

## Лекция 4. Механизмы реального времени.

1 Механизмы реального времени: система приоритетов и алгоритмы диспетчеризации, механизмы межзадачного взаимодействия, время реакции системы

2 Механизмы реального времени: ядра реального времени, время переключения контекста, средства для работы с таймерами

3 Стандарты POSIX на ОСРВ

### 1 Механизмы реального времени: система приоритетов и алгоритмы диспетчеризации, механизмы межзадачного взаимодействия, время реакции системы

Важным параметром при оценке операционных систем реального времени является набор инструментов, механизмов реального времени, предоставляемых системой.

Процесс проектирования конкретной системы реального времени начинается с тщательного изучения объекта.

Разработчики проекта:

- исследуют объект и возможные события на нем,
- определяют критические сроки реакции системы на каждое событие
- и разрабатывают алгоритмы обработки этих событий.

Затем следует проектирование как таковое и разработка программных приложений.

Мечтой каждого разработчика является идеальная ОСРВ, в которой приложения реального времени разрабатываются на языке событий объекта. Хотя такая система и существует только в теории, она имеет свое название: система, управляемая критическими сроками.

Разработка приложений реального времени в этой системе сводится к описанию возможных событий на объекте. В каждом описании указываются два параметра:

- временной интервал - критическое время обслуживания данного события,
- адрес подпрограммы его обработки.

Всю дальнейшую заботу о том, чтобы подпрограмма обработки стартовала до истечения критического интервала времени, берет на себя операционная система.

Но это - мечта. В реальности же разработчик должен перевести язык событий объекта в сценарий многозадачной работы приложений ОСРВ, стараясь при этом оптимально использовать предоставленные ему специальные механизмы и оценить времена реакций системы на внешние события.

Какие же механизмы в операционных системах реального времени делают ОСРВ предсказуемой?

#### 1 Система приоритетов и алгоритмы диспетчеризации.

Базовыми инструментами разработки сценария для системы являются

1. система **приоритетов процессов** (задач),
2. **алгоритмы планирования** (диспетчеризации) ОСРВ.

В многозадачных ОС общего назначения используются, как правило, различные модификации алгоритма круговой диспетчеризации, основанные на понятии непрерывного кванта времени (time slice), которое предоставляется для исполнения данного процесса. Планировщик по истечении каждого кванта времени просматривает очередь активных процессов и принимает решение, какому из них передать управление, основываясь на приоритетах.

**Приоритеты** могут быть

1. фиксированными
2. или меняться со временем

- это зависит от алгоритмов планирования в данной ОС, но рано или поздно процессорное время получат все процессы в системе.

Алгоритмы круговой диспетчеризации в чистом виде в ОСРВ неприменимы. Основным их недостатком является то, что в течение непрерывного кванта времени процессором владеет только один процесс. Планировщики же ОСРВ имеют возможность сменить процесс до истечения time slice, если в этом возникла необходимость. Один из возможных **алгоритмов планирования** при этом - **приоритетный с вытеснением**. Мир ОСРВ отличается богатством различных алгоритмов планирования:

- динамические,
- приоритетные,
- монотонные,
- адаптивные и пр.,

цель же всегда преследуется одна - предоставить инструмент, позволяющий в нужный момент времени исполнять именно тот процесс, который необходим.

## **2 Механизмы межзадачного взаимодействия.**

Другой набор механизмов реального времени относится к средствам синхронизации процессов и передачи данных между ними. В него входят семафоры, мьютексы (mutex), события, сигналы, средства для работы с разделяемой памятью, каналы данных (pipes), очереди сообщений. В ОСРВ эти механизмы очень развиты. Многие из них используются и в ОС общего назначения, но их реализация в ОСРВ имеет свои особенности: время исполнения системных вызовов почти не зависит от состояния системы, и в каждой ОСРВ есть, по крайней мере, один быстрый механизм передачи данных от процесса к процессу.

## **3 Время реакции системы**

Почти все производители систем реального времени приводят такой параметр, как время реакции системы на прерывание (interrupt latency). В самом деле, если главным для системы реального времени является ее способность вовремя отреагировать на внешние события, то такой параметр, как время реакции системы является ключевым. Однако, что это за время и как к нему относиться? В настоящий момент нет, к сожалению, общепринятых методологий измерения этого параметра, поэтому он является полем битвы маркетинговых служб производителей систем реального времени. Есть надежда, что в скором времени положение изменится, так как уже стартовал проект сравнения операционных систем реального времени, который включает в себя в том числе и разработку методологии тестирования. Как быть в этой ситуации? Поймем, какие времена мы должны знать для того, чтобы предсказать время реакции системы.

- События, происходящие на объекте, регистрируются датчиками, данные с датчиков передаются в модули ввода-вывода (интерфейсы) системы.
- Модули ввода-вывода, получив информацию от датчиков и преобразовав ее, генерируют запрос на прерывание в управляющем компьютере, подавая ему тем самым сигнал о том, что на объекте произошло событие.
- Получив сигнал от модуля ввода-вывода, система должна запустить программу обработки этого события.

Интервал времени - от события на объекте и до выполнения первой инструкции в программе обработки этого события и является временем реакции системы на события, и, проектируя систему реального времени, разработчики должны уметь вычислять этот интервал. Из чего он складывается? Время выполнения цепочки действий - от события на объекте до генерации прерывания - никак не зависит от операционных систем реального времени и целиком определяется аппаратурой, а вот интервал времени - от возникновения запроса на прерывание и до выполнения первой инструкции обработчика определяется целиком свойствами операционной системы и архитектурой компьютера. Причем это время нужно уметь оценивать в худшей для системы ситуации, то есть в предположении, что процессор загружен, что в это время могут происходить другие прерывания, что система может выполнять какие-то действия, блокирующие прерывания. Неплохим основанием для оценки времен реакции системы могут служить результаты тестирования с подробным описанием архитектуры целевой системы, в которой проводились измерения, средств измерения и точным указанием, какие промежутки времени измерялись. Некоторые производители операционных систем реального времени результаты

такого тестирования предоставляют. Их можно отыскать на WEB-страницах, в документах технической поддержки, в публикациях фирм, проводящих независимое тестирование.

## **2 Механизмы реального времени: ядра реального времени, время переключения контекста, средства для работы с таймерами**

### **1 Ядра реального времени**

В этот класс входят системы с монолитным ядром, где и содержится реализация всех механизмов реального времени этих операционных систем. Исторически системы этого типа были хорошо спроектированы. В отличие от систем других классов, которые появлялись как временные компромиссы и затем "наращивали мускулы" благодаря первым удачным реализациям (исполнительные системы реального времени и UNIX'ы реального времени), разработчики систем этого класса имели время для разработки систем именно реального времени и не были изначально ограничены в выборе средств (например, фирма "Microware" имела в своем распоряжении три года для разработки первого варианта OS-9). Системы этого класса, как правило, модульны, хорошо структурированы, имеют наиболее развитый набор специфических механизмов реального времени, компактны и предсказуемы. Наиболее популярные системы этого класса: OS9, QNX. Одна из особенностей систем этого класса - высокая степень масштабируемости. На базе этих ОС можно построить как компактные системы реального времени, так и большие системы серверного класса. Как правило, ядра реального времени имеют два типа систем разработки - кроссовую и резидентную.

### **2 Размеры системы**

Для ОСРВ важным параметром является размер системы исполнения - суммарный объем минимально необходимого для работы приложения системного набора (ядро, системные модули, драйверы и т. д.). Хотя надо признать, что постепенно значимость этого параметра уменьшается, тем не менее он остается важным и производители ОСРВ стремятся к тому, чтобы размеры ядра и обслуживающих модулей были невелики.

Например, размер ядра ОСРВ OS9 на микропроцессорах MC68xxx составляет 22 Кб, VxWorks - 16 Кб.

### **Возможность исполнения системы из ПЗУ (ROM)**

Это свойство ОСРВ - одно из базовых. Оно позволяет создавать компактные встроенные СРВ повышенной надёжности, с ограниченным энергопотреблением, без внешних накопителей.

### **3 Время переключения контекста**

В операционные системы реального времени заложен параллелизм, возможность одновременной обработки нескольких событий, поэтому все операционные системы реального времени являются многозадачными (многопроцессными, многонитевыми). Для того чтобы уметь оценивать накладные расходы системы при обработке параллельных событий, необходимо знать время, которое система затрачивает на передачу управления от процесса к процессу (от задачи к задаче, от нити к нити), то есть время переключения контекста.

### **4 Средства для работы с таймерами**

Такие инструменты, как средства работы с таймерами, необходимы для систем с жестким времененным регламентом, поэтому развитость средств работы с таймерами - необходимый атрибут ОСРВ. Эти средства, как правило, позволяют:

- измерять и задавать различные промежутки времени (от 1 мкс и выше),
- генерировать прерывания по истечении временных интервалов,
- создавать разовые и циклические будильники

Здесь описаны только базовые, обязательные механизмы, использующиеся в ОСРВ. Кроме того, почти в каждой операционной системе реального времени вы найдете целый набор

дополнительных, специфических только для нее механизмов, касающийся системы ввода-вывода, управления прерываниями, работы с памятью.

Каждая система содержит также ряд средств, обеспечивающих ее надежность:

- встроенные механизмы контроля целостности кодов,
- инструменты для работы с Watch-Dog таймерами.

### **3 Стандарты POSIX на ОСРВ**

Стандарт POSIX (Portable Operating System Interface) 1003.1b, ранее существовавший под рабочим именем POSIX 1003.4 и разработанный IEEE (Institute of Electrical and Electronical Engineers), определяет расширения стандарта POSIX 1001 на операционные системы UNIX, которые позволяют использовать последние в качестве ОСРВ. Большинство приложений UNIX могут быть перенесены в такие системы, поскольку стандарт POSIX 1003.1b обеспечивает единый с системами UNIX программный интерфейс (API, application interface).

Стандарт POSIX 1003 состоит следующих частей.

1. POSIX 1003.1 определяет стандарт на основные компоненты операционной системы, API для процессов, файловой системы, устройств и т.д.
2. POSIX 1003.2 определяет стандарт на основные утилиты.
3. POSIX 1003.1b определяет стандарт на основные расширения реального времени.
4. POSIX 1003.1c определяет стандарт на задачи (threads).
5. POSIX 1003.1d определяет стандарт на дополнительные расширения реального времени (такие, как, например, поддержка обработчиков прерываний); этот стандарт еще официально не утвержден, но некоторые ОСРВ заявляют о своей поддержке некоторых его частей.

Стандарту POSIX 1003 с расширением 1003.1b удовлетворяют такие системы, как Lynx, VxWorks, QNX. Некоторые системы, например, CHORUS, обеспечивают поддержку стандарта 1003.1b при загрузке поставляемого программного обеспечения, т.е. имеют как бы два типа API: оригинальный собственный и стандартный.

#### **Упражнения к лекции.**

1. Какой наиболее часто используемый в ОСРВ принцип планирования?
2. Какие задачи чаще всего решаются при планировании?
3. Как происходит взаимное согласование задач?
4. Какие проблемы возникают при использовании ресурсов? Что такое критическая секция?
5. Какие основные тенденции используются при синхронизации с внешними событиями?
6. Для чего используется синхронизация по времени?
7. Что относится к средствам синхронизации процессов и передачи данных?
8. Что является временем реакции системы на события?
9. Какой интервал времени во времени реакции системы определяется аппаратурой?
10. Какой интервал времени во времени реакции системы определяется операционной системы и архитектурой компьютера?

#### **Литература к лекции.**

- 2.1 Мартин Дж. Программирование для вычислительных систем реального времени. Пер. с англ. Изд-во "Наука", 1975.- 360с.
- 2.2 Тенанбаум Э. Современные операционные системы. пер. с англ. 2-е изд. – М.: СПБ.: Нижний Новгород: Питер, 2005. – 1037с.
- 2.3 Олифер В.Г. Олифер Н.А. Сетевые операционные системы. М.: СПБ.: Нижний Новгород: Питер, 2006. – 538с.

- 2.4 Грибанов В.П., Дробин С.В., Медведев В.Д. Операционные системы. - М.: Финансы и статистика, 1990. - 239 с.
- 2.5 Дейтел Х.М., Чофтес Р.Д. Операционные системы. пер. с англ. – М.: БИНОМ, 2006. – 704с.