

ОСРВ Багет 2.7. Функциональное описание программы

Операционная система реального времени Багет 2.7 (далее – ОСРВ Багет 2.7) предназначена для создания программного обеспечения вычислительных систем, комплексов и средств автоматизированного управления, работающих в режиме реального масштаба времени. В настоящее время она может применяться на ЭВМ с микропроцессорами R3081, 1890BM2T (КОМДИВ32), RM7000, 1890BM5Ф (КОМДИВ64-СМП), 1890BM6Я (КОМДИВ64-РИО), 1890BM7Я (КОМДИВ128-РИО), 1890BM8Я (КОМДИВ64-М), 1890BM9Я, 1890BM108, 1890BM118, 1890BM128, K5500BK018, Intel (микропроцессор i486 и совместимые с ним).

Соответствие стандартам

При разработке операционной системы использовались следующие международные стандарты:

- POSIX 1003.1, стандарт на мобильные операционные системы (программный интерфейс);
- стандарт C, описывающий язык и библиотеки языка C.

Изначально POSIX разрабатывался с целью устранить разноречивость в различных UNIX-системах и тем самым способствовать мобильности прикладных программ. UNIX был ориентирован не на решение задач реального времени, а на обеспечение одновременного доступа к ЭВМ со стороны нескольких пользователей, которые слабо взаимодействуют друг с другом или вообще не взаимодействуют.

Редакция POSIX 1996 года уже охватывает (в качестве необязательной части) основные функции операционных систем реального времени:

- потоки управления (threads),
- сигналы реального времени,
- средства синхронизации (семафоры, мьютексы и условные переменные);
- очереди сообщений,
- высокоточные таймеры,
- асинхронный ввод/вывод.

Операционная система полностью соответствует стандарту POSIX 1996 г. в части, относящейся к реальному времени. Те части стандарта, которые не относятся к системам реального времени (традиционный UNIX) реализованы не полностью.

В рамках стандарта C 1990 г. реализованы математические функции (sin, exp, log и др.), функции обработки символов и строк, функции распределения памяти и др. Эти функции хорошо знакомы всем тем, кто разрабатывает программы на языке C. Они входят в состав таких хорошо известных средств разработки программ на языке C, как Borland C и Microsoft Development Studio C/C++.

Средства протоколирования в основном соответствуют стандарту POSIX 2001 г. При разработке средств обработки исключительных ситуаций с плавающей точкой использовался стандарт IEEE 754, а также стандарт C 1999 г.

Для эмуляции протокола Ethernet в многопроцессорных системах, использующих шину VME, используется стандарт ANSI/VITA. Этот стандарт позволяет взаимодействовать через шину VME и общую память различным как по аппаратуре, так и по используемой операционной системе процессорным платам.

Использование стандарта при разработке операционной системы облегчает создание мобильных прикладных программ. Если операционная система имеет уникальный интерфейс, то перенос прикладной программы с одной операционной системы на другую является довольно трудоемким делом. В этом случае пользователь ОС попадает в зависимость от ее разработчика. Эта зависимость особенно ощутима, если нет версии ОС для аппаратной платформы, на которую нужно перенести прикладную программу. Перенос прикладных программ с одной ОС на другую значительно проще, если обе они удовлетворяют стандартам.

Отметим также, что использование стандартов облегчает освоение системы и помогает при подготовке кадров программистов.

Мобильность

С целью повышения мобильности операционная система разбита на три части:

- не зависящая от аппаратуры,
- зависящая только от типа центрального процессора,
- пакет поддержки модуля (поставляется отдельно).

Не зависящая от аппаратуры часть ОС имеет самый большой объем и написана полностью на языке C. Перенос этой части на другие платформы несложен.

Та часть ОС, которая зависит только от типа процессора, написана на языке С или на Ассемблере и имеет сравнительно небольшой объем. Туда входят, например, функции запоминания и восстановления контекста, пролог и эпилог диспетчера прерываний.

Пакет поддержки модуля (ППМ) содержит ту часть ОС, которая зависит от конкретной ЭВМ (модуля). ППМ, в частности, содержит драйверы устройств и диспетчер прерываний (за исключением пролога и эпилога).

Отметим, что граница между этими частями не является жесткой. Например, некоторые, не зависящие от аппаратуры функции, могут быть переписаны с использованием Ассемблера с целью повышения скорости. В этом случае они станут зависимыми от типа процессора.

Внесение изменений в драйверы, а также разработка новых драйверов и включение их в операционную систему производится путем внесения изменений в исходные тексты ППМ. При этом нет необходимости вносить изменения в ядро операционной системы.

Архитектура программного обеспечения

При использовании операционной системы программное обеспечение системы реального времени состоит из операционной системы и прикладной программы. В системах реального времени граница между операционной системой и прикладной программой не так резко очерчена, как в случае традиционных операционных систем. В частности, в системах реального времени прикладная программа может непосредственно вызывать те функции, которые в случае традиционных ОС могут выполняться только операционной системой (например, запрет или разрешение прерываний). Такая "свобода" позволяет повысить эффективность системы реального времени, но накладывает большую ответственность на прикладного программиста.

Прикладная программа выполняется в режиме ядра и виртуальная память не применяется. Это позволяет заметно сократить время обращения к функциям операционной системы, сократить время переключения контекста и увеличить скорость выполнения прикладных программ и сделать ее предсказуемой.

Прикладная программа представляет собой совокупность потоков управления (пользовательских потоков управления). При инициализации системы порождается корневой поток управления прикладной программой, при необходимости другие потоки управления прикладная программа порождает динамически.

Операционная система состоит из ядра и системных потоков управления. Ядро выполняет функции планирования, синхронизации и взаимодействия потоков управления, а также низкоуровневые операции ввода/вывода. Функции ядра выполняются в контексте вызвавшего его потока управления или функции обработки прерывания. Микроядро представляет собой небольшую часть ядра ОС, функциями которой пользуются другие части ОС. Микроядро содержит функции управления потоками нижнего уровня (включая планировщик) и быстрые средства синхронизации (взаимоисключения). Все другие функции (например, захват и освобождение семафора, низкоуровневые операции ввода/вывода) выполняются вне микроядра, используя его функции.

Системные потоки выполняют более сложные функции операционной системы, такие как ввод/вывод информации по сети или обмен информацией с файловой системой. Использование системных потоков для сложных и протяженных во времени функций ОС позволяет продолжать работу в параллель с выполнением этих функций. В рамках таких потоков выполняется часть функций, которые обычно выполняются в рамках обработчиков прерываний драйвера. Во-первых, такой подход дает определенные удобства при программировании, так как не все функции ОС, доступные в контексте потока, можно выполнять в контексте прерывания. Во-вторых, сокращается время выполнения функций обработки прерываний, что особенно важно для систем реального времени. Действительно, если низкоприоритетная задача ведет интенсивный ввод/вывод, то она будет мешать выполнению высокоприоритетной задачи, так как функция обработки прерывания драйвера более приоритетна, чем любой поток управления. Потери (времени) будут тем больше, чем больше времени будет занимать функция обработки прерываний.

Объектно-ориентированный подход активно применялся при разработке ядра ОС, хотя объектно-ориентированные языки программирования не использовались. Каждый объект представляет собой структуру. Однотипные объекты объединяются в классы.

Класс имеет описатель класса – структуру, которая содержит идентификатор класса, функции обработки объектов класса (методы), а также общие для всех объектов класса переменные. Каждый объект содержит указатель на описатель класса, к которому он относится.

Временные характеристики

При оценке систем реального времени используются две важнейшие характеристики:

- время ответа на прерывание,
- время ответа потока управления.

Время ответа на прерывание – это время между моментом, когда был выставлен запрос на прерывание, и моментом, когда начала выполняться первая команда функции обработки прерывания. Оно включает в себя, в частности, задержку прерывания – время, на которое ядро запрещает прерывания.

Время ответа потока управления – это время между моментом, когда был выставлен запрос на прерывание, и моментом, когда начала выполняться первая команда потока, который должен отреагировать на это прерывание. Оно включает в себя, в частности, время ответа на прерывание, задержку планирования и время переключения контекста.

Задержка планирования – это промежуток времени, когда планирование (переключения потоков управления) запрещено. Запрещение планирования является широко используемым методом взаимного исключения для защиты критических ресурсов. Для повышения скорости реакции системы важно, чтобы длительность задержки прерывания и задержки планирования была минимальна.

Время переключения контекста – это время, требующееся для того, чтобы одна задача прекратила свою работу, а другая начала.

ОСРВ Багет 2.7 спроектирована так, что наибольшая задержка прерывания происходит при входе в прерывания. В остальных случаях задержка прерывания незначительна. Прерывания запрещаются на время установки в (изъятия из) неупорядоченную очередь, на время увеличения (уменьшения) некоторых счетчиков, на время установки или снятия некоторых флагов. Отметим, что диспетчер прерываний зависит от архитектуры ЭВМ и входит в состав ППМ (пакета поддержки модуля). Таким образом, максимальная задержка прерывания определяется ППМ.

Запрещение планирования в ОС производится на время установки описателя потока, а также некоторых других объектов в упорядоченную очередь. В силу этого переключение потоков управления возможно и при выполнении функций ОС.

Средства разработки

Для разработки прикладного программного обеспечения используется комплекс, состоящий из двух ЭВМ, соединенных по сети:

- инструментальная ЭВМ (компьютер, с операционной системой типа UNIX),
- целевая ЭВМ (ЭВМ, для которой разрабатывается программное обеспечение).

Разработка программного обеспечения ведется на инструментальной ЭВМ. Средства разработки позволяют оттранслировать программу, написанную на языках С и Ассемблер, а также отлаживать программу, загруженную в целевую машину.

Установка программного изделия «ОСРВ БАГЕТ 2.7»

Для установки ОСРВ Багет 2.7 на инструментальную ЭВМ (ИЭВМ) необходимо выполнить следующие операции.

Перейти на машинный носитель дистрибутива в каталог выбранной архитектуры.

Выполнить команду

```
sh install <dir>
```

где <dir> – каталог, в который будет установлена ОС РВ Багет 2.7.

Перейти на машинный носитель дистрибутива в каталог «doc». Выполнить команду

```
sh install <dir>
```

где <dir> – каталог, в который будет установлена интерактивная документация ОСРВ Багет 2.7.

Руководство пользователя

1. Функциональные характеристики

Операционная система реального времени Багет 2.7 (далее ОСРВ Багет 2.7) предназначена для создания программного обеспечения вычислительных систем, комплексов и средств автоматизированного управления, работающих в режиме реального масштаба времени.

1.1 Соответствие стандартам

Большая часть интерфейсов стандарта POSIX (the Portable Operating System Interface/Мобильной Операционной системы) 1003.b, определяющего услуги операционных систем реального времени, реализована в ОСРВ Багет 2.7. Использование этих интерфейсов облегчает пользователям переход на другую операционную систему, также поддерживающую указанный стандарт.

В состав ОСРВ Багет 2.7 входят:

- ядро;
- командный интерпретатор;
- средства конфигурирования и масштабирования;
- загрузчик модулей;
- средства отладки;
- локальную файловую систему;
- средства обеспечения удаленного доступа;
- сетевые средства,
- веб-сервер,
- интерфейс системы безопасности,
- средства протоколирования событий операционной системы и прикладной программы.

2 Эксплуатационные характеристики

2.1 Системные требования

ОСРВ Багет 2.7 может применяться на ЭВМ с микропроцессорами R3081, 1890ВМ2Т (КОМДИВ32), RM7000, 1890ВМ5Ф (КОМДИВ64-СМП), 1890ВМ6Я (КОМДИВ64-РИО), 1890ВМ7Я (КОМДИВ128-РИО), 1890ВМ8Я (КОМДИВ64-М), 1890ВМ9Я, 1890ВМ108, 1890ВМ118, 1890ВМ128, K5500BK018, Intel (микропроцессор i486 и совместимые с ним).

2.2 Входные и выходные данные

Входными данными ОСРВ Багет 2.7 являются исходные, объектные и библиотечные файлы, подключаемые в проект сборки образа ОСРВ.

Выходными данными является исполняемый образ ОСРВ Багет 2.7, предназначенный для загрузки на целевую ЭВМ.

2.3 Основные возможности

Ядро ОСРВ Багет 2.7 обеспечивает многопоточность с динамическим планированием по приоритетам, синхронизацию потоков и обработку прерываний. Если поток создается с режимом квантования времени, то управление по истечении кванта передается готовому к выполнению потоку равного приоритета (если таковой имеется).

Командный интерпретатор позволяет в режиме диалога вызывать как функции ОСРВ Багет 2.7, так и пользовательские функции.

Средства конфигурирования и масштабирования позволяют создавать образ ОСРВ Багет 2.7, отвечающий требованиям пользователя.

Загрузчик модулей дает возможность динамически грузить объектные модули на целевую ЭВМ по мере необходимости, а также удалять ранее загруженные модули.

ОСРВ Багет 2.7 содержит набор средств отладки, предоставляющих возможность:

- видеть текущее состояние системы;
- дизассемблировать загруженный код;
- размещать точки останова и выполнять код в пошаговом режиме;
- смотреть и изменять значения переменных и областей памяти.

Локальная файловая система поддерживает устройства с жесткими и гибкими дисками в формате VFAT.

ОСРВ Багет 2.7 располагает клиентскими частями сетевых файловых систем NFS и FTP, что допускает работу с удаленными файлами и каталогами.

Программный интерфейс ОСРВ Багет 2.7 обеспечивает возможность функционирования ОС в режиме веб-сервера с поддержкой НТТР.

Пользовательский интерфейс средств обеспечения безопасности для автоматизированного режима в ОСРВ Багет 2.7 поддерживает два режима работы:

- режим специализированного командного интерпретатора;
- режим веб-сервера.

В состав ОС имеются средства протоколирования событий операционной системы и прикладной программы. Результаты протоколирования направляются в файл, который впоследствии можно просматривать и анализировать с помощью программного изделия «Трассировщик ОСРВ 2.7».

3 Разработка прикладных программ

Разработка пользовательских программ, работающих под управлением ОСРВ Багет 2.7, предполагает наличие двух ЭВМ, одна из которых выполняет функции инструментальной машины, а другая – целевой. На инструментальной ЭВМ IBM PC устанавливается операционная система Astra Linux Special Edition версии 1.7 или Debian 12.4 и пакет кросс-разработки, она обеспечивает разработку пользовательских модулей, их компиляцию, выполняет создание образа ОСРВ с включением в него объектных файлов пользователя.

Инструментальная машина должна быть соединена с целевой ЭВМ с помощью локальной сети Ethernet. Это позволяет загрузить образ ОСРВ Багет 2.7 с инструментальной ЭВМ в целевую. После загрузки целевой ЭВМ пользователь с помощью локальных или удаленных средств отладки может наблюдать и корректировать поведение своего программного обеспечения.

В тех случаях, когда архитектура целевой ЭВМ – Intel, инструментальная ЭВМ может выполнять и роль целевой, при этом, после создания желаемого образа системы она перезагружается и образ ОСРВ Багет 2.7 с помощью загрузчика ОСРВ Багет 2.7 загружается в оперативную память, после чего можно начинать отладку. Описываемая процедура предполагает наличие на ЭВМ операционной системы Astra Linux Special Edition версии 1.7 или Debian 12.4 и установленного в соответствии с документацией загрузчика ОСРВ Багет 2.7 для Intel-архитектуры, и возможности поочередной их загрузки.

Сама ОСРВ Багет 2.7 не зависит от архитектуры целевой ЭВМ. Функционирование ее на целевой ЭВМ конкретной архитектуры обеспечивается пакетом поддержки модуля (ППМ), поставляемым вместе с операционной системой.