

Конфигурируемая ОС PB Embox для встроенных систем

Embox – операционная система жесткого реального времени, ориентированная на применение во встроенных системах

Сегодня существует большой класс задач, для решения которых применяют встроенные системы. Основными требованиями к таким системам часто являются их надежность и предсказуемость, поэтому в них применяют различные ОС реального времени. Кроме того, как правило, такие системы сильно отличаются друг от друга в аппаратной части как архитектурой процессора, так и доступными ресурсами. Естественно, что в подобных системах применяют различные алгоритмы планирования этих ресурсов, оптимизированные под конкретную задачу или класс задач.

ЗАО «Ланит-Терком» имеет богатый опыт в разработке встроенных систем различного уровня. Совместно с кафедрой системного программирования СПбГУ был организован исследовательский проект по созданию ОС реального времени для применения во встроенных системах. Проект основан на принципах свободного ПО и имеет свободную лицензию BSD.

Целью исследовательского проекта является создание ОС реального времени для применения в широком спектре встроенных систем. Код проекта уже используется в коммерческих продуктах ЗАО «Ланит-Терком» и других компаний. Кроме того, кафедра теоретической кибернетики СПбГУ планирует применять ядро Embox в различных роботизированных устройствах, в которых, как известно, нужна ОС жесткого реального времени.

Ядро ОС Embox построено на принципах модульности и конфигурируемости. В отличие от монолитных ядер (таких, как Linux) весь код организован в различные модули (библиотеки). Каждый модуль реализует свой интерфейс, например, интерфейс диспетчера памяти или планировщика задач. При конфигурации системы можно выбирать, какие модули будут включены в образ. Таким образом, достигается создание образа, нацеленного на конкретную задачу, будь то система для тестирования аппаратного обеспечения либо полноценная ОС, включающая сетевую и файловую подсистемы, а также управление виртуальной памятью.

Подобные принципы построения упрощают разработку и отладку отдельных модулей системы. Причем это относится как к драйверам, так и к различным модулям, обеспечивающим функции ядра (распределение ресурсов, межпроцессорный обмен и т.д.).

Еще одним важным свойством таких принципов построения является возможность применять различные стратегии планирования ресурсов. Например, можно создать образ ядра системы, обладающий классическими менеджерами ресурсов (диспетчером памяти, потоков и т.д.), а можно создать образ, запускающий определенный поток по определенному событию, что уменьшит накладные расходы на планирование задач. Или, например, создать образ, позволяющий привязывать задачу к каким-то ресурсам системы, процессорному ядру, области памяти и т.д. Такие свойства ОС Embox достигаются за счет возможности задать различные реализации системных диспетчеров ресурсов.

Как уже отмечалось, во встроенных системах используют различные процессорные архитектуры, иногда специально нацеленные на какой-то класс задач. Например, сетевые процессоры, построенные по принципам несимметричной многоядерности.

Embox портирован на такие широко применяемые во встроенных системах архитектуры, как MicroBlaze, ARM, SPARC v8 (LEON3), а также имеется возможность запускать его на x86 архитектуре. Вследствие модульной структуры процесс портирования на новые платформы – достаточно легкая задача. В отличие от ядра Linux, где необходимо портировать сразу все архитектурные части (прерывания, таймеры, виртуальную память и т.д.), Embox может быть запущен с минимальным набором подсистем, например, можно даже не реализовывать подсистему прерываний, а портировать только ввод/вывод и уже с помощью него отладить остальные части.

Важную роль для ОС играют средства разработки и отладки пользовательских программ, драйверов, а также самого ядра ОС.

Ядро ОС Embox написано на языке C, при этом использовался компилятор с открытым исходным кодом GCC, позволяющий осуществлять сборку системы для различных архитектурных платформ. Остальные средства

разработки также доступны в свободном доступе под свободной лицензией; в качестве IDE используется Eclipse, в качестве отладчика – GDB.

Сейчас для разработки пользовательских приложений используется язык C. Проект частично поддерживает стандарт POSIX, но может быть выбран другой интерфейс взаимодействия с пользовательским приложением. Например, для разработки управляющих программ роботов на кафедре теоретической кибернетики СПбГУ используют собственный интерфейс, нацеленный на выполнение подобных задач.

Соответственно можно для различных классов задач задавать специфичные интерфейсы взаимодействия ядра и пользовательского приложения. Это позволяет разрабатывать проблемно-ориентированные языки (DSL, domain-specific language) и на их основе графические CASE-средства. В итоге можно легко создавать и моделировать алгоритмы, ориентированные на конкретный тип задач. Например, кафедра кибернетики СПбГУ планирует использовать интегрированную среду визуального проектирования QReal.

Еще одной важной характеристикой для ОС является набор служебных библиотек, которые упрощают разработку и расширяют возможности данной ОС. К таким библиотекам прежде всего относят сетевой стек и файловую подсистему. Обе этих подсистемы поддерживаются в Embox.

Сетевой стек содержит реализацию протоколов стандартного IP-стека (IP, ICMP, UDP, TFTP, ARP), а также набор стандартных сетевых утилит, таких как ifconfig, route, arp, ping и так далее. Кроме того, Embox поддерживает стандартный интерфейс сокетов.

Файловая подсистема Embox основана на логической файловой системе, к которой можно подключать различные драйверы, реализующие ту или иную конкретную файловую систему. Это позволяет обращаться к файлам или физическим устройствам с помощью единого интерфейса (файлового интерфейса POSIX).

Таким образом, Embox – это ОС жесткого реального времени (время отклика – менее 10 мкс на процессоре AT91SAM7 с частотой 48 MHz) с открытым исходным кодом, которая обладает следующими свойствами:

- **Модульность** – позволяет включать только необходимые модули в образ системы, что важно для систем с ограниченными ресурсами.
- **Конфигурируемость** – позволяет задавать реализацию и параметры каждой подсистемы в отдельности, что дает возможность тонкой настройки под конкретную задачу. Это также позволяет создавать ОС, избавленные от основных дисциплинирующих принципов: разделения задач по времени и по используемой памяти.
- **Портируемость** – позволяет использовать наиболее удобную для конкретной задачи платформу или процессорную архитектуру.

Поскольку областью применения Embox являются прежде всего встраиваемые системы, а в данном сегменте часто применяются архитектуры процессоров, отличные от x86 (например, ARM), то Embox задумывался как портируемая ОС, т.е. предназначенная для переноса на различные архитектуры. Также предусмотрена возможность добавления новой платформы. Для этого основной код ядра разрабатывался на платформонезависимом языке C, а код, привязанный к конкретной платформе, вынесен в отдельную папку.

Embox позволяет гибко конфигурировать ядро под конкретную задачу и платформу. В том числе выбирать стратегии. В определенных случаях можно вообще отказаться от этих частей и создать образ ядра с необходимым минимумом интерфейсов. Например, контроллер на роботе Lego содержит flash-память объемом 256 kb и оперативную память 64 kb.

Данная ОС разрабатывается в рамках проекта с открытым исходным кодом, что способствует привлечению как студентов и разработчиков, так и потенциальных пользователей. Одним из ее основных преимуществ является нацеленность на задачи именно российского сегмента рынка.