

ВСТРОЕННЫЕ ОС ДЛЯ ОДНОПЛАТНЫХ КОМПЬЮТЕРОВ

С.В. Щекин (ЗАО "ИнтраСофт")

Рассматриваются возможности создания на основе модификаций свободно распространяемого ПО встроенных ОС для одноплатных компьютеров. Анализируются мероприятия, которые могут потребоваться при адаптации ОС общего назначения (Linux, BSD, Open Solaris) и их модификаций (RT-Linux, RTAI, XENOMAI, ADEOS) для инсталляции на одноплатных компьютерах.

В настоящее время просматривается тенденция к увеличению использования свободно распространяемого системного ПО в качестве основы для построения встроенных ОС, в том числе и для одноплатных компьютеров. Немаловажным фактором при этом является возможность легально использовать ОС без дополнительных затрат на покупку лицензий. Но это далеко не единственная причина. Существуют проблемы, связанные с переносом ранних версий прикладных программ, при использовании коммерческих ОС и т.д. В ряде случаев ОС нуждается в доработках, требующих наличия исходных кодов системы, что не всегда возможно или экономически целесообразно при работе с коммерческими продуктами (QNX, VX-Works).

Развитие одноплатных компьютеров накладывает свою специфику на ОС, которые используются вместе с ними. Как правило, размещение ОС, прикладного ПО и данных осуществляется на Flash-носителях (Compact-flash, PCMCIA-flash, Disk-on-Chip, встроенные Nor и Nand Flash). Следовательно, ОС должна иметь возможность запуска с Flash, поддерживать файловые системы на Flash и иметь средства защиты или восстановления файловой системы при внезапных выключениях питания. Одноплатные компьютеры часто работают в длительном необслуживаемом режиме, и поэтому файловые системы должны быть более стойкими к повреждениям, чем аналогичные системы на ПК. Несмотря на удешевление и увеличение емкости Flash-носителей, ОС должны занимать небольшой объем, гораздо меньший, чем ОС, размещаемые на жестких дисках. Один из наиболее действенных способов защитить ОС от повреждений является использование раздела памяти, содержащего ОС на Flash-носителе, в режиме "только чтение" и только в момент загрузки ОС. В этом случае файловая система и ОС размещаются в оперативной памяти, объем которой еще более ограничен. Ресурсы одноплатных компьютеров не так велики, особенно это относится к системам со сверхнизким энергопотреблением. В то же время набор периферийных устройств может быть достаточно большим. Часть периферийных устройств одноплатных компьютеров исключают возможность использования готовых драйверов и требуют их доработки или портирования из других ОС или версий системы. В зависимости от специфики решаемых задач к ОС могут предъявляться различные требования, в числе которых и временные ограничения, присущие ОС РВ. Однако далеко не все свободно распространяемые ОС им удовлетворяют. Для воз-

можности переноса прикладных программ на другие платформы необходимо соответствие интерфейсов прикладных программ (API – Application Programming Interface). Полезным свойством является соответствие API системы стандарту POSIX.

Свободно распространяемые ОС можно разделить на два класса: системы, которые изначально были предназначены для использования исключительно в качестве встроенных ОС (RTEMS, eCOS, ROME и др.), системы общего назначения (Linux, BSD, Open Solaris), которые могут быть использованы для создания на основе их модификаций встроенных ОС. Именно системы подобного типа в последнее время получают наибольшую популярность, в основном это относится к модификациям Linux.

Операционная система RTEMS имеет четкое разделение на аппаратно-зависимую и аппаратно-независимую части ядра ОС. При этом аппаратно-зависимая часть имеет очень небольшой объем, вследствие чего адаптация ОС к другим платформам значительно облегчается по сравнению с другими ОС.

В свою очередь аппаратно-независимая часть ядра не является монолитной и состоит из так называемых менеджеров, которые представляют компоненты ядра, отвечающие за ту или иную функцию ОС. Например, в RTEMS 4.6.6 существуют менеджеры периодического планирования, задач, таймеров, семафоров, регионов памяти, прерываний, ввода/вывода, мультипроцессорирования, сигналов, сообщений, событий и т.д. Многие менеджеры могут быть удалены из системы без влияния на другие функции ОС. Такой компонентный подход позволяет достаточно просто конфигурировать структуру ОС в зависимости от наличия сервисов, требуемых для решения конкретных задач.

Планировщик РВ является основной частью ядра ОС и может настраиваться в процессе выполнения. Он поддерживает: алгоритмы периодического планирования RMSA (Rate Monotonic Scheduling Algorithm) и циклического планирования RR (Round Robin) с настраиваемой дискретностью кванта времени; алгоритм планирования FIFO. Планирование является приоритетным, параметры РВ, например, минимальный период планирования периодического запуска задач, разрешение таймеров, точность определения времени являются изменяемыми при настройке ОС RTEMS.

Существует специальный компонент RTEMS, эмулирующий работу аппаратно-зависимой части в ОС UNIX (Linux, Solaris, HPUX), благодаря чему возможна установка аппаратно-независимых компонент ядра

Свободно распространяемого ПО - это двигатель программно-технического прогресса и характерная черта цивилизации

Журнал "Автоматизация в промышленности"

RTEMS поверх ОС Linux. Временные характеристики системы при этом не могут быть выше характеристик Linux, но становится возможным использование UNIX-средств для разработки и отладки приложений РВ RTEMS и самих компонент RTEMS. Система RTEMS использует представление процессов, средств синхронизации и коммуникации как отдельных объектов ОС, которые могут быть созданы, использованы и удалены при помощи соответствующих менеджеров. В этом смысле данную ОС РВ можно отнести к классу объектно-ориентированных. Каждый объект имеет символичный и внутренний идентификатор ОС. Внутренний идентификатор содержит идентификаторы класса объекта, данного экземпляра объекта и узла. Такая система позволяет быстро получить доступ к объекту по его идентификатору в мультипроцессорной системе. В большинстве менеджеров RTEMS предусмотрена поддержка мультипроцессорной обработки.

Отличительной особенностью RTEMS является поддержка различных типов API. Для большинства функций имеются интерфейсы стандартов POSIX и TRON, кроме того, возможен доступ через собственный API-интерфейс RTEMS. Встроенная ОС RTEMS после конфигурирования и настройки компонуется вместе с откомпилированным прикладным ПО в единый загружаемый образ, который можно загрузить и запустить с носителя.

Преимущества RTEMS по отношению к системам общего назначения: меньшие временные задержки; предсказуемый характер поведения планировщика и других подсистем; высокое разрешение функций таймера и получения времени для процессов RTEMS; компактность системы; относительно простой перенос системы на другую платформу; возможность настройки структуры и параметров ОС для конкретной системы; поддержка API POSIX, TRON, RTEMS одновременно; запуск ОС вместе со своими задачами как процесса UNIX, прозрачность компонент системы для отладочных средств UNIX.

Недостатки RTEMS: неполное соответствие POSIX; временные задержки и особенности реализации планировщика не позволяют в полной мере отнести RTEMS к системам с "жестким РВ"; меньшее число поддерживаемых устройств по отношению к системам общего назначения.

Одной из основных проблем, с которой сталкиваются при использовании RTEMS на новых одноплатных компьютерах, является необходимость доработки или создания драйверов устройств. Другой проблемой является ограниченный набор файловых систем, поддерживаемый ОС. Поэтому достаточно сложно защитить flash-разделы большого объема от повреждений.

Ядро ОС Linux развивалось как ядро системы общего назначения. Оно дополнено поддержкой потоков,

механизмами межпроцессных коммуникаций и управления планированием задач POSIX-1b. По умолчанию все процессы (прикладные программы пользователя и утилиты ОС) имеют нулевой приоритет, низший для ОС Linux. Алгоритм планирования задач, используемый по умолчанию, использует квантование времени и вытесняющую многозадачность. Для каждого процесса может быть выбрана индивидуальная дисциплина планирования в соответствии со спецификацией POSIX 1b (sched other, sched fifo, sched RR). Алгоритм по умолчанию (Sched Other) в Linux-реализации не допускает изменения приоритета процесса.

Для процессов РВ в Linux определены два алгоритма планирования Sched fifo и Sched RR. Процессы РВ могут иметь приоритеты 1...99. Таким образом, самый низкий приоритет любого процесса РВ Linux в любом случае будет выше приоритета обычного процесса Linux. Работа с потоками в Linux реализована через специальные библиотеки, часть из которых использует поддержку потоков в ядре системы. Соответствие API POSIX-1c (расширение POSIX-1, добавляющее потоки threads) обеспечивается на уровне библиотеки потоков libpthread. В Linux реализована поддержка средств межпроцессной коммуникации и синхронизации в соответствии с POSIX 1, 1b, 1c. В ОС поддерживаются семафоры, сигналы, таймеры, мьютексы, условные переменные, разделяемая память, каналы FIFO, очереди сообщений. Ввод/вывод реализован в соответствии со стандартом POSIX. Расширение стандарта POSIX для РВ поддерживается не в полном объеме, например, отсутствуют таймеры РВ с высоким разрешением. Для периодического запуска процессов в Linux единственным механизмом является использование в процессах обработчика сигнала от однократного или многократного таймера, предельное разрешение которого составляет 10000 мкс. Предельное разрешение функций запроса времени составляет 1000 мкс, в то время как специализированные ОС РВ позволяют работать с временными разрешениями на 3...4 порядка ниже. Неприятным моментом является существенная зависимость задержек от загрузки системы.

Основными достоинствами Linux как встроенной ОС являются: соответствие функций API спецификациям UNIX и POSIX; относительно простой перенос ПО с других UNIX-платформ; прозрачность приложений для отладочных средств UNIX; большое число драйверов устройств.

Основные недостатки Linux — использование функций таймера и времени с низким разрешением; большие задержки на функциях получения времени, ввода/вывода, обработки прерываний, межпроцессной коммуникации и синхронизации; нестабильность ряда временных параметров; чрезмерная избыточность системы.

С целью улучшения временных параметров появился ряд ОС РВ на основе Linux (RT-Linux, RTAI, XENO-MAI), используемых в качестве систем жесткого РВ.

Операционная система RT-Linux представляет собой ядро РВ, поддерживающее высокоскоростную обработку прерываний и заменяющее планировщик и некоторые другие сервисы Linux. Планировщик РВ RT-Linux при инициализации перехватывает прерывания таймера и за счет этого позволяет представить всю остальную часть ОС Linux как отдельную задачу, выполняемую наравне с другими задачами РВ в RT-Linux. Планировщиком RT-Linux поддерживается планирование с вытеснением на основе приоритетов с квантованием времени. Планировщик поддерживает однопроцессорные, многопроцессорные и симметричные системы, однако полная симметричность не поддерживается. Приложения Linux планируются штатным планировщиком Linux и выполняются во время, не используемое процессами РВ, когда RT-Linux передает управление Linux. Процессы Linux могут использовать библиотеки и средства взаимной синхронизации и коммуникации Linux.

Приложения РВ не могут использовать библиотеки и стандартные функции API Linux. ОС RT-Linux реализует собственный API-интерфейс ввода/вывода, функции работы с таймером, механизмами взаимной синхронизации и коммуникации процессов РВ.

Ввод/вывод API RT-Linux соответствует POSIX в минимальном наборе (поддерживает только режим чтения/записи с рядом ограничений). Процессы РВ могут взаимодействовать с обычными процессами Linux через каналы FIFO.

Необходимость введения отдельного API в RT-Linux, как и в других системах подобного типа, обусловлена невозможностью выполнения многих функций API Linux с малыми временными задержками или нестабильностью времени их выполнения.

По сравнению с ОС Linux система RT-Linux обеспечивает ряд преимуществ:

- очень небольшие временные задержки API RT-Linux для процессов RT-Linux;
- высокое разрешение функций таймера и получения времени для процессов RT-Linux;
- возможность запуска и параллельной работы обычных процессов Linux и RT-Linux.

Недостатками системы являются: невозможность запуска приложений, использующих API Linux как процессов РВ; нестандартная реализация ряда функций RT-Linux; неполное соответствие POSIX.

ОС RTAI так же, как и RT-Linux представляет собой ядро РВ. Она перехватывает все аппаратные прерывания за счет использования механизма RTHAL (Run-Time Hardware Abstraction Layer – уровень абстрактного оборудования РВ). Прерывания анализируются на предмет наличия их обработчиков среди процессов ядра RTAI и лишь при отсутствии таковых передаются задачам Linux.

Таким образом, RTHAL перехватывает прерывания аппаратного таймера, что позволяет запускать

планировщик RTAI ранее, чем управление может быть передано в Linux (кроме того, RTHAL способен обновлять часы РВ Linux, так что пропуск прерываний таймера никак не сказывается на их работе). Таким образом, Linux представляется как одна из задач ядра RTAI, обладающая наименьшим приоритетом.

Планировщик RTAI основан на алгоритмах вытесняющей многозадачности с разделением времени. Возможно периодическое планирование с высоким временным разрешением. Поддерживается мультипроцессорное планирование для полностью симметричных систем (автоматическая миграция задач по процессорам). Задачи RTAI используют интерфейс API RTAI, а не API Linux, имеющий собственную реализацию ввода/вывода, механизмов коммуникации и синхронизации процессов, таймеров, функций времени. Функции работы с таймером и временем позволяют определять интервалы с точностью до наносекунд.

В RTAI существуют дополнительные механизмы, позволяющие использовать планировщик РВ RTAI для приложений, использующих API Linux. Этот механизм называется LXRT (Linux Real-Time – РВ Linux). Фактически механизм позволяет использовать интерфейс между RTAI и процессами Linux, использующими LXRT. Механизм LXRT основан на задачах-агентах. Процессу в Linux, использующему LXRT, ставится в соответствие специальная задача-агент в RTAI. Она необходима только для инициализации структур данных, используемых RTAI для задач РВ, и служит для представления процесса Linux в ядре RTAI. Управление процессу Linux передается планировщиком RTAI через задачу-агент.

В результате процессы LXRT имеют больший приоритет, чем любые другие процессы Linux, и считаются выполняемыми в режиме РВ. Они до определенной степени могут использовать механизмы API RTAI и в то же время являются процессами Linux. Механизм LXRT также может использоваться и в целях отладки процессов жесткого РВ, которые впоследствии будут перенесены в ядро RTAI.

Помимо процессов RTAI и LXRT возможно выполнение обычных процессов Linux, которые планируются штатным планировщиком Linux в моменты времени, свободные от процессов RTAI и LXRT. Каналы FIFO позволяют осуществлять к ним доступ для процессов любого типа через любые API-интерфейсы (RTAI, Linux). Таким образом, в системе могут одновременно выполняться три типа процессов с возможностью взаимодействия между собой.

Преимущества RTAI по отношению к ОС Linux:

- небольшие временные задержки времени API RTAI для процессов RTAI и LXRT;
- очень высокое разрешение функций таймера и получения времени для процессов RTAI и LXRT;
- возможность запуска и параллельной работы обычных процессов Linux, RTAI, LXRT.

Преимущества RTAI по отношению к RT-Linux:

- компоненты ядра RTAI компилируются как отдельные модули, которые можно загружать (выгру-

жать, заменять) без перекомпиляции ядра и перезапуска системы;

- процессы RTAI могут быть сначала отлажены в Linux через механизм LXRT;
- сохраняется возможность запуска приложений Linux в PB RTAI с помощью LXRT.

Недостатки RTAI: отсутствие поддержки сигналов в API RTAI; нестандартная реализация некоторых функций синхронизации и коммуникации, неполное соответствие POSIX.

Проект ADEOS использует механизм очень схожий с механизмом RTHAL. Однако ADEOS потенциально позволяет иметь одновременно несколько расширений PB. В этом случае RTAI становится всего лишь одним из таких расширений.

Свободно распространяемая система XENOMAI реализована на основе ADEOS, характеризуется множественностью интерфейсов API, имитирующих интерфейсы и поведение других ОС PB в Linux. Данное свойство может оказать неоценимую помощь при переносе прикладного ПО из других встроенных ОС PB в Linux-XENOMAI. В конце 2006 г. очередная версия XENOMAI 2.3 обладала интерфейсами API Native, RTAI, Vx-Works, PSOS+, RTDM, VRTX, POSIX. Для RTAI и XENOMAI реализован вариант стека PB TCP/IP, выполняющийся вместе с ядром PB, ведется разработка стека USB и подсистемы Fire-Wire PB.

Рассмотренные доработки Linux позволяют приблизить временные параметры к аналогичным параметрам ОС PB, традиционно используемым в качестве встроенных ОС. Следует отметить, что такие доработки осуществляются и для систем BSD. Например, коммерческая встроенная система PB RTMK OS является расширением системы Open BSD, улучшающим временные параметры и добавляющим интерфейс API в соответствии с POSIX.

Однако остаются вопросы защиты системы, значительный размер и потребляемые ресурсы. Для размещения на Flash-диске одноплатного компьютера необходимо выделить подмножество необходимых компонент базовой инсталляции обычной Linux-системы. Не лишним будет оставить (или добавить) средства удаленного доступа и отладки ПО, что значительно упростит разработку и отладку прикладного ПО. Следует сразу определиться со средствами разработки и отладки на инструментальной системе, чтобы потом не пришлось пересматривать состав встроенной ОС. Ядро, библиотеки и исполняемые программы целесообразно перекомпилировать из исходных текстов с оптимизацией под конкретный процессор. Ядро системы лучше конфигурировать с включением только необходимых модулей. Из библиотек и исполняемых программ следует удалить отладочную информацию. Для снижения потребляемых ресурсов может быть использовано ядро более низкой версии, недостающие драйверы могут быть перенесены из ядер последних версий. Примерно по такой техноло-

гии осуществлялась поддержка ОС RedHat Linux, когда вместо использования ядра новой версии использовались дополнения старого ядра изменениями, заимствованными из текстов ядер других версий.

С точки зрения защиты файловой системы могут быть использованы различные способы. Размещение файловой системы в оперативной памяти и режим работы с Flash в режиме защиты от записи не всегда возможны. Если на носителе информации должна производиться запись данных или используется ПО и данные очень большого объема, то должны быть использованы другие приемы. Можно использовать комбинированный подход, когда корневая файловая система создается в оперативной памяти, а на flash-носителе имеются защищенные от записи разделы с ПО и незащищенные – для данных. При выборе файловой системы для разделов, не защищенных от записи, следует сбалансировать дополнительные временные затраты от журнализации и эффект, который она реально может дать. В этом смысле стандартная журналируемая файловая система Linux ext3fs показывает не самые лучшие результаты особенно на одноплатных компьютерах с Flash-носителями. Для ряда носителей имеет смысл использовать файловые системы jffs, jffs2. В некоторых случаях хорошей альтернативой ext3 является JFS.

Производители одноплатных компьютеров все чаще предлагают поставки с предустановленной ОС Linux, наборы инструментария для разработки ПО (Development Kit) на основе Linux или образы файловых систем Linux для Flash-носителей. С Linux используются технологии, применяемые с другими встроенными ОС. Например, некоторые одноплатные компьютеры на базе процессоров, не совместимых с x86, используют для загрузки Linux системный загрузчик Red Boot, который является минимальной реализацией ОС eCos. Предлагаемые системы имеют очень разную степень готовности для использования в качестве рабочей встроенной ОС. Иногда они служат только для демонстрации возможности работы компьютера в Linux, а иногда являются полноценной платформой для разработки встроенных систем.

Готовые разработки на основе систем BSD (Open BSD, Net BSD, Free BSD) значительно более редки и предлагаются компаниями, специализирующимися на разработке ПО. Отчасти это связано с общей тенденцией доминирования Linux по отношению к BSD. ОС Solaris не так давно стала свободно распространяемой системой, и попытки адаптировать ее для одноплатных компьютеров пока не получили широкого распространения, но тем не менее такие работы ведутся. Чтобы Open Solaris можно было рассматривать как приемлемое решение для одноплатных компьютеров должен пройти значительный период времени или должно существенно измениться распространение самой системы Open Solaris по отношению к другим ОС.

Щекин Сергей Валерьевич – канд. техн. наук, руководитель направления встроенных систем ЗАО "ИнтраСофт".

Контактный телефон (812) 575-06-09. <http://www.intrasoft-spb.com> E-mail: info@intrasoft-spb.com