

УДК: 004.451.9

Бесплатные операционные системы Microsoft: Singularity и Verve

Ореликов В.В., Госуниверситет - УНПК, 11-БС

Singularity — начатый в 2003 году проект исследовательского подразделения корпорации Майкрософт по созданию высоконадёжной операционной системы, в которой микроядро, драйверы устройств и приложения написаны на управляемом коде.

ОС Verve непосредственно связана с проектом Singularity. Ее принцип — никаких уязвимостей и поразительная скорость работы. Verve — это вполне комплексная исследовательская система, которая, например, может выполнять параллельно несколько программ. Работать при этом Verve будет работать на порядок быстрее Windows, поэтому можно смело полагать что именно Verve станет основой для Windows 9. Однако сейчас она находится на стадии разработки и количество приложений для этого проекта пока ограничено, и на данный момент существуют тестовые версии. Еще не реализовано соединение с Интернетом через TCP, а также отсутствуют браузер и вебсервер. Кроме того, эта операционная система пока работает только на одноядерном процессоре.

Новая операционная система от Microsoft бесплатна, надежна, не подвержена вирусным угрозам и работает намного быстрее, чем Windows, Mac OS и Linux. Тем не менее многие пользователи не принимают ее во внимание. Tiberium объяснит почему.

Шесть лет назад корпорация Microsoft решила начать разработку новой операционной системы: этот проект получил название Singularity. При этом специалисты исследовательского подразделения Microsoft создали абсолютно новую ОС, основным достоинством которой стала надежность. Во время презентации Singularity на выставке разработок TechFest корпорация стремилась донести следующую мысль: Singularity — это не очередная версия Windows, а источник идей и технических нововведений для нее. По всей видимости, данное заявление было сделано для того, чтобы умерить желание

многих пользователей получить новую ОС. Ведь всего того, на что способна Singularity благодаря своей архитектуре, в Windows нет. Нет этого даже в Windows 7.

Практически любая существующая сегодня операционная система, установленная в персональных компьютерах, будь то Windows, Mac OS или Linux, имеет в своей основе «древнюю» архитектуру. Все используемые концепции относятся к тому времени, когда еще не было ни Интернета, ни вирусов, а компьютер по величине походил на современный серверный шкаф.

Основная часть традиционных операционных систем, прежде всего ядро, написана на языке программирования C. Современные разработчики этого бы делать не стали, так как команды на языке C и его преемнике C++ оказывают прямое влияние на регистры процессорного ядра и оперативную память. Из-за высокой производительности вначале это кажется преимуществом, но одновременно с этим система испытывает трудности с устранением программных ошибок или остановки исполнения вредного кода, так как ей неподконтрольно то, что делает код.

Известным примером, иллюстрирующим данную ситуацию, является переполнение буфера, при котором программа записывает данные в области памяти других, «соседних» приложений — типичный способ действия вредоносного ПО, которое стремится заполучить на компьютере права администратора. И даже если вся ответственность за брешь в системе безопасности ложится на небрежно написанное приложение, в первую очередь несовершенство ОС делает такую программу по-настоящему опасной.

В Singularity разрешен только безопасный программный код, написанный на языках, обеспечивающих собственную среду выполнения (Runtime Environment). На уровне приложений она работает как виртуальная машина. Известными языками программирования, поддерживающими Runtime, являются прежде всего Java и все .Net-языки — к примеру, C#. Код Singularity также написан практически полностью на C# (читается как «Си-шарп»), точнее

специально разработанном для нее языке Sing#. В отличие от C и C++, компилятор не

переводит программный код сразу на машинный язык. Вместо этого среда выполнения преобразует его вначале в байт-код. Благодаря данному промежуточному этапу система может проверить правильность кода и избежать «вторжения» в другие области памяти. Это, конечно, отнимает некоторые системные ресурсы, но Singularity позволяет компенсировать потери — каким образом, вы узнаете из раздела «Производительность».

В отличие от традиционных систем, Singularity способна на программном уровне изолировать друг от друга все выполняющиеся процессы и обеспечить безопасность их выполнения. Для каждого запущенного приложения, включая все необходимые программные библиотеки, создается программно-изолированный процесс (SIP), для которого выделяются требуемые ресурсы.

Среда выполнения каждого SIP также работает независимо от системы и может быть изменена в соответствии с требованиями приложения. В Windows или Linux, как правило, имеется одна среда выполнения для всех приложений. К примеру, ответственность за все запущенные Java-приложения ложится на Java Runtime Environment (JRE) — исполнительную среду для написанных на языке Java приложений. Сбой одного потянет за собой всю JRE, и работа всех приложений Java будет прервана.

Среда выполнения каждого SIP также работает независимо от системы и может быть изменена в соответствии с требованиями приложения. В Windows или Linux, как правило, имеется одна среда выполнения для всех приложений. К примеру, ответственность за все запущенные Java-приложения ложится на Java Runtime Environment (JRE) — исполнительную среду для написанных на языке Java приложений. Сбой одного потянет за собой всю JRE, и работа всех приложений Java будет прервана.

По причине того, что программно-изолированные процессы в большинстве случаев работают самостоятельно, обмен данными с ядром посредством двоичного интерфейса приложений (ABI)

ограничивается несколькими командами. Для сравнения: ABI в Singularity обходится 163

функциями, а для программного интерфейса приложений в Windows (API) предусмотрено около 14 000 функций.

Программно-изолированные процессы, отделенные друг от друга, общаются между собой посредством каналов. Они проходят через контролируемое ядром пространство — «обменную область» (Exchange Heap). В ней также находятся файлы, которые обрабатываются процессами в настоящий момент. Каждый бит данных в «обменной области» принадлежит только одному процессу. Совместное использование страницы памяти в Singularity запрещено, разрешено только неоднократное получение доступа на чтение.

Файл, загруженный через браузер из Интернета, будет помещен в «обменную область». Для его сохранения на диск браузер открывает канал подключения к драйверу файловой системы. Она также посредством канала устанавливает связь с драйвером жесткого диска. Одновременно с открытием канала SIP передает права владения файлом.

В существующих ОС системы каналов нет, так что Singularity в ближайшем будущем едва ли сможет прийти на смену Windows. Для этого пришлось бы переписать все существующие приложения. Такие известные программы, как Photoshop и Nero, в Singularity работать не будут, так как они написаны на небезопасных языках C или C++. Но даже Java- или .Net-приложения, которые полностью совместимы со средой Singularity, пришлось бы дополнить возможностью работы с каналами.

Список используемой литературы

1. Проект Singularity: обзор [Электронный ресурс]
/http://www.rsdn.ru/article/singularity/singularity.xml – режим доступа : (дата обращения 05.12.12)

2. Singularity: лучше, чем Windows [Электронный ресурс]
/http://www.open-life.org/blog/exotic_os/695.html – режим доступа : (дата обращения 05.12.12)

3. ОС Verve вместо Windows [Электронный ресурс]
/http://vestytechnology.ru/computer/4343-VerveWindows.html – режим доступа : (дата обращения 05.10.12)