

Специальность 09.02.07
«Информационные системы и программирование»



«Внедрение и поддержка
компьютерных систем»
Модуль №2

Группа
С 121к, С 122

Модуль 2 Методы решения проблем совместимости

- Настройка управления питанием.
- Оптимизация использования процессора.
- Оптимизация использования памяти.
- Оптимизация использования жесткого диска.
- Оптимизация использования сети.
- Инструменты повышения производительности программного обеспечения.

Тема 1 Совместимость программного обеспечения

- Настройка управления питанием.

Настройка управления питанием

Настройка схемы управления питанием по умолчанию

- На обслуживающем компьютере откройте командную строку с повышенными привилегиями.
- Чтобы использовать схему управления питанием с другого компьютера, импортируйте ее. Например, чтобы импортировать схему управления питанием «Схема для улицы», в командной строке введите: powercfg -IMPORT C:\OutdoorPlan.pow

Настройка управления питанием

Настройка схемы управления питанием по умолчанию

- Чтобы узнать идентификаторы GUID всех схем управления питанием на компьютере, введите: powercfg -LIST

Отобразится список доступных схем управления питанием. В следующих примерах эти схемы обозначены guidPlan1 и guidPlan2.

Existing Power Schemes (* Active)

Power Scheme GUID: {guidPlan1} (Balanced) *

Power Scheme GUID: {guidPlan2} (Power saver)

- 5. Обратите внимание на GUID рядом со схемами управления питанием, которые вы хотите изменить.

Настройка управления питанием

Подтверждение схемы управления питанием по умолчанию

- Нажмите кнопку Пуск и выберите Панель управления.
- Щелкните Оборудование и звук, а затем выберите Электропитание. Откроется окно Электропитание, и появится схемы управления питанием.
- Просмотрите каждую схему.
- Убедитесь, что в качестве текущей схемы управления питанием выбрана нужная схема. Текущая схема управления питанием отмечена звездочкой (*).

Настройка управления питанием

Подтверждение схемы управления питанием по умолчанию

- Нажмите кнопку Пуск и выберите Панель управления.
- Щелкните Оборудование и звук, а затем выберите Электропитание. Откроется окно Электропитание, и появится схемы управления питанием.
- Просмотрите каждую схему.
- Убедитесь, что в качестве текущей схемы управления питанием выбрана нужная схема. Текущая схема управления питанием отмечена звездочкой (*).

Тема 1 Совместимость программного обеспечения

- Оптимизация использования процессора.

Настройка управления питанием

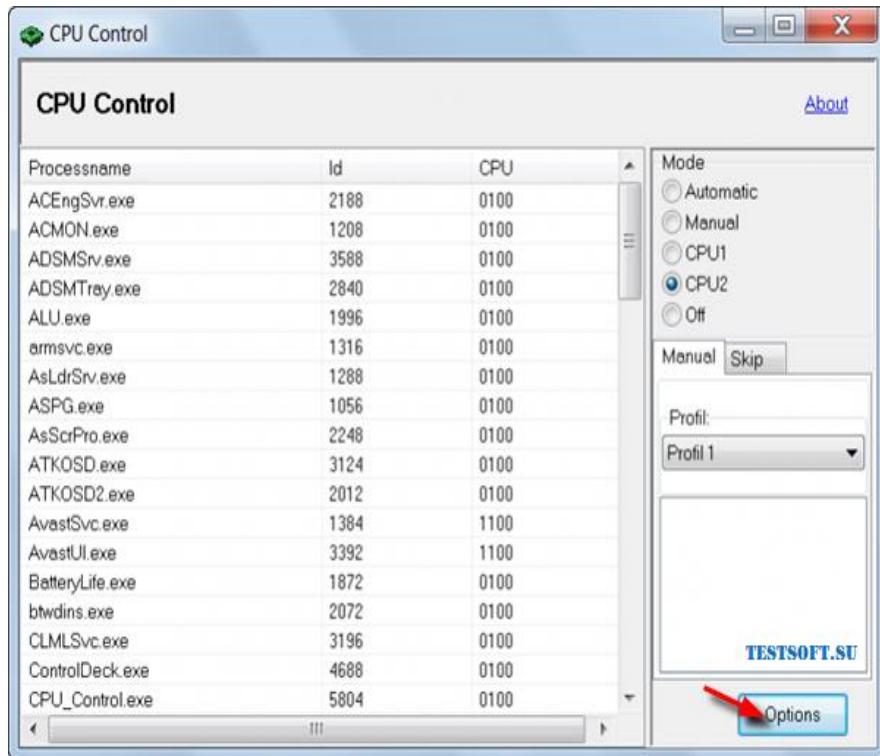
Оптимизация работы процессора: утилита CPU Control

- Некоторые приложения не поддерживают многоядерный режим, поэтому для корректной работы необходимо вручную выделить им отдельное процессорное ядро. Другой "проблемный" вариант – материнская плата асинхронно инициализирует ядра. В обоих случаях требуется оперативное вмешательство пользователя.
- По умолчанию, Windows присваивает всем запускаемым процессам равный приоритет в пользовании процессорных мощностей. Что не совсем правильно в случае запуска ресурсоемких приложений, например, когда вашему любимому Call of Duty "мешает" антивирус вкупе с дюжиной другого "второстепенного" ПО. CPU Control поможет пропорционально разнести процессы по ядрам, что напрямую скажется на общей производительности системы.

Утилита поддерживает двух- и четырехъядерные процессоры, но, по отзывам, корректно работает и на 6–8 ядрах.

Настройка управления питанием

Оптимизация работы процессора: утилита CPU Control

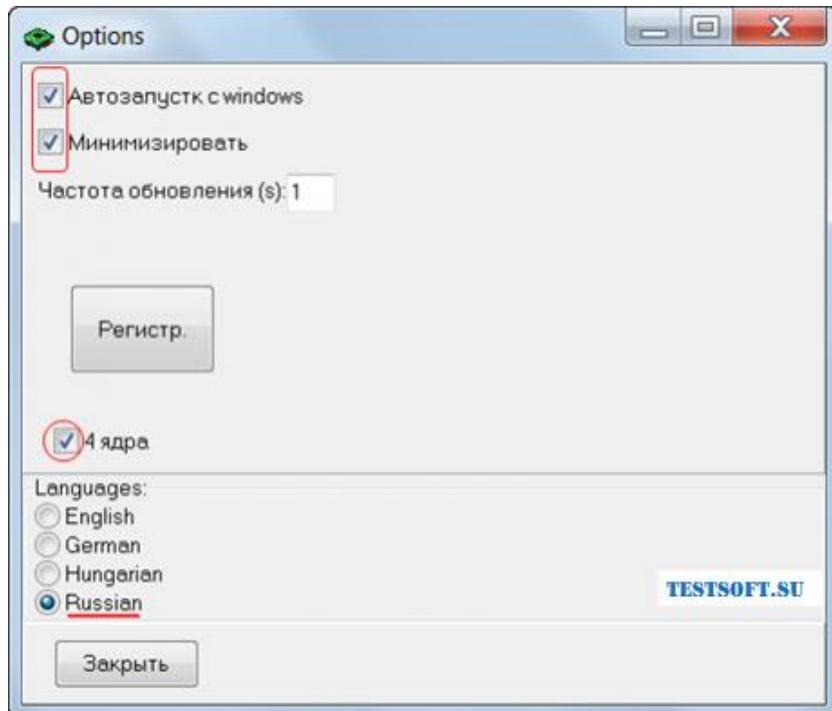


- Для начала необходимо скачать утилиту. Никаких "хитростей" в установке нет, поэтому последовательно жмем "Next" ("Далее") до полной инсталляции. Кстати, программа не "сорит" в реестре, т.е. портативна и носима на съемных накопителях.

После запуска первым делом кликаем на "Options" ("Опции") и русифицируем CPU Control через "Languages" ("Языки") → "Russia". Далее, в соответствии со следующим скриншотом, ставим галочки напротив "Автозапуск с Windows", "Минимизировать" и "4 ядра" (для четырех и более ядер). Понятно, что если ЦП двухъядерный, активировать "4 ядра" без надобности.

Настройка управления питанием

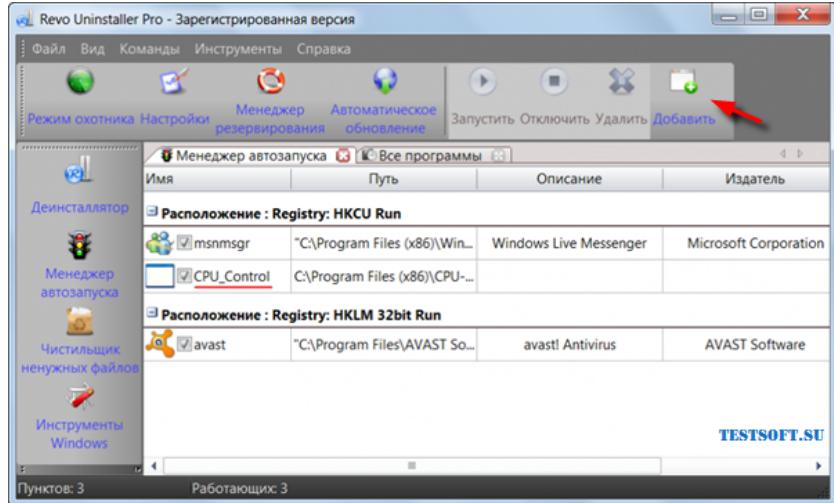
Оптимизация работы процессора: утилита CPU Control



- После запуска первым делом кликаем на "Options" ("Опции") и русифицируем CPU Control через "Languages" ("Языки") → "Russian". Далее, в соответствии со следующим скриншотом, ставим галочки напротив "Автозапуск с Windows", "Минимизировать" и "4 ядра" (для четырех и более ядер). Понятно, что если ЦП двухъядерный, активировать "4 ядра" без надобности.

Настройка управления питанием

Оптимизация работы процессора: утилита CPU Control

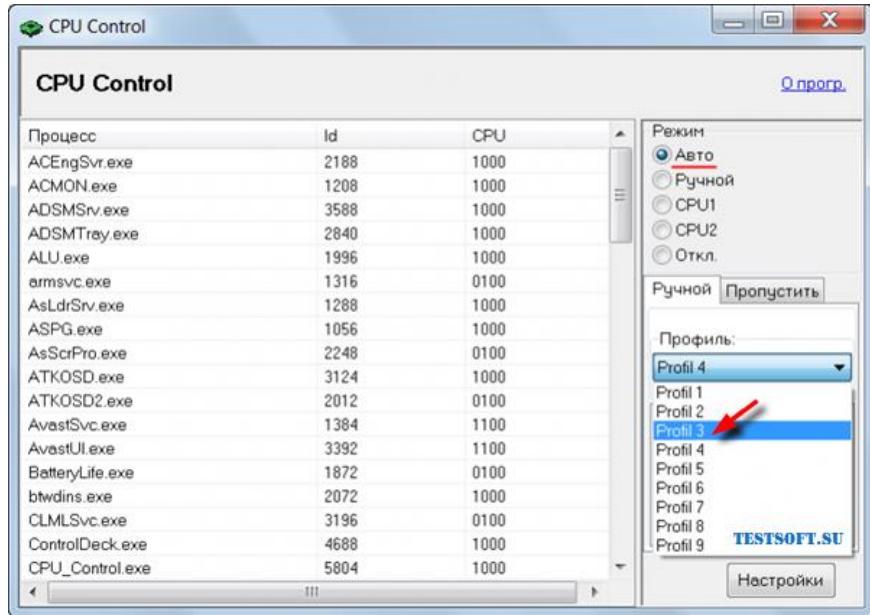


- Закрыв "Опции", в главном окне приложения видим 5 режимов контроля работы ЦП: "Авто", "Ручной", "CPU1" (все процессы выполняются на 1-ом ядре), "CPU2" (все процессы выполняются на 2-ом ядре) и "Отключено" (без оптимизации). Последние три режима нам не интересны, поэтому сосредоточимся на первых двух.

Для начинающих пользователей рекомендуется просто выбрать режим "Авто" и, закрыв окно, проверить для будущих загрузок Windows, чтобы CPU Control был в списке автозагружаемых приложений (!). Это легко сделать, например, через Revo Uninstaller любой версии, смотрите в панели меню "Инструменты" → "Менеджер автозапуска" (при необходимости добавить приложение кликаем аплет "Добавить").

Настройка управления питанием

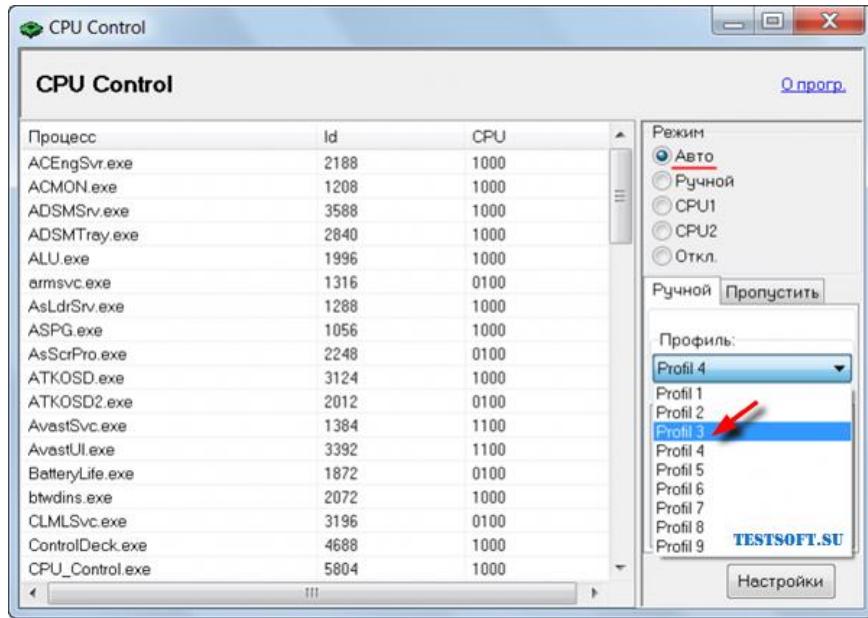
Оптимизация работы процессора: утилита CPU Control



- Для любителей "покопаться" эта простенькая утилита дает шанс "развернуться": в авторежиме имеется 9 базовых профилей-вариаций распределения процессов по ядрам (скриншот ниже). При наличии времени и открытом на вкладке "Быстродействие" Диспетчере задач ("Ctrl + Alt + Del"), для мониторинга результатов, можно выбрать оптимальный профиль под свою систему.

Настройка управления питанием

Оптимизация работы процессора: утилита CPU Control

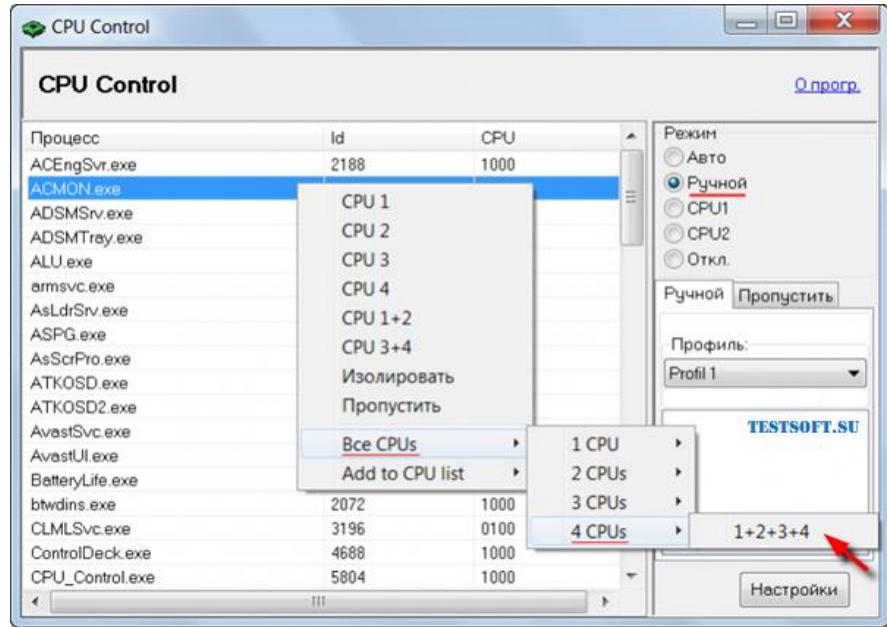


- Самые "продвинутые" пользователи, желающие добиться максимальных результатов в деле повышения производительности системы данным способом, наверняка заинтересуются режимом "Ручной". В этом режиме вы самостоятельно распределяете процессы по ядрам или группам ядер.

Все просто: выделяете процесс(-ы) и через правую клавишу мыши "вешаете" на нужное ядро или комбинацию ядер. Например, при выборе "CPU1" процесс будет "висеть" на первом ядре, при выборе "CPU3+4" – на третьем и четвертом ядрах и т.д.

Настройка управления питанием

Оптимизация работы процессора: утилита CPU Control



- Вот хороший вариант ранжирования для 4-ядерного ПК: системные процессы оставьте 1-ому ядру, самые ресурсоемкие приложения (типа Photoshop) "распылите" между всеми ядрами (вариант как на последнем скриншоте), а "середнячков" типа антивирусного сканера отдайте в руки комбинации на 2 ядра, скажем, "CPU3+4".

Диспетчер задач Windows поможет сделать правильный выбор. Для удобства мониторинга, при открытой в Диспетчере вкладке "Быстродействие", проверьте, чтобы была активирована функция "По графику на каждый ЦП" (см. "Вид" → "Загрузка ЦП" → искомое "По графику на каждый ЦП").

Тема 1 Совместимость программного обеспечения

- Оптимизация использования памяти.

Тема 1 Совместимость программного обеспечения

- Оптимизация использования жесткого диска.

Оптимизация использования жесткого диска

Отключение индексирования

- Данный способ подойдет для тех, кто не использует функцию поиска в операционной системе.
- Основой метода является закрытие раздела винчестера от индексирования, что обеспечит ускорение его работы. Для этого:
 1. Переходим в «Мой компьютер». Там нужно вызвать контекстное меню на главном диске, зачастую это диск C:\. Затем нажимаем «Свойства».
 2. В меню «Общие» выбираем пункт «Разрешить индексировать содержимое...». Нужно снять галочку и подтвердить действие.
 3. После данных действий появится оповещение, где будет сообщаться про возможность использования новой политики только к определенным файлам. Выбираем «Продолжить», а затем «Пропустить все».

Оптимизация использования жесткого диска

Отказываемся от ведения журнала

- В стандартных настройках в системе имеется функция ведения журнала. Там отображаются все сбои и произведенные изменения. Если данную систему Вы не используете или даже не понимаете, что это значит, то можете смело ее отключить.
- Инструкция по отключению:
 1. Запуск командной строки. Наиболее простейший способ: нажатие «Win+R», а в появившемся окне «Выполнить» ввести «cmd».
 2. Откроется приложение, где требуется ввести: fsutil usn deletejournal /D C:, где C – это буква необходимого жесткого диска.
 3. Для подтверждения своих намерений нужно нажать на «Enter». Затем командную строку необходимо закрыть.
 4. Для вступления в силу изменений, компьютер необходимо перезагрузить.

Оптимизация использования жесткого диска

Отказываемся от ведения журнала

- В стандартных настройках в системе имеется функция ведения журнала. Там отображаются все сбои и произведенные изменения. Если данную систему Вы не используете или даже не понимаете, что это значит, то можете смело ее отключить.
- Инструкция по отключению:
 1. Запуск командной строки. Наиболее простейший способ: нажатие «Win+R», а в появившемся окне «Выполнить» ввести «cmd».
 2. Откроется приложение, где требуется ввести: fsutil usn deletejournal /D C:, где C – это буква необходимого жесткого диска.
 3. Для подтверждения своих намерений нужно нажать на «Enter». Затем командную строку необходимо закрыть.
 4. Для вступления в силу изменений, компьютер необходимо перезагрузить.

Оптимизация использования жесткого диска

Очистка от ненужных файлов

- Довольно часто показатели скорости работы жестких дисков заметно улучшаются после удаления ненужных файлов, которые за длительное время могли собраться на накопителе. Чем больше свободного пространства на винчестере, тем быстрее будет происходить поиск файлов, а это ускорит работу.
- Если этот процесс рассмотреть более подробно, то получается, что система не будет производить долгий поиск в своих таблицах адреса определенных единиц информации. При удалении не нужных данных количество таких таблиц уменьшится, чем при заполненном пространстве.
- Поэтому нужно чаще удалять файлы с компьютера, которые не используются. Если они важные, то можно просто их собрать в одну папку и заархивировать. Подобная процедура освободит место, а в параметрах основной таблицы будет только один адрес.

Оптимизация использования жесткого диска

Очистка от ненужных файлов

- Чтобы очистить диски в Виндовс 7 и иных версиях, используются специальные инструменты.
- Они располагаются в меню «Пуск» в категории служебных программ. Точное название – «Очистка диска». Программа производит поиск неиспользуемых файлов в разделе, а затем предлагает пользователю их удалить.
- Кроме этого существует большое количество программ, обеспечивающих проведение данной процедуры.

Оптимизация использования жесткого диска

Дефрагментация

- Когда производится запись новой информации на устройство, она не всегда размещается рядом. Размещение отдельных пакетов производится по свободному месту тома. После проведения многочисленных записей, один файл может быть расположен по всему носителю. Это приводит к тому, что система постоянно обращается к разным секторам. Используя дефрагментацию, можно искать подобные данные и собирать их вместе. Затем система будет обращаться к соседним местам, что увеличит скорость работы.
- Подобный инструмент способен помочь только при использовании HDD дисков.
- Функция имеется в различных программах, которые предлагают сервисы по ускорению работы компьютера.

Оптимизация использования жесткого диска

Дефрагментация

- Операционные системы от Microsoft имеют возможность проводить дефрагментацию с использованием встроенных возможностей.
- Для этого в «Компьютере» необходимо выбрать нужный диск и произвести запуск контекстного меню.
- Затем выбирается «Свойства», далее пункт «Сервис» и указывается «Оптимизировать». Затем ОС предложит выполнить дефрагментацию.

Оптимизация использования жесткого диска

Переразметка

- В случаях, когда компьютер используется длительное время и работает винчестер на полную мощность, можно произвести переразметку. Это поможет при наличии двух и более разделов.
- Это форматирование диска, а некоторые разработчики указывают, что заново «режется» жесткий диск на сектора и дорожки.
- Идеальным является вариант, когда в системе применяется один жесткий диск с единственным томом. Нормальной считается система с двумя разделами.
- При большем количестве – система будет более медленно работать, тратя время на поиск необходимых файлов. В данном процессе поиск осуществляется по всем жестким дискам поочередно. Можно один жесткий диск использовать для установки ОС и софта, на другой записывать фильмы и прочие мультимедийные файлы.

Оптимизация использования жесткого диска

Перемещение файлов на съемный носитель

- Неплохим вариантом считается перемещение неиспользуемых данных на отдельные носители. Зачастую для этого применяются внешние накопители. Важно выбирать надежные внешние HDD, которые будут отличаться высокой скоростью обработки информации.
- Изделия можно использовать не только для хранения устаревших файлов, но и в виде флешки. Для подобных целей 500 Гб будет достаточно. С компьютером он соединяется по USB интерфейсу и отличается бесшумной работой.

Оптимизация использования жесткого диска

Отключение автозагрузки программ

- Наличие лишних программ в автозагрузке существенно замедляет работу компьютера и использует его процессор и память. Если их отключить, то можно увеличить скорость работы диска и в целом производительность системы.
- В разных версиях Виндовс управление автозагрузкой располагается в разных местах. Наиболее удобно для подобной цели использовать утилиту «CCleaner». Требуется осуществить запуск утилиты и перейти в меню «Сервис/Автозагрузка».
- В автозагрузке нужно оставить только те программы, которые нужны для работы компьютера, например, антивирус. Остальные компоненты можно отключить кнопкой «Выключить» в правой панели.
- Любую из отключенных программ можно всегда снова запустить вручную. Если пропадет значок в системном трее, то ее запуск можно вернуть с использованием кнопки «Включить».

Оптимизация использования жесткого диска

Обновление драйвера контроллера диска

- От актуальности драйверов зависит стабильность всей системы. Рекомендуется обновлять драйверы до новейшей версии, особенно это касается SSD дисков.
- Для проверки версии установленных драйверов нужно зайти в «Диспетчер устройств», вызвать командную строку, а в ней ввести «devmgmt.msc» и нажать «Enter». Затем нажать правой кнопкой мыши по SATA контроллеру и выбрать пункт «Свойства».
- Если используется старый драйвер, то новый можно найти на официальном сайте производителя системной платы или ноутбука, скачать и обновить его.
- Существуют специальные утилиты, которые позволяют обновить драйверы. Можно выбрать софт «Driver Booster», который при запуске покажет список драйверов, для которых требуется обновление.

Тема 1 Совместимость программного обеспечения

- Оптимизация использования сети.

Тема 1 Совместимость программного обеспечения

- Инструменты повышения производительности программного обеспечения.

Инструменты повышения производительности программного обеспечения.

Общие вопросы быстродействия программ.

- Быстродействие программ (ПО) зависит от многих факторов, но основными из них являются два:
 - Соотношение между реальными системными требованиями ПО и существующей аппаратной конфигурацией ЭВМ;
 - Алгоритмы работы ПО.
- Если низкое быстродействие обусловлено первым фактором, то решением является модернизация аппаратной части (hardware). В некоторых случаях проблему можно решить также с помощью тонкой настройки hardware и операционной системы. Однако этот путь имеет ряд недостатков:
 1. Увеличивается производительность hardware, а вовсе не быстродействие ПО;
 2. Производительность hardware ограничена возможностями существующих в данный момент элементной базы и инженерных решений в данной области;
 3. Большие финансовые затраты на модернизацию и настройку по причине высокой стоимости комплектующих ЭВМ и услуг специалистов требуемой квалификации.

Инструменты повышения производительности программного обеспечения.

Общие вопросы быстродействия программ.

- По этим причинам при разработке ПО прибегают к увеличению его быстродействия с помощью различных средств программной инженерии. Это позволяет:
 1. Обеспечить работу нового ПО на уже существующем оборудовании;
 2. Разработать масштабируемое ПО;
 3. Значительно уменьшить финансовые и трудовые затраты при внедрении.
- **Вместе с тем и у этого пути имеется ряд недостатков:**
 1. Значительно усложняется процесс разработки ПО, так как более «быстрые» алгоритмы сложнее более «медленных» (на пример алгоритм бинарного поиска сложнее, чем алгоритм линейного поиска);
 2. Реализация более сложных алгоритмов, как правило, требует привлечения специалистов более высокой квалификации;
 3. В случае работы с большими объёмами данных или выполнении задач требующих больших и сложных вычислений, ресурсоёмкость ПО всё равно остаются достаточно высокой. Несмотря, на какие либо способы увеличения быстродействия.

Инструменты повышения производительности программного обеспечения.

Общие вопросы быстродействия программ.

- **Таким образом, в общем случае обеспечение быстродействия ПО является комплексной задачей.**
- Однако следует заметить, что среди существующих задач, очень немногие обладают высокой ресурсоёмкостью. Вследствие этого в большинстве случаев не требуется никаких действий относительно hardware и требуемого результата можно достичь, прибегая только к программной инженерии.
- **Программная инженерия предоставляет несколько способов увеличения быстродействия программ.**
- Рассмотрим их на примере языков программирования Delphi и Assembler.

Инструменты повышения производительности программного обеспечения.

Увеличение быстродействия программ.

- Увеличить быстродействие ПО можно соответствующим образом реализовав его алгоритмы.
- Качественным показателем быстродействия алгоритма (а, следовательно, и ПО) является время его выполнения, измеренное по специальной методике, так называемого профилирования.
- Таким образом, в общем случае выбор наиболее «быстрых» алгоритмов сводится к измерению времени их выполнения и сравнении полученных результатов между собой. Такой способ анализа быстродействия является наиболее объективным.

Инструменты повышения производительности программного обеспечения.

Увеличение быстродействия программ.

- На протяжении многих лет программистами был накоплен большой опыт профилирования, который позволяет сделать определённые выводы относительно возможности оптимизации быстродействия ПО ещё на стадии написания.
- Эти выводы были обобщены и представлены в виде определённых рекомендаций. Если программист будет следовать данным рекомендациям, то написанная программа вероятнее всего будет обладать большим быстродействием, чем в случае их игнорирования.
- Однако следует ещё раз подчеркнуть, что достоверные сведения о быстродействии может дать только профилирование.
- Это обусловлено тем, что быстродействие алгоритма определяет в первую очередь его конкретная реализация.

Инструменты повышения производительности программного обеспечения.

Увеличение быстродействия программ.

- В чём же состоят выше упомянутые рекомендации? Их краткое содержание применительно к языку программирования Delphi приведено ниже.
- 1. При написании кода программ рекомендуется избегать процедур, состоящих из сотен строк. Практически всегда в них можно выделить блоки, которые лучше оформить в виде отдельной процедуры. Возможно, позже вы ей даже воспользуетесь где-то в другом месте. Не говоря уже о том, что это повышает понимание программы и вами, и другими программистами. К тому же так проще искать «узкие» места в программе.
- 2. Использование оператора case (switch) вместо многократных if... then... else (if... else). Во втором варианте компилятор будет выполнять проверку условия столько раз, сколько у вас вариантов. В первом проверка выполняется лишь однажды.

Инструменты повышения производительности программного обеспечения.

Увеличение быстродействия программ.

- 3. Некоторые действия могут быть довольно продолжительными, поэтому рекомендуется выносить за рамки цикла всё, что можно выполнить вне его, чтобы избежать большого числа повторений внутри цикла.
- 4. В циклах типа `for` нужно стараться, чтобы значение счетчика уменьшалось до нуля, а не наоборот — начиналось с нуля. Это связано с особенностями процессора. Сравнение с нулём выполняется гораздо быстрее, чем с другим числом.
- 5. Пользоваться типом `Variant` только при необходимости. Операции над этим типом сложнее, чем, например, над `Integer` или `String`.
- 6. Не злоупотреблять «программированием на компонентах». В частности не использовать компонент `TTreeView` для хранения древовидных структур данных — он работает очень медленно и предназначен только для визуального отображения. В случае работы со структурами данных лучше использовать алгоритмы, созданные самостоятельно на основе фундаментальных.

Инструменты повышения производительности программного обеспечения.

Увеличение быстродействия программ.

- 7. Сохранение и загрузка свойств компонентов с помощью методов ReadComponent и WriteComponent работает довольно медленно, поэтому по возможности рекомендуется сохранять и восстанавливать состояние программы между сессиями при помощи других способов.
- 8. Заменить простой в реализации алгоритм на более сложный, но с большим быстродействием. Например, если заранее известно, что в списке для поиска будет много элементов, лучше его отсортировать и применять бинарный поиск вместо линейного.
- 9. В критических с точки зрения быстродействия местах программы делать вставки на ассемблере. Команды ассемблера напрямую транслируются в машинный код. Таким образом, в отличие от высокоуровневых языков при компиляции отсутствует проблема синхронизации и ряд других негативных обстоятельств.
- Для других языков программирования вышеприведённый список может несколько отличаться, в частности отсутствием поддержки ассемблера и как следствие возможности оптимизации с его помощью (Java, Visual C# и др.).

Инструменты повышения производительности программного обеспечения.

Увеличение быстродействия программ.

- Особо следует отметить, что рекомендации 3 и 4 применяются не только для языков высокого уровня, но и для ассемблера.
- Помимо вышеуказанных для увеличения быстродействия программ написанных на ассемблере, в том числе и вставок, существуют следующие рекомендации:
 1. Замещение универсальных инструкций на учитывающие конкретную ситуацию, например, замена умножения на степень двойки на команды сдвига (отказ от универсальности).
 2. Уменьшение количества передач управления в программе: за счет преобразования подпрограмм в макрокоманды для непосредственного включения в машинный код; за счет преобразования условных переходов так, чтобы условие перехода оказалось истинным значительно реже, чем условие для его отсутствия; перемещение условий общего характера к началу разветвленной последовательности переходов; преобразование вызовов, сразу за которыми следует возврат в программу, в переходы («сращивание хвостов» и «устранение рекурсивных хвостов») и т.д.

Инструменты повышения производительности программного обеспечения.

Увеличение быстродействия программ.

- 3. Максимальное использование всех доступных регистров за счет хранения в них рабочих значений всякий раз, когда это возможно, чтобы минимизировать число обращений к памяти, упаковка множественных значений или флагов в регистры и устранение излишних продвижений стека (особенно на входах и выходах подпрограмм).
- 4. Использование специфических для данного процессора инструкций, например, инструкции засылки в стек непосредственного значения, которая имеется в процессоре 80286 и более поздних. Другие примеры – двухсловные строковые инструкции, команды перемножения 32-разрядных чисел, деление 64-разрядного на 32-разрядное число и умножение на непосредственное значение, которые реализованы в процессорах 80386 и 80486. Программа должна, разумеется, вначале определить, с каким типом процессора она работает!