



СИСТЕМЫ РЕАЛЬНОГО ВРЕМЕНИ

Лекция 2



- 1. ОСНОВНЫЕ ПОНЯТИЯ СИСТЕМ РЕАЛЬНОГО ВРЕМЕНИ.**
- 2. ТИПЫ ЗАДАЧ СИСТЕМ РЕАЛЬНОГО ВРЕМЕНИ.**
- 3. КЛАССЫ СИСТЕМ РЕАЛЬНОГО ВРЕМЕНИ.**



ОСНОВНЫЕ ПОНЯТИЯ СИСТЕМ РЕАЛЬНОГО ВРЕМЕНИ

1. ПРОЦЕСС
2. ЗАДАЧА
3. РЕСУРС
4. СТЕК, ВИРТУАЛЬНАЯ ПАМЯТЬ
5. МЕЖПРОЦЕССНОЕ ВЗАИМОДЕЙСТВИЕ
6. СОБЫТИЕ



ПРОЦЕСС

ПРОЦЕСС - ОТДЕЛЬНО ЗАГРУЖАЕМЫЙ ПРОГРАММНЫЙ МОДУЛЬ (ФАЙЛ), КОТОРЫЙ, КАК ПРАВИЛО, ВО ВРЕМЯ ИСПОЛНЕНИЯ ИМЕЕТ В ПАМЯТИ СВОИ НЕЗАВИСИМЫЕ ОБЛАСТИ ДЛЯ КОДА И ДАННЫХ, Т.Е. ПРОЦЕСС — ЭТО ПРОГРАММА НА СТАДИИ ВЫПОЛНЕНИЯ.

В ДАННОМ СЛУЧАЕ ПОД ПРОГРАММОЙ ПОНИМАЕТСЯ ОПИСАНИЕ НА НЕКОТОРОМ ФОРМАЛИЗОВАННОМ ЯЗЫКЕ АЛГОРИТМА, РЕШАЮЩЕГО ПОСТАВЛЕННУЮ ЗАДАЧУ. ПРОГРАММА ЯВЛЯЕТСЯ СТАТИЧЕСКОЙ ЕДИНИЦЕЙ, ТО ЕСТЬ НЕИЗМЕНЯЕМОЙ С ТОЧКИ ЗРЕНИЯ ОПЕРАЦИОННОЙ СИСТЕМЫ, ЕЕ ВЫПОЛНЯЮЩЕЙ.

ПРОЦЕСС ИМЕЕТ:

- СОБСТВЕННЫЕ ОБЛАСТИ ПАМЯТИ, ГДЕ ОСУЩЕСТВЛЯЕТСЯ ХРАНЕНИЕ КОДА И ДАННЫХ;
- СОБСТВЕННЫЙ СТЕК;
- СОБСТВЕННОЕ ОТОБРАЖЕНИЕ ВИРТУАЛЬНОЙ ПАМЯТИ НА ФИЗИЧЕСКУЮ (В СИСТЕМАХ С ВИРТУАЛЬНОЙ ПАМЯТЬЮ);
- СОБСТВЕННОЕ СОСТОЯНИЕ.



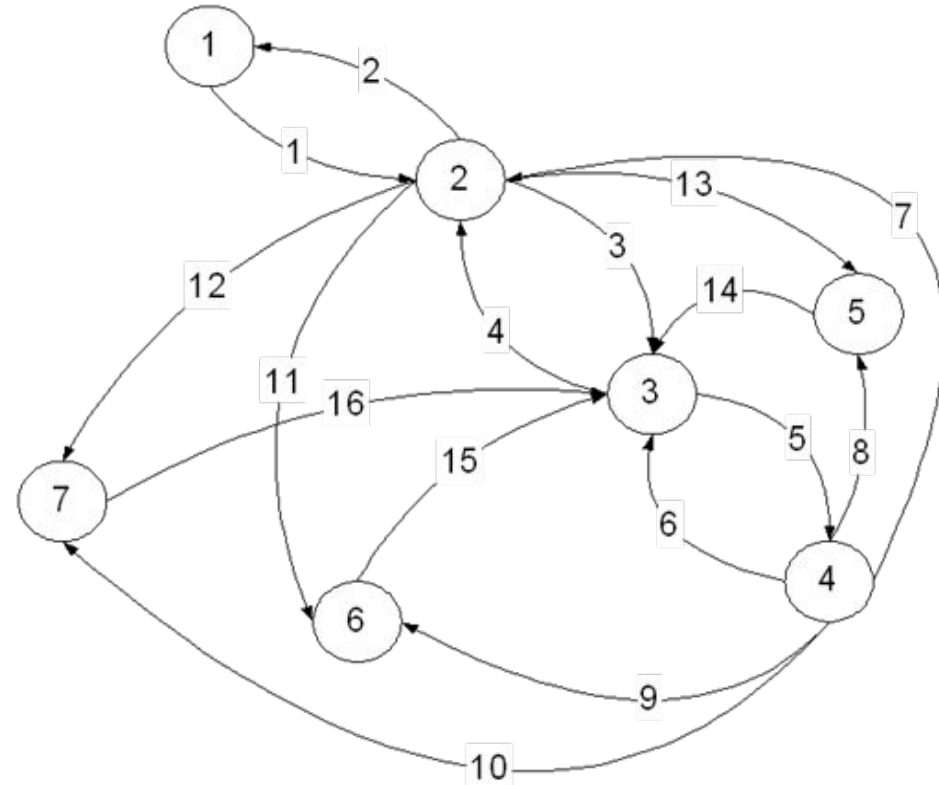
ПРОЦЕСС

ПРОЦЕСС МОЖЕТ НАХОДИТЬСЯ В ОДНОМ ИЗ СЛЕДУЮЩИХ ТИПИЧНЫХ СОСТОЯНИЙ:

- 1) «ОСТАНОВЛЕН» - ПРОЦЕСС ОСТАНОВЛЕН И НЕ ИСПОЛЬЗУЕТ ПРОЦЕССОР (НАПРИМЕР, В ТАКОМ СОСТОЯНИИ ПРОЦЕСС НАХОДИТСЯ СРАЗУ ПОСЛЕ СОЗДАНИЯ);**
- 2) «ТЕРМИНИРОВАН» - ПРОЦЕСС ТЕРМИНИРОВАН И НЕ ИСПОЛЬЗУЕТ ПРОЦЕССОР (НАПРИМЕР, ПРОЦЕСС ЗАКОНЧИЛСЯ, НО ЕЩЕ НЕ УДАЛЕН ОПЕРАЦИОННОЙ СИСТЕМОЙ);**
- 3) «ЖДЕТ» - ПРОЦЕСС ЖДЕТ НЕКОТОРОГО СОБЫТИЯ (КОТОРЫМ МОЖЕТ БЫТЬ АППАРАТНОЕ ИЛИ ПРОГРАММНОЕ ПРЕРЫВАНИЕ, СИГНАЛ ИЛИ ДРУГАЯ ФОРМА МЕЖПРОЦЕССОРНОГО ВЗАИМОДЕЙСТВИЯ);**
- 4) «ГОТОВ» - ПРОЦЕСС НЕ ОСТАНОВЛЕН, НЕ ТЕРМИНИРОВАН, НЕ ОЖИДАЕТ, НЕ УДАЛЕН, НО И НЕ РАБОТАЕТ (НАПРИМЕР, ПРОЦЕСС МОЖЕТ НЕ ПОЛУЧАТЬ ДОСТУПА К ПРОЦЕССОРУ, ЕСЛИ В ДАННЫЙ МОМЕНТ ВЫПОЛНЯЕТСЯ ДРУГОЙ, БОЛЕЕ ПРИОРИТЕТНЫЙ ПРОЦЕСС);**
- 5) «ВЫПОЛНЯЕТСЯ» - ПРОЦЕСС ВЫПОЛНЯЕТСЯ И ИСПОЛЬЗУЕТ ПРОЦЕССОР (В ОСРВ ЭТО ОБЫЧНО ОЗНАЧАЕТ, ЧТО ЭТОТ ПРОЦЕСС ЯВЛЯЕТСЯ САМЫМ ПРИОРИТЕТНЫМ СРЕДИ ВСЕХ ПРОЦЕССОВ, НАХОДЯЩИХСЯ В СОСТОЯНИИ «ГОТОВ»).**



ПРОЦЕСС



ПЕРЕХОДЫ:

1. ПЕРЕХОД 1-2 СОЗДАНИЕ ПРОЦЕССА
2. ПЕРЕХОД 2-1 УНИЧТОЖЕНИЕ ПРОЦЕССА
3. ПЕРЕХОД 2-3 АКТИВИЗАЦИЯ ПРОЦЕССА ДИСПЕТЧЕРОМ
4. ПЕРЕХОД 3-2 ДЕАКТИВИЗАЦИЯ ПРОЦЕССА
5. ПЕРЕХОД 3-4 ЗАГРУЗКА НА ВЫПОЛНЕНИЕ ПРОЦЕССА ДИСПЕТЧЕРОМ

6. ПЕРЕХОД 4-3 ТРЕБОВАНИЕ ОБСЛУЖИВАНИЯ ОТ ПРОЦЕССОРА ДРУГИМ ПРОЦЕССОМ (ПРЕЕМПТИОН – ПРИОРИТЕТНОЕ ПЕРЕКЛЮЧЕНИЕ)

7. ПЕРЕХОД 4-2 ЗАВЕРШЕНИЕ ПРОЦЕССА

8. ПЕРЕХОД 4-5 БЛОКИРОВКА ПРОЦЕССА ДО ОСВОБОЖДЕНИЯ ТРЕБУЕМОГО РЕСУРСА

9. ПЕРЕХОД 4-6 БЛОКИРОВКА ПРОЦЕССА ДО ИСТЕЧЕНИЯ ЗАДАННОГО ВРЕМЕНИ

10. ПЕРЕХОД 4-7 БЛОКИРОВКА ПРОЦЕССА ДО ПРИХОДА СОБЫТИЯ

11. ПЕРЕХОД 2-6 АКТИВИЗАЦИЯ ПРОЦЕССА ПРИВОДИТ К ОЖИДАНИЮ ВРЕМЕННОЙ ЗАДЕРЖКИ

12. ПЕРЕХОД 2-7 АКТИВИЗАЦИЯ ПРОЦЕССА ПРИВОДИТ К ОЖИДАНИЮ СОБЫТИЯ

13. ПЕРЕХОД 2-5 АКТИВИЗАЦИЯ ПРОЦЕССА ПРИВОДИТ К ОЖИДАНИЮ ОСВОБОЖДЕНИЯ РЕСУРСА

14. ПЕРЕХОД 5-3 АКТИВИЗАЦИЯ ПРОЦЕССА ИЗ-ЗА ОСВОБОЖДЕНИЯ ОЖИДАВШЕГОСЯ РЕСУРСА

15. ПЕРЕХОД 6-3 АКТИВИЗАЦИЯ ПРОЦЕССА ПО ИСТЕЧЕНИИ ЗАДАННОГО ВРЕМЕНИ

16. ПЕРЕХОД 7-3 АКТИВИЗАЦИЯ ПРОЦЕССА ИЗ-ЗА ПРИХОДА ОЖИДАВШЕГОСЯ СОБЫТИЯ



Задача

СОВОКУПНОСТЬ СВЯЗАННЫХ МЕЖДУ СОБОЙ И ОБРАЗУЮЩИХ ЕДИНОЕ ЦЕЛОЕ ПРОГРАММНЫХ МОДУЛЕЙ И ДАННЫХ, ТРЕБУЮЩИХ РЕСУРСОВ ВЫЧИСЛИТЕЛЬНОЙ СИСТЕМЫ ДЛЯ СВОЕЙ РЕАЛИЗАЦИИ.

ЕДИНИЦА РАБОТЫ, ДЛЯ ВЫПОЛНЕНИЯ КОТОРОЙ ПРЕДОСТАВЛЯЕТСЯ ЦЕНТРАЛЬНЫЙ ПРОЦЕССОР. ВЫЧИСЛИТЕЛЬНЫЙ ПРОЦЕСС МОЖЕТ ВКЛЮЧАТЬ В СЕБЯ НЕСКОЛЬКО ЗАДАЧ..

ПРОЦЕССОР

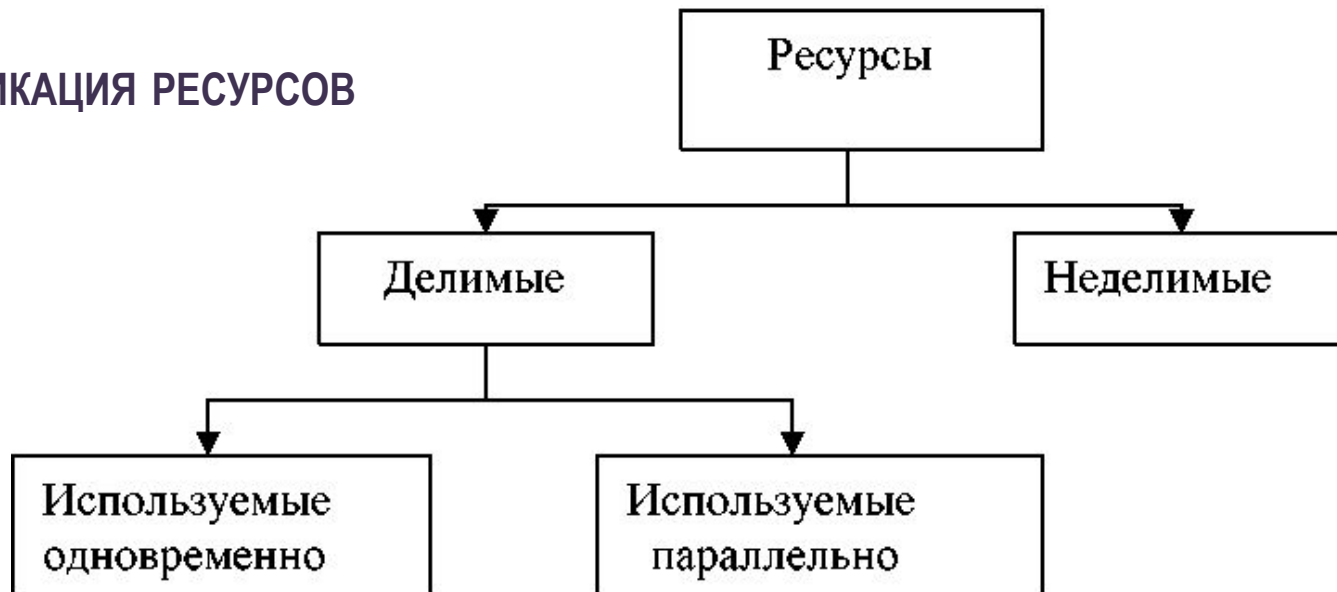
НОСИТЕЛЬ
ДАННЫХ

ВЫПОЛНЯЕТ
ОПЕРАЦИИ,
СВЯЗАННЫЕ С
ОБРАБОТКОЙ
ДАННЫХ



Термин ресурс обычно применяется по отношению к повторно используемым, относительно стабильным и часто недостающим объектам, которые запрашиваются, используются и освобождаются процессами в период их активности. Другими словами, ресурсом называется всякий объект, который может распределяться внутри системы. Ресурс - это объект, необходимый для работы процессу или задаче.

Классификация ресурсов



Различного рода программные и информационные ресурсы также могут быть определены для системы как объекты, которые могут разделяться и распределяться, и доступ к которым, необходимо соответствующим образом контролировать. В настоящее время понятие ресурса превратилось в абстрактную структуру с целым рядом атрибутов, характеризующих способы доступа к этой структуре и ее физическое представление в системе.



Если в операционной системе могут одновременно существовать несколько процессов или/и задач, находящихся в состоянии «выполняется», то говорят, что это многозадачная система, а эти процессы называют параллельными.

При мультипрограммировании повышается пропускная способность системы, но отдельный процесс никогда не может быть выполнен быстрее, чем, если бы он выполнялся в однопрограммном режиме (всякое разделение ресурсов замедляет работу одного из участников за счет дополнительных затрат времени на ожидание освобождения ресурса).

Система поддерживает мультипрограммирование и старается эффективно использовать ресурсы путем организации к ним очередей запросов, составляемых тем или иным способом. Это требование достигается поддержанием в памяти более одного процесса, ожидающего процессор, и более одного процесса, готового использовать другие ресурсы, как только последние станут доступными.

При необходимости использовать какой-либо ресурс (оперативную память, устройство ввода/вывода, массив данных и т.п.), задача обращается к супервизору операционной системы – ее центральному управляющему модулю, который может состоять из нескольких модулей и сообщает о своем требовании. При этом указывается вид ресурса и, если надо, его объем. Директива обращения к операционной системе передает ей управление переводя процессор в привилегированный режим работы, который

Ресурс может быть выделен задаче, обратившейся к супервизору с соответствующим запросом, если:

- ❖ **он свободен и в системе нет запросов от задач более высокого приоритета к этому же ресурсу;**
- ❖ **текущий запрос и ранее выданные запросы допускают совместное использование ресурсов;**
- ❖ **ресурс используется задачей низшего приоритета и может быть временно отобран (разделяемые ресурсы).**

Получив запрос, система либо удовлетворяет его и возвращает управление задаче, выдавшей данный запрос, либо, если ресурс занят, ставит задачу в очередь к ресурсу, переводя ее в состояние ожидания (блокируя).

В общем случае при организации управления ресурсами в СРВ всегда требуется принять решение о том, что в данной ситуации выгоднее: быстро обслуживать отдельные наиболее важные запросы, предоставлять всем процессам равные возможности, либо обслуживать максимально возможное количество процессов и наиболее полно использовать ресурсы.



Стек (STACK) - это область памяти, в которой размещаются локальные переменные, аргументы и возвращаемые значения функций. Вместе с областью статических данных полностью задает текущее состояние процесса.

Виртуальная память - это «память», в адресном пространстве которой работает процесс. Виртуальная память:

- 1) позволяет увеличить объем памяти, доступной процессам за счет дисковой памяти;
- 2) обеспечивает выделение каждому из процессов виртуально непрерывного блока памяти, начинающегося (виртуально) с одного и того же адреса;
- 3) обеспечивает изоляцию одного процесса от другого.

Трансляцией виртуального адреса в физический адрес занимается операционная система. Для ускорения этого процесса многие компьютерные системы имеют поддержку со стороны аппаратуры, которая может быть либо прямо в процессоре, либо в специальном устройстве управления памятью. Среди механизмов трансляции виртуального адреса преобладает страничный, при котором виртуальная и физическая память разбиваются на части равного размера, называемые страницами (типичный размер - 4Кб), между страницами виртуальной и физической памяти устанавливается взаимно однозначное (для каждого процесса) отображение.



МЕЖПРОЦЕССНОЕ ВЗАИМОДЕЙСТВИЕ - ЭТО ТОТ ИЛИ ИНОЙ СПОСОБ ПЕРЕДАЧИ ИНФОРМАЦИИ ИЗ ОДНОГО ПРОЦЕССА В ДРУГОЙ. **НАИБОЛЕЕ РАСПРОСТРАНЕННЫМИ ФОРМАМИ ВЗАИМОДЕЙСТВИЯ ПРОЦЕССОВ ЯВЛЯЮТСЯ:**

1) РАЗДЕЛЯЕМАЯ ПАМЯТЬ - ДВА (ИЛИ БОЛЕЕ) ПРОЦЕССА ИМЕЮТ ДОСТУП К ОДНОМУ И ТОМУ ЖЕ БЛОКУ ПАМЯТИ. В СИСТЕМАХ С ВИРТУАЛЬНОЙ ПАМЯТЬЮ ОРГАНИЗАЦИЯ ТАКОГО ВИДА ВЗАИМОДЕЙСТВИЯ ТРЕБУЕТ ПОДДЕРЖКИ СО СТОРОНЫ ОПЕРАЦИОННОЙ СИСТЕМЫ, ПОСКОЛЬКУ НЕОБХОДИМО ОТОБРАЗИТЬ СООТВЕТСТВУЮЩИЕ БЛОКИ ВИРТУАЛЬНОЙ ПАМЯТИ ПРОЦЕССОВ НА ОДИН И ТОТ ЖЕ БЛОК ФИЗИЧЕСКОЙ ПАМЯТИ); СЕМАФОРЫ - ДВА (ИЛИ БОЛЕЕ) ПРОЦЕССА ИМЕЮТ ДОСТУП К ОДНОЙ ПЕРЕМЕННОЙ, ПРИНИМАЮЩЕЙ ЗНАЧЕНИЕ 0 ИЛИ 1. САМА ПЕРЕМЕННАЯ ЧАСТО НАХОДИТСЯ В ОБЛАСТИ ДАННЫХ ОПЕРАЦИОННОЙ СИСТЕМЫ И ДОСТУП К НЕЙ ОРГАНИЗУЕТСЯ ПОСРЕДСТВОМ СПЕЦИАЛЬНЫХ ФУНКЦИЙ;

3) СИГНАЛЫ - ЭТО СООБЩЕНИЯ, ДОСТАВЛЯЕМЫЕ ПОСРЕДСТВОМ ОПЕРАЦИОННОЙ СИСТЕМЫ ПРОЦЕССУ. ПРОЦЕСС ДОЛЖЕН ЗАРЕГИСТРИРОВАТЬ ОБРАБОТЧИК ЭТОГО СООБЩЕНИЯ У ОПЕРАЦИОННОЙ СИСТЕМЫ, ЧТОБЫ ПОЛУЧИТЬ ВОЗМОЖНОСТЬ РЕАГИРОВАТЬ НА НЕГО. ЧАСТО ОПЕРАЦИОННАЯ СИСТЕМА ИЗВЕЩАЕТ ПРОЦЕСС СИГНАЛОМ О НАСТУПЛЕНИИ КАКОГО-ЛИБО СБОЯ, НАПРИМЕР, ДЕЛЕНИИ НА 0, ИЛИ О КАКОМ-ЛИБО АППАРАТНОМ ПРЕРЫВАНИИ, НАПРИМЕР, ПРЕРЫВАНИИ ТАЙМЕРА;

4) ПОЧТОВЫЕ ЯЩИКИ - ЭТО ОЧЕРЕДЬ СООБЩЕНИЙ (ОБЫЧНО ТЕХ ИЛИ ИНЫХ СТРУКТУР ДАННЫХ), КОТОРЫЕ ПОМЕЩАЮТСЯ В ПОЧТОВЫЙ ЯЩИК ПРОЦЕССАМИ И/ИЛИ ОПЕРАЦИОННОЙ СИСТЕМОЙ. НЕСКОЛЬКО ПРОЦЕССОВ МОГУТ ЖДАТЬ ПОСТУПЛЕНИЯ СООБЩЕНИЯ В ПОЧТОВЫЙ ЯЩИК И АКТИВИЗИРОВАТЬСЯ ПОСЛЕ ЕГО ПОСТУПЛЕНИЯ. ТРЕБУЕТ ПОДДЕРЖКИ СО СТОРОНЫ ОПЕРАЦИОННОЙ



СОБЫТИЕ - ЭТО ОПОВЕЩЕНИЕ ПРОЦЕССА СО СТОРОНЫ ОПЕРАЦИОННОЙ СИСТЕМЫ О ТОЙ ИЛИ ИНОЙ ФОРМЕ МЕЖПРОЦЕССНОГО ВЗАИМОДЕЙСТВИЯ.

СОЗДАНИЕ, ОБЕСПЕЧЕНИЕ ВЗАИМОДЕЙСТВИЯ, РАЗДЕЛЕНИЕ ПРОЦЕССОРНОГО ВРЕМЕНИ ТРЕБУЕТ ОТ ОПЕРАЦИОННОЙ СИСТЕМЫ ЗНАЧИТЕЛЬНЫХ ВЫЧИСЛИТЕЛЬНЫХ ЗАТРАТ, ОСОБЕННО В СИСТЕМАХ С ВИРТУАЛЬНОЙ ПАМЯТЬЮ, ПОСКОЛЬКУ КАЖДЫЙ ПРОЦЕСС ИМЕЕТ СВОЕ ОТОБРАЖЕНИЕ ВИРТУАЛЬНОЙ ПАМЯТИ НА ФИЗИЧЕСКУЮ, КОТОРОЕ НАДО МЕНЯТЬ ПРИ ПЕРЕКЛЮЧЕНИИ ПРОЦЕССОВ И ПРИ ОБЕСПЕЧЕНИИ ИХ ДОСТУПА К ОБЪЕКТАМ ВЗАИМОДЕЙСТВИЯ. ЧАСТО БЫВАЕТ ТАК, ЧТО ТРЕБУЕТСЯ ЗАПУСТИТЬ НЕСКОЛЬКО КОПИЙ ОДНОЙ И ТОЙ ЖЕ ПРОГРАММЫ, НАПРИМЕР, ДЛЯ УПРАВЛЕНИЯ НЕСКОЛЬКИМИ ЕДИНИЦАМИ ОДНОГО И ТОГО ЖЕ ОБОРУДОВАНИЯ. В ЭТОМ СЛУЧАЕ МЫ НЕСЕМ ДВОЙНЫЕ НАКЛАДНЫЕ РАСХОДЫ: ДЕРЖИМ В ОПЕРАТИВНОЙ ПАМЯТИ НЕСКОЛЬКО КОПИЙ КОДА ОДНОЙ ПРОГРАММЫ И ТРАТИМ ДОПОЛНИТЕЛЬНОЕ ВРЕМЯ НА ОБЕСПЕЧЕНИЕ ИХ ВЗАИМОДЕЙСТВИЯ.

Задача (или поток, или нить, thread) - ЭТО КАК БЫ ОДНА ИЗ ВЕТВЕЙ ИСПОЛНЕНИЯ ПРОЦЕССА:

- РАЗДЕЛЯЕТ С ПРОЦЕССОМ ОБЛАСТЬ ПАМЯТИ ПОД КОД И ДАННЫЕ;
- ИМЕЕТ СОБСТВЕННЫЙ СТЕК;
- РАЗДЕЛЯЕТ С ПРОЦЕССОМ ОТОБРАЖЕНИЕ ВИРТУАЛЬНОЙ ПАМЯТИ НА ФИЗИЧЕСКУЮ (В СИСТЕМАХ С ВИРТУАЛЬНОЙ ПАМЯТЬЮ);
- ИМЕЕТ СОБСТВЕННОЕ СОСТОЯНИЕ.

ТАКИМ ОБРАЗОМ, У ДВУХ ЗАДАЧ В ОДНОМ ПРОЦЕССЕ ВСЯ ПАМЯТЬ ЯВЛЯЕТСЯ РАЗДЕЛЯЕМОЙ И ДОПОЛНИТЕЛЬНЫЕ РАСХОДЫ, СВЯЗАННЫЕ С РАЗНЫМ ОТОБРАЖЕНИЕМ ВИРТУАЛЬНОЙ ПАМЯТИ НА ФИЗИЧЕСКУЮ, СВЕДЕНЫ К НУЛЮ. ДЛЯ ЗАДАЧ ТАК ЖЕ, КАК ДЛЯ ПРОЦЕССОВ, ОПРЕДЕЛЯЮТСЯ ПОНЯТИЯ СОСТОЯНИЯ ЗАДАЧИ И МЕЖЗАДАЧНОГО ВЗАИМОДЕЙСТВИЯ.



ПРИОРИТЕТ. СВЯЗЫВАНИЕ

ПРИОРИТЕТ - ЭТО ЧИСЛО, ПРИПИСАННОЕ ОПЕРАЦИОННОЙ СИСТЕМОЙ КАЖДОМУ ПРОЦЕССУ И ЗАДАЧЕ. ЧЕМ БОЛЬШЕ ЭТО ЧИСЛО, ТЕМ ВАЖНЕЕ ЭТОТ ПРОЦЕСС ИЛИ ЗАДАЧА И ТЕМ БОЛЬШЕ ПРОЦЕССОРНОГО ВРЕМЕНИ ОН ИЛИ ОНА ПОЛУЧИТ. При неправильном планировании приоритетов в ОСРВ, задача с меньшим приоритетом может вообще не получить управления при наличии в состоянии готовности задачи с большим приоритетом.

СВЯЗЫВАНИЕ (линковка, LINKAGE) - ЭТО ПРОЦЕСС ПРЕВРАЩЕНИЯ СКОМПИЛИРОВАННОГО КОДА (ОБЪЕКТНЫХ МОДУЛЕЙ) В ЗАГРУЗОЧНЫЙ МОДУЛЬ (ТО ЕСТЬ ТО, ЧТО МОЖЕТ ИСПОЛНЯТЬСЯ ПРОЦЕССОРОМ ПРИ ПОДДЕРЖКЕ ОПЕРАЦИОННОЙ СИСТЕМЫ). Различают:

- СТАТИЧЕСКОЕ СВЯЗЫВАНИЕ, КОГДА КОД НЕОБХОДИМЫХ ДЛЯ РАБОТЫ ПРОГРАММЫ БИБЛИОТЕЧНЫХ ФУНКЦИЙ ФИЗИЧЕСКИ ДОБАВЛЯЕТСЯ К КОДУ ОБЪЕКТНЫХ МОДУЛЕЙ ДЛЯ ПОЛУЧЕНИЯ ЗАГРУЗОЧНОГО МОДУЛЯ;
- ДИНАМИЧЕСКОЕ СВЯЗЫВАНИЕ, КОГДА В РЕЗУЛЬТИРУЮЩЕМ ЗАГРУЗОЧНОМ МОДУЛЕ ПРОСТАВЛЯЮТСЯ ЛИШЬ ССЫЛКИ НА КОД НЕОБХОДИМЫХ БИБЛИОТЕЧНЫХ ФУНКЦИЙ; САМ КОД БУДЕТ РЕАЛЬНО ДОБАВЛЕН К ЗАГРУЗОЧНОМУ МОДУЛЮ ТОЛЬКО ПРИ ЕГО ИСПОЛНЕНИИ.

При статическом связывании загрузочные модули получаются очень большого размера. Поэтому подавляющее большинство современных операционных систем использует динамическое связывание, несмотря на то, что при этом начальная загрузка процесса на исполнение медленнее, чем при статическом связывании из-за необходимости поиска и загрузки кода нужных библиотечных функций (часто только тех из них, которые не были загружены для других процессов). При этом для избежания недетерминированной задержки на загрузку программы на исполнение все необходимые процессы реального времени запускают при старте системы (заранее, а не по требованию).



2. ТИПЫ ЗАДАЧ СИСТЕМ РЕАЛЬНОГО ВРЕМЕНИ



КАТЕГОРИИ ЗАДАЧ СИСТЕМ РЕАЛЬНОГО ВРЕМЕНИ

Всякий процесс содержит одну или несколько задач. ОПЕРАЦИОННАЯ СИСТЕМА ПОЗВОЛЯЕТ ЗАДАЧЕ ПОРОЖДАТЬ НОВЫЕ ЗАДАЧИ. Задачи, по своей манере действовать, можно разделить на 3 категории:



Циклические задачи.
Характерны для процессов управления и интерактивных процессов.



Периодические задачи. Характерны для многих технологических процессов и задач синхронизации.



Импульсные задачи. Характерны для задач сигнализации и асинхронных технологических процессов.



ИДЕНТИФИКАТОР ПРОЦЕССА (PID - PROCESS IDENTIFICATOR)

ТИП (ИЛИ КЛАСС) ПРОЦЕССА

ПРИОРИТЕТ ПРОЦЕССА

ПЕРЕМЕННУЮ СОСТОЯНИЯ ПРОЦЕССА

ЗАЩИЩЕННУЮ ОБЛАСТЬ ХРАНЕНИЯ РЕГИСТРОВ ПРОЦЕССА

ИНФОРМАЦИЮ О РЕСУРСАХ

АДРЕС ДЛЯ ОРГАНИЗАЦИИ ОБЩЕНИЯ С ДРУГИМИ ПРОЦЕССАМИ

ПАРАМЕТРЫ ВРЕМЕНИ ЗАПУСКА

АДРЕС ЗАДАЧИ НА ДИСКЕ И АДРЕС ВЫГРУЗКИ ИНФОРМАЦИИ



СИСТЕМА ПОДДЕРЖИВАЕТ ИХ ОБОСОБЛЕННОСТЬ ПРОЦЕССОВ ДЛЯ ТОГО, ЧТОБЫ ЗАЩИТИТЬ ОДИН ПРОЦЕСС ОТ ДРУГОГО, ПОСКОЛЬКУ ОНИ, СОВМЕСТНО ИСПОЛЬЗУЯ ВСЕ РЕСУРСЫ ВЫЧИСЛИТЕЛЬНОЙ СИСТЕМЫ, КОНКУРИРУЮТ ДРУГ С ДРУГОМ: У КАЖДОГО ПРОЦЕССА ИМЕЕТСЯ СВОЕ ВИРТУАЛЬНОЕ АДРЕСНОЕ ПРОСТРАНСТВО, КАЖДОМУ ПРОЦЕССУ НАЗНАЧАЮТСЯ СВОИ РЕСУРСЫ – ФАЙЛЫ, ОКНА, СЕМАФОРЫ И Т.Д. В ОБЩЕМ СЛУЧАЕ ПРОЦЕССЫ (ЗАДАЧИ) НИКАК НЕ СВЯЗАНЫ МЕЖДУ СОБОЙ И МОГУТ ПРИНАДЛЕЖАТЬ ДАЖЕ РАЗНЫМ ПОЛЬЗОВАТЕЛЯМ, РАЗДЕЛЯЮЩИМ ОДНУ ВЫЧИСЛИТЕЛЬНУЮ СИСТЕМУ. ПРИ ЭТОМ ИМЕННО СИСТЕМА БЕРЕТ НА СЕБЯ РОЛЬ АРБИТРА В КОНКУРЕНЦИИ МЕЖДУ ПРОЦЕССАМИ ПО ПОВОДУ РЕСУРСОВ.

ЖЕЛАТЕЛЬНО ИМЕТЬ ВОЗМОЖНОСТЬ ЗАДЕЙСТВОВАТЬ ВНУТРЕННИЙ ПАРАЛЛЕЛИЗМ, КОТОРЫЙ МОЖЕТ БЫТЬ В САМИХ ПРОЦЕССАХ. ТАКОЙ ВНУТРЕННИЙ ПАРАЛЛЕЛИЗМ ВСТРЕЧАЕТСЯ ДОСТАТОЧНО ЧАСТО И ЕГО ИСПОЛЬЗОВАНИЕ ПОЗВОЛЯЕТ УСКОРИТЬ ИХ РЕШЕНИЕ. В ОДНОПРОЦЕССОРНОЙ СИСТЕМЕ ЗАДАЧИ РАЗДЕЛЯЮТ МЕЖДУ СОБОЙ ПРОЦЕССОРНОЕ ВРЕМЯ ТАК ЖЕ, КАК ЭТО ДЕЛАЮТ ОБЫЧНЫЕ ПРОЦЕССЫ, А В МУЛЬТИПРОЦЕССОРНОЙ СИСТЕМЕ МОГУТ ВЫПОЛНЯТЬСЯ ОДНОВРЕМЕННО, ЕСЛИ НЕ ВСТРЕЧАЮТ КОНКУРЕНЦИИ ИЗ-ЗА ОБРАЩЕНИЯ К ИНЫМ РЕСУРСАМ.

ТАКИМ ОБРАЗОМ, ГЛАВНОЕ, ЧТО ОБЕСПЕЧИВАЕТ МНОГОПОТОЧНОСТЬ, - ЭТО ВОЗМОЖНОСТЬ ПАРАЛЛЕЛЬНО ВЫПОЛНЯТЬ НЕСКОЛЬКО ВИДОВ ОПЕРАЦИЙ В ОДНОЙ ПРИКЛАДНОЙ ПРО-ГРАММЕ. ПАРАЛЛЕЛЬНЫЕ ВЫЧИСЛЕНИЯ (А, СЛЕДОВАТЕЛЬНО, И БОЛЕЕ ЭФФЕКТИВНОЕ ИСПОЛЬЗОВАНИЕ РЕСУРСОВ ЦЕНТРАЛЬНОГО ПРОЦЕССОРА, И МЕНЬШЕЕ СУММАРНОЕ ВРЕМЯ ВЫПОЛНЕНИЯ ЗАДАЧ) ТЕПЕРЬ УЖЕ ЧАСТО РЕАЛИЗУЮТСЯ НА УРОВНЕ ЗАДАЧ, И ПРОГРАММА, ОФОРМЛЕН-НАЯ В ВИДЕ НЕСКОЛЬКИХ ЗАДАЧ (ПОТОКОВ, НИТЕЙ) В РАМКАХ ОДНОГО ПРОЦЕССА, МОЖЕТ БЫТЬ ВЫПОЛНЕНА БЫСТРЕЕ ЗА СЧЕТ ПАРАЛЛЕЛЬНОГО ВЫПОЛНЕНИЯ ЕЕ ОТДЕЛЬНЫХ ЧАСТЕЙ.



3. КЛАССЫ СИСТЕМ РЕАЛЬНОГО ВРЕМЕНИ



Количество операционных систем реального времени, несмотря на их специфику, очень велико. Сама специфика применения операционных систем реального времени требует гарантий надежности, причем гарантий, в том числе и юридических - этим, видимо, можно объяснить тот факт, что среди некоммерческих систем реального времени нет сколько-нибудь популярных.

Среди коммерческих систем реального времени можно выделить группу ведущих систем - по объемам продаж и по популярности. Это системы: **VxWorks**, **OS-9**, **rSOS**, **LynxOS**, **QNX**, **VRTX**.

Различают следующие классы СРВ:

ИСПОЛНИТЕЛЬНЫЕ СИСТЕМЫ
РЕАЛЬНОГО ВРЕМЕНИ

ЯДРА РЕАЛЬНОГО ВРЕМЕНИ

UNIX'Ы РЕАЛЬНОГО
ВРЕМЕНИ



Признаки систем этого типа - различные платформы для систем разработки и исполнения. Приложение реального времени разрабатывается на host-компьютере (компьютере системы разработки), затем компонуется с ядром и загружается в целевую систему для исполнения. Как правило, приложение реального времени - это одна задача и параллелизм здесь достигается с помощью нитей (threads).

Системы этого типа обладают рядом достоинств, среди которых основным достоинством является высокая скорость и реактивность системы.

Главная причина высокой реактивности систем этого типа - наличие только нитей (потоков) и, следовательно, малое время переключения контекста между ними (в отличие от процессов). С этим главным достоинством связан и ряд недостатков:

- зависание всей системы при зависании нити;
- проблемы с динамической загрузкой новых приложений.

Кроме того, системы разработки для продуктов этого класса традиционно дороги (порядка \$20000). Однако, необходимо отметить, что качество и функциональность систем разработки в этом классе традиционно являются хорошими.

Наиболее ярким представителем систем этого класса является операционная система VxWorks. Область применения - компактные системы реального времени с хорошими временами реакции.



В ЭТОТ КЛАСС ВХОДЯТ СИСТЕМЫ С МОНОЛИТНЫМ ЯДРОМ, ГДЕ И СОДЕРЖИТСЯ РЕАЛИЗАЦИЯ ВСЕХ МЕХАНИЗМОВ РЕАЛЬНОГО ВРЕМЕНИ ЭТИХ ОПЕРАЦИОННЫХ СИСТЕМ. **И**СТОРИЧЕСКИ СИСТЕМЫ ЭТОГО ТИПА БЫЛИ ХОРОШО СПРОЕКТИРОВАНЫ. **В** ОТЛИЧИЕ ОТ СИСТЕМ ДРУГИХ КЛАССОВ, 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: OS-9, 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.



Исторически системы реального времени создавались в эпоху расцвета и бума UNIX'a и поэтому многие из них содержат те или иные заимствования из этой красивой концепции операционной системы (пользовательский интерфейс, концепция процессов).

Часть разработчиков операционных систем реального времени попыталась просто переписать ядро UNIX, сохранив при этом интерфейс пользовательских процессов с системой, насколько это было возможно. Реализация этой идеи не была слишком сложной, поскольку не было препятствия в доступе к исходным текстам ядра, а результат оказался замечательным. Получили и реальное время, и весь набор пользовательских приложений - компиляторы, пакеты, различные инструментальные системы.

В этом смысле создателям систем первых двух классов пришлось потрудиться не только при создании ядра реального времени, но и продвинутых систем разработки.

Однако Unix'ы реального времени имеют следующие недостатки: системы реального времени получаются достаточно большими и реактивнее, чем реактивность систем первых двух классов.

Наиболее популярным представителем систем этого класса является операционная система реального времени Lynx OS.





После появления **Windows NT**, сразу несколько фирм объявили о создании расширений реального времени для **Windows NT**. Появление в свое время **UNIX**'ов реального времени означало попытку применить господствующую программную технологию для создания приложений реального времени. Появление расширений реального времени для **Windows NT** имеет те же корни, ту же мотивацию. Огромный набор прикладных программ под **Windows**, мощный программный интерфейс **WIN32**, большое количество специалистов, знающих эту систему. Сблзнительно было получить в системе реального времени все эти возможности.

Несмотря на то, что **Windows NT** создавалась как сетевая операционная система, и сочетание слов "**Windows NT**" и "реальное время" многими воспринимается как нонсенс, в нее при создании были заложены элементы реального времени. Она имеет двухуровневую систему обработки прерываний (**ISR** и **DPC**), классы реального времени (процессы с приоритетами **16-32** планируются в соответствии с правилами реального времени). Причина появления этих элементов кроется в том, что у разработчиков **Windows NT** за плечами был опыт создания классической для своего времени операционной системы реального времени **RSX11M** (для компьютеров фирмы **DEC**).

Анализ возможностей **Windows NT** показывает, что эта система не годится для построения систем жесткого реального времени (система непредсказуема - время выполнения системных вызовов и время реакции на прерывания сильно зависит от загрузки системы; система велика; нет механизмов защиты от зависаний). Поэтому даже в системах мягкого реального времени **Windows NT** может быть использована только при выполнении целого ряда рекомендаций и ограничений.

Разработчики расширений пошли двумя путями.

В первом случае они использовали ядра классических операционных систем реального времени в качестве дополнения к ядру Windows NT. Таковы решения фирм "LP ELEKTRONIKS" и "RADISYS". В первом случае параллельно с Windows NT (на одном компьютере) работает операционная система VxWORKS, во-втором случае - INTIME. Кроме того, предоставляется набор функций для связи приложений реального времени и приложений Windows NT. Вот как, например, это выглядит у LP ELEKTRONIKS: вначале стандартным образом загружается Windows NT, затем с помощью специального загрузчика загружается операционная система VxWORKS, распределяя под себя необходимую память Windows (что в дальнейшем позволяет избежать конфликтов памяти между двумя ОС). После этого полной "хозяйкой" на компьютере уже становится VxWORKS, отдавая процессор ядру Windows NT только в случаях, когда в нем нет надобности для приложений VxWORKS. В качестве канала для синхронизации и обмена данными между Windows NT и VxWORKS служат псевдодрайверы TCP/IP в обеих системах. Технология использования двух систем на одном компьютере следующая - работу с объектом выполняет приложение реального времени, передавая затем результаты приложениям Windows NT для обработки, передачи в сеть, архивирования.



ВТОРОЙ ВАРИАНТ РАСШИРЕНИЙ РЕАЛЬНОГО ВРЕМЕНИ ФИРМЫ VENTURCOM ВЫГЛЯДИТ ИНАЧЕ: ЗДЕСЬ СДЕЛАНА ПОПЫТКА "ИНТЕГРИРОВАТЬ" РЕАЛЬНОЕ ВРЕМЯ В WINDOWS NT ПУТЕМ ИССЛЕДОВАНИЯ ПРИЧИН ЗАДЕРЖЕК И ЗАВИСАНИЙ И УСТРАНЕНИЯ ЭТИХ ПРИЧИН С ПОМОЩЬЮ ПОДСИСТЕМЫ РЕАЛЬНОГО ВРЕМЕНИ. РЕШЕНИЯ ФИРМЫ "VENTURCOM" (RTX 4.2) БАЗИРУЮТСЯ НА МОДИФИКАЦИЯХ УРОВНЯ АППАРАТНЫХ АБСТРАКЦИЙ WINDOWS NT (HAL - HARDWARE ABSTRACTION LAYER) - ПРОГРАММНОГО СЛОЯ, ЧЕРЕЗ КОТОРЫЙ ДРАЙВЕРЫ ВЗАИМОДЕЙСТВУЮТ С АППАРАТУРОЙ. МОДИФИЦИРОВАННЫЙ HAL И ДОПОЛНИТЕЛЬНЫЕ ФУНКЦИИ (RTAPI) ОТВЕЧАЮТ ТАКЖЕ ЗА СТАБИЛЬНОСТЬ И НАДЕЖНОСТЬ СИСТЕМЫ, ОБЕСПЕЧИВАЯ ОТСЛЕЖИВАНИЕ КРАХА WINDOWS NT, ЗАВИСАНИЯ ПРИЛОЖЕНИЙ ИЛИ БЛОКИРОВКУ ПРЕРЫВАНИЙ. В СОСТАВ RTX ВХОДИТ ТАКЖЕ ПОД-СИСТЕМА РЕАЛЬНОГО ВРЕМЕНИ RTSS, С ПОМОЩЬЮ КОТОРОЙ WINDOWS NT РАСШИРЯЕТСЯ ДОПОЛНИТЕЛЬНЫМ НАБОРОМ ОБЪЕКТОВ (АНАЛОГИЧНЫМ СТАНДАРТНЫМ, НО С АТТРИБУТАМИ РЕАЛЬНОГО ВРЕМЕНИ). СРЕДИ НОВЫХ ОБЪЕКТОВ -НИТИ (ПОТОКИ, ПРОЦЕССЫ) РЕАЛЬНОГО ВРЕМЕНИ, КОТОРЫЕ УПРАВЛЯЮТСЯ СПЕЦИАЛЬНЫМ ПЛАНИРОВЩИКОМ РЕАЛЬНОГО ВРЕМЕНИ (256 ФИКСИРОВАННЫХ ПРИОРИТЕТОВ, АЛГОРИТМ - ПРИОРИТЕТНЫЙ С ВЫТЕСНЕНИЕМ). ПОБОЧНЫМ РЕЗУЛЬТАТОМ RTX ЯВЛЯЕТСЯ ВОЗМОЖНОСТЬ ПРОСТОГО СОЗДАНИЯ ПРОГРАММ УПРАВЛЕНИЯ УСТРОЙСТВАМИ, ТАК КАК СРЕДИ ФУНКЦИЙ RTAPI ЕСТЬ И ФУНКЦИИ РАБОТЫ С ПОРТАМИ ВВОДА-ВЫВОДА И ФИЗИЧЕСКОЙ ПАМЯТЬЮ. РЕШЕНИЯ VENTURCOM ХАРАКТЕРНЫ ЕЩЕ И ТЕМ, ЧТО ОНИ ПРЕДОСТАВЛЯЮТ ДЛЯ NT ВОЗМОЖНОСТЬ КОНФИГУРИРОВАНИЯ WINDOWS NT И СОЗДАНИЯ ВСТРОЕННЫХ КОНФИГУРАЦИЙ (БЕЗ ДИСКОВ, КЛАВИАТУРЫ И МОНИТОРА, ИНТЕГРАТОР КОМПОНЕНТОВ - CI).

НЕСМОТРИ НА ВСЮ НЕОДНОЗНАЧНОСТЬ ОТНОШЕНИЯ ТРАДИЦИОННЫХ ПОЛЬЗОВАТЕЛЕЙ СИСТЕМ РЕАЛЬНОГО ВРЕМЕНИ КО ВСЕМУ, ЧТО СВЯЗАНО С "MICROSOFT", НЕОБХОДИМО КОНСТАТИРОВАТЬ ФАКТ: ПОЯВИЛСЯ НОВЫЙ КЛАСС ОПЕРАЦИОННЫХ СИСТЕМ РЕАЛЬНОГО ВРЕМЕНИ - А ИМЕННО РАСШИРЕНИЯ РЕАЛЬНОГО ВРЕМЕНИ ДЛЯ WINDOWS NT. РЕЗУЛЬТАТЫ НЕЗАВИСИМЫХ ТЕСТИРОВАНИЙ ЭТИХ ПРОДУКТОВ ПОКАЗЫВАЮТ, ЧТО ОНИ МОГУТ БЫТЬ В ПЕРСПЕКТИВЕ ИСПОЛЬЗОВАНЫ ДЛЯ ПОСТРОЕНИЯ СИСТЕМ ЖЕСТКОГО РЕАЛЬНОГО ВРЕМЕНИ ПОСЛЕ СООТВЕТСТВУЮЩЕЙ ДОРАБОТКИ. ОБЛАСТЬ ПРИМЕНЕНИЯ РАСШИРЕНИЙ РЕАЛЬНОГО ВРЕМЕНИ - БОЛЬШИЕ СИСТЕМЫ РЕАЛЬНОГО ВРЕМЕНИ, ГДЕ ТРЕБУЕТСЯ



THANK YOU!

