

О проекте

Что такое osFree?

osFree это проект по созданию  **открытой** операционной системы, основанной на микроядре  **L4**, целью которого является бинарная совместимость с  **OS/2** (IA32). Также, возможно параллельное сосуществование API различных операционных систем, построенных поверх одного и того же микроядра.

Зачем нужно микроядро?

- микроядро может служить базой для параллельных API, реализованных поверх него. Эти API могут существовать независимо, имея минимальную общую базу, состоящую из самого микроядра и некоторого набора общих сервисов (называемых personality-нейтральными сервисами). Это позволяет иметь параллельную (non-layered) реализацию параллельных API.
- API ОС поверх микроядра реализуется полностью в пространстве пользователя, оставляя только микроядро в режиме ядра. Такие компоненты ОС как планировщики процессов, менеджеры памяти, процесс подкачки, и даже компоненты, осуществляющие прямой доступ к оборудованию и обработку прерываний, также перемещены в пространство пользователя.
- это позволяет создать устойчивую систему с очень стабильным отлаженным крошечным ядром и менее стабильными компонентами из пространства пользователя, которые (как показывает, например, дизайн **Minix**) могут быть перезапущены (причем, даже автоматически) после того, как возникла неисправность.
- это также позволяет использовать обычные (“прикладные”) технологии разработки для драйверов, так как драйвер в микроядерной системе не отличается кардинально от обычных приложений.
- микроядерная архитектура также улучшает структурированность системы, то есть, зависимости между ее компонентами более четкие. Серверы работают поверх микроядра, взаимодействуют только через жестко заданные интерфейсы и скрывают свое внутреннее устройство, что имеет четкие параллели с объектно-ориентированным подходом.
- микроядерная система улучшает изоляцию ошибок внутри компонентов системы, так как все серверы исполняются в отдельных адресных пространствах.
- микроядро абстрагирует аппаратные интерфейсы от серверов пространства пользователя, что дает возможности для создания переносимых операционных систем, в которых все компоненты пространства пользователя остаются неизменными (на уровне исходного кода), необходима только их recompilation.

Почему L4?

- L4 это микроядро второго поколения, значительно улучшающее общую производительность системы. Это может быть наглядно заметно на примере l4linux. L4linux – это, в сущности, порт обычного ядра Linux на новую архитектуру “l4”. Его код, напрямую работающий с оборудованием, изменен таким образом, что оно становится доступно не напрямую, а через механизмы L4. Производительность ядра Linux может

быть оценена при помощи тестов (benchmarks), которые показывают потерю всего около 2% производительности, по сравнению с “родным” ядром Linux. Мы пробовали запускать l4linux на реальной машине и не заметили никакого различия в производительности. Это показывает, что L4 дает замечательную производительность с очень маленькими накладными расходами.

- Его минимальность и перемещение всей логики работы (policies) вне ядра, с оставлением внутри него только минимального набора механизмов, делает его почти универсальным и позволяет реализовать практически любое нужное API.
- Мы не хотим изобретать велосипед, и L4 содержит практически все нужные нам механизмы.
- Для L4 уже реализованы наборы готовых сервисов общего назначения, то есть, у нас уже есть готовый “конструктор”, а не только голое ядро.
- В качестве базы для Linux personality можно использовать уже готовый l4linux. Он все еще в стадии отладки, но почти все, тем не менее, работает. На нашем ноутбуке, только PCMCIA модем не заработал, из всего оборудования (это можно, впрочем, исправить, сделав ioremap на другие адреса – исходные адреса, используемые драйвером, заняты в L4-based системе). Wi-fi, bluetooth стек, USB стек, файловые системы, запись CDROM – все это заработало! Да, еще одно неудобство: так как видео работает поверх родной графической L4 консоли, которая на данный момент работает в режиме VESA (только избранные видеокарты поддерживаются с акселерацией), то поддержка видео, конечно, ограниченная.
- Device Driver Environment (DDE) может быть использовано в будущем в качестве модели драйверов устройств. Оно делает возможным портирование драйверов Linux (DDE/Linux) и FreeBSD (DDE/FreeBSD) в L4 userlevel. Таким образом, большая база драйверов Linux может быть использована в будущем. В отличие от драйверов Windows, они доступны в виде исходного кода и могут быть портированы для работы в user-level.

Зачем нужна реимплементация OS/2 с нуля?

OS/2 имеет одно из самых стабильных, надежных и высокопроизводительных ядер. Написанное примерно на 40% на ассемблере, оно очень хорошо оптимизировано и на 100% задействует возможности архитектуры i386. Его модульная структура позволяет легко заменять компоненты более улучшенными и менее ресурсоемкими, отрывать GUI, или настраивать систему под надобности пользователя. Она является очень гибко настраиваемой. Нам нравится ее компактное и “вылизанное” API, простота использования и интуитивный гибкий [объектно-ориентированный интерфейс](#). Она использует один из лучших общеупотребительных скриптовых языков – [REXX](#) – в качестве встроенного в систему и многие приложения скриптового механизма. OS/2 рекламировалась фирмой IBM как “DOS лучше чем DOS и Windows лучше чем Windows”. Это правда – ее VDM была практически лучшей из существующих. И это верно не только для DOS/Windows. Поддержка Java и XFree86 была также очень мощной. Поэтому, мы полюбили OS/2 как мощную интегрирующую платформу (Integration Platform, (TM) by IBM) на основе единого десктопа. Она широко использовалась множеством маргиналов и нонконформистов многие годы и всегда имела свой собственный “way of doing things”. Мы хотим, продолжать следовать этому пути ;) Мы можем спать спокойно, зная, что наша система не популярна среди хакеров и вирусописателей – они обычно атакуют мейнстрим... Но мы не можем спокойно жить, как раньше – начиная с декабря 2006 года IBM-овское начальство решило похоронить OS/2, и оставило OS/2 сообществу быстро устаревающую систему и ядро без исходников, и, самое главное, ухудшающаяся поддержка оборудования. Драйверов стали писать все меньше и меньше, и, в основном, новые драйвера являются портами с Linux. [Петиции](#) к IBM об открытии исходников также не дали результатов.

Тем не менее, мы хотим продолжить жить и работать в OS/2. Самая важная задача – это написание нового ядра. Нам всегда были интересны эксперименты IBM's с OS/2 поверх микроядра. Мы читали редбук от IBM про OS/2 Warp (PowerPC edition). Поэтому мы с энтузиазмом поддержали идею использования L4 как основы для реализации OS/2 API. Это дало толчок началу этого проекта.

Почему бы не мигрировать на другую ОС?

IBM, Netlabs и другие компании пытаются заставить пользователей мигрировать на другую ОС – GNU/Linux, BSD, Windows и др. Мы согласны с тем, что TCO текущих версий OS/2 становится все больше и больше для домашнего пользователя. В области серверов OS/2 также быстро сдает позиции (но по-прежнему является зрелой и стабильной). Но нам нравится подход IBM к дизайну ОС. OS/2 нам нравится **продуманностью**, а не только ее дизайном, как он есть. Она проста в использовании. Ее API компактный и чистый. Мы хотим продолжать работать и программировать в OS/2.

Совместимая

Планируется сделать osFree совместимой с основной частью текущего OS/2 API. Но мы не планируем поддерживать совместимость с API для драйверов. Драйверы OS/2 все больше и больше устаревают и не поддерживают множество новых устройств, но мы хотим иметь поддержку нового оборудования. Согласно своей идее, osFree может работать поверх множества ядер, таких как L4, Linux, Windows, etc. В результате, мы можем использовать подсистему драйверов от этого ядра.

API compatibility allow us to have still clean and small API and reuse existent applications. We have no plan for full support of 16-bit part of OS/2 because not so many applications which true 16-bit. For most mixed 16/32 applications we will provide on-the-fly patching of 16-bit calls to true 32-bit calls. As result we will have true 32-bit applications (after many years of mixed 16/32 applications).

We have a plan for limited support of DOS and Win16 personalities (for historic reasons). But not limit you to add more personalities.

Lightweight

OS/2 is one of most lightweight 32-bit OS. osFree also will be as lightweight as possible. We don't want to request 1 Gb minimum for work. We want to make it work on as minimal hardware as it possible. This will allow to use osFree in embeded area.

Open

osFree components is comes under open licenses like BSD and (L)GPL licenses. And we try to document interfaces as much as possible. So you are free to extend it as you wish.

Object-oriented

osFree tries to extend object-oriented design of desktop to other parts of the system. We want to introduce CPI+, GPI+, PM+ which is object-oriented representation of current CPI, GPI and PM services. Access to low-level services of the kernel also planned to be represent as objects.

Contributing

We have a lot of things still to do, so any help is welcome. Not only for code development, but also documentation writing, web page maintenance, distribution maintenance, and much more. See the [Project Roadmap](#) for more information about where we intend to go.

We are also looking for developers. For newbies, we have a number of [small tasks](#). If you're an experienced developer, there are plenty of [complex tasks](#) awaiting your talents! Check out the [development page](#) for more information about developing for us, and look at our [licensing](#).

osFree IRC channel is #osFree at [EFnet](#) and [eCSnet](#).

From:

<http://osfree.org/doku/> - **osFree wiki**

Permanent link:

<http://osfree.org/doku/doku.php?id=ru:about&rev=1363282244>

Last update: **2013/03/13 23:00**

