом) по очереди переходят в состояние выполнения и пребывают в нем в течение одного системного кванта, после чего возвращаются в состояние готовности.

Задача находится в заблокированном состоянии, если она ожидает наступления временного или внешнего события (*event*). Например, вызвав

API-функцию задержки, задача переведет себя в блокированное состояние до тех пор, пока не пройдет временной период задержки (delay): это будет временное событие. Задача блокирована, если она ожидает события, связанного с другими объектами ядра — очередями и семафорами: это будет внешнее (по отношению к задаче) событие. Нахождение задачи в блокированном состоянии ограничено тайм-аутом. То есть если ожидаемое внешнее событие не наступило в течение тайм-аута, то задача возвращается в состояние готовности к выполнению. Это предотвращает «подвисание» задачи при ожидании внешнего события, которое по каким-то причинам никогда не наступит. Блокированная задача не получает процессорного времени.

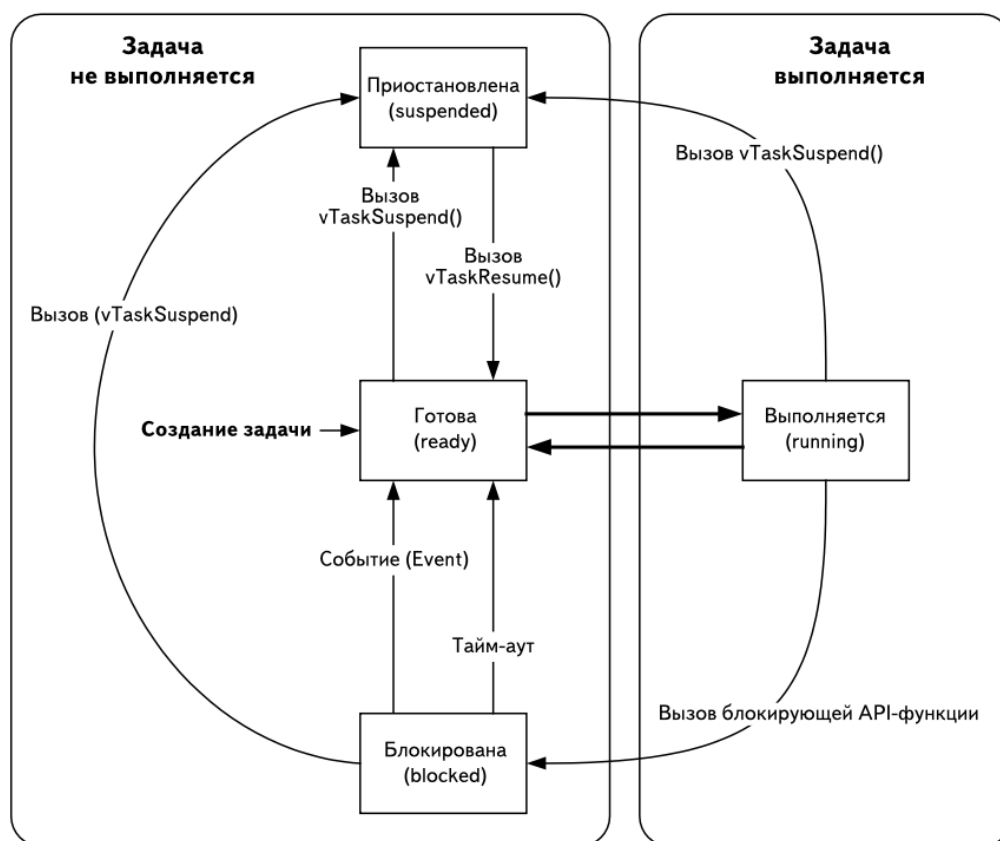


Рис. 4. Схема состояний задач и их переходов из одного состояния в другое на примере OCPB FreeRTOS

Приостановленная (suspended) задача также не получает процессорного времени, однако, в отличие от блокированного состояния, переход в приостановленное состояние и выход из него осуществляется в явном виде

вызовом соответствующих API-функций. Тайм-аут для приостановленного состояния не предусмотрен, и задача может оставаться приостановленной сколько угодно долго.

В любой программе реального времени есть как менее, так и более ответственные задачи. Под «ответственностью» задачи здесь понимается время реакции программы на внешнее событие, которое обрабатывается задачей. Например, ко времени реакции на срабатывание датчика в производственной установке предъявляются куда более строгие требования, чем ко времени реакции на нажатие клавиши на клавиатуре. Для обеспечения преимущества на выполнение более ответственных задач во ОСРВ применяется механизм приоритетов задач (Task priorities). Среди всех задач, находящихся в состоянии готовности, планировщик отдаст управление той задаче, которая имеет наивысший приоритет. Задача будет выполняться до тех пор, пока она не будет блокирована или приостановлена или пока не появится готовая к выполнению задача с более высоким приоритетом.

СПИСОК ИНФОРМАЦИОННЫХ ИСТОЧНИКОВ

1. Курниц А. FreeRTOS – операционная система для микроконтроллеров // Компоненты и технологии № 3, 2011, с. 109-114.