

Развитие микропрограммного программного обеспечения компьютерной платформы: от BIOS ко всем компонентам Intel®

Винсент Зиммер (Vincent Zimmer)
Инженер-специалист
Подразделение Enterprise Platforms Group
Корпорация Intel

Содержание

(Для перехода к соответствующему разделу нажмите на номер страницы)

РАЗВИТИЕ МИКРОПРОГРАММНОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ КОМПЬЮТЕРНОЙ ПЛАТФОРМЫ: ОТ BIOS КО ВСЕМ КОМПОНЕНТАМ INTEL®	3
ОБЗОР: В ПОИСКАХ АЛЬТЕРНАТИВЫ BIOS	3
НЕМНОГО ИСТОРИИ	3
СУЩЕСТВУЮЩАЯ ПРОБЛЕМА	3
НОВАЯ ТЕХНОЛОГИЯ	4
ПРИМЕР СИСТЕМЫ	6
ВЫВОДЫ	7
ДОПОЛНИТЕЛЬНАЯ ИНФОРМАЦИЯ	8
ОБ АВТОРЕ	8

ОТКАЗ ОТ ОТВЕТСТВЕННОСТИ: МАТЕРИАЛЫ ПРЕДОСТАВЛЯЮТСЯ ПО ПРИНЦИПУ «КАК ЕСТЬ», БЕЗ КАКИХ-ЛИБО ЯВНЫХ ИЛИ ПОДРАЗУМЕВАЕМЫХ ГАРАНТИЙ, ВКЛЮЧАЯ, В ЧИСЛЕ ПРОЧЕГО, ГАРАНТИЙ В ОТНОШЕНИИ ИХ РЫНОЧНЫХ КАЧЕСТВ, НЕНАРУШЕНИЯ ПРАВ НА ИНТЕЛЛЕКТУАЛЬНУЮ СОБСТВЕННОСТЬ ИЛИ ПРИГОДНОСТИ К ИСПОЛЬЗОВАНИЮ В ТЕХ ИЛИ ИНЫХ КОНКРЕТНЫХ ЦЕЛЯХ. НИ ПРИ КАКИХ ОБСТОЯТЕЛЬСТВАХ КОРПОРАЦИЯ INTEL ИЛИ ЕЕ ПОСТАВЩИКИ НЕ НЕСУТ КАКОЙ-ЛИБО ОТВЕТСТВЕННОСТИ ЗА УЩЕРБ (ВКЛЮЧАЯ, В ЧИСЛЕ ПРОЧЕГО, УПУЩЕННУЮ ВЫГОДУ, ПОСЛЕДСТВИЯ ПРИОСТАНОВКИ ПРЕДПРИНИМАТЕЛЬСКОЙ ДЕЯТЕЛЬНОСТИ ИЛИ ПОТЕРЮ ДАННЫХ), ВЫТЕКАЮЩИЙ ИЗ ФАКТА ИСПОЛЬЗОВАНИЯ МАТЕРИАЛОВ, ЛИБО НЕВОЗМОЖНОСТИ ИХ ИСПОЛЬЗОВАТЬ, ЧТО РАСПРОСТРАНЯЕТСЯ И НА ТЕ СЛУЧАИ, КОГДА КОРПОРАЦИЯ INTEL БЫЛА ПРЕДУПРЕЖДЕНА О ВОЗМОЖНОСТИ НАНЕСЕНИЯ ТАКОГО УЩЕРБА. УЧИТЫВАЯ, ЧТО ЗАКОНОДАТЕЛЬСТВО, ДЕЙСТВУЮЩЕЕ В РЯДЕ ЮРИСДИКЦИЙ, НЕ ДОПУСКАЕТ ОГРАНИЧЕНИЯ ИЛИ ОТКАЗА ОТ ОТВЕТСТВЕННОСТИ ЗА ПОБОЧНЫЙ ИЛИ КОСВЕННЫЙ УЩЕРБ, ИЗЛОЖЕННОЕ ВЫШЕ ПОЛОЖЕНИЕ МОЖЕТ К ВАМ НЕ ОТНОСИТЬСЯ. КОРПОРАЦИЯ INTEL И ЕЕ ПОСТАВЩИКИ НЕ ГАРАНТИРУЮТ ТОЧНОСТИ ИЛИ ПОЛНОТЫ ТЕКСТОВОЙ ИЛИ ГРАФИЧЕСКОЙ ИНФОРМАЦИИ, ССЫЛОК И ИНОГО СОДЕРЖАНИЯ МАТЕРИАЛОВ. КОРПОРАЦИЯ INTEL ВПРАВЕ В ЛЮБОЕ ВРЕМЯ И БЕЗ ПРЕДВАРИТЕЛЬНОГО УВЕДОМЛЕНИЯ ВНОСИТЬ ЛЮБЫЕ ИЗМЕНЕНИЯ В УКАЗАННЫЕ МАТЕРИАЛЫ, А РАВНО И В ПРОДУКЦИЮ, ОПИСАНИЕМ КОТОРОЙ ОНИ СЛУЖАТ. КОРПОРАЦИЯ INTEL НЕ БЕРЕТ НА СЕБЯ КАКИХ-ЛИБО ОБЯЗАТЕЛЬСТВ ПО ОБНОВЛЕНИЮ МАТЕРИАЛОВ.

Примечание: Корпорация Intel не контролирует содержание сайтов других компаний и не может нести ответственности за продукцию и услуги других компаний. Все ссылки, выводящие Вас за пределы Web-сайта Intel, предоставляются только для Вашего удобства.

Развитие микропрограммного программного обеспечения компьютерной платформы: от BIOS ко всем компонентам Intel®

Винсент Зиммер (Vincent Zimmer)

Инженер-специалист

Подразделение Enterprise Platforms Group

Корпорация Intel

Обзор: в поисках альтернативы BIOS

С 1980-х гг. платформы персональных компьютеров прошли большой путь развития. Это позволило на порядки увеличить производительность компьютеров, простоту их использования, объем дисковой памяти и возможности внешних подключений. Однако есть элемент ПК, который не меняется уже 23 года, и это – BIOS.

Представленная корпорацией Intel концепция Platform Innovation Framework for the Extensible Firmware Interface (EFI) («Рамочная инфраструктура для развития платформы на базе расширяемого микропрограммного интерфейса», далее – просто «Рамочная инфраструктура») предлагает альтернативу BIOS, которая ускоряет загрузку, улучшает управляемость и расширяет возможности компьютера.

Немного истории

Задача загрузочного микропрограммного ПО (будь то BIOS или микропрограммное ПО на базе рамочной инфраструктуры) – в ходе загрузки объединить разрозненный набор аппаратных средств, который представляет собой компьютер до загрузки, в единую систему. Из соображений экономии сегодня и в обозримом будущем оказывается целесообразно создавать компоненты и платы, не имеющие собственных механизмов инициализации при включении. В результате при перезагрузке система, состоящая из таких компонентов, оказывается в весьма «первобытном» состоянии.

Такая система сильно зависит от загрузочной микропрограммы, которая готовит компьютер к загрузке операционной системы, предоставляет операционной системе сервисы (в особенности на ранних стадиях процесса загрузки) и сообщает данные об управляемости системы.

Существующая проблема

Для начала рассмотрим роль BIOS в современном компьютере. BIOS хранится в энергонезависимой памяти платформы и начинает работать сразу же после перезагрузки системы. Именно BIOS отвечает за инициализацию системы. Это процесс обычно называют тестированием при включении (POST).

Процедура POST для BIOS обычно представляет монолитную, статически связанную 16-разрядную процедуру на языке ассемблера реального режима, исполняемую в небольшой области памяти. Структура языка ассемблера и отсутствие развитых системных сервисов, таких как современный диспетчер памяти, в сочетании с ограниченностью пространства для исполнения препятствуют разработке более развитых алгоритмов и функциональных возможностей.

Помимо процедуры POST, BIOS отвечает за первоначальное обращение к операционной системе и предоставление ей сервисов. При этом сервисы для операционной системы реализуются на основе 16-разрядных программных прерываний. В частности, сюда относятся прерывания Int 13h для обращения к диску, Int 10h для обращения к графическому адаптеру и Int 16h для обращения к клавиатуре.

Наличие этих сервисов необходимо для загрузки операционной системы. Такая организация сервисов BIOS порождает ряд проблем, включая сложность введения новых сервисов, ограниченные возможности передачи параметров через регистры и ограничения реального режима процессора. Спецификация Extensible Firmware Interface дает возможность создать общий загрузчик операционной системы для платформ с различными архитектурами, включая платформы на базе процессоров с 32-разрядной архитектурой Intel® и семейства процессоров Intel® Itanium®. Возможности существующих сегодня загрузчиков ОС ограничиваются платформами ПК на базе 32-разрядной архитектуры Intel®.

Новая технология

В архитектуре рамочной инфраструктуры задача инициализации системы и предоставления сервисов операционной системы решается в несколько этапов. Каждый этап характеризуется ресурсами, которые доступны на этом этапе, правилами, которым должен подчиняться программный код для этого этапа, и результатами исполнения этапа. Эти этапы показаны на **рис. 1**. Следует отметить, что каждый этап опирается на результаты работы предыдущего.



Рис. 1. Этапы работы рамочной архитектуры

Инфраструктура, доступная на каждом этапе, предоставляется центральной рамочной инфраструктурой, а платформенно-зависимые функции реализуются в виде взаимодействующих между собой модулей. На этапе Pre-EFI (PEI) эти модули носят название модулей PEI (PEIM). В случае среды исполнения драйверов (DXE) модули представляют собой драйверы DXE или EFI. Соотношение между PEI и DXE показано на **рис. 2**. Инфраструктура и модули почти полностью выполнены в виде переносимого программного кода на языке C.

Драйверы EFI в некоторой степени аналогичны драйверам устройств в операционных системах. Они обеспечивают архитектурную расширяемость рамочной инфраструктуры и позволяют решить ряд задач:

- Удовлетворить требования широкого спектра платформ
- Реализовать новые технологии и исправления, а также поддержку новых аппаратных средств
- Реализовать поддержку модульной программной архитектуры

Драйверы EFI могут разрабатываться в разное время и разными организациями. Это порождает проблемы, отсутствовавшие в монолитных традиционных BIOS. Рамочная инфраструктура предусматривает эффективные средства для управления последовательностью исполнения драйверов EFI, абстрагирования интерфейсов драйверов EFI и управления совместно используемыми ресурсами. Перед началом работы возможна криптографическая проверка как самой рамочной инфраструктуры, так и драйверов, что позволяет построить доверительную цепочку от момента включения до загрузки ОС и далее.

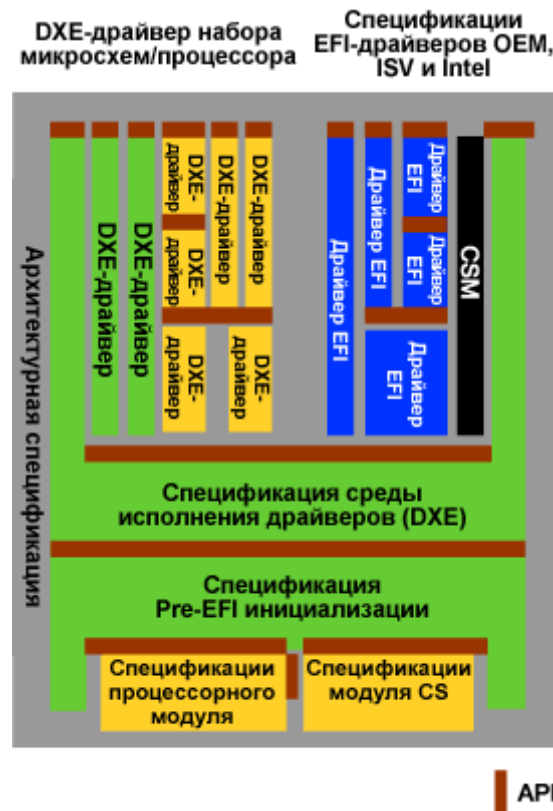


Рис. 2. Инфраструктурные двоичные модули

Модули PEIM и драйверы могут быть выполнены в виде самостоятельных двоичных модулей. Модули располагаются в логическом пространстве долговременной памяти, называемом микропрограммным томом. Понятие микропрограммного тома может охватывать как энергонезависимую память платформы, так и другие средства хранения.

Модули взаимодействуют с системой и между собой посредством вызываемых интерфейсов, именами которых служат глобально уникальные идентификаторы (GUID). GUID представляет собой статистически уникальное 128-разрядное число. Такая уникальность позволяет создавать расширяемые сервисы без ограничений и конфликтов между стандартными и платформенно-зависимыми сервисами.

Принцип интерфейсов, заложенный в основу конструкции EFI и рамочной инфраструктуры, позволяет отделить программные абстракции от конкретной микроархитектуры и топологии платформы. В результате рамочную инфраструктуру удалось успешно перенести на настольные системы, серверы, встраиваемые и мобильные системы с 32-разрядной архитектурой Intel®. Кроме того, рамочная инфраструктура была реализована в диапазоне от серверов на базе семейства процессоров Itanium® до платформ на основе технологии Intel XScale®. Чтобы не затруднять развертывание, компоненты инфраструктуры при этом просто подвергали кросс-компиляции и создавали новые модули (или модифицировали имеющиеся) с учетом особенностей платформ на базе технологии Intel XScale.

Этапы PEI и DXE обеспечивают инициализацию платформы, в частности, оперативной памяти и средств управления ресурсами шин ввода/вывода, а также обеспечивают обнаружение устройств ввода/вывода, которое в старых системах на базе BIOS происходило на этапе POST. Применение двоичных модулей и динамическое обнаружение сервисов позволяют повторно использовать компоненты и реализовать различные функциональные возможности платформы без полной перекомпоновки всего кода, как это было в случае монолитной BIOS.

Кроме инициализации платформы аналогично процедуре POST, Рамочная инфраструктура также предоставляет интерфейс для операционной системы. В самом начале этапа DXE инициализируется набор интерфейсов EFI. Это дает возможность компактно реализовать набор EFI-совместимых интерфейсов. Кроме того, рамочная инфраструктура имеет набор драйверов для поддержки старых интерфейсов ОС. Такая двойная структура интерфейсов ОС показана на **рис. 3**.



Рис. 3. Уровневая структура ПО

Пример системы

Предложенная инфраструктура может быть использована в качестве основы для построения платформы. Ниже на **рис. 4** показан пример системы, похожей на платформу современного персонального компьютера. Среди важнейших составляющих системы можно отметить полупроводниковые компоненты (синий цвет), оперативную память (зеленый цвет), шины ввода/вывода (сиреневый цвет), и процессор (оранжевый цвет). Для построения платформы на базе предложенной рамочной инфраструктуры предусмотрены модули PEIM, ответственные за инициализацию базовой коммутационной системы платформы, в том числе инициализацию оперативной памяти и установку процессора и набора микросхем в начальные состояния.



Рис. 4. Блок-схема системы

Этап PEI начинается с того, что базовая инфраструктура PEI вызывает ряд модулей PEIM. В это число входят модули инициализации северного моста, модуль инициализации оперативной памяти, модуль инициализации платформы и модуль инициализации процессора. Такое разделение позволяет, например, использовать PEIM-модуль процессора в широком спектре платформ с разными конструкциями и разными наборами системной логики.


После того как платформа приведена в начальное состояние, начинается этап DXE. Именно на этапе DXE происходят основные операции по инициализации платформы. Здесь требуется драйвер для распределения ресурсов шины PCI, другой драйвер для абстрагирования отказоустойчивого доступа на запись в флэш-память, средства консольного доступа (видео, USB-клавиатура), и т. п. Также на этапе DXE реализуется и поддержка загрузки операционной системы традиционным способом и через интерфейсы EFI.

Так же как и в случае PEI, на этапе DXE легко обеспечить поддержку различных модификаций платформы, заменяя соответствующие наборы драйверов. Например, для поддержки разных компонентов флэш-памяти или разных видеоустройств потребуется обновление всего одного соответствующего драйвера. Аналогично, сервер с высокой готовностью может отличаться от встраиваемой системы с фиксированной конфигурацией всего одним драйвером, описывающим порядок загрузки.

Выводы

Рамочная инфраструктура Intel Platform Innovation Framework for the Extensible Firmware Interface представляет собой новую, ориентированную на практическое применение реализацию интерфейса EFI. Она представляет собой альтернативу BIOS, которая позволяет ускорить загрузку, улучшить управляемость и расширить возможности компьютера. Эта инфраструктура представляет собой набор надежных архитектурных интерфейсов, реализованных на языке C, который позволяет строить микропрограммное обеспечение платформы из модульных составляющих. Назначение инфраструктуры – дать производителям BIOS и клиентам Intel возможность ускорить создание принципиально новых, нетрадиционных конструкций вычислительных платформ.

Обратная связь

[Сообщите нам](#)  Ваше мнение об этой статье.

Дополнительная информация

Дополнительная информация об инфраструктуре [Intel Platform Innovation Framework for the Extensible Firmware Interface](#) .

Дополнительная информация об интерфейсе [Extensible Firmware Interface](#) .

Об авторе

Винсент Зиммер работает в корпорации Intel семь лет и в настоящее время занимает должность инженера-специалиста подразделения Enterprise Platforms Group (EPG). Он имеет более чем 12-летний опыт разработки программного обеспечения для встраиваемых систем и является обладателем более 100 действующих и находящихся в стадии рассмотрения патентов. Винсент Зиммер получил степень бакалавра по электротехнике в Корнелльском университете и степень магистра компьютерных наук в университете штата Вашингтон в Сиэтле.

— Конец статьи из журнала *Technology@Intel* —