

# экстремальный разгон XP

крис каперски ака мышцх

**кто это у тебя там фырчит? а... винчестер между переключениями с одной задачи на другую. а Windows разгонять не пробовал? у тебя ведь Windows стоит, верно? если это модель семейства NT'вых (NT, W2K, XP, 2003) эта статья для тебя!**

## введение

Все началось с того, что на одном из форумов появилось сообщение о новом способе разгона Windows XP суть которого сводилась к замене стандартного ядра на ядро от i486, для чего предлагалось в процессе установки Windows нажать <F5>, в появившемся диалоговом окне выбрать пункт "Others" (другие) и указать "Standard PC with C-Step i486", после чего производительность системы существенно возрастет. Автор заметки напирал на то, что дескать Microsoft умышленно замедляет быстродействие процессора в новых ядрах и потому старое ядро намного предпочтительнее.

Информация не осталась незамеченной и вызвала бурную дискуссию, быстро переросшую во флейм. Спорящие стороны с умным видом обсуждали темы в которых ни хрена не смыслят и оперировали тезисами в стиле "если бы это было правой, MS уже давно засудили" и "где такую траву брал?!". Реальную замену ядра осуществили единицы. У одних система воспряла духом и завращалась быстрее прежнего, другие же не обнаружили никаких изменений в производительности.

Сам автор экспериментировал с ядрами еще во времена Windows NT 4.0, но не считал эту тривиальную операцию чем-то экстраординарным, тем более, что она официально документирована самой Microsoft и никакого подвоха тут нет. Ни для кого не секрет, что старые версии программного обеспечения обычно работают намного быстрее, но вместе с тем уступают в функциональных возможностях.

## подготовка

Для смены ядра переустанавливать Windows совершенно необязательно. Эту операцию можно осуществить и на лету. Если вы не устанавливали никаких Service Pack'ов, откройте каталог WINNT\System32\Driver Cache\i386\driver.cab и вытащите из него файлы, в начале которых стоит HAL. Это – различные версии библиотеки аппаратных абстракций или в оригинале *Hardware Abstraction Layer*. Теперь найдите файлы, содержащие "nt" и "kr" (в FAR'е для этого достаточно нажать Gray-<+> и "\*nt\*kr\*"). Это – различные исполнительные системы ядра. Скопируйте их в каталог WINNT\System32. Для тех, у кого Service Pack установлен: ищите указанные файлы в каталоге WINNT\ServicePackFiles\i386.

Вооружившись "блокнотом" или любым другим аналогичным редактором, откройте файл boot.ini, находящийся в корневом каталоге загрузочного диска и найдите в нем строку следующего вида:

```
multi(0)disk(0)rdisk(0)partition(1)\WINNT="Windows 2000 Professional" /fastdetect /SOS
```

### Листинг 1 строка загрузочных параметров

Теперь либо добавьте к ней два новых ключа /KERNEL= и /HAL= указав имена файлов исполнительной системы ядра (по умолчанию ntoskrnl.exe) и уровня аппаратных абстракций (по умолчанию hal.dll), либо выделите всю строку целиком и вставьте ее в конец файла boot.ini, изменив текст "Windows 2000 Professional" на что-то в стиле "Windows 2000 hacked". Тогда при старте системы на экране появится меню многовариантной загрузки, позволяющее быстро переключаться между различными версиями ядер, не рискуя при этом обрушить основную конфигурацию системы (если это меню не появится, нажмите <F8>).

Какую версию ядра выбрать? Это зависит от архитектурных особенностей компьютера и ваших непосредственных потребностей. **Таблица 1** описывает назначения некоторых наиболее популярных ядер, из которых в первую очередь хотелось бы обратить внимание на NTKRNLMPS.EXE/HALMPS.DLL, ориентированных на многопроцессорные системы. Если вы установили Windows на компьютер без поддержки Hyper-Threading, а затем неожиданно решили ее поддержать, купив новехонький Pentium-4, или обновив BIOS, Windows не захочет работать со вторым процессором до тех пор, пока вы не переустановите ее или... не замените ядро.

Второе, естественно, проще и быстрее. Кстати, о быстроте. Ядра с поддержкой ACPI проигрывают не-ACPI ядрам по скорости, однако, всех "вкусностей" ACPI-режима они, естественно, лишены. Они не могут приостанавливать работу компьютера, вгоняя его в сон, и не всегда корректно отключают питание. Многопроцессорные ядра самые медленные. Использовать их на однопроцессорных машинах не рекомендуется (ну разве что вы хотите поэкспериментировать). Ядро, доставшееся в "наследство" от 486-машин, самое быстрое, однако, и самое ограниченное в своих функциональных возможностях. При использовании современного оборудования и некоторых навороченных игрушек могут появиться серьезные проблемы от полного нежелания работать, до периодических сбоев или чудовищного торможения всей системы. Но не будем нагнетать обстановку и сгущать краски, ведь в большинстве случаев i486-ядро и на современных процессорах "летает" вполне нормально. Только имейте ввиду, что в штатный комплект поставки Windows 2000 оно не входит, однако, его вполне реально найти в Интернете. Можно попробовать выдернуть HAL486C.DLL из дистрибутива Windows XP, но никаких гарантий, что оно нормально встанет на Windows 2000 ни у меня, ни у вас нет. Экспериментируйте на свой страх и риск.

Ключ /PCLOCK запрещает системе использовать динамическую адресацию IO/IRQ на PCI-шине, что в некоторых случаях значительно повышает производительность, однако, препятствует совместному использованию системных ресурсов несколькими устройствами одновременно. Теоретически, BIOS и Ось должны равномерно распределять порты/прерывания между устройствами, однако, практически они нередко впадают в грубые ошибки.

Сохранив изменения в boot.ini файле, перезагрузитесь. Имейте ввиду, что если вы сделаете что-нибудь неправильно, система может наотрез отказаться загружаться! Если это произошло, подключите винчестер с упавшей Windows к системе с работающей NT, повестив его вторым (Secondary Master или Primary/Secondary Slayer) и скопируйте исправный boot.ini на свой диск. Ну или просто переустановите систему.

компонент ядра	целевое назначение
NTOSKRNL.EXE	исполнительная система для однопроцессорных ПК с физической памятью 4 Гб или меньше
NTKRNLMP.EXE	исполнительная система для многопроцессорных ПК с физической 4 Гб или меньше
NTKRNLPA.EXE	исполнительная система для однопроцессорных ПК с физической памятью свыше 4 Гб
NTKRPAMP.EXE	исполнительная система для многопроцессорных ПК с физической памятью свыше 4 Гб
HAL.DLL	стандартный HAL (не ACPI, не APIC)
HAL486C.DLL	HAL для i486 C-Step систем
HALAPIC.DLL	однопроцессорная версия HALMPS.DLL (не ACPI, APIC)
HALAST.DLL	для симметричных многопроцессорных систем от компании AST
HALMPS.DLL	для большинства многопроцессорных систем на базе Intel (не ACPI, APIC)
HALACPI.DLL	однопроцессорный HAL с поддержкой ACPI, не APIC
HALAACPI.DLL	однопроцессорный HAL с поддержкой ACPI и APIC
HALMACPI.DLL	многопроцессорный HAL с поддержкой ACPI и APIC

**Таблица 1 описание наиболее распространенных ядер**

## ***разгон и его последствия***

Смена ядра не заставит процессор вычислять синус угла быстрее и уж точно не расширит пропускную способность интерфейсных шин. Популярные тестовые программы для измерения быстродействия ядер так же не годятся, поскольку не обнаруживают никакого прироста производительности, даже тогда, когда разница видна невооруженным взглядом. Почему так происходит? Все очень просто. "Быстрые" ядра отличаются от "медленных" прежде всего накладными расходами на диспетчеризацию прерываний и переключения контекста. Промежуток времени между двумя переключениями (условно называемый квантом) это целая вечность для процессора, в течении которой он успевает обсчитать множество тестовых задач, в результате чего длительность переключения контекста не учитывается. К тому же, при малом числе потоков, издержки от переключений между ними достаточно невелики, но стоит запустить параллельно с тестовой программой десяток-другой тяжеловесных приложений, как все измениться!

Практика показывает, что на компьютерах, оснащенных SDR-памятью и процессорами с частотой менее 1 ГГц, i486-ядро существенно повышает "реакционноспособность" системы и работать с ней становится значительно приятнее. Естественно, с ростом быстродействия компьютера этот выигрыш стремительно тает, хотя все же остается заметным. Попутно

исчезают конфликты присущие не вполне ACPI-совместимым устройствам, драйверам и BIOS'ам. На смену им приходят конфликты с древней версией ядра, на совместимость с которой ни устройства, ни драйвера, ни BIOS'ы вообще никем не тестировались.

Как поведет себя 486-ядро в вашей системе – заранее неизвестно и выяснить это можно лишь экспериментально.

## **>>> врезка внутри ядра**

Существуют ли теоретические обоснования почему i486-ядро увеличивает быстродействие компьютера? Да, существуют и хотя до сих пор не ясен конкретный вклад каждого из них, общая картина событий выглядит скорее тривиальной, чем удивительной.

Начнем с того, что поддержка многопроцессорности не проходит бесследно и налагает на архитектуру ядра определенный отпечаток, заставляя его решать те задачи, которые на однопроцессорных машинах просто не возникают. Взять хотя бы проблему когерентности, т.е. согласованности данных. Представьте себе, что произойдет, если один процессор обратится к недостроенной структуре данных, конструируемой другим процессором. Чтобы этого избежать в каждый момент времени только один процессор может модифицировать данные, а остальные блокируются при помощи спинлуков (от английского spin lock – взаимоблокировка). В однопроцессорных системах спинлуки лишены смысла и должны заменяться на NOP'ы, однако, в Windows так не происходит и профилировка показывает, что большую часть отведенного ему процессорного времени ядро расходует именно на спинлуки! И это еще далеко не все! Алгоритмы, эффективно исполняющиеся на двух или даже четырех процессорах далеко не всегда демонстрируют свою эффективность на одном. Судя по всему, во времена 486-процессоров Microsoft поддерживала независимые ядра, оптимизируя каждую конфигурацию по отдельности. Когда же аппаратные возможности персональных компьютеров возросли, она с облечением свела все ядра воедино. Дизассемблирование доказывает, что однопроцессорная версия ядра практически во всем повторяет многопроцессорную, и работает намного медленнее, чем хотелось бы.

Другой источник тормозности – этот пресловутый Plug and Pray. Древние ядра самостоятельно обслуживали шины, DMA и прочие устройства, за каждым из которых жестко закреплялось свое пространство адресов ввода-вывода, свой IRQ и свой канал DMA. Теперь же все значительно усложнилось и... затормозилось. Ядро абстрагировалось от конкретного оборудования и перешло на виртуальные шины, эмулируемые драйверами соответствующих устройств. Билли ухитрился испоганить все, включая схему обработки прерываний. Это раньше диспетчер знал все прерывания в лицо, а теперь вынужден постоянно обращаться к базе данных, выясняя какой вектор какому устройству принадлежит (ведь системные ресурсы могут динамически переназначаться во время работы)!

## **заклучение**

Единого мнения по поводу целесообразности перехода на 486-ядро как не было, так и нет. Сообщения о реально проведенных экспериментах носят единичный характер и статистики по ним не постоишь. Тем не менее, разгон Windows это достаточно тривиальная и безопасная операция, намного более безопасная, чем разгон процессора, например. Обращаясь ко всем читателям, автор предлагает организовать широкомасштабный эксперимент, – меняйте ядра и оставляйте свои впечатления на сайте журнала. Должны же мы в конце концов узнать дает ли 486-ядро реальный выигрыш или нет!