

ЛИТЕРАТУРА

1. Лемешко Ю.Б., Чимитова Е.В. Максимизация мощности критериев типа χ^2 // Доклады СО АН ВШ. – Новосибирск: 2000. – №2. – С. 53-61.
2. Лемешко Ю.Б. Асимптотически оптимальное группирование наблюдений – это обеспечение максимальной мощности критериев согласия // Надежность и контроль качества. 1997. – №8. – С. 3-14.
3. Лемешко Ю.Б. Асимптотически оптимальное группирование наблюдений в критериях согласия // Заводская лаборатория. – 1998. – Т.64, №1. – С. 56-64.
4. Лемешко Ю.Б., Чимитова Е.В. О выборе числа интервалов в критериях согласия типа χ^2 // Заводская лаборатория/Диагностика материалов. – 2003. – Т.69. – С. 61-67.
5. Ртищева М.В., Разумейчик В.С., Дереченник С.С. Анализ топологических характеристик неплотных неупорядоченных монодисперсных структур / Проблемы проектирования и производства радиоэлектронных средств: Сб. материалов IV Международной НТК (25-26 мая 2006, г. Новополоцк). – Т.2. – Новополоцк: ПГУ, 2006. – С. 214-217.

УДК 004.514.62

Калиновский Р.В.

Научный руководитель: к.т.н. Костюк Д.А.

ПАРАВИРТУАЛИЗАЦИЯ ДЛЯ СМАРТФОНОВ С ОПЕРАЦИОННОЙ СИСТЕМОЙ SYMBIAN

Современные многофункциональные сотовые телефоны, известные также как смартфоны, обладают аппаратными и программными ресурсами, позволяющими сравнивать их с некоторым подмножеством персональных компьютеров. Пользователь смартфона имеет относительно широкий выбор прикладных программных средств, однако жестко привязан к одной единственной операционной системе. Поскольку операционная система (ОС) типичного смартфона в той или иной степени выполняет коммуникационные функции, ее замена приводит к неспособности устройства работать по прямому назначению.

Технология паравиртуализации позволяет с помощью специальной программы гипервизора, обеспечивающей распределенный доступ к оборудованию, одновременно работать двум операционным системам, каждая из которых при этом может выполнять собственные функции. При этом одна из операционных систем называется базовой, а другая - гостевой. Реализации метода паравиртуализации требует модифицировать гостевые операционные системы для того, чтобы они могли получить доступ к оборудованию через гипервизор. В отличие от гостевой, базовая ОС требует только написания специального драйвера, взаимодействующего с гипервизором. Поэтому в настоящий момент проекты паравиртуализации реализованы для операционных систем с открытым исходным кодом.

Проект Cooperative Linux [2] является одной из реализаций данной технологии на архитектуре i386. При использовании Cooperative Linux каждое ядро имеет собственный контекст центрального процессора и адресное пространство, и каждое ядро решает, когда отдать управление ядру-партнеру. Работа Cooperative Linux возможна под любой операционной системой, которая поддерживает загрузку драйверов.

Только одно из двух ядер может управлять физическими аппаратными средствами, а другое обеспечивается виртуальной аппаратной абстракцией. Ведущим может быть любое ядро операционной системы, которое предоставляет возможность работы с драйвером, обеспечивающим режим CPL0 - режим нулевого кольца - и выполнение распределения памяти.

Большинство изменений в патче Cooperative Linux находится в дереве i386 (единственная поддерживаемая архитектура для Cooperative на текущий момент). Другие изменения – по большей части добавления виртуальных драйверов: cobd (блочное устройство), соnet (сеть), и сосоп (консоль). Большинство изменений в дереве i386 – добавление кода инициализации и установки.

Cooperative Linux использует драйвера устройств для того, чтобы обратиться к привилегированному режиму и использовать примитивы, экспортируемые ядром ведущей ОС. Основная часть драйвера – независимый от ОС код, который связывается с примитивами, зависящими от ОС: распределением страниц памяти, выводом отладочной информации и связью с пространством пользователя. Виртуальная машина (VM) Cooperative Linux связана с ресурсами процесса – с описателем (дескриптором) файла.

Вся физическая оперативная память предоставляется ядру при запуске, а некоторое фиксированное количество физических страниц выделяется для Cooperative Linux. Затем средствами трансляции обеспечивается прозрачная работа Cooperative Linux со страницами памяти. Поэтому всю память, которую Cooperative Linux считает физической, следует рассматривать как псевдо-физическую оперативную память (PRAM).

Память распределяется ведущей ОС соответствующей функцией ядра. Выделенные страницы всегда резидентны и не освобождаются, пока VM не будет завершена.

Таблицы страниц создаются для того, чтобы отображать распределенные страницы в виртуальном адресном пространстве ядра VM. Адресное пространство VM сходно с адресным пространством основного ядра – нормальная зона оперативной памяти (RAM) отображена рядом в 0xc0000000.

Адресное пространство VM также имеет собственные специальные фиксированные отображения – сами таблицы страниц отображаются в 0xfef00000, чтобы обеспечить способность транслировать адреса PPRAM в физические адреса при создании элемента таблицы страниц (PTE) для пространства пользователя. Специальная карта распределения physical-to-PPRAM, отображенная в 0xff000000, ускоряет обработку событий, таких как ошибка страницы, требующих трансляции физических адресов в адреса PPRAM.

Cooperative Linux использует только один процесс ведущей ОС, чтобы обеспечить контекст для себя и своих процессов. Этот процесс, называемый colinux-демоном, часто вызывает драйвер ядра, чтобы выполнить переключение контекста между ядром ведущей ОС и ядром гостевой ОС Linux. При частых входах (10^9 раз в секунду) в ведущее ядро, VM в состоянии полностью управлять центральным процессором и MMU, не влияя на ядро ведущей ОС. Для архитектуры i386 чтобы выполнить переключение контекста необходимо изменить значение регистра - указателя таблицы каталога верхней страницы – CR3.

Промежуточное адресное пространство определяется специально созданными таблицами страниц и в гостевом, и в ведущих контекстах и отображает тот же самый код, который используется для переключения (код прохода) в обеих используемых виртуальных адресациях. Когда происходит переключение контекста, сначала изменяется CR3, чтобы указать на промежуточное адресное пространство. Далее используя безусловный переход, EIP перемещается к другому отображению кода прохода. Наконец, CR3 изменяется так, чтобы он указывал на каталог таблицы верхней страницы другой ОС. Одна страница MMU, которая содержит код страницы прохода, также содержит сохраненное состояние одной ОС, в то время как другая выполняется. В начале переключения контекста запрещены прерывания, и текущее состояние сохраняется в странице прохода кодом страницы прохода.

При анализе аналогичных возможностей архитектуры ARM, являющейся основой современных устройств на базе ОС Symbian, были выявлены аналогичные возможности, заключающиеся в аппаратно-ускоренном быстром переключении контекста процесса – Fast Context Switch Extension (FCSE). FCSE позволяет многим программам, запущенным на процессоре ARM, использовать идентичные диапазоны адресов, гарантируя обеспечение различия предоставляемых адресов для процесса и остальной части системы памяти. FCSE обрабатывает каждый виртуальный адрес при запросе доступа к памяти для предоставления измененного виртуального адреса.

В середине кода страницы прохода восстанавливается состояние другой ОС и разрешаются прерывания. Этот процесс является родственным "нормальному" процессу переключения контекста. Так как управление возвращается ведущей ОС при каждом аппаратном прерывании, то планировщик ведущей ОС отвечает за задание интервалов времени для VM Cooperative Linux (так, как если бы VM была постоянным процессом).

Cooperative Linux должен установить таблицу векторов прерываний, чтобы обрабатывать аппаратные прерывания, которые происходят в системе. Однако Cooperative Linux только пересылает запросы прерываний ведущей ОС.

Внутренние векторы прерывания процессора (0x0-0x1f) и векторы системных вызовов (0x80) содержатся как есть, так что Cooperative Linux обрабатывает собственные ошибки страниц и другие исключения, а остальные вектора прерывания указывают на специальные программы обработки прерывания (ISR).

Когда такая ISR вызывается внешним аппаратным прерыванием в течение контекста Cooperative Linux, выполняется переключение контекста в ведущую ОС, используя код прохода. С другой стороны, адрес этого ISR ведущей ОС определен в таблице дескрипторов прерываний (IDT) ведущей ОС. Стек вызова по прерыванию подменяется и происходит переход по адресу ISR. Между запросом ISR на стороне Linux и обработкой прерывания на стороне ведущей ОС заблокирован флаг прерывания. Эти манипуляции добавляют небольшое время запаздывания обработки прерываний в ведущей ОС, но им можно пренебречь. Беря во внимание тот факт, что эта методика отсылки прерываний также связывает аппаратное прерывание по таймеру, ведущая ОС не может обнаружить, что его CR3 был подменен на небольшое время, и поэтому в результате переключения контекста не происходит никаких исключений на ведущей стороне. Чтобы обеспечить прерывания для виртуальных драйверов устройств гостевой ОС, производят изменения в коде архитектуры; добавляют виртуальный контроллер прерываний, который получает сообщения от ведущей ОС по случаю переключения и вызывает `do_IRQ()` с подмененной структурой `pt_args`. Номера прерываний виртуальны и распределены в зависимости от устройства.

Гипервизор и загрузчик Cooperative Linux выполняется базовой ОС в качестве специализированного драйвера / расширения ядра. В случае ОС Symbian этот код разделяется на две относительно независимые составляющие, которые в свою очередь могут состоять из двух компонент – интерфейса с аппаратной частью / ядром и API.

Модель драйвера устройства ОС Symbian использует два типа DLL ядра – логический драйвер устройства (LDD) и физический драйвер устройства (PDD). Это соглашение обеспечивает уровень абстракции, который помогает в переносе программного обеспечения между платформами и в добавлении новых возможностей драйверов устройств без изменения общего кода и API.

LDD обычно разрабатывается для выполнения общих функциональных возможностей, а PDD используют для того, чтобы реализовать код, зависимый от устройства.

Symbian обеспечивает стандартные LDD для периферийных устройств (драйвера медиа-устройств, контроллер USB и последовательные устройства связи). Однако изготовители телефонов часто разрабатывают собственные интерфейсы для специальных аппаратных средств, что требует разработки дополнительных LDD.

PDD управляет конкретным периферийным устройством от имени его LDD и содержит код, зависимый от устройства. PDD взаимодействует только с соответствующим ему LDD, таким образом приложение пользовательской стороны не может получить доступ непосредственно к PDD. На PDD возлагают обязанности взаимодействия с вариантными и простыми расширениями или непосредственно аппаратными средствами. Как и LDD, PDD может быть сконфигурирован для выполнения инициализации во время загрузки.

К тому времени, когда ядро готово запустить планировщик, оно требует ресурсов, которые строго не определены архитектурой центрального процессора, которые обеспечиваются расширениями. Эти расширения являются специфичными для конкретной платформы, на которой запускается ОС Symbian, и разрешаются для портирования изготовителем телефона, без непосредственной перекомпиляции ядра ОС.

Расширение состоит из двух компонентов: слой, независимый от платформы (PIL), и слой, определяемый платформой (PSL). Перечисленные слои аналогичны слоям LDD/PDD у драйверов устройств. PIL в основном отвечает за обеспечение функциональных возможностей, одинаковых у версий расширения, и определяет API с PSL, который обеспечивает связь непосредственно с аппаратными средствами.

ЛИТЕРАТУРА

1. Танненбаум Э. Современные операционные системы. СПб: Питер, 2002. - 1040 стр.
2. Aloni D. Cooperative Linux. // Proceedings of the Linux Symposium, vol. 1. Ottawa, Canada, 2004. - p. 3 - 11
3. Столлинс В. Операционные системы. - Мн.: Вильямс, 2004. - 848 стр.
4. Symbian OS system definition - detailed view. Symbian developer network. http://developer.symbian.com/main/downloads/papers/SymbOS_cat/SymbianOS_cat.html. 2007.
5. Горнаков С. Symbian OS. Программирование мобильных телефонов на C++ и Java 2 ME. - Мн.: ДМК, 2005. - 448 стр.
6. Brash D. The ARM Architecture Version 6 (ARMv6). ARM White Paper, January 2002. - 15 pp.

УДК 004.514.62

Калиновский Р.В.

Научный руководитель: к.т.н. Костюк Д.А.

ТЕХНОЛОГИЯ ВИРТУАЛИЗАЦИИ ДЛЯ СМАРТФОНОВ

В настоящее время наблюдается бурное развитие технологий виртуализации операционных систем, позволяющих получить все многообразие программных платформ пользователя на единой аппаратной платформе вне зависимости от ее типа. Всплеск популярности вызван в первую очередь возросшими вычислительными возможностями аппаратуры, а также наличием значительного количества системного программного обеспечения с открытым исходным кодом, что облегчает его изучение и модификацию.

Современные многофункциональные устройства мобильной связи, известные также как смартфоны, при портативных размерах имеют аппаратные и программные ресурсы, зачастую сравнимые с таковыми у персональных компьютеров начального уровня (а по