

Настройка параметров виртуальной памяти и дискового кэша

Сбрось память на диск!

Игорь Лейко, Среда, 07 Июль 2004, 17:03

Вступление

Можно без преувеличения сказать, что оптимальная настройка виртуальной памяти и дискового кэша - одна из наиболее непонятных для пользователя областей. Множество рекомендаций кочует из одной журнальной статьи в другую, оттуда в книги, из книг - на страницы интернета, а из книг - снова в журналы.

Если заглянуть в русскую часть интернета, то окажется, что ошибочные рекомендации и неточная информация на эту тему составляют не менее 95 % от общего количества публикаций.

Некоторые из рекомендаций остаются неизменными чуть ли не со времен Windows 3.0. В чем заключается выигрыш от их применения или не объясняется вовсе, или объясняется поверхностно, исходя из общих соображений и без учета того, как данное изменение параметров скажется на системе в целом.

К числу таких рекомендаций относятся установление фиксированного размера файла подкачки, кратного размеру ОЗУ компьютера, перемещение файла подкачки на отдельный раздел того же диска, на который установлена Windows, установку параметра ConservativeSwapFileUsage=1 для более эффективного использования памяти (хотя на самом деле он ухудшает ее использование), уменьшение размеров дискового кэша вплоть до двух мегабайт и пр.

Чтобы вам стало ясно, почему следование этим советам снижает эффективность работы Windows 98/Me, расскажем об особенностях использования виртуальной памяти и дискового кэша в этой ОС. Затем будут даны рекомендации, справедливые для подавляющего большинства пользователей. Существуют некоторые очевидные исключения, например редактирование графических или видеофайлов размером во многие десятки и сотни мегабайт. Или, наоборот, выполнение на компьютере задач расчетного характера, производящих большие объемы вычислений с не очень большим количеством исходных данных (большинство игр типа "убей все, что движется" относится именно к этому классу задач). В этих случаях оптимальные настройки могут оказаться другими, но если вы поймете, как Windows использует виртуальную память, то легко сможете сделать необходимые изменения сами.

Но вначале вернемся в весну 1998 года, чтобы стать свидетелями переписки разработчиков Windows 98 с пользователями.

[newpage]

Беседа с разработчиками

Пользователь Windows 98: - многих смущает необычное поведение файла подкачки по сравнению с Windows 95. Например, его увеличенный размер и повышенная активность. Не мог бы кто-нибудь дать квалифицированное разъяснение?

системы необходимо изменить размер файла подкачки, мы внесли два изменения по сравнению с Windows 95.

Когда я разрешаю Windows управлять виртуальной памятью, я ожидаю, что диспетчер памяти будет создавать файл подкачки только тогда, когда **вся** оперативная память будет использована. Именно так должен работать динамический файл подкачки. Он должен быть резервом на тот случай, когда системе не хватит оперативной памяти и не должен

Второй разработчик: - Я хотел бы уточнить, что Windows 98 не рассчитана на работу с отключенной виртуальной памятью и последствия такого отключения непредсказуемы. Это вовсе не означает, что система сразу перестанет работать. Работать она будет, но до некоторого момента. Невозможно предсказать, где и как проявит себя отключение виртуальной памяти. (Замечание автора статьи: известны случаи, когда после отключения виртуальной памяти пропадал звук. После создания файла подкачки фиксированного размера в один мегабайт все приходило в норму).

Разработчик: - к сказанному ранее добавлю, что при завершении работы Windows 98 мы не уменьшаем файл подкачки до минимально необходимого значения, а оставляем его таким, как есть. А теперь давайте обсудим некоторые нюансы поведения системы, касающиеся отключения виртуальной памяти и логики работы диспетчера памяти.

Если виртуальная память включена, то значительная доля выделяемой памяти (в частности, вся память, выделяемая приложениям, в том числе и программный код, и данные) помечается как выгружаемая. Это означает, что в некоторый момент (который невозможно определить заранее) мы можем выгрузить содержимое этой памяти в файл подкачки.

Если у вас 96 Мб ОЗУ и вы используете лишь одно или два приложения, тогда необходимость такой выгрузки вряд ли когда-нибудь возникнет. Но в какой момент следует резервировать место в файле подкачки на случай возникновения такой необходимости? После ее возникновения? А вдруг к этому моменту на диске не останется свободного места? Тогда единственным выходом будет аварийное завершение работы приложения.

Поэтому мы резервируем место в файле подкачки в тот момент, когда приложению выделяется оперативная память. В этом случае мы можем получить отказ в распределении памяти, если не удастся увеличить размер файла подкачки.

То есть нам либо не удастся запустить приложение, либо приложение, обратившись к системе с запросом о выделении памяти, получит отказ. Любой из этих вариантов намного более приемлем, чем аварийное завершение работы приложения.

А если мы разрешим диспетчеру памяти выделять место под файл подкачки только после того, как вся память будет использована, то столкнемся с описанной выше проблемой.

(Замечание автора статьи. При эксплуатации Win95 в случае исчерпания свободного места на диске появлялась вероятность зависания системы из-за необходимости подгрузить какой-либо модуль в условиях невозможности выделения дополнительной виртуальной памяти, и я сам с этим сталкивался.)

Второй разработчик: - предыдущий ответ не совсем полон. Главный его недостаток - ничего не говорится о дисковом кэше. Приложениям, которые запускает пользователь, вполне хватает 96 Мб памяти, раз система работает с отключенной виртуальной памятью. И если количество распределенной виртуальной памяти меньше, чем физической, то, конечно, прямой необходимости в файле подкачки нет. Единственная причина, по которой система в этих условиях все же использует файл подкачки - то, что, как мы обнаружили, система

работает лучше, если часть виртуальной памяти (не используемая в данный момент)

выгружена в файл подкачки, чтобы освободить физическую память для дискового кэша, повысив тем самым его эффективность.

Даже если с технической точки зрения системе нет необходимости выгружать данные в файл подкачки и она прекрасно может работать без этого, она будет работать лучше, если неиспользуемые данные все же окажутся выгружены. Нет никакого смысла занимать физическую память невостребованными блоками виртуальной памяти.

Приведенное выше обсуждение состоялось, когда разработка Windows 98 еще не была завершена. Вероятно, поэтому никто не упомянул параметр ConservativeSwapfileUsage. Но о нем мы поговорим позже.

[newpage]

Виртуальная память

Сначала изложим объяснение работы виртуальной памяти в общем. Когда запускается какое-либо приложение, ему выделяется некоторое количество оперативной памяти. Эта память выделяется блоками (обычно их называют страницами), и в специальной области памяти ведется таблица, учитывающая все выделенные страницы. Размер страницы памяти определяется аппаратной реализацией процессора. В процессорах семейства x86 этот размер равен четырем килобайтам.

Когда для загрузки очередного приложения физической памяти уже не хватает, страницы, которые выделены приложению, простаивающим в данный момент, или менее важным, чем текущее (имеющим меньший приоритет), записываются на диск, о чем делается пометка в таблице распределения страниц.

Высвободившаяся память выделяется запускаемому приложению. Когда программе требуется та часть программного кода или данных, которая в данный момент выгружена на диск, операционная система загружает их вновь, при необходимости освободив память выгрузкой других страниц.

Такая схема неплохо работает в многопользовательских операционных системах, в которых реально выполняется много задач одновременно. При этом задачи, ожидающие загрузки страниц в память, приостанавливаются в ожидании этой загрузки, а процессорное время передается другим программам.

В однопользовательской же операционной системе, которой преимущественно является Windows 9x/Me, такой алгоритм работы оказывается неоптимальным. Цель, стоящая перед Windows, совсем иная: не обеспечение максимальной загрузки процессора для наиболее полного использования его вычислительной мощности, а создание наибольших удобств для пользователя, то есть в первую очередь минимизация времени, требующегося на переключение между задачами. Естественно, не за счет существенного замедления работы задач.

Если вы работаете в Windows 3.x и переключаетесь на задачу, которая в данный момент выгружена на диск, вам приходится ждать, пока операционная система высвободит

Настройка параметров виртуальной памяти и дискового кэша

<http://www.osrc.info/plugins/content/content.php?content.34>

оперативную память, выгрузив часть страниц на диск, и затем загрузит необходимые

страницы с диска. При запуске нового приложения перед выгрузкой еще необходимо увеличить файл подкачки (если он временный, а не постоянный), чтобы в нем появилось место для выгрузки.

Чтобы ускорить переключение между приложениями и запуск новых, в Windows 9x используются следующие приемы:

Размер файла подкачки динамически увеличивается и уменьшается (во время, когда компьютер не загружен работой), чтобы в любой момент в нем было достаточно места для выгрузки всех страниц, находящихся в оперативной памяти и не являющихся невыгружаемыми.

Страницы с измененными данными переписываются в файл подкачки (не выгружаются, а переписываются, оставаясь в памяти) во время, когда компьютер не загружен другой работой. Это обеспечивает возможность мгновенного освобождения таких страниц для использования другими задачами в случае необходимости.

Учтите при этом еще одну особенность Windows 95/98/Me (и Windows NT 4/2000/XP тоже), неизвестную подавляющему большинству пользователей. Программы и программные модули, имеющие формат PE (Portable Executable) и хранящиеся на локальном диске с несменяемым носителем, запускаются особым образом. Их запуск начинается не с загрузки программного кода в память, а с распределения памяти и сопоставления страниц виртуальной памяти участкам файла программы. То есть с точки зрения Windows непосредственно перед запуском программа оказывается выгруженной на диск, причем не в файл подкачки, а в свой собственный файл, который становится как бы частью файла подкачки. Затем начинается исполнение кода из первой страницы, и остальные страницы подгружаются в память только при необходимости. Участки программы, которые в данный момент не используются, в память не загружаются. Тем самым достигается минимально необходимый расход оперативной памяти (ОП) и существенная ее экономия по сравнению с обычной процедурой считывания в память всей программы и последующего ее запуска. К тому же программа начинает выполняться раньше на время, почти равное требуемому на ее загрузку в память. К типу PE относится подавляющее большинство программ, написанных для 32-разрядных ОС семейства Windows (NT, 2000, XP, 95, 98, Me).

Запуск программ как бы из файла подкачки дает особенно заметный выигрыш в случае размещения программного кода на диске в соответствии с порядком его загрузки в память. Такое размещение обеспечивает программа дефрагментации диска Windows 98/Me.

Еще одним преимуществом такого подхода является уменьшение потребности в файле подкачки. Программный код уже хранится на диске, и, поскольку он не изменяется, нет необходимости выгружать его на диск. Можно сразу отдать эту страницу ОП под другую страницу виртуальной памяти, а затем при необходимости заново считать программный код с диска. Но, в отличие от файла подкачки, загрузка программного кода в память происходит через дисковый кэш, что при малых размерах дискового кэша может привести к его перегрузке и резкому снижению эффективности. Как следствие, Windows станет работать медленнее.



Стоит заметить, что здесь имеется одно существенное различие между Windows 95 и Windows 98/Me. Первая передает копии страниц из дискового кэша в память, распределяемую диспетчером виртуальных машин, из-за чего в памяти фактически имеется два экземпляра таких страниц. Однако, не спешите возмущаться - так поступают почти все ОС. Windows 98 и Windows Me могут выполнять программный код непосредственно из кэша. Это означает, что та часть кэша, которая занята отображенными в память участками программ, одновременно оказывается обычной оперативной памятью, выделенной этим программам.

Системный монитор может показать количество таких двояко используемых страниц и тем самым помочь определить, какая часть дискового кэша занята программами. На компьютере автора этот показатель колеблется от 500 до 1500 страниц, обычное значение - 800 страниц (т.е. 2-6 Мб и 3,2 Мб соответственно). В ваших условиях количество таких страниц может быть другим. Но очевидно, что искусственное уменьшение размера дискового кэша до нескольких мегабайт почти наверняка сделает кэш неэффективным.

В результате совокупности мер, принятых при разработке Windows 95 и, особенно, Windows 98, в большинстве случаев запись в файл подкачки выполняется довольно редко и небольшими порциями. К тому же выполняется она преимущественно в то время, когда диск не загружен другой работой. Интенсивность чтения данных из файла подкачки, как правило, также невысока и редко превышает несколько сотен килобайт в секунду, а размер считываемого за один прием блока обычно не превышает нескольких десятков килобайт.

[newpage]

Многие ОС, разработанные в семидесятых-восемидесятых годах при нехватке памяти выгружали на диск программы целиком. Так поступала даже Windows 3.x при работе в стандартном режиме. Такое поведение обуславливалось тем, как именно организовано в процессоре управление оперативной памятью. Обладая некоторыми преимуществами (размещение выгруженной на диск программы, как правило, в одном непрерывном блоке), такое решение имеет и существенный недостаток - увеличение обмена с диском. При нехватке даже нескольких килобайт физической памяти одна из программ выгружается на диск целиком, а позже снова загружается целиком же.

В процессорах 80386 и совместимых с ними моделях при работе в защищенном режиме процессора 80386 (расширенный режим Windows) управление памятью осуществляется постранично. На диск выгружается не вся область кода и данных программы целиком, а только отдельные страницы по мере надобности. Хотя, конечно, в результате вся память, выделенная некоторой программе, может оказаться выгруженной. Зато исключается возможность избыточной выгрузки. Постраничное управление памятью имеет и побочный эффект: выгруженная программа часто не занимает непрерывную область в файле подкачки, а разбросана по нему, подобно тому, как разбросан по разным участкам жесткого диска фрагментированный файл.

В этих условиях увеличение скорости чтения из файла подкачки даст лишь незначительный прирост скорости. Это означает, что умеренная фрагментация файла подкачки практически не изменит времени, требующегося на подкачку страниц в память. Windows 98 увеличивает файл подкачки ступенями по 4 Мб, стремясь при этом, чтобы каждый такой участок был непрерывным. Этой меры обычно оказывается достаточно для того, чтобы непрерывный файл подкачки практически не имел преимуществ в скорости по сравнению с файлом,

Размещение большого файла подкачки в начале диска может увеличить скорость работы в основном из-за того, что оно совмещается с расположением программных файлов в начале диска. Такое размещение обеспечивается дефрагментатором и несколько уменьшает время, требующееся для перемещения головок диска от программных файлов к файлу подкачки и обратно. А вот расположение файла подкачки на отдельном разделе того же диска, на котором установлена Windows, гарантированно снижает скорость работы, поскольку требует постоянного перемещения головок на довольно большое расстояние.

Наилучшим вариантом является размещение файла подкачки на другом физическом диске, на котором нет активно используемых программ. Этот диск не обязательно должен быть самым быстрым, достаточно, чтобы он не был намного медленнее основного диска. Несколько отвлекаясь от темы скажем, что если программа, с которой вы в основном работаете, создает еще и очень большие временные файлы, то целесообразно каталоги для временных файлов перенести на третий (физический!) диск.

Замечание

Иногда можно встретить рекомендацию не размещать файл на сжатом диске без крайней необходимости. Якобы затраты времени на упаковку и распаковку подкачиваемых страниц заметно снизят производительность системы. Эта рекомендация не имеет под собой никаких оснований. Если файл подкачки размещается на сжатом диске, то он помечается как не подлежащий сжатию, и замедления доступа не происходит.

Несколько слов о том, как определить, какой размер файла подкачки вам нужен. Часто предлагаемая формула "трехкратный размер оперативной памяти" основана на особенностях использования виртуальной памяти в Windows 3.x, но не в Windows 9x. В последнее время, правда, даже самые ярые сторонники данного совета понимают, что числа получаются просто абсурдные, и уменьшают рекомендуемый размер файла до двух-двух с половиной размеров ОП, а в последнее время – даже до полутора.

Лучше всего поступить следующим образом. Запустите системный монитор, добавьте показатель «Размер файла подкачки» и установите интервал времени обновления в 10 минут. Проработайте с запущенным монитором весь день и посмотрите, какой размер принимал файл подкачки. Теперь установите для файла подкачки такой минимальный размер, который оказался бы достаточен для работы в течение 90-95 процентов времени. Максимальный размер не устанавливайте (укажите размер, соответствующий количеству свободного места на диске), чтобы не столкнуться с сообщением о нехватке памяти для запуска программ.

Такой настройкой вы избавите Windows от необходимости часто менять размер файла подкачки и не лишитесь возможности запустить столько программ, сколько вам нужно. А файл подкачки не будет занимать на диске лишнего места.

Нет никакого смысла в установке размера файла подкачки в зависимости от размера оперативной памяти, поскольку потребность в нем определяется не столько имеющейся

Если в вашем компьютере имеется оперативная память очень большого объема, то у вас может появиться соблазн отключить виртуальную память (использование файла подкачки). Windows 98/Me не рассчитана на работу с отключенной виртуальной памятью, и последствия такого отключения непредсказуемы. Это не означает, что она перестанет работать, но нормальная работа системы будет нарушена и невозможно предсказать, когда и как это нарушение проявится. Вместо отключения можно задать минимальный и максимальный размер файла подкачки равным одному мегабайту. Но задавать верхнюю границу, как уже говорилось, нежелательно.

[newpage]

Существует еще одно различие между Windows 95 и 98/Me. Первая из них при завершении работы уменьшает размер файла подкачки до минимума, а после запуска вновь его увеличивает. Следующие версии этого не делают. Причина достаточно очевидна. Во время разработки Windows 95 программы для ДОС были еще весьма распространены, а размеры дисков - не слишком велики. Чтобы при работе в режиме ДОС (или в старой версии ДОС) место на диске не терялось напрасно, файл подкачки решили уменьшать. К моменту появления Windows 98 необходимость в экономии места на диске для режима ДОС отпала. Как из-за того, что размеры дисков значительно выросли, так и из-за того, что на подавляющем большинстве компьютеров режим ДОС уже не использовался. Поэтому для уменьшения времени загрузки и завершения работы было решено отказаться от уменьшения размера файла подкачки при выходе из Windows. Зачем делать напрасную работу, ведь этот файл все равно потребуется увеличить при следующем включении компьютера.

Совет

Если у вас установлен дисковод компакт-дисков, не уменьшайте размер его кэша, чтобы сэкономить память. Вы сэкономите не столько оперативную, сколько виртуальную память, то есть место в файле подкачки. Дело в том, что кэш компакт-дисков в Windows 9x является выгружаемым. То есть если он не используется, а память, которую он занимает, требуется другим программам, то он вытесняется в файл подкачки. При обращении к компакт-диску кэш снова загружается в память.

Совет

Если вам остро не хватает места на диске, то, возможно, вы сталкивались с ситуацией, когда файл подкачки занимал все свободное место и некуда было сохранить результаты работы. В этом случае вам поможет строка

```
<tt>MinUserDiskSpace=количество_</tt>
```

добавленная в раздел [386 Enh] файла System.ini. После этого Windows будет оставлять на диске свободное место указанного размера, ограничивая увеличение размера файла подкачки.

Теперь несколько слов о том, как на скорость работы влияет частое изменение размера файла подкачки. Начнем с экскурса в историю. В Windows 3.1 файл подкачки мог быть либо постоянным, либо временным. Постоянный файл находился на диске и занимал на нем место всегда. А временный создавался лишь на то время, когда у Windows появлялась нужда в виртуальной памяти, что несколько экономило место на диске. Зато с постоянным файлом подкачки система работала быстрее.

В представлении основной массы пользователей и многих так называемых "" это ускорение работы было связано с тем, что Windows тратила много времени на увеличение и уменьшение размера файла подкачки. Как сказал один мудрый человек, нет ничего легче, чем найти простое и ясное для понимания неправильное решение (в данном случае - объяснение).

На самом деле разница в скорости работы объяснялась отнюдь не увеличением и уменьшением размера файла подкачки. Эти процедуры занимали мало времени и выполнялись, в основном, при запуске и завершении работы программ. С постоянным файлом подкачки система работала в обход подпрограмм обслуживания дисков, записанных в ПЗУ компьютера (BIOS). Доступ к временному файлу подкачки осуществлялся через эти процедуры (если такие процедуры вынуждена использовать Windows 9x, то на вкладке «Быстродействие» появляется сообщение "Страничный обмен в режиме MS-DOS снижает").

Windows 9x использует временный файл подкачки, но обращается к нему в обход процедур ДОС и BIOS. Для нее это нормальный режим работы с диском. Создать в этой системе постоянный файл подкачки невозможно. Хотя она и может использовать постоянный файл подкачки, оставшийся от Windows 3.x, но использует его как временный. Можно создать файл подкачки постоянного размера, задав для него одинаковые верхнюю и нижнюю границы, но постоянным он от этого не станет.

Постоянный файл подкачки имел заданный размер и был непрерывным, но помимо этого располагался в строго определенном месте диска, которое указывалось во вспомогательном файле. Если файла подкачки на этом месте не оказывалось, то вы получали сообщение, что он недоступен.

Чтобы убедиться, что изменение размера файла подкачки не оказывает сколько-нибудь заметного влияния на скорость работы программ, выполните небольшой эксперимент. Возьмите файл с какой-нибудь небольшой программой для ДОС, например редактор Edit, находящийся в папке Command. Откройте окно свойств этой программы и установите очень большие требования (например 32 768 КБ) к дополнительной (XMS), отображаемой (EMS) памяти и памяти защищенного режима ДОС (DPMI). Если вы установили файлу подкачки постоянный размер, уберите верхнюю границу и перезагрузите компьютер.

Запустите системный монитор и задайте отслеживание загрузки процессора и размера файла подкачки и обновление информации через одну секунду. Теперь запустите программу для ДОС, свойства которой вы устанавливали, и посмотрите, много ли времени потребовалось системе на увеличение размера файла подкачки. Если вы задали довольно большой минимальный размер этого файла, то, возможно, придется запустить не одну, а несколько копий программы, прежде чем система увеличит его размер.

Теперь закройте запущенную программу (программы) и подождите, пока размер файла

подкачки уменьшится. И опять это произойдет почти моментально. Правда, если в "" части файла подкачки окажутся выгруженные страницы, то тогда система предварительно переместит их в оставляемую часть файла, а на это потребуется некоторое время и довольно большое число операций ввода-вывода. Но значительную часть этого времени процессор будет простаивать.

[newpage]

ConservativeSwapfileUsage=1

Этот параметр был документирован некоторое время спустя после выхода Windows 98. В описании к нему сказано, что он предназначен для обеспечения совместимости с некоторыми программами для Windows 95, которые отслеживают обращения Windows к файлу подкачки.

Внутренний механизм работы с файлом подкачки в Windows 98 изменен. При необходимости выгрузки какой-либо области памяти в файл подкачки Windows 95 ждала момента, когда система в целом оказывалась в состоянии простоя, а Windows 98 ждет момента, когда простаивает VFAT, то есть лишь одна из подсистем - дисковая.

Такой подход должен повышать быстродействие системы. На практике повышение оказывается практически незаметным из-за относительной редкости операций записи в файл подкачки. К тому же дополнительное условие для заметности - высокая загруженность процессора при невысокой интенсивности обращений к дискам, что встречается не так уж часто.

Впрочем, как заявлял герой одного анекдота, "десять старушек - уже ". И именно набор таких небольших выигрышей приводит к тому, что быстродействие новых систем (подчеркну: при прочих равных условиях) оказывается выше, чем у старых. В предвидении возмущенных писем с утверждениями, что Win95 всегда работает быстрее, чем Win98, напомним один факт. ZDNet, которая отнюдь не с симпатией относится к "", в 1998 году обнародовала результаты своих исследований. Согласно им, при 16 МБ ОЗУ связка Win95+IE4 работает быстрее, чем Win98. Но уже при 32 МБ Win98 оказывается на 9 процентов быстрее своей предшественницы. С ростом объема памяти выигрыш увеличивается, хотя уже и не так быстро.

Но вернемся к нашим баранам. Помимо документированного эффекта параметр ConservativeSwapfileUsage=1 обладает еще недокументированным. Он также включает использование алгоритма управления файлом подкачки от Windows 95. То есть отменяет предварительное увеличение размера файла подкачки и выгрузку в файл подкачки неиспользуемых модулей ради увеличения размера дискового кэша.

Народная молва, заметив это, тут же приписала ему чудодейственный эффект: якобы, параметр заставляет Windows максимально эффективно использовать оперативную память, минимизируя использование файла подкачки. Внешне объяснение действительно логичное: если у вас 128 Мб памяти, то после загрузки, хоть несколько десятков мегабайт физической памяти и свободно, Windows 98 создает файл подкачки в 20-30 Мб. А если добавить в файл system.ini упомянутый параметр, то размер файла подкачки оказывается нулевым. Казалось бы, уменьшение подкачки налицо?

В том то и дело, что нет. Подкачки не было и в первом случае (системный монитор

<http://www.osrc.info/plugins/content/content.php?content.34>
показывает, что занято в файле подкачки 0 байт). Но кому нужно запускать системный

монитор и разбираться в его показаниях? А команда DIR, такая простая и наглядная, всегда под руками.

Эксперимент

Так что же в действительность дает этот параметр? Я решил выяснить это экспериментально. Но чем измерять? Имеющаяся у меня версия WinStone уж очень старая и нормально на Windows 98 работать не хочет. SiSoft Sandra и схожая программа из NU не годятся, поскольку не измеряют быстродействие компьютера, а оценивают его. Они меряют отдельные характеристики, каждую независимо от других, а потом, по каким-то своим соображениям, выводят итоговое значение. В результате получаем некое конкретное число, но на него можно только ориентироваться.

Я выбрал в качестве тестового задания печать большого документа Word (30 Мб, свыше 400 страниц) с множеством рисунков и таблиц.

Word запускался командой печати данного документа, после чего автоматически закрывался. В свойствах принтера была включена отложенная печать, так что собственно печать документа не выполнялась, а только формировались данные для печати.

Сама процедура эксперимента выглядела так. Использовалась рабочая копия Win98SE на машине следующей конфигурации: Пентиум III 667 МГц, 128 Мб ОЗУ, винчестер 7200 об/мин с двухмегабайтным кэшем. Настройки виртуальной памяти и кэша - принятые по умолчанию.

В этой системе последовательно выполнялась печать сначала с отключенным параметром ConservativeSwapfileUsage=1, затем, после перезагрузки, - с включенным. Перед каждой перезагрузкой файл подкачки и файл с данными для печати удалялись.

Такая пара экспериментов была для накопления статистики повторена трижды. Затем то же самое я проделал для памяти, ограниченной размером 48 Мб.

Параметры системы измерялись системным монитором, включенным на запись данных в файл. Периодичность замеров была задана равной 0,1 секунды.

Итого - 12 перезагрузок и 12 тестов.

[newpage]

Результат

Прежде всего, меня удивило то, что во всех 12 случаях после печати размер файла подкачки был одинаков: 19 четырехмегабайтных кусков. Исходя из общепринятых представлений логично было бы ожидать, что при меньшем объеме памяти файл подкачки должен был бы быть больше.

Добавление параметра действительно уменьшало исходный размер файла подкачки: с 68 Мб до 0 при 128 Мб памяти и с 68 Мб до 52 Мб - при 48 Мб ОЗУ.

При 128 Мб занято в файле подкачки перед началом печати (т.е. после загрузки) в обоих случаях было 0 байт, при 48 Мб - около 2 Мб при выключенном и 0 - при включенном

одинаков во всех 12 экспериментах. То есть, место на диске добавлением этого параметра сэкономить не удалось.

Среднее время выполнения задания (около минуты для 128 Мб и примерно 70 секунд для 48 Мб) при включенном параметре незначительно отличалось в меньшую сторону на 128 Мб и в большую - на 48 Мб. Но статистически эти отличия были недостоверны: различия между сериями оказались меньше или сопоставимы с разбросом значений внутри серий (пришлось вспоминать правила обработки результатов экспериментов, которые я когда-то изучал в курсе матстатистики).

 $\langle u|$

количество прочитанных (с диска) страниц виртуальной памяти, что нелогично, если считать общепринятое мнение о параметре правильным. Подкачка должна была бы быть меньше:

При этом уменьшалось, также примерно на 300, число очищенных страниц, то есть количество страниц, которые могут быть высвобождены без записи их содержимого в виртуальную память.

Обсуждение результатов

Одинаковое значение размера файла подкачки в конце работы показывает, что потребность системы в виртуальной памяти не зависит от параметра. Выделяется ли память до запуска программы или после - влияет только на первоначальный размер файла подкачки. Количество необходимой системе виртуальной памяти не меняется.

То, что размер файла подкачки оказывается одинаковым и при 128 Мб ОЗУ, и при 48 Мб, скорее всего, означает, что общее количество места, зарезервированного в файле подкачки (как до запуска программ, так и в процессе работы), зависит от запущенных программ, а вовсе не от объема ОЗУ. Это опять-таки соответствует логике использования файла подкачки, изложенной разработчиками.

Дополнительный эксперимент

Когда результаты эксперимента уже были обработаны, я задумался над увеличением количества выгрузок страниц в файл подкачки. А что, если это увеличение является "" и в дальнейшей работе не проявляется? То есть при увеличении загрузки памяти содержимое ненужных страниц выгружается на диск - в процессе работы выполняется именно та операция, которую Windows 98 в нормальных условиях выполняет в ходе загрузки. Впрочем, это легко проверить. Поскольку в данном случае вполне достаточно качественной картины, без тщательных измерений можно было обойтись.

Описанный выше эксперимент был проделан еще два раза, оба раза при 48 Мб, один раз с выключенным параметром, второй раз - с включенным. На этот раз печать выполнялась дважды. Между операциями печати была сделана небольшая пауза, чтобы в протоколе системного монитора можно было легко отличить одну операцию от другой.

Результат: при отключенном параметре выгрузка страниц в файл подкачки происходила только во время первой печати. При включенном параметре - и во время первой печати, и во время второй. Правда, при второй печати интенсивность выгрузки была в несколько раз меньшей, но все же достаточно заметной. Мое предположение оказалось неверным.

[newpage]

Выводы и рекомендации

Изменение размера файла подкачки происходит относительно редко и занимает небольшую часть ресурсов процессора, из-за чего практически не сказывается на быстродействии системы. Поэтому нет смысла задавать файлу подкачки постоянный размер. Все, чего вы этим добьетесь - это вероятность получить сообщение о нехватке памяти для запуска программ. А если размер файла подкачки выбран чрезмерно большим, то вы впустую тратите место на диске.

Наибольший эффект оказывает размещение файла подкачки на другом физическом жестком диске (размещение на другом разделе того же диска уменьшит скорость работы). Если жесткий диск один, и на нем достаточно свободного места, чтобы Windows 9x могла избежать чрезмерной фрагментации файла подкачки, то принятие специальных мер по оптимизации этого файла, как правило, не даст заметного эффекта.

Добавление параметра в раздел [386Enh] файла system.ini приводит к уменьшению размера файла подкачки перед началом объемной работы или через некоторое время после нее, не влияет (или мало влияет) на размер файла подкачки во время работы, не оказывает заметного влияния на скорость работы, приводит к более интенсивному использованию виртуальной памяти.

В целом, изменения, внесенные в Windows 98 в алгоритм работы с виртуальной памятью, как

А как же быть с множеством случаев, когда при добавлении параметра ConservativeSwapfileUsage=1 наблюдалось улучшение работы системы? Увы, могу объяснить это только эффектом плацебо. Тем более что во всех известных мне таких случаях оценка производилась на глаз, без каких-либо измерений. Подобным оценкам принято доверять только при сравнении двойным слепым методом.

Приведу пример влияния субъективности из моего личного опыта. Небезызвестный в англоязычном Интернете Эндрю Кэмерон писал мне по поводу программы Win2cache: "The log file says 'not installed' - however I am SURE it is, because things seem A LOT faster" (в файле протокола сказано «не установлена», хотя я УБЕДЕН, что она установилась, поскольку видно, что все работает НАМНОГО быстрее). И это при том, что программа не только не установилась, но и в принципе не могла дать никакого улучшения на Pentium III. Налицо обычное самовнушение, когда ожидаемое воспринимается как происходящее в действительность.

Так вреден или полезен параметр ConservativeSwapfileUsage=1? Я не стану давать категорического ответа. Конечно, его добавление ухудшает параметры системы, но ухудшение не настолько велико, чтобы быть критически важным. С другой стороны, Windows 9x/Me - это персональная ОС, и настраивать ее следует так, как удобнее и приятнее вам. Если вам кажется, что при добавлении параметра система работает быстрее, и вам при такой настройке работать комфортнее - добавление необходимо. В конце концов, существуют люди, которые оценивают скорость системы не по скорости работы приложений, а по тому, насколько быстро выскакивают окошки или насколько быстро система загружается.

Каждый вправе настраивать свою систему так, как нравится именно ему. Вот только навязывать свои предпочтения другим, если эти предпочтения не подкреплены фактами, не стоит.

Несколько слов о дисковом кэше

Выше уже говорилось о дисковом кэше, но вскользь. Давайте рассмотрим его работу и настройки подробнее. Дисковый кэш - это область памяти, в которую записываются данные, ранее прочитанные с диска. Если эти данные требуются повторно, то они берутся из кэша, а не прочитываются с диска заново. Часто можно услышать, что дисковый кэш ускоряет работу диска. Но такое утверждение в корне неверно. Дисковый кэш ускоряет работу системы с диском, если случаются попадания в кэш. Обращение же к диску занимает при работающем кэше занимает ровно столько же времени, сколько и при отсутствии кэша: примерно 10-15 миллисекунд (для жесткого диска).

Рассмотрим процесс чтения более детально. Если системе требуются какие-то данные с диска, а эти данные есть в кэше (попадание в кэш), то данные будут получены практически моментально и обращения к жесткому диску не будет. Если же данные в кэше отсутствуют (промах), то система обращается к диску (здесь и далее все расчеты будем делать для жесткого диска, для конкретности возьмем мой IBM DJNA-371350 - 13 Гб, 7200 об/мин). Поскольку система у нас многозадачная, то мы не можем определить, к какому участку диска выполнялось предыдущее обращение и должны предполагать, что головки находятся над каким-то случайным цилиндром. Сначала головки должны установиться на нужный цилиндр -

от 2,2 до 15,5 мс, среднее время – 9,0 мс. Затем необходимо дождаться, когда под головками окажется нужный сектор. Согласно теории вероятности среднее время ожидания – половина оборота диска, т.е. 4,17 мс. Затем надо прочитать запрашиваемый блок данных и передать его в оперативную память. Продолжительность чтения и передачи зависит от скорости работы диска, скорости передачи данных по шине и размера прочитываемых данных. Средний размер одной порции данных при работе в Windows составляет примерно 15-20 килобайт, возьмем для расчетов 20 Кб. Чтобы определить время чтения, делим 20 Кб на параметр media transfer rate (скорость обмена данными с поверхностью) и еще примерно на 1,2 – чтобы учесть межсекторные промежутки и служебные зоны. Получаем приблизительно 1 мс. Скорость передачи данных по шине зависит от характеристик шины и составит ~ 0,7 мс при шине 33 Мб/с и 0,2 мс при шине 100 Мб/с. Итого – от 14,4 до 14,9 мс в зависимости от скорости шины.

Итак, данные прочитаны, переданы программе и сохранены в кэше. Следующая операция чтения опять займет примерно столько же времени. Даже если потребуются данные, хранящиеся на том же самом цилиндре, диск наверняка повернется уже достаточно далеко, чтобы снова пришлось ждать появления под головками нужных данных. И чтение займет 5-6 миллисекунд. Но вероятность такого хода событий не слишком велика: системе может понадобиться обратиться к файлу подкачки, программному файлу, сохранить данные во временном файле, прочитать данные для другой программы...

Но миллисекунды – гигантское время по меркам компьютера. Поэтому чем реже системе приходится обращаться к диску – тем лучше. Значит, надо повышать эффективность кэша, иначе говоря – соотношение числа попаданий к числу промахов. Один из путей – улучшать алгоритм работы кэша – нам недоступен. Это привилегия разработчиков. Второй путь – увеличивать размер памяти, отведенной под кэш. Именно поэтому кэш стремится занять столько памяти, сколько возможно. И такого поведения не надо пугаться – если запускаемым программам требуется память, кэш сразу же отдаст требуемое количество памяти. В нормальных условиях, однако, кэш будет уменьшаться примерно до 10-15 процентов от общего объема памяти. Иначе снижение эффективности кэша будет более значительным, нежели потери времени от обмена содержимым памяти с файлом подкачки.

Поэтому искусственное ограничение размера кэша – это сознательное ухудшение характеристик системы. Если во время работы имеется свободная оперативная память – это неиспользуемая (напрасно установленная память). И неважно, почему память свободна – то ли потому, что вы установили некую программу, принудительно освобождающую память, то ли потому, что вы не разрешили кэшу ее использовать. С таким же успехом вы можете физически удалить эту свободную память. Раз она все равно не используется, ее отсутствие ни на что не повлияет.

Статья опубликована с любезного разрешения Игоря Лейко.

[Посетить сайт Игоря.](#)

Статья также публиковалась в журнале [Upgrade](#).