



Журнал
для
пользователей
компьютеров

#2 (36)
февраль 2001

Издательство "Техно-ПРЕСС", Санкт-Петербург

Компьютер для
нового века

Экономьте на...
процессоре

Виртуальное
диджейство

Win2k.
Программная
совместимость

Записки на манжетах

"Полуось"

Рукопись,
найденная
в ZIP-архиве



Связанные одной сетью

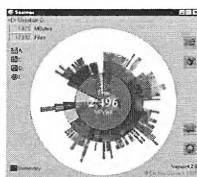
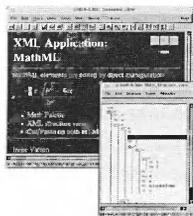
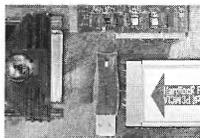
E-mail:

tmt@mail.wplus.net

http://www.magicpc.spb.ru

Поддержку сайта осуществляет "Ланк-Интернет"

№ 2(36) февраль '2001



КОМПЬЮТЕРЫ

Alt+F4 — надпись на майке Победителя.....	2
Компьютер для нового века.....	4
Экономьте на... процессоре.....	6
Локальная сеть малого предприятия.....	8
Что нам стоит сеть построить?.....	8

НАЧИНАЮЩИМ

Не слушайте "профессионалов".....	12
-----------------------------------	----

ФОТОИСКУССТВО И ПК

Image Sprayer Tool — кисть фотореализма.....	14
--	----

МУЗЫКАЛЬНЫЙ ПК

Почти настоящая гитара.....	17
Виртуальное диджейство.....	19

БДИ!

HASP, HARDLOCK и другие. Что новенького?.....	22
---	----

ИНТЕРНЕТ

Поиск информации в Интернете.....	25
Индексирование веб-документов.....	27
Эффективная работа в Интернете.....	30
Прокси-сервер. Обратная сторона медали.....	32
Какие бывают скрипты.....	34
Создание CGI-скриптов.....	37
Копилка веб-мастера.....	40

НОМО COMPUTERUS

Сто выдающихся деятелей IT ушедшего столетия.....	44
Джон Непер: 450 лет в истории науки.....	46
Профессиональная пригодность.....	48

ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ

Секреты файловой системы.....	50
Win2k. Программная совместимость.....	52
Искусство жить по средствам.....	55
"Полуось".....	58
Без If-ов и Switch-ей.....	60
Записки на манжетах.....	62
Паскаль.....	64

МУЛЬТИМЕДИА

Королевство сгорело, короли разбежались.....	66
--	----



Часть вторая, субъективная



И.Е. Сетевой

Alt+F4 — надпись на майке Победителя

Итак, в первой части было высказано предположение, что именно агрессивная политика MicroSoft в немалой степени способствовала вытеснению с рынка большей части профессиональных программистов, породила пиратство и компьютерные вирусы.

Этапы "большого" пути

В нашей стране в конце 80-х — начале 90-х годов прошлого века (как уже принято писать в контексте 2001 года) лишь немногие программисты в первых рядах получили доступ к архитектуре Intel в лице первых персональных компьютеров IBM PC XT/AT-286. В основном это были программисты, работавшие в сфере ВПК, который мог выделять сумасшедшие по тем временам средства на, как это тогда называли, ПЭВМ. Для сравнения: первый ПК AT-286, за которым я начал освоение новых технологий, стоил в 1989 году 82000 рублей, а "Жигули" 5-й модели официально стоили 5700 рублей.

Воспитанный на системе команд IBM 360/370 и микрокоде отечественной ЭВМ М6000, я, помнится, пришел в неопишуемый восторг от возможностей настольного чуда техники. Завораживали скорость, с которой можно было получать конеч-

ный результат, и полный доступ к внутренностям компьютера, состоявшего всего из нескольких плат.

Счастье "копания в кодах" и безоблачного освоения новых технологий продолжалось недолго, вплоть до полноценного вступления новой России в рынок. До этого момента я мог еще считаться полноценным специалистом — необходимость адаптации программных продуктов требовала детально разбираться в коде программ, а скромные вычислительные мощности первых ПК заставляли бороться за каждый байт, что без детального знания предмета было невозможно. В начале 90-х "почтовый ящик", в котором я имел честь процветать, потихоньку "дал дуба", и многие специалисты в области информатики, подобно мне, поддались на вольные хлеба, а именно — в торговлю компьютерами.

Здесь ставка делалась исключительно на результат, и фирмы, вагонами по бартеру ввозившие тогда персональные компьютеры и делавшие на этом шальные деньги, требовали "завести" технику и поскорее избавиться от нее, чтобы обернуть средства для закупки следующей партии ПК. Так из системных программистов получались "железячники". И дело не в том, что они были семи пядей во лбу и сумели освоить основы электроники, которые дол-

жен был знать специалист по "железу". Просто архитектура ПК в большинстве случаев позволяла выдавать нужный результат без какой-либо специальной подготовки!

Потом начался бум на базы данных, от бухгалтерских программ до всевозможных справочно-нормативных систем, и бывшие "железячники" становились свежеспеченными прикладными программистами. Затем был период расцвета компьютерной верстки печатных изданий, затем спрос на администраторов локальных компьютерных сетей, теперь, как не трудно догадаться, на веб-мастеров — рынок, направляемый тандемом Intel и MicroSoft, диктует свои условия, не заботясь о том, что на подготовку даже посредственного специалиста требуется несколько больше времени, чем время между релизами Windows и новыми поколениями процессоров.

Бурное развитие информационных технологий (ИТ) в сторону потребительского рынка сформировало аналогичный подход и к прикладным системам. На первое место выдвигаются не функциональность, эффективность и безопасность кода, а его потребительские качества — "навороты" интерфейса, скорость разработки и быстрота окупаемости программного продукта.

Из-под пера той же MicroSoft вы-

ходят visual-среды разработки программ, а одним из основных языков программирования становится Basic, что означает Beginner's All-purpose Symbolic Instruction Code — язык для начинающих, первоначально предназначенный для обучения программированию.

Упрощение инструментальных средств, в свою очередь, привело к тому, что в сферу ИТ ринулось огромное число этих самых beginners, способных за короткий срок создать то, что выглядит "законченным продуктом" в глазах руководителей больших и малых фирм, не имеющих и понятия о том, каким этот самый продукт должен быть в действительности.

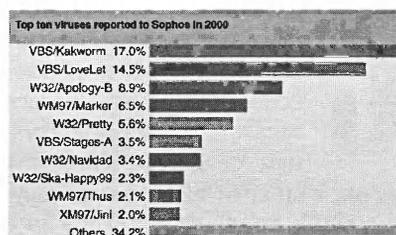
Сложилась ситуация, когда подмена настоящих специалистов недочками-универсалами стала обычным делом. В этих условиях было утеряно главное — культура производства. Ведь носитель профессии — это не просто человек, обученный определенным навыкам. Это в первую очередь специалист, изучивший историю и накопленный опыт в своей сфере, концентрирующий передовые методы. Применительно к ИТ это становится еще более актуальным. Именно поэтому диплом "университета профсоюзов" никогда не встанет на одну доску с дипломом Ленинградского Технологического института или МГУ — купить можно все, кроме истории и культуры образования. Впрочем, это к слову — просто "за державу обидно".

В итоге на ИТ-рынке возникла неконкурентоспособная среда, в которой профессионалы борются за право достойно представить доведенный до ума программный продукт, а "блшинный рынок", в котором большое число невостребованных в своих областях специалистов, вооруженных "простыми и эффективными" с их точки зрения инструментами, борются за то, чтобы оказаться в нужное время в нужном месте и выдать на-гора кучу красивых "окошек", не имеющих никакого отношения к законченной системе, способной работать самостоятельно и эффективно.

На фоне этой армии "специалистов" профессионалам все реже находится постоянное и достойное

место — их работа не выглядит более впечатляющей, а стоит дороже. Здесь следует упомянуть и еще одну "национальную особенность": у нас работник в глазах руководителя и государства не является "средством производства", тогда как на Западе "качество" персонала входит в оценочную стоимость компании, а затраты на обучение приравнены к производственным затратам.

И ныне продолжаются рыночные ротации реальных специалистов из одной ИТ-области в другую, но вопрос, насколько очень хороший прикладной программист сможет стать очень хорошим сетевым администратором или веб-мастером в условиях цейтнота, остается открытым.



VBS — VisualBasic Script;

WM97 — макровирус MS Word'97;

XM97 — макровирус MS Excel'97,

W32 — вирус, использующий системные средства Windows; LoveLet — самый шумевший вирус "Love It" и его многочисленные клоны, распространяемые через e-mail.

Так обстоят дела у нас. А что же просвещенная Европа? Тут другие проблемы и, в противоположность нашим, они идут исключительно от высокого уровня жизни и социальной защищенности всех без исключения слоев населения.

К примеру, какой смысл добропорядочному немецкому бюргеру изучать основы построения сложных систем и языки программирования, если, окончив за два года то, что у нас называлось ПТУ, он может почти за те же деньги (с разницей в 10—15 тыс. марок в год) встать на 6 часов в день, включая обеденный перерыв и ланч, к конвейеру Mercedes-Benz и 25 лет следить за тем, как автомат закручивает одну и ту же гайку. Его обязанности сводятся к тому, чтобы вовремя нажать большую красную кнопку, если гайка вдруг перестала

закручиваться. Если гайку по технологическим причинам вдруг заменят на плашку, в дело тут же вступают профсоюзы и заставляют фирму либо вернуть гайку, либо провести ускоренные курсы обучения "Основные отличия закручивания плашек по сравнению с гайками", а кроме того — либо пересмотреть время реакции работника при нажатии на кнопку, либо повысить ему зарплату.

Кстати, таких специалистов по всякому роду гайкам в последнее время интенсивно набирают в России: известно заявление правительства Германии о потребности в более чем 30-тысячной армии программистов. Я, конечно, утрирую, но в общем дела обстоят именно так.

Вирусы и пиратство

Все остальное плавно вытекает из того, что описано выше. Строго говоря, есть достаточное количество объективных причин, помимо политики MicroSoft, которые привели к удручающему состоянию дел в ИТ-сфере.

Возьмем, например, компьютерные вирусы. Если в начале 90-х написание кода вируса было прерогативой действительно высококлассных программистов, то через 10 лет картина резко изменилась. Вот как примерно по оценке компании Sophos (www.sophos.com) выглядит хит-парад современных вирусов.

Просуммировав первые три типа вирусов, получим, что почти в 48% случаях практически не требовалось знаний операционной системы. Поэтому при появлении очередного вируса советую усомниться в гениальности его 10-летнего автора, а направить вопрос в службу поддержки MicroSoft, если, конечно, вы обладаете легальной копией Windows и того продукта, который подвергся заражению.

Нелегальное копирование программ имеет ту же природу, что и вирусы, плюс социальную страсть всех слоев общества к халяве.

В основе проблем вирусов и компьютерного пиратства, на мой взгляд, лежит одна и та же проблема, связанная с созданием больших си-

стем: один человек создать систему не в состоянии, а много специалистов не в состоянии осуществить тотальный контроль ее целостности, одной из составляющих которой является собственно безопасность. Все это сильно зависит от времени создания и тестирования системы, что в условиях потогонного рынка, созданного MicroSoft, приводит к устойчивости сложившейся ситуации.

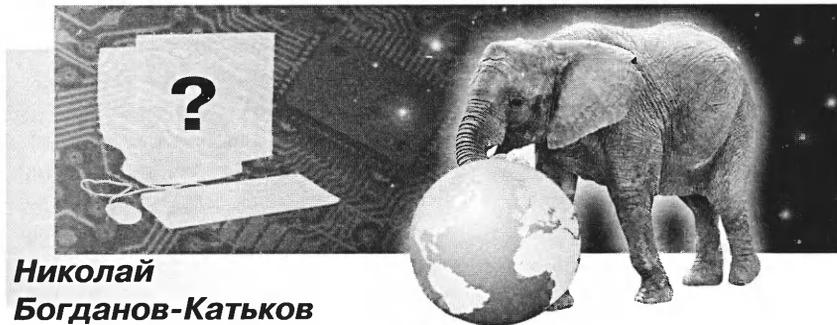
С другой стороны, упомянутое выше вымывание профессионалов уже вызвало дефицит менеджеров высшего звена ИТ-индустрии, в обязанности которых как раз и входит контроль за целостностью системы, состоящей из большого количества так или иначе влияющих друг на друга модулей.

Итак, низы по разным причинам не могут, а верхи просто отсутствуют. Не знаю, можно ли назвать это революционной ситуацией, но до тех пор, пока потребительская гонка будет искусственно взвинчиваться усилиями Microsoft+Intel, мы будем иметь то, что имеем, с каждым разом переходя качественно новый уровень информационной безответственности. И ситуация, показанная в фильме



“Хакеры”, в ближайшем будущем может превратиться в реальность.

Ну, и анекдот напоследок, чтобы не было так грустно. *Попадает Билл Гейтс в рай. Апостол Петр водит его по саду — все как надо, тепло, пляж, девушки и еда в изобилии. Договорились, что после смерти Билл за свои заслуги именно сюда и попадет. Пришло время, попадает он, так сказать, по назначению и видит — пустыня, дубак и есть нечего. Как же это, Петр, — спрашивает Гейтс, — а где девушки, море, пляж? — Извини, Билл, — отвечает апостол, — то была демо-версия.*



**Николай
Богданов-Катков**

Компьютер для нового века

Сейчас в прогнозах нет недостатка. Если на протяжении последних лет все компьютерные издания добросовестно подводили итоги года, публикуя перечни лучших товаров, появившихся за год, и строили прогнозы на следующий год, то теперь пытаются прогнозировать пути развития компьютерной техники кто на десять лет, а кто и на сто. И это несмотря на то, что всеми признано: техника и технологии делают немислимые пируэты, каждые 4—5 лет появляются новые, неожиданные направления.

О том, каким будет персональный компьютер в новом веке, аналитики начали задумываться еще два—три года назад. Многие прогнозы остаются актуальными и сейчас.

К прогнозам надо прислушиваться. В продаже одновременно есть и новинки, которые как правило стоят дорого, а в скором времени заметно подешевеют, и то, что уже устарело или устареет в ближайшее время. Оптимум лежит где-то посередине.

Двигатель технического прогресса

“В 2001 г. благодаря развитию микропроцессорной техники даже 700-долларовые компьютеры будут более мощными, чем лучшие машины 1998 года”. Разумеется, это так. 700 долларов стоил тогда один только процессор Pentium II на 400 МГц, а “лучшая машина 1998 года” на его

основе, стоившая 2—3 тысячи, в подметки не годится нынешнему компьютеру с Duron’ом. Но...

Появляется Pentium IV на 1.4 ГГц и обгоняет гигагерцовый Pentium III ровно в 1.4 раза, причем только на тестах, специально для него написанных, а в реальных задачах различие между тем и другим исчисляется процентами, причем не всегда в пользу Pentium IV... Чтобы выжать максимум производительности из этого процессора, надо не только укомплектовать компьютер 256 Мб очень дорогой памяти RDRAM, но и написать программы, специально под него адаптированные.

Но на других испытаниях тот же гигагерцовый P III отстает и от 800 МГц Thunderbird (на тесте Photoshop), и даже от 500 МГц Power Mac... Чем больше появляется таких данных, тем чаще звучат пессимистические высказывания: гонки подходят к финишу.

После выхода статьи “Гонка за скоростью. Один гигагерц” один читатель задал мне вполне естественный вопрос: за чем же следует гнаться, если не за частотой и не за скоростью?

Общий ответ можно сформулировать так. Нужна не производительность, а функциональность. Компьютер должен уметь делать все, но не “вообще все”, а все то, что нужно его владельцу. Но дальше начинаются сложности.

Все слышали, что согласно закону Мура тактовая частота процессо-

ров растет вдвое за полтора года. На протяжении последних лет этот закон соблюдался. Другой вопрос, будет ли так и в дальнейшем, ведь существуют ограничения, налагаемые законами физики и атомно-молекулярной структурой вещества. Но вот другая, малоизвестная часть закона Мура. Стоимость разработки процессора с тактовой частотой вдвое большей, чем у данного, требует расходов на порядок больше...

Между тем, цены на комплектующие постоянно меняются. Снижение цен на процессоры и другие комплектующие компенсирует дороговизну вновь появляющихся моделей. И сейчас, и три года, и шесть лет назад добротный компьютер, пригодный для огромного большинства применений, можно купить за \$600—800 (с монитором), а мощный стоит несколько больше тысячи. Разумеется, можно собрать и значительно более дорогую машину, если брать все "самое-самое".

Принято считать, что двигателем технического прогресса служит рост потребностей. Это верно лишь отчасти. Да, современный пользователь привык и к графическому интерфейсу, и к 3D-графике, и хорошая акустика ему кажется чем-то само собой разумеющимся. Но не секрет, что спрос на высокопроизводительные системы в значительной мере создается искусственно.

Программы становятся все более ресурсоемкими, а их достоинства увеличиваются отнюдь не пропорционально объему. Когда вышла восьмая версия CorelDraw, многие были разочарованы: объем возрос значительно, а возможностей прибавилось немного, да и работать стало сложнее. Теперь продвигают девятую версию. Сколько возможностей сделать документ "покрасивше" заложено в MSWord, начиная с версии 7.0! А появляется Office 97, затем 2000... С каждым новым этапом "украшательства" становится все больше, а рабочие функции задвигают все глубже, так, что ни один пользователь всех и не знает. Можно ли считать, что это вызвано реальными потребностями?

Да, но только не потребностями

пользователя. Конвейеры Intel и бригады программистов Microsoft должны работать, приносить прибыль. Что же касается реальных потребностей, то их можно оценить на примере операционной системы Linux, которая была создана людьми, работавшими не ради прибыли.

Сама операционная система занимает на диске около 100 Мб — в два с половиной раза меньше, чем Windows 98. Если добавить комплект прикладных программ, содержащих все, что только может понадобиться пользователю, получится 500 Мб. Аналогичные программы под Windows займут вдвое больше. Linux нормально работает на 486-м, а Pentium 90 под ним выполняет операции с любыми программами как Celeron.

Остаются, правда, требования к оперативной памяти. Графика, видео, обработка звука требуют ее много. Но память — дело наживное. Если бы не диктат Microsoft на рынке, если бы большинство пользователей довольствовались Linux и написанными под него программами, последние два—три года Intel работала бы впустую, самый современный процессор имел бы частоту 400—500 МГц, и этого хватило бы самым притязательным пользователям. Объемы винчестеров также были бы меньше.

Но программы становятся все более ресурсоемкими, и появляется новое "железо". Более скоростное, более емкое, более...

А куда девать старое? Широко известен основной закон маркетинга — производить то, что покупают, а не продавать то, что произведено. Куда реже вспоминают другой закон: все, что уже произведено, должно быть продано. Даже то, что устарело.

Яблоко для Евы

Осенью в одном из петербургских рекламных изданий появилась реклама известной компьютерной фирмы. Под надписью "Соблазнительное предложение" змей, высушенный из экрана монитора, протягивал румяное яблоко обнаженной женщине. Тут же приводились конфигурации и цены двух компьютеров

— мощного, с процессором Pentium III, и "ширпотребовского" с Celeron.

Достаточно сравнить предлагаемые цены с ценами других фирм на компьютеры аналогичных конфигураций в тот период, чтобы сделать вывод: коварный змей пытался надуть бедняжку Еву долларов на 30—40. Но дело даже не в этом и не в том, что для мощного компьютера 32 Мб RAM маловато. Первый компьютер имел винчестер на 4.3, а второй — на 3.2 Гб. К тому времени такие винчестеры практически исчезли из продажи, а в "средние" и мощные компьютеры чаще всего устанавливали "винты" на 10 Гб.

На то были веские причины. Цена винчестера не может упасть ниже некоторой величины, определяемой стоимостью корпуса, мотора, контроллера, кэш-памяти. Цены на винчестеры снижаются, приближаясь к этой величине, быстрее всего дешевеют более емкие модели. К осени прошлого года цены сблизились: винчестеры на 4.3 и 10.2 Гб стоили в среднем \$75 и \$80. Таким образом, все модели объема менее 10 Гб можно было смело относить к устаревшим.

Ясно, что когда разница в ценах между "лучшим" и "худшим" устройством составляет всего несколько процентов, а потребительские качества (в данном случае это форматированная емкость) различаются более чем вдвое, покупать следует "лучшее".

Но реклама не всегда пытается навязать покупателю старье. В середине января процессоры Pentium III 800 и 1000 МГц стоили соответственно \$200 и \$550, а компьютеры на их основе различаются по производительности (в данном случае это потребительские качества) всего на 8—10%. Разумеется, цены будут снижаться, но при столь малом различии в производительности можно предположить, что по соотношению цена/качество гигагерцовый P III НИКОГДА не будет опережать более ранние модели. С некоторой натяжкой можно сказать, что он устарел с момента своего появления на рынке! Между тем на его долю пришлось больше рекламы, чем на любой другой процессор Intel.

Казалось бы, что за глупость — экономить на процессоре, и это при том, что мощь и "крутость" компьютера сейчас определяются именно его характеристиками? И почему же тогда все гонятся сейчас за скоростью, почему "разгоняют" процессоры?

Безусловно, процессор является одним из важнейших компонентов компьютера. От его производительности и класса примерно на 40% зависят функциональные возможности всего компьютера. Сложность выбора обусловлена наличием в продаже нескольких поколений процессоров от нескольких производителей. Выбор производителя — дело субъективное, каждый волен выбирать, исходя из объема наличности и стоящих перед ним задач. Любой процессор предназначен для конкретного типа материнских плат и разъема на них. В свою очередь, материнская плата будет определять то, какими дополнительными устройствами вы сможете воспользоваться в будущем и насколько эти устройства окажутся вам полезными при смене процессора.

Наиболее стабильные и совместимые процессоры выпускают две фирмы — Intel и AMD. До недавнего времени процессоры Intel считались достаточно дорогими, но наиболее производительными. Сейчас положение изменилось. Продукция AMD стала качественней и производительней аналогов от Intel. Надо ли платить лишние 30—50 уе. за имя конторы?

Давайте взглянем на вопрос выбора процессора более разумно. Во-первых, рассмотрим цены. Скажем, Pentium-166 три года назад стоил около \$100. Столько же стоила и материнская плата для него. Сейчас как Pentium-166, так и его материнская плата стоят по \$25. Pentium-II еще недавно продавался за \$400—500, сейчас же его можно купить уже за \$100, а вскоре цена упадет еще заметнее.

Во-вторых, оценим потребности среднего пользователя. Для чего обычно используется компьютер? Во-первых, для работы с текстовой информацией. Во-вторых, как сред-



Экономьте на... процессоре

Антон Орлов, Петр Устинов

ство работы с Интернетом, с электронной почтой. В-третьих, для работы с мультимедийными ресурсами — прослушивания CD-дисков и файлов MP3, для работы с интерактивными CD-энциклопедиями. В-четвертых, для игр. И, в-пятых, довольно редко, — для обработки видеоматериала, записи и редактирования звука, для работы со сложной графикой.

Для работы с текстом хватит и 486-го процессора. Для работы с Интернетом и электронной почтой, а также для создания сайтов вполне достаточно младшей модели Pentium, скажем, Pentium-100. FineReader тоже будет вполне спокойно работать на таком процессоре. CD-Audio проигрываются даже на старой "четверке" (так как участие процессора минимально), а MP3-файлам хватит Pentium-166. Интерактивные энциклопедии редко требуют что-либо мощнее, чем Pentium-150. Созданием видеоэффектов заниматься в домашних условиях вряд ли кому-нибудь придется, тем более, что мощности даже самого современного процессора для этого не хватит — нужны специальные и значительно более дорогие графические станции. То же самое можно сказать и о работе с музыкой и звуком, причем для этого нужно еще и студийное оборудование.

Таким образом, для подавляющего большинства задач, которые

могут возникнуть перед пользователем домашнего ПК, вполне хватит Pentium-233 MMX или младшего Celeron. Тем более, что цены на эти процессоры значительно ниже, чем на модные Pentium-III-900 и др. Если только не прожигать жизнь, тратя время на игры. Они-то как раз требуют огромных ресурсов.

Вот на чем действительно не стоит экономить, так это на оперативной памяти. Ее стоимость практически не снижается, а при любом upgrade она может использоваться повторно. Поэтому вполне логично потратить деньги не на более новый процессор, а на большее количество "оперативки". Это, кстати, может дать компьютеру куда как лучшую производительность. Так, работа в Adobe Photoshop с большими картинками идет гораздо быстрее на Pentium-133 с 256 Мб ОЗУ, чем на Pentium-III-600 с 32 Мб ОЗУ (в последнем случае активно используется виртуальная память на винчестере, работа с которой значительно медленнее, чем с оперативной памятью).

По личному опыту на процессоре Cyrix PR-166 при 64 Мб оперативной памяти, и Windows95 в качестве операционной системы вполне можно одновременно загружать три файла из Интернета, получать почту, грузить пару страничек в Internet Explorer, слушать файл в формате MP3, вводить текст в Word и архивировать 40—50

МБ в фоновом режиме WinRar'ом. Кстати, правильный выбор операционной системы может серьезно снизить аппаратные требования.

Итак, при покупке компьютера экономить деньги, которых всегда не хватает, следует именно на процессоре, смело покупая дешевые модели, считающиеся "устаревшими". На сегодня таковыми являются процессоры Celeron и Duron с низкой тактовой частотой. Однако это не относится к материнской плате — крайне желательно, чтобы она поддерживала стандарты AGP и USB, допускала установку самых новых моделей процессоров (чтобы поставить их потом, когда цена на них упадет).

Процессоры от Intel нижней ценовой категории, аналоги процессоров Pentium II, называемые Celeron (Mendocino) имеют кэш второго уровня (L2) размером 128 Кб и тактовые частоты от 300 МГц и выше. Их достойные аналоги — процессоры AMD K6-2 с частотами от 300 до 550 МГц, но при равных тактовых частотах они обладают меньшей производительностью по сравнению с Celeron. Приобретать процессоры данного класса стоит только в расчете на скорую замену.

Процессоры Intel Pentium II с частотами 333 МГц и выше имеют кэш L2 512 Кб, работающий на половине частоты ядра процессора. Аналоги от AMD маркируются AMD K6-III и имеют хотя и в два раза меньший кэш L2, чем у Pentium II, но работающий в два раза быстрее. Зато кэш первого уровня (L1) у K6-III в два раза больше, чем у Pentium II (64 Кб).

Производительность процессоров от AMD по многим тестам выше, чем у Pentium II, а стоимость гораздо ниже (правда, найти эти процессоры в продаже довольно сложно, хотя и можно). Процессоры этого поколения очень хорошо вписываются в конфигурацию приличного во всех отношениях компьютера. На сэкономленные деньги вы сможете купить хороший 17" монитор или хорошую видеокарту с ускорителем.

И, наконец, самые передовые достижения в области процессоростроения — это процессоры AMD Athlon, Intel Pentium III и их аналоги. Именно Athlon, побив рекорды производительности, принес фирме AMD славу и признание, заслуженные вследствие умелой ценовой политики и высокого технологического потенциала. Характерные черты Athlon, соперника безликого Intel Pentium III, — стандартная частота системной шины Alpha EV6 200 МГц, громадный кэш L1 (128 Кб), новый высокопроизводительный математический сопроцессор вещественных чисел (FPU).

Та стратегия, которой следует Intel, безусловно, имеет право на существование. Добавляя новые инструкции для работы с вещественными числами, Intel добивается высокой производительности на оптимизированных под этот процессор программах. Вопрос в том, будут ли восприняты SIMD-инструкции программистами. Даже если так, произойдет это не скоро. А вот технологического преимущества у Intel нет, о чем свидетельствует тот факт, что архитектура осталась той же, что и у Pentium II.

мерена добиться увеличения производительности на 40%.

Стали наконец известны некоторые подробности о втором процессоре с архитектурой IA-64 от компании Intel с кодовым именем McKinley. Процессор будет использовать 7-ступенчатый конвейер (у Merced — 10-ступенчатый). Первые образцы McKinley появятся с тактовой частотой ГГц (против 800 МГц у Merced) и первоначально будут производиться по тому же технологическому процессу (0,18 мкм, 6 слоев, алюминиевые проводники).

Наиболее доступными можно считать следующие аналоги этих процессоров. AMD Duron — то же, что и Athlon, но с урезанным до 64 Кб кэшем L2 (хотя он достаточно мал, но работает на частоте ядра и является эксклюзивным, то есть данные, хранящиеся в кэше L1, в L2 не дублируются) и предназначен для разъема Socket A. Вторая разновидность Athlon в варианте для Socket A именуется AMD Thunderbird — полноценный процессор своего поколения с кэшем L2 256 Кб, также работающим на частоте ядра. Дешевый аналог Pentium III — Celeron Coppermine. В отличие от своего младшего брата (Celeron Mendocino) данный процессор содержит набор SIMD-инструкций SSE, ускоряющих работу с 3D-графикой, и печально известный индивидуальный номер процессора. От Pentium III для Slot 1 данный процессор отличается в два раза меньшим кэшем L2, но работающим на частоте ядра.

Однако какой бы процессор вы ни выбрали, помните, что угнаться за техническим прогрессом все равно невозможно. Народная мудрость гласит: *самый современный компьютер устареет за время, в течение которого покупатель везет его из магазина домой*. Поэтому выбирайте процессор, исходя из ваших реальных потребностей и финансовых возможностей, не опасаясь оказаться в числе владельцев "устаревшего" компьютера: главное, чтобы ваша машина работала стабильно и надежно, выполняя те функции, которые вам нужны.

McKinley будет иметь 214 млн транзисторов, поскольку кэш третьего уровня также будет размещен на кристалле. Кэши L1, L2 и L3 будут увеличены и оптимизированы для уменьшения латентности и достижения большей пропускной способности. В дальнейшем планируется перевод производства на технологию 0,13 мкм. Intel заявляет, что за счет улучшения системы кэширования и увеличения тактовой частоты McKinley будет как минимум вдвое производительнее своего предшественника Itanium.

Дмитрий Добрый

Между тем...

Transmeta планирует выпустить в этом квартале два новых процессора серии 5600 (667 и 700 МГц) с новой версией специального ПО, оптимизирующего работу процессора и тем самым экономящего энергию. Процессоры серии 5800 будут обладать тактовой частотой, начиная с 800 МГц, и меньшим энергопотреблением, поскольку будут базироваться на 0,13 мкм технологии от IBM, которая производит все чипы Transmeta. С помощью повышения частоты и нового ПО компания на-

Net? Да!



Валентин Холмогоров

Локальная сеть малого предприятия

Задумываясь о возможности создания локальной сети, руководители многих фирм зачастую сталкиваются с недостатком информации о необходимом для этого оборудовании и о технологиях, позволяющих построить такую сеть с минимальными затратами. Помочь вам разобраться в наиболее распространенных в нашей стране сетевых стандартах и призвана эта статья. Безусловно, здесь рассмотрены далеко не все возможные варианты компоновки небольшой локаль-

ной сети, а лишь те, которые можно встретить в офисах современных российских предприятий чаще всего.

10Base2

Начнем, пожалуй, с одного из самых дешевых вариантов объединения нескольких компьютеров в локальную сеть, — со схемы, которую принято называть стандартом 10Base2.

Компьютеры объединяются в локальную сеть 10Base2 последова-

тельной "цепочкой" или "гирляндой" с использованием специального экранированного коаксиального кабеля. Каждый отрезок кабеля имеет на концах круглые металлические разъемы с проворачивающимся замком-фиксатором, отдаленно напоминающие телевизионный антенный штекер. Смонтировать такой разъем и зафиксировать его на кабеле не составляет труда, для этого необходимо лишь уметь обращаться с паяльником и пассатижами.

В каждый ПК, который вы планируете подключить к локальной сети,

Что нам стоит сеть построить?

Дмитрий Апарцев

Все, о чем пойдет речь ниже, относится к маленьким фирмам. Обычно они располагают небольшим количеством компьютеров, от 2—3 до 10—15, устаревших как морально, так и физически. Именно у таких фирм есть шанс снизить эксплуатационные затраты, установив в офисе небольшую локальную сеть. Разговор пойдет только о прокладке и настройке сети. В результате вы сможете подсчитать, во сколько обойдется установка сети для уже имеющегося оборудования.

Зачем это надо?

"А действительно, зачем?" — спросит любой хозяин небольшой конторы, маленького магазина. Подобные вопросы не возникают разве что в юридических консультациях, нотариальных конторах, консалтинговых фирмах: с их огромным документооборотом они просто нуждаются в создании небольшой локальной сети. Локальная сеть дает следующие преимущества:

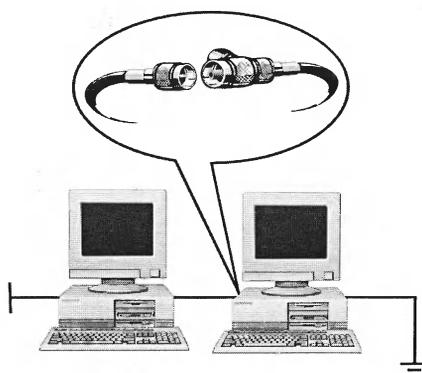
- совместное использование жестких дисков позволит избавиться

от постоянной беготни с дискетами от одного компьютера к другому;

- совместная работа над документами сэкономит кучу бумаги;
- сетевой принтер поможет обойтись без очередей на компьютер, к которому подключен принтер, и не нужно будет тратиться на содержание зверинца из разномастных принтеров;
- резервное копирование текущих и архивное копирование старых документов со всех компьютеров на CD-R или CD-RW (см. статью "Покупаем пишущий CD-ROM своей меч-

вставляется сетевая карта — устройство, обеспечивающее двусторонний обмен данными между машиной и локальной сетью. Для работы в сети 10Base2 вполне подойдут самые дешевые модели сетевых карт, например, NE2000-совместимые, ценой всего \$5—10 за штуку, или более современные карты семейства LAN Planet по цене от \$10 до \$25. Для каждого из подключаемых к сети компьютеров необходимо приобрести так называемый T-коннектор. Это металлическое устройство в форме буквы "Т", состоящее из трех разъемов: один из них вставляется в порт сетевой карты, к двум другим присоединяются кабели, идущие от соседних машин. Свободные разъемы, которые останутся на T-коннекторах крайних в "цепочке" компьютеров, закрываются специальными "заглушками", образующими на выходе сетевой карты необходимое для нормальной работы локальной сети сопротивление нагрузки. Одна из таких "заглушек" должна быть укомплектована небольшой металлической цепочкой — это "украшение" выполняет роль заземления для всей сети, обеспечивая беспрепятственный сток накапливающегося на сетевых разъемах статического напряжения. Сетевые принтеры подключаются к стандартным портам работающих в сети компьютеров.

Одно из достоинств сети 10Base2 в том, что коаксиальные сетевые кабели не требуют прокладки: их можно "набросить" между столами сотрудников или закрепить на плинтусах офисного помещения. В такую сеть может входить произвольное количество ПК, надо лишь следить за тем, чтобы длина кабеля, соединяющего отдельные узлы сети, не превышала



120 метров. Максимальная пропускная способность канала связи в такой системе составляет 10 Мбит/с, однако на практике она, как правило, значительно ниже.

К недостаткам сети 10Base2 следует отнести крайне низкую надежность: чем больше в ней компьютеров, тем больше вероятность отказа одного из сегментов сети. Достаточно небольшого смещения любого разъема в каком-либо из T-коннек-

торов (неосторожного касания сетевого кабеля), и вся сеть мгновенно перестает работать. После этого администратору придется последовательно проходить все участки сети, отыскивая плохой контакт.

10BaseT

Более дорогостоящим вариантом компоновки локальной сети является система с архитектурой Ethernet 10BaseT. В отличие от 10Base2, сеть 10BaseT имеет звездообразную структуру, в которой роль "центра" всего комплекса играет специальное устройство, называемое хабом, или концентратором. В упрощенном виде компоновка сети 10BaseT выглядит следующим образом. В офисном помещении прокладывается сетевая кабель, который принято называть "витой парой". Обычно используется кабель пятой категории с пропускной способностью 10 Мбит/с. Он представляет собой жгут изолированных проводников, запаянных в экранированную оплетку из полимерного материала.

Разводка кабеля осуществляется так, чтобы с одной стороны линии подходили к каждому рабочему месту, с другой стороны — объединялись в некой точке офиса, где будет установлен хаб. При этом длина каж-

ты", №10/1999) уберезет головы подчиненных от гнева шефа;

- доступ в Интернет с каждого компьютера упростит переписку с партнерами и контрагентами.

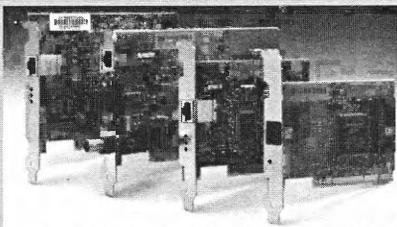
Конечно, это потребует некоторых затрат на начальном этапе и на поддержку сети в дальнейшем. Более того, результат не будет столь ощутимым, как хотелось бы, но это вполне объяснимо: сеть не увеличит напрямую выручку фирмы, при разумном подходе она поможет лишь сократить расходы (главным образом — времени и нервов).

Что включить в сеть?

Я думаю, вы не настолько богаты, как программисты из Bell Labs, которые ставят у чайника веб-камеру и

следят, не закипел ли он. Однако в любом офисе найдется 5—10 устройств, которые имеет смысл объединить в единое целое.

Прежде всего — это компьютеры. Действительно, иначе зачем все это нужно? Для подключения компьюте-



Сетевые карты

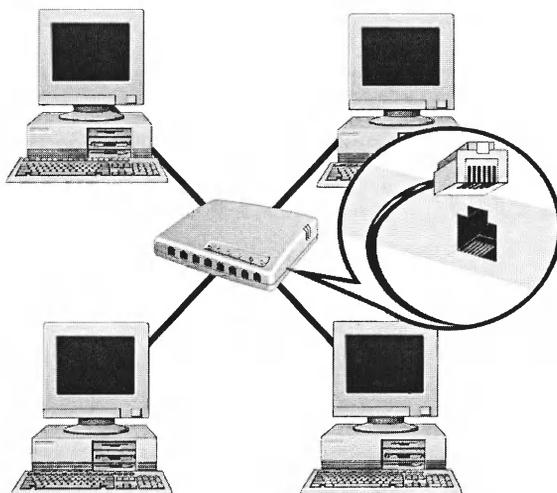
ра к сети нужен сетевой адаптер (сетевая карта). Его можно приобрести за цену от \$3—5 на рынке Юнона до \$20—30 в хорошем магазине.

Принтеры. Подключить принтер в сеть можно двумя способами. Первый, самый дешевый и простой, — подключить принтер к компьютеру, а компьютер к сети, и настроить принтер так, чтобы на нем можно было печатать с других компьютеров. Недостаток этого способа в том, что для печати на этом принтере необходимо, чтобы работал компьютер, к которому он подключен. Правда, в это время на данном компьютере можно делать еще что-нибудь. Второй способ — подсоединение принтера напрямую в сеть через принт-сервер (или, как у некоторых принтеров, — через специальную сетевую карту, которая вставляется в принтер). Стоит принт-сервер от \$80 и после уста-

дого отрезка сети должна быть менее 120 метров. На концах проводов, ведущих к хабу, с помощью специального инструмента, напоминающего пассатижи, монтируется пластмассовый разъем с замком, устройство которого аналогично разъемам на модальных шнурах или в современных электронных телефонных аппаратах. С другой стороны каждого отрезка кабеля устанавливается специальная сетевая розетка с ответным контактным отверстием. Для всех ПК сети изготавливаются небольшие отрезки кабеля, заканчивающиеся двумя аналогичными пластмассовыми разъемами (на жаргоне компьютерщиков они называются "поводками"). Каждый "поводок" одним концом соединяется непосредственно с приемным портом сетевой карты ПК, а другим вставляется в сетевую розетку. Осталось только подключить идущие от розеток отрезки кабеля к соответствующим портам концентратора, и сеть готова к работе.

Основные недостатки сети с архитектурой 10BaseT — высокая стоимость хабов (цена зависит от изготовителя и количества доступных

портов), а также сложность монтажа разъемов: инструмент для их "обжима" стоит от \$60 и выше, взять же его в аренду, как показывает практика, зачастую гораздо сложнее, чем купить. К тому же, чтобы правильно расположить тонкие проводники на



контактных площадках, необходим определенный навык.

Однако достоинства такой системы с лихвой покрывают все ее недостатки. Во-первых, сети 10BaseT гораздо надежнее и "быстрее" своих аналогов: проблем с плохими контак-

тами, от которых постоянно страдают пользователи сетей 10Base2, здесь практически никогда не возникает. Во-вторых, такая сеть мобильна: для того чтобы переставить компьютеры, уже не нужно менять всю компоновку "локалки" — вполне достаточно отключить соответствующий "поводок" и присоединить его к порту другого ПК. К такой сетевой системе очень просто присоединить новые сегменты: для этого нужно лишь связать между собой концентраторы двух различных локальных сетей, расположенных, скажем, в соседних офисах.

Часто в единую локальную сеть объединяют подсети с различной архитектурой. Это возможно в случае, если выбранная вами модель хаба имеет порт для подключения T-коннектора: присоединив к нему коаксиальный кабель, идущий от сети 10Base2, и перекрыв второй контакт коннектора заглушкой, вы можете использовать остальные порты для подключения компьютеров по витой паре по стандарту 10BaseT. Важно лишь, чтобы обе сети использовали один и тот же протокол и в них не встречались ПК с одинаковыми именами.

новки не требует к себе особого внимания.

Модем. Для использования модема надо выбрать самый мощный компьютер, так как на нем придется ставить прокси-сервер и, возможно, факс-сервер. Прокси-сервер — это программа, которая позволяет выходить в Интернет компьютерам, не имеющим собственного модема, а факс-сервер — отправить факс через модем с любого компьютера, не заботясь о том, что линия занята. Таким образом, вам уже не нужен

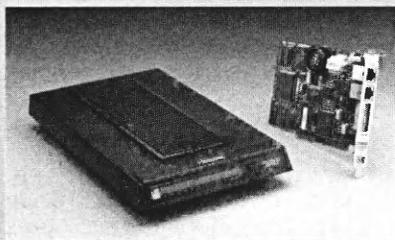
факсимильный аппарат, не расходуется лишняя бумага, да и времени на отправку факса уходит меньше. Нужно еще отметить, что модем может служить как факсовый аппарат, но это решение порождает несколько проблем: невозможность передать подпись, неустойчивая работа на плохих линиях, проблемы с программным обеспечением.

Необходимо также выбрать тип сети: одноранговая сеть или сеть с выделенным сервером. В одноранговой сети все компьютеры равноправны, на каждом компьютере можно открыть часть диска для чтения всем остальным компьютерам. Это избавит от необходимости бегать по офису с дискетками, но приведет к хаосу в хранении документов: нужный документ может оказаться на любом из компьютеров. Кроме того, в такой сети слож-

но организовать работу прокси- и факс-серверов. Одноранговую сеть можно порекомендовать фирмам с количеством компьютеров не более пяти. Сеть с выделенным сервером подразумевает, что для хранения документов, подключения принтера и модема будет выделен специальный довольно мощный компьютер, который будет заниматься только этими задачами (не стоит сажать за него секретаршу или играть на нем в игры, у него и так достаточно работы). Однако такой сервер нуждается в серьезной настройке и стоит будет довольно дорого — до \$1000.

Чем и как соединять?

Наиболее простой и доступной является сеть Ethernet, а из ее вариантов — Ethernet 10BaseT. Не надо пугаться страшных названий, это



Модем (внешний и внутренний)

Протоколы

Сетевым протоколом называют набор спецификаций и стандартов, описывающих правила обмена информацией между компьютерами, объединенными в сеть. В большинстве российских локальных сетей используется один из трех протоколов, входящих в комплект поставки MS Windows: это TCP/IP для больших сетей или сетей с возможностью подключения к Интернету, NetBEUI для небольших локальных сетей и гораздо реже — IPX/SPX-совместимый протокол, в основном для сетей, включающих в себя серверы с ПО производства компании Novell.

Протокол NetBEUI привлекателен прежде всего простотой настройки: указав название рабочей группы и уникальное имя для каждого ПК, вы, по большому счету, уже подключаете его к сети. Этот протокол подходит для небольших локальных сетей, состоящих из 5—10 ПК, однако он обладает достаточно "капризным" характером: зачастую компьютеры неожиданно перестают "видеть" друг друга в сетевом окружении, причем ничем иным, кроме проявления потусторонних сил, такие "спецэффекты" не могут объяснить даже системные

администраторы с многолетним опытом работы. После переустановки протокола и воссоздания всех прежних настроек связь столь же неожиданно восстанавливается.

Если вы хотите, чтобы все компьютеры вашего предприятия имели выход в Интернет через одну "головную" машину, оснащенную модемом, а также если ваша локальная сеть насчитывает достаточно большое количество узлов, вам необходим TCP/IP. В этом случае каждому ПК, входящему в сеть, назначается уникальный IP-адрес из четырех числовых регистров, разделенных точкой, например, 195.98.105.1. Адреса узлов одной сети должны различаться только значением последнего регистра. Сложности могут возникнуть лишь в том случае, когда ваша сеть имеет непосредственный доступ к высокоскоростной интернет-магистральной: тогда IP-адрес сети придется регистрировать в соответствующих инстанциях, тогда как адреса сетей, замкнутых "сами на себя" или соединяющихся с Интернетом по модему можно брать "с потолка" в пределах, допускаемых стандартами.

Для организации доступа к Интернету с использованием модема потребуется установка специальной программы — прокси-сервера, са-

мая популярная из которых называется WinGate. Подробные инструкции по настройке протокола TCP/IP и прокси можно найти в справочной литературе.

Установив сетевое оборудование и настроив протоколы, необходимо совершить последний шаг к оживлению вашей локальной сети: открыть доступ к ресурсам входящих в сеть ПК. Для этого в окне "Мой Компьютер" следует указать мышью диск, использовать который вы хотите разрешить всем пользователям сети, выбрать в меню, появившемся по нажатию правой кнопки мыши, пункт "Свойства", установить режим совместного использования диска (под его ярлыком появится символ горизонтально развернутой ладони) и указать вид доступа к хранящимся на диске данным. Аналогичную процедуру следует проделать с принтерами, после чего потребуется установить режим использования сетевого принтера на всех подключенных к сети компьютерах. При необходимости можно настроить на каждом ПК доступ к сетевым дискам — это необходимо, например, при использовании бухгалтерских программ 1С в многопользовательском режиме. Вот теперь сеть полностью готова к работе.

всего лишь означает, что для сети вам понадобится специальный провод (так называемая витая пара 5-й

категории) и небольшое устройство — хаб (hub), к которому с помощью этих проводов будут подсоединены все устройства. Хаб представляет собой небольшую коробку, в которой есть специальные отверстия (порты) для подключения проводов. Обычно на хабе бывает 4, 8, 12, 16 или 24 порта. Выбирайте с запасом. В крайнем случае можно соединить два хаба, но при этом на соединении вы потеряете



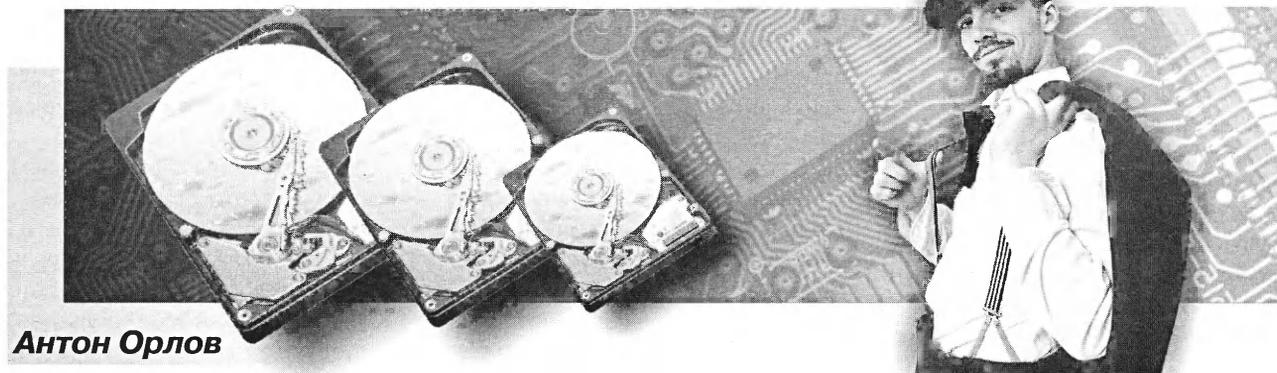
два порта. По цене хабы могут существенно различаться: от \$20 до \$500, но для небольшого офиса подойдут самые дешевые. Провод стоит от \$0,2 за метр. Кроме него понадобятся еще специальные сетевые розетки (от \$1) и вилки (от \$0,2).

Разводку кабеля и подключение всех устройств в сеть следует дове-

рнуть настоящим специалистам. На прокладку сети существуют очень жесткие стандарты, и если вы не хотите проблем в будущем, не доверяйте мастерам-самоучкам. Прокладка сети стоит в среднем около \$50 за одно подключенное устройство.

Итого

Установка и настройка сети обойдется примерно в \$75—100 за подключенный компьютер, плюс \$200—500 общих расходов. Более "продвинутый" вариант потребует дополнительно \$1000—2000. Поддержка сети штатным специалистом обойдется в 1500—2000 рублей в месяц. Но при этом в вашем офисе будет царить спокойствие, все будут заняты своей непосредственной работой, и вы будете успевать делать за день гораздо больше дел.



Антон Орлов

Не слушайте “профессионалов”

Вспомните, как вы обустроили свой первый компьютер. Наверняка многие из вас при этом сталкивались с “особо продвинутыми” пользователями, которые не могли упустить шанс дать вам “советы бывалого профессионала”. Но действительно ли все эти советы такие уж полезные? И не окажется ли так, что вам придется потом долго жалеть о том, что вы им последовали? Может быть, все эти “советы” — лишь мифы, существующие в общественном сознании?

Разбору одного из таких мифов и посвящена данная статья.

Объем жестких дисков в последнее время стал расти с огромной скоростью. Еще несколько лет назад диск емкостью 2 Гб считался очень большим, а сегодня не редкость — винчестер на 15—20 Гб. Но очень часто при первой настройке компьютера с большим винчестером все дисковое пространство отводится под один большой логический раздел, максимум два. Если вы спросите того, кто занимается разбиением винчестера, почему он делает именно так, то услышите ответы вроде “так принято”, “так лучше”, “большой логический диск — это хорошо”. Но если вы попросите аргументировать

ответ, то вряд ли услышите что-нибудь логичное.

Правильнее всего разделить ваш винчестер на несколько логических дисков и размещать на разных дисках файлы разного назначения. Скажем, на первом — системные файлы и программы, на втором — временные файлы и файл подкачки, на третьем — файлы, с которыми вы работаете, на четвертом — архивы, на пятом — игры, песни, видео. Размеры им установите соответствующие: системному — гигабайта два, для временных файлов — полутора хватит, играм — больше всего, гигабайт 4—10, сколько не жалко, а рабочим файлам и архивам — золотую середину.

В качестве файловой системы выберите FAT16 для всех дисков, кроме того, что с играми и песнями, — ему можно и FAT32. Вот увидите, работать будет значительно удобнее и безопаснее.

Имя	Тип	Полный объем	Свободно
Диск 3.5 (A:)	Диск 3.5		
Windows_95 (C:)	Локальный диск	1.99 Гбайт	91.6 Мбайт
Archive (D:)	Локальный диск	448 Мбайт	105 Мбайт
Information (E:)	Локальный диск	1.99 Гбайт	1.04 Гбайт
Information (F:)	Локальный диск	1.99 Гбайт	1.36 Гбайт
Entrance (G:)	Локальный диск	1.50 Гбайт	1.06 Гбайт
Multimedia (H:)	Локальный диск	6.80 Гбайт	4.98 Гбайт
(I:)	Компакт диск	643 Мбайт	0 байт
Панель управлен...	Системная папка		
Принтеры	Системная папка		

Вполне рациональная организация хранения данных на диске: все на своих местах, удобно и безопасно

Почему же не стоит поддаваться на уговоры “опытных пользователей”, которые иной раз делают один логический диск на винчестере в 20 Гб и считают, что так и надо? Причины следующие.

1. Разбив винчестер на несколько логических дисков, вы повышаете защищенность данных от случайного форматирования.

Многие вирусы в качестве своего разрушающего действия производят форматирование диска C:. Отформатировать диск C: можно и случайно, если, скажем, ошибиться буквой при вводе команды format из DOS. Восстановить же информацию с диска окажется очень сложно: работа утилиты Unformat затянется на долгие часы. Если же ее не будет под рукой, то компьютер вообще окажется выведенным из строя надолго, поскольку в этом случае произвести запись какой-либо информации на винчестер, в том числе устанавливать операционную систему, будет нельзя: создаваемые файлы могут легко затереть остатки старых, и тогда восстановить будет уже абсолютно невозможно. Если же винчестер разбит на несколько логических разделов, то на восстановление работоспособности машины потребуется ровно столько времени, сколько занимает простая установка операционной системы и не-

обходимых приложений. Даже в том трагическом случае, когда случайно окажутся отформатированными все логические диски (скажем, из-за действий детей, получивших доступ к отцовскому компьютеру), работу на компьютере можно возобновить, снова установив операционную систему и нужные программы. Просто на тот логический диск, где ранее находились рабочие файлы, не надо ничего записывать — пусть он дожидается утилиты Unformat или специалиста по восстановлению данных.

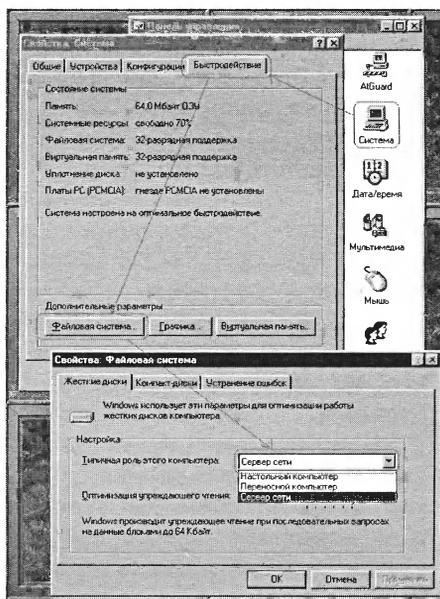
При удалении файлов с жесткого диска (в Windows — при очистке "Корзины") сама информация в ячейках жесткого диска не уничтожается. Операционная система помечает кластер, ранее занятый частью удаленного файла, как свободный, но все его содержимое остается на месте. Если в кластеры, где размещался удаленный файл, больше ничего не записывать, то восстановить файл можно с помощью специальных утилит (к примеру, Norton UnErase Wizard).

При быстром форматировании диска очищается таблица размещения данных FAT (в ней все кластеры помечаются как свободные) и корневой каталог — из него удаляется вся информация о содержащихся в нем файлах и других каталогах. Сами данные на винчестере также не стираются, и их можно восстановить программами типа Unformat.

2. При наличии нескольких логических дисков повышается быстродействие!

Когда операционной системе необходимо считать с диска файл, она узнает из таблицы FAT, в каких местах диска этот файл физически размещен, а затем уже считывает файл. При этом происходит перемещение головок по диску от области FAT до места расположения файла. Чем больше логический диск, тем большее расстояние приходится преодолевать для этого головок и тем больше тратится на это времени. Если диск разбит на несколько логических разделов, таблицы FAT размещаются в начале каждого из них, и расстояние между таблицей FAT и данными на диске становится меньше.

Для повышения быстродействия (чтобы головкам жесткого диска не приходилось постоянно перемещаться от нее к области, занимаемой файлами) таблица FAT кэшируется операционной системой — частично переписывается в оперативную память. Точно так же можно кэшировать и каталоги. Указать количество запоминаемых путей к файлам можно в параметре "Панель управления —> Система —> Быстродействие —> Файловая система —> Типичная роль этого компьютера". Установите значение этого параметра как "Сервер сети", и ваша система будет запоминать информацию о пути к 64 папкам и 2729 файлам, к которым в последнее время происходило обращение (при значении "Настольный компьютер" запоминается информация лишь о 32 папках и 677 файлах).



Настройка числа кэшируемых путей

Если размер FAT достаточно мал, она сможет уместиться в памяти целиком, и головкам жесткого диска не придется постоянно перемещаться к FAT при чтении разных файлов. А если объем FAT велик, как это обычно бывает при большом объеме логического диска, то вся она не уместится в оперативной памяти, и считывать данные с винчестера все же придется.

Когда диск разбит на несколько логических разделов, таблица FAT

становится значительно компактнее и может целиком уместиться в памяти машины.

3. При наличии нескольких жестких дисков меньше фрагментация файлов!

Если системные файлы расположены на одном диске, архивы на другом, а рабочие файлы на третьем, то их фрагментация будет меньше, чем при записи на один диск, а для чтения даже фрагментированных файлов потребуется меньше времени, поскольку системные и рабочие файлы не будут перемешаны. Кроме того, фрагменты любого файла окажутся в пределах одного логического диска, и для их чтения головкам жесткого диска придется преодолевать меньшее суммарное расстояние.

4. Обслуживание нескольких маленьких логических дисков проще и быстрее, чем одного большого!

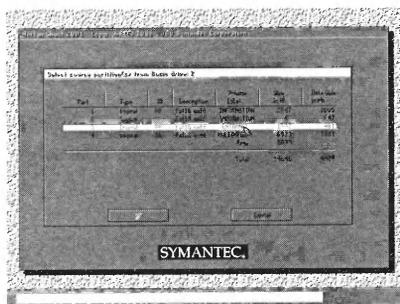
Попробуйте запустить программу дефрагментации или проверки диска во время интенсивной работы с располагающимися на нем файлами, и вы увидите, что после каждой операции записи на диск любая из этих программ начнет работать заново. Располагая несколькими независимыми разделами, вы можете работать с одним из них в то время, когда компьютер производит дефрагментацию другого.

Разбив свой винчестер на несколько логических дисков разного назначения, вы сможете запускать программу дефрагментации значительно реже и тратить на нее меньше времени, чем при одном большом логическом диске (согласитесь, дефрагментация двух гигабайт пройдет быстрее, чем двадцати).

5. Архивировать целый большой диск долго и неэффективно.

Существуют специальные программы типа Norton Ghost или Microsoft Backup, которые создают архив логического диска и помещают его на резервный носитель. Ясно, что архивировать следует лишь свои рабочие файлы и архивы — операционную систему, программы и игры проще переустановить. Но Norton Ghost может создавать архив только целого логического диска. И если у

вас и системные, и рабочие файлы находятся на одном диске, то все они окажутся в архиве. Сколько же места для этого понадобится! Если же применить Norton Ghost для архивирования установки операционной системы, а на диске с восстанавливаемой системой окажутся ваши рабочие файлы, то они будут безжалостно удалены! А если создать несколько дисков, то можно независимо архивировать и восстанавливать каждый из них.

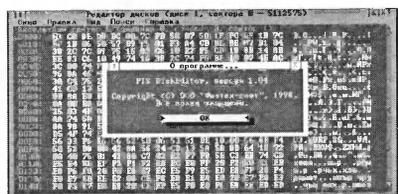


Norton Ghost. Внешний вид оправдывает имя

6. Восстановить информацию с нескольких небольших дисков проще, чем с одного большого.

Вследствие действия вируса, некорректно написанной программы, случайного форматирования файловой система, увы, бывает уничтоженной. Однако данные с диска никуда не денутся и их, в принципе, можно достать прямым редактором диска вроде PTS DiskEditor (программа бесплатная, загрузить можно с сайта компании-производителя <http://www2.PhysTechSoft.com> или с адреса <http://filetools.nm.ru/discedit.rar>). Если на диске с уничтоженной файловой системой надо найти какую-то текстовую информацию, то обязательно использовать утилиты типа Unformat, это может сделать и DiskEditor.

В ходе поиска DiskEditor просматривает все сектора диска последовательно. Ясно, что на диске



PTS DiskEditor

малого объема поиск пройдет быстрее. Если хранить рабочие файлы на отдельном логическом диске, то в случае катастрофы стоит обработать лишь его, а остальные, содержавшие неэксклюзивные данные, просто отформатировать и установить все заново.

Заключение

Итак, купив новый большой винчестер, не поддавайтесь на уговоры "профессионалов" и организуйте на нем несколько отдельных логических разделов, например, с использованием DOS-овской утилиты fdisk. Если компьютер вам нужен для работы и разумного отдыха, а не для массового убийства монстров в играх, если удобство работы на нем для вас означает прежде всего хороший результат, настоятельно рекомендую задуматься над приведенными выше соображениями. И пусть ваш компьютер служит вам долго и качественно.

Как разбить жесткий диск?

Проще всего это сделать, когда жесткий диск еще новый и на нем нет никакой информации. В этом случае следует воспользоваться программой Microsoft Fdisk, взяв ее вместе с загрузочным диском с другого компьютера, где уже установлена Windows. Запустив Fdisk, согласитесь на поддержку больших дисков (хоть на одном логическом диске FAT32 ведь вам нужна), выделите Primary Dos Partition — основной раздел DOS в 2 Гб, а все остальное место отведите под Extended DOS Partition — дополнительный раздел DOS. В последнем выделите 3—5 логических дисков объемом до 2 Гб, а остальное отведите под один большой диск. После окончания работы Fdisk отформатируйте полученные логические диски командой format (из них диск C: — как системный).

Если же на диске уже есть информация, но его разбиение на логические диски вас не устраивает, необходимо воспользоваться программой типа Partition Magic (www.powerquest.com).

Программа Image Sprayer Tool — это интересный инструмент в стандартной поставке PhotoPaint. Что же кроется за этим названием? За наше отставание в информатике нам приходится расплачиваться изучением терминологии не на своем, родном, а на английском языке. Image — это образ, термин Spray у нас уже вошел в обиход как спрей — некий дезодорант-распылитель, ну, а Tool — это инструмент. Вместе получается нечто вроде "распылителя образов".

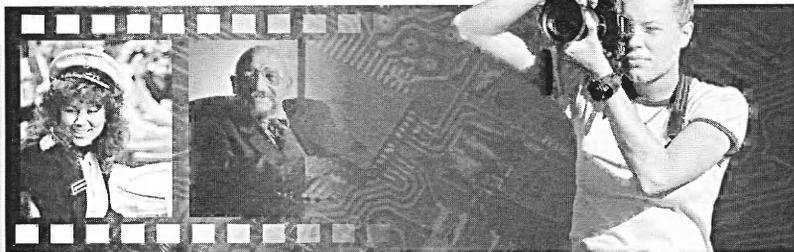
При активизации иконки Image Sprayer Tool (на ней изображены картина и баллончик) на экране возникает изображение кисточки. Особенность ее вот в чем. Художник, рисуя кистью, оставляет на холсте мазок, являющийся, так сказать, неделимой единицей. В Image Sprayer Tool виртуальное прикосновение кисти к экрану (щелчок левой кнопкой мышки) приводит к появлению некоторого изображения, которое тоже является неделимой единицей. Назовем его примитивом (в смысле неделимости, поскольку на самом деле изображение может быть достаточно сложным). Что же это дает?

В работе дизайнера и, особенно, веб-дизайнера постоянно возникает необходимость в разнообразных фонах. Чтобы не ломать голову, где найти нужное изображение и не блуждать в его поисках по Интернету, хорошо иметь под рукой все, что может понадобиться. Вот здесь-то и пригодится инструмент Image Sprayer.

Возможности

Это не просто емкий сундук, содержащий множество таких примитивов, как всевозможные шары, спортивные мячи, бабочки, листья деревьев и кустов, подсолнухи, облака и т.п. Открыв Tool Settings, вы увидите три диалоговые таблицы. В первой выводится список примитивов для текущего использования. Причем вы можете изменять способ вывода изображений на экран — размер примитива, степень его прозрачности.

Естественно, что оперативной

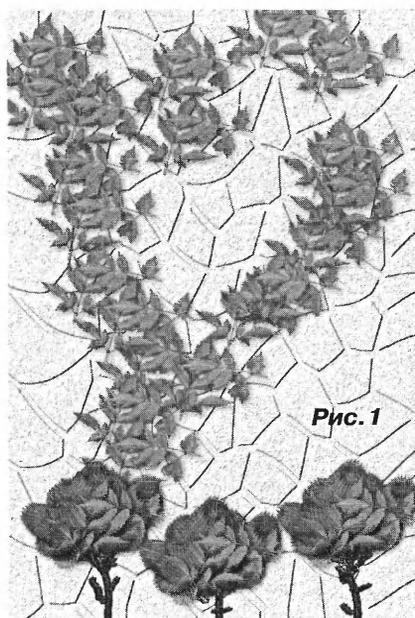


Виталий Шнейдеров

Image Sprayer Tool — кисть фотореализма

работы можно достичь лишь при небольшом объеме хранимых примитивов. На все случаи жизни примитивов заранее не напасешься. Поэтому в Image Sprayer предусмотрена возможность пополнения списка самим пользователем. При этом ограниченность списка примитивов компенсируется гибкостью системы, то есть возможностью "подгонки" примитива по месту.

Во второй таблице задается количество примитивов, которые выводятся на экран при каждом нажатии на кнопку мышки. Можно управлять расстоянием между примитива-



ми. Выбрав из списка примитивы, используемые в текущей работе, можно задать порядок их вывода, — либо случайный, либо детерминированный, последовательный.

Последняя таблица позволяет задавать орбиты примитивов, которые будут вращаться вокруг оси движения виртуальной кисти. Значение радиуса определяет величину этих орбит.

Рисовать с помощью Image Sprayer — истинное удовольствие. При легком прикосновении кисточки к экрану тут же вырисовывается готовый примитив или даже несколько примитивов сразу, а при движении кисти — череда сменяющих друг друга изображений. Изменяя яркость, прозрачность, положение объекта в пространстве, добавляя блики и тени, можно получать фон в виде трехмерных картин.

Режим одиночного вывода

Этот режим наиболее прост и напоминает технику "вырезать—вставить". Отличие в том, что здесь вырезать изображения уже не надо, так как вы имеете дело со списком готовых примитивов. Чтобы вставить примитив в рисунок, достаточно щелкнуть левой клавишей мышки.

Первое, с чего обычно начинают пользователи после знакомства с Image Sprayer, — это работа с бабочками. Если это портрет, то ба-

бочку можно посадить на любое место — на плечо, на руку, на голову. Это создает ощущение покоя и созерцательности.

Если двигать мышку, то следом за перемещением кисточки по экрану остается своего рода борозда, в которую "высеиваются" примитивы. Так, выбрав список Foliage (листва), можно "озеленить" любую каменную стену (рис. 1).

Чтобы получить "борозду" в виде прямой линии, нужно воспользоваться клавишами Ctrl или Alt. Таким путем удобно создавать оригинальные рамки для фотопортретов или для текста. Рамка из бабочек особенно хороша для женского портрета. Увы, в черно-белом варианте все краски потухли, и рамка из бабочек у меня стала похожей на невзрачную связку сушеных грибов. Поэтому для иллюстрации возможностей построения рамок в Image Sprayer я выбрал шуточный портрет девушки — часовых дел мастера (рис. 2). Если продолжить аналогии, то рамка из овощей подойдет к портрету огородника, из фруктов или листьев — садовода, из пчел — пчеловода и т.д.



Режим строки

Траекторию "борозды", в которой располагаются примитивы, можно не только вычерчивать самому, но и выбрать готовую. Тогда при одном щелчке левой клавишей мышки из списка выбранных примитивов будет построено кольцо, звезда, латин-

ская буква V, пятно случайной формы из примитивов (режим Blot) или простой столбец.

На рис. 3 приведен пример использования режима строки с каплями дождя (Rain Drops), а на рис. 4 — пример построения олимпийских колец (режим строки — кольцо).

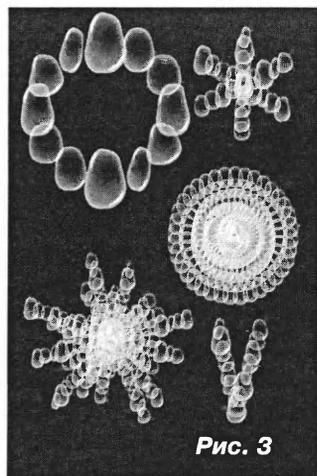


Рис. 3

Режим орбит

Совершенно новые возможности открывает режим орбит. В этом случае при движении кисточки по экрану примитивы совершают вращательное движение вокруг траектории движения кисточки. Так, выбрав в качестве примитивов шары небольшого размера и задав их количество, мы получим наглядную интерпретацию процедуры закручивания шнура или каната. Изменяя количество орбит примитивов, их радиус, скорость вращения примитива по орбите, можно навивать шнур, кабель, канат, спираль или сплошную трубу.

Рисуем толпу

Image Sprayer оказывается очень удобным для создания своего рода толпы, некоего безликого фона, состоящего из отдельных единиц множества, в котором индивидуальные черты отходят на второй план. Таким множеством птиц является стая, множеством деревьев — лес, травинки — газон или поле и т.п.

На фотографии мы видим группу из девяти девушек (рис. 5). С помощью маскирования можно выделить



Рис. 5



Рис. 4

девушек в виде девяти объектов, но мы упростим задачу и выделим два объекта, которые затем введем в Image Sprayer в виде примитивов. Первый объект — группа из трех девушек справа и второй объект — группа из шести девушек слева.

Естественно, чем больше примитивов, тем разнообразнее, естественнее будет толпа, но тем более трудоемкой будет подготовительная работа. Самый простой способ достичь квазиразнообразия — продублировать эти два объекта, а затем их зеркально отобразить, чтобы те, кто повернут влево, оказался повернут вправо и наоборот. В результате количество

объектов удвоится и станет равным четырем.

Введем эти объекты в список Image Sprayer в виде четырех примитивов и присвоим ему имя Crowd (толпа). Теперь начинается самое интересное. Двигая мышку, можно легко построить толпу или большую группу людей любой формы и любой степени плотности.

Поскольку в толпе передние ряды заслоняют собой задние, начинать рисование нужно с задних рядов. Чтобы выдержать перспективу, следует увеличивать размер примитивов при движении к передним рядам в зависимости от ракурса. Так, на рис. 6 глаз виртуального фотографа находится ниже, чем на рис. 7, поэтому здесь перспектива оказывается сильнее. На рис. 7 внимательный глаз выявит повторяющиеся персонажи в группе девушек, но как их уменьшить, вы уже знаете.

Таким путем можно создавать большие панорамные произведения — сводный хор, слет туристов, собрание акционеров, военный парад и многое другое.

Аналогичным образом можно строить деревья, лесные поляны, клумбы, газоны и т.п. Примеры применения Image Sprayer для создания фотореалистичных изображений природы можно найти в Интернете на сайте <http://www.unleash.com>.



Рис. 6



Рис. 7



Юрий Петелин

Почти настоящая гитара

В журнале открыта новая и постоянная (хочется надеяться) рубрика, которая называется "Музыкальный компьютер". Вести ее буду я, Юрий Петелин, — один из авторов серии книг "Компьютер и творчество".

Основой рубрики станет цикл статей "Уроки музыки на компьютере", рассчитанный прежде всего на начинающих компьютерных музыкантов. "Продвинутым" пользователям будут адресованы статьи, посвященные отдельным программам, наиболее интересным в своей функциональной нише. В перспективе мы надеемся привлечь к сотрудничеству с журналом участников проекта "Музыкальный компьютер".

В рамках этого проекта на радиостанции "Гардарика" в эфир вышли уже 40 выпусков одноименной передачи. В них впервые прозвучали ров-

но 100 интереснейших композиций. Проведены два конкурса, причем победители получили неплохие призы. Во Дворце культуры "Гавань" Васильевского острова проводит встречи и концерты клуб проекта "Музыкальный компьютер". А для того чтобы помочь не только питерцам, но и коллегам из других городов, создан сайт <http://musicalpc.chat.ru/>, на сегодня — один из наиболее посещаемых российских музыкальных сайтов. Участниками проекта "Музыкальный компьютер" уже стали многие увлеченные и талантливые люди. Вы, уважаемый читатель, если захотите, тоже можете оказаться в их числе.

В этом номере предлагаю познакомиться с двумя по-своему интересными программами. Одна из них — виртуальный гитарист, а другая — виртуальный DJ.

Думается, что все человеческое можно условно поделить на три группы: люди, умеющие играть на гитаре, мечтающие научиться и сожалеющие, что этого не удалось сделать. По собственному опыту знаю, что человеку с гитарой и живется веселее. Гитара привлекала, привлекает и будет привлекать. Так что, если не удается стать виртуозом и не получается "плести кружева", то можно попробовать научиться хотя бы аккомпанировать.

Освоить гитарный аккомпане-

мент вроде бы не так уж и трудно. И в самом деле, из чего складывается игра гитариста? Пальцами левой руки он берет на грифе аккорды, а правой рукой извлекает из струн звук, формируя тем самым ритм аккомпанемента и тембр.

Все просто, если ограничиться тремя аккордами и одним перебором. И все сложно, когда занимаешься гитарой всерьез. Судите сами. Аккорды гитарист считывает из партитуры, которая в случае аккомпанирующей гитары принимает вид аккордовой сетки — последовательности условных обозначений

аккордов. Поэтому человек, претендующий на роль гитариста, во-первых, должен уметь превращать условное обозначение аккорда в таблатуру (схему расположения пальцев на грифе). Во-вторых, он должен владеть различными приемами извлечения звука: удар сверху вниз, снизу вверх, приглушенный удар, арпеджио и т.д. Кроме того, если говорить о мастере, то он обязан уметь использовать все лучшее из того, что заложено в гитаре: неисчерпаемое множество возможных гармонических ходов и выразительность звучания. Это значит, что он должен свободно брать не только несколько основных аккордов, но также все их расширения и обращения, уметь акцентировать звуки, выполнять множество приемов игры и гитарных штрихов.

Все эти проблемы приходится преодолевать, если вы взяли в руки настоящую гитару, сделанную из дерева и стали. А музыканту, пишущему гитарную партию с помощью MIDI-редактора, работать еще труднее. Для того чтобы имитировать характерные для гитары приемы извлечения звука, потребуются долгие часы попутного редактирования MIDI-сообщений нескольких типов, кропотливой и ювелирной коррекции их временных параметров.

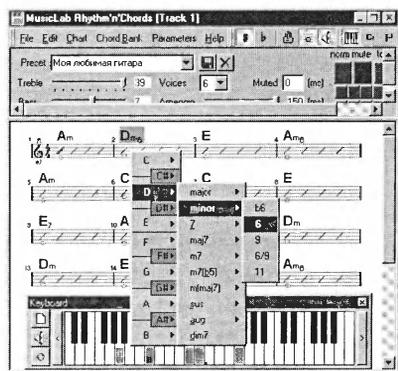
В общем, в сфере программного редактирования гитарных MIDI-партий давно назрела необходимость в достижении реалистичного звучания при относительно невысоких трудозатратах. И вот теперь эта проблема, похоже, решена. Имя этому решению — Rhythm'n'Chords MIDI FX Plugin for CakewalkR, плагин для Cakewalk Pro Audio, созданный фирмой MusicLab Inc (www.musiclab.com). При помощи этой программы любой человек, даже не владеющий техникой игры на реальной гитаре, сможет получить желаемый результат — довольно естественно звучащий гитарный аккомпанемент.

Rhythm'n'Chords

Основной принцип, лежащий в основе работы с этой программой, — раздельное представление ритма

и гармонии, которое, кстати говоря, часто применяется при написании партии ритм-гитары как наиболее удобное для чтения и исполнения. Гармония записывается аккордовыми символами, а ритм — одногласной ритмической линией. Правда, в этом случае реальному гитаристу придется самому строить каждый аккорд: выбирать мелодическую позицию, интервальный состав, количество голосов.

При работе с программой аккорды записываются в партитурном окне Chord Chart стандартными аккордовыми символами. Пользуясь системой меню, вы можете выбирать любой из мыслимых аккордов. Программа автоматически построит гитарные аккорды и свяжет их в аккордовую последовательность, подбирая ближайшие позиции (обращения).



Аккорды в партитурном окне Chord Chart

Ритм пишется в треке музыкального редактора Cakewalk Pro Audio, к которому подключен плагин, в окне Piano Roll стандартными сообщениями типа Note (в виде отпечатков клавиш). Специальная таблица ударов (Stroke Map) преобразует эти сообщения в удары, имитирующие различные приемы игры на гитаре (названия ударов расположены в левой части окна). Так формируется ритм-трек.

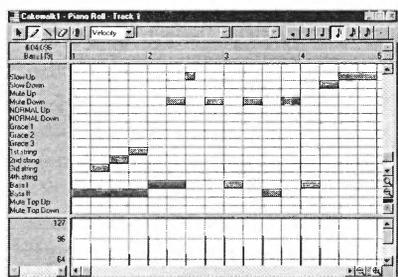
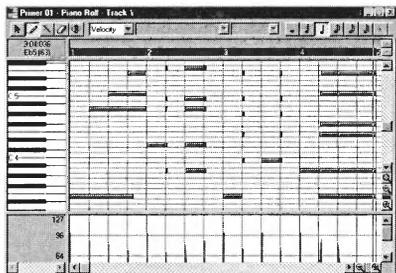


Таблица ударов и ритм-трек

Результатом работы плагина является готовая аккордовая партия — либо звучащая в реальном времени в режиме Play секвенсора, либо преобразованная командой Apply MIDI Effects прямо в трек Cakewalk Pro Audio.



Аккорды партии, созданной с помощью плагина

Принцип отдельного представления ритма и гармонии позволяет создавать аккордовые партии быстро и качественно. Назначив гармоническую схему композиции по тактам и размножив в ритм-треке подходящий ритмический рисунок, можно мгновенно получить, например, партию ритм-гитары любой гармонической и ритмической сложности и протяженности.

В программе предусмотрена масса средств, способствующих созданию таких гитарных MIDI-партий, которые трудно отличить от исполненных на настоящей гитаре. Вы можете выбирать обращения аккордов, создавать свои собственные аккорды, задавать силу удара по каждой из струн, скорость исполнения арпеджио, количество звучащих струн и многое другое.

И еще об одной интересной особенности программы. Установив плагин и подключив MIDI-клавиатуру, можно играть на ней в реальном времени, почти как на настоящей гитаре: левой рукой выбирать аккорд, а правой, нажатием на единственную клавишу, исполнять один из предусмотренных ударов (или их комбинацию).

Looper

Начинающему компьютерному гитаристу может оказаться полезной еще одна разработка фирмы MusicLab Inc. Это программа Looper, которая позволяет применять библиотеку ритмов, входящую в поставку плагина Rhythm'n'Chords, и со-

здавать свои ритмические паттерны (шаблоны).

Щелкните мышью в поле Effects окна Track программы Cakewalk Pro Audio, затем двойным щелчком на названии Looper запустите и этот плагин. Откроется окно MusicLab Looper. Расположите окна обоих плагинов и окно Piano Roll так, чтобы с ними было удобно работать.

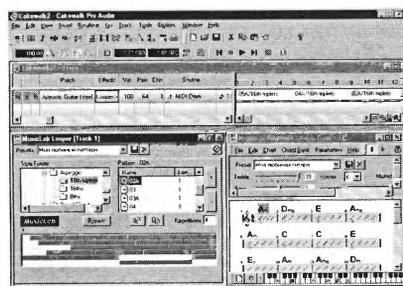
Таким образом, открыты два плагина, подключенные к одному треку. В самом треке нот нет — он пока пуст. После того как вы выберете стиль аккомпанемента и включите секвенсор на воспроизведение, Looper будет передавать ритмические паттерны, находящиеся в нем, в плагин Rhythm'n'Chords. Тот, в свою очередь, в реальном времени превратит паттерны в аккордовую гитарную партию в соответствии с записанной в партитурном окне Chord Chart последовательностью аккордов.

Для быстрого и удобного поиска, выбора и прослушивания паттернов в Looper использована файловая система. При запуске плагина в левой его части в секции Style Folder открывается корневая папка, содержащая набор стилевых папок с названиями музыкальных стилей и указанием рекомендованного темпа. В стилевых папках содержатся наборы ритмических паттернов, характерных для каждого стиля. Открыв стилевую папку, в правой секции (Pattern) окна плагина вы увидите список паттернов, находящихся в этой папке. Щелчком мыши выберите любой паттерн из списка, при этом в нижней секции окна появится графическое отображение его содержания — пропорциональное изображение отпечатков нот и контроллеров. Если включить Cakewalk Pro Audio в режим Play, то вы услышите исполнение аккордов, установленных в партитурном окне Chord Chart, выбранным вами ритмическим паттерном. При этом паттерн будет воспроизводиться циклически по ходу композиции, до остановки секвенсора.

Теперь можно составить из коротких паттернов ритмическую структуру для партии гитары в треке Cakewalk Pro Audio. В режиме Play секвенсора, слушая композицию, выберите под-

ходящий паттерн в окне MusicLab Looper (например, 05A из папки Pop(RnC)/Pop Ballad (T-90)/Arpeggio/16th triplets). Остановите секвенсор. Задайте необходимое количество повторений (например, 4) в поле ввода Repetitions, находящемся в правом нижнем углу окна MusicLab Looper. Теперь перетащите мышью паттерн в нужное место гитарного трека (в окне Piano Roll). Для переноса паттерна в трек можно использовать также команды Copy и Paste.

Размноженный паттерн установится в трек в виде четырехтактового клипа с указанием в названии исходного паттерна и его стилиевой папки (05/16th triplets). Таким же образом подберите паттерны для других частей композиции и перенесите их в трек.



В треке SAKEWALK Pro Audio размещена последовательность паттернов

Если захотите прослушать созданную ритмическую структуру, отключите Looper, нажав кнопку Bypass. В этом случае в Rhythm'n'Chords будут поступать MIDI-сообщения с трека SAKEWALK Pro Audio, то есть вы будете слышать гитарную партию, исполняемую ритм-треком, составленным из паттернов. Если звучание вас устраивает, можно импортировать партию в трек SAKEWALK Pro Audio командой Apply MIDI Effects и убрать плагины с трека. Партия ритм-гитары готова.

Вот и завершен краткий рассказ о двух плагинах фирмы MusicLab Inc. к SAKEWALK Pro Audio. Освоив их, вы станете настоящим виртуозом гитарного MIDI-аккомпанемента, а гитарные партии в ваших композициях зазвучат естественно и неповторимо. Подробности вы можете найти в книге: Петелин Р.Ю., Петелин Ю.В. SAKEWALK. "Примочки" и плагины.



Роман Петелин

Виртуальное диджейство

Кто такие DJ?

Тема этой статьи — виртуальное диджейство. Но перед уходом в виртуальную реальность давайте разберемся, кто же они такие — эти самые DJ. В оксфордском толковом словаре говорится, что DJ, он же диск-жокей — это человек, презентующий популярную музыку. Доисторический DJ выглядел примерно так: на дискотеке выходил этакий DJ с характерной для своего времени прической и говорил: "А сейчас вы услышите совершенно новый хит такой-то группы", — и ставил пластинку. С тех пор прошло много времени, и понятие стало очень расплывчатым. Сейчас на любом радио есть люди, именующие себя диджеями. Однако эти девушки с зовущими голосами и профессионально жизнерадостные парни, презентующие современную музыку в промежутках между рекламными роликами, не имеют ничего общего с истинными DJ.

Основной атрибут DJ — коллекция дисков (виниловых или лазерных). Однако на современных радиостанциях диски используются все реже. Вместо них теперь серверы, хранящие в своих базах данных звуковые файлы и воспроизводящие их в соответствии play-листом, подготовкой которого занимаются другие сотрудники радиостанции или даже компьютерная программа. Та-

ким DJ не остается больше никакой работы, кроме как объявлять названия песен, высвечиваемых на мониторе, и развлекать слушателей своими монологами или разговорами по телефону. Правильнее этих людей назвать радио-жокеями (RJ), а ведущих музыкальных телепрограмм — видео-жокеями (VJ).

Сейчас же речь пойдет о других, настоящих современных DJ.

Теперь DJ не презентует популярную музыку, он создает ее сам, причем обычно без помощи музыкальных инструментов. Вместо них у него два (а иногда и больше) проигрывателя виниловых дисков, называемых "вертушками". От обычных проигрывателей они отличаются повышенной надежностью, большим вращающим моментом (чтобы диски раскручивались быстрее) и широкими возможностями в управлении скоростью вращения. Постепенно на смену проигрывателям виниловых дисков приходят специализированные DJ's CD-плееры. От обычных проигрывателей CD они отличаются опять-таки надежностью и возможностью регулировки темпа. Главный атрибут такого CD-плеера — манипулятор типа JOG, внешне представляющий собой свободно вращающийся диск. С помощью этого манипулятора DJ может "подкрутить" или "притормозить" компакт-диск вручную так, как

это он сделал бы с виниловыми дисками. Современные DJ's CD-плееры поддерживают и множество дополнительных функций, в том числе — сэмплирование и воспроизведение лупов (о них мы еще поговорим).



Диджейский CD-плеер
Pioneer CDJ-500S

Еще один обязательный для DJ атрибут — микшер. Это устройство, в котором сигналы смешиваются в определенных пропорциях, задаваемых положением регуляторов. В микшер могут быть встроены эквалайзеры и другие устройства обработки звука. Современные DJ's микшеры могут совмещать в себе функции процессора эффектов и сэмплера.



Диджейский микшер Pioneer DJM-600

Классической считается следующая схема: микшер располагается между двумя плеерами. Кроме микшера и вертушек у любого DJ имеются наушники и, конечно же, набор отобранных для выступления пластинок. Кстати, все это вместе стоит довольно дорого.

Итак, DJ ставит одну из своих пластинок. Вполне возможно, что эту пластинку DJ записал сам, и не

без помощи компьютера и синтезатора, но чаще всего пластинка приобретается в составе DJ's наборов. На таких пластинках могут быть записаны какие-то современные и не очень ритмы, грувы, специальные звуки или повторяющиеся фразы наподобие "Come on! Let's do it! Check the sound".

Пластинка эта начинает звучать, а народ — танцевать. Тогда DJ берет другую пластинку и запускает вторую вертушку. Танцующие люди ее пока не слышат, ее слышит только сам DJ. Для этого-то ему и нужны наушники. Различными манипуляциями DJ добивается, чтобы темп второй пластинки совпадал с темпом первой. Кстати, на современных вертушках и диджейских микшерах обычно присутствуют счетчики, отображающие текущее значение темпа (единица измерения темпа BPM — количество музыкальных долей в минуту).

Если на обеих пластинках присутствуют мелодии, то задача DJ усложняется — пластинки должны быть подобраны так, чтобы не было дисгармонии. Допустим, микс (суммарный сигнал, поступающий от обеих вертушек) DJ нравится. Тогда он выводит его на публику, плавно добавляя звучание второй пластинки. Обе пластинки воспроизводятся одновременно до тех пор, пока это не надоест DJ. Тогда он плавно выводит первую пластинку из микса, постепенно понижая громкость ее звучания. Пока играет вторая пластинка, DJ ставит вместо первой новую, и вся процедура повторяется, при этом танцевальный марафон не прерывается ни на минуту. Музыка звучит в режиме non stop.

Конечно, опытный DJ мог бы поведать вам о своей работе гораздо больше, но вряд ли он станет выдавать свои профессиональные тайны.

Программа **Groove Maker 2.0**

Итак, мы с вами знаем, что DJ микширует звук с разных, заранее отобранных пластинок, и следит, чтобы не было нарушения синхро-

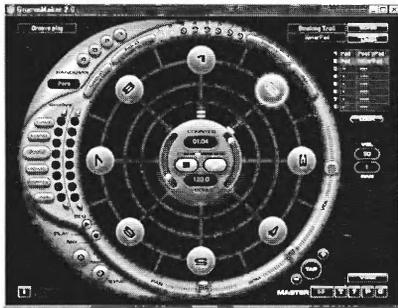
низации ритмов. Хотя с этой работой вполне может справиться и компьютер, без человека здесь никак не обойтись, ведь удовольствие от своей игры должен получать именно он, а не компьютер. И в этом ему помогут специальные программы, иногда называемые виртуальными DJ. Об одной из таких программ (итальянской фирмы IK Multimedia) — **Groove Maker 2.0**, название которой можно перевести как "изготовитель грувов", я и хочу рассказать.

В основе этой и большинства других подобных программ лежит использование сэмплерных лупов. По-английски loop означает "петля". В звукозаписи под лупом понимается фрагмент фонограммы, который повторяется в цикле. Причем нет никакой границы между концом предыдущего и началом последующего повтора. Длительность лупа



DJ Groove за работой

обычно соответствует одному или нескольким музыкальным тактам. Лупы могут содержать партии барабанов, какие-то музыкальные фразы, грувы, перкуссию, специальные эффекты и, в общем-то, любые звуки, которые имеет смысл воспроизводить в цикле. Сами по себе лупы звучат несколько монотонно, однако не настолько монотонно, чем пластинки из диджейского набора. Несомненное преимущество лупов в том, что их можно комбинировать: воспроизводить одновременно несколько лупов, постепенно заменяя некоторые из них. Если в распоряжении традиционного диджея имеется всего две (в редких случаях три) вертушки, то у виртуального диджея, вооруженного **Groove**



Groove Maker 2.0

Maker, — целых восемь одновременно звучащих треков.

Для начала нажмите кнопку Song, затем выберите на CD-ROM Groove Maker (кнопка так и называется — CD) какую-нибудь библиотеку сэмплов и перегрузите ее на винчестер (кнопка COPY TO DISK). Кстати, для Groove Maker создано несколько десятков дисков с библиотеками лупов. Затем, чтобы вернуться в локальный каталог Groove Maker (папку, где находятся загруженные библиотеки) нужно нажать кнопку HOME, еще раз выбрать библиотеку сэмплов (теперь уже окончательно) и нажать кнопку LOAD.

В центре окна две кнопки: Stop и Play. Запускаете воспроизведение. Каждому из треков соответствует своя большая круглая кнопка с его номером. Нажимаете ее — трек становится активным. С помощью рычажков в центре окна или меню в правом верхнем меню выберите тип лупа (Bass — басы; BD — басовые барабаны; FX — спецэффекты; Line — основная мелодическая линия; Loop — ритмические лупы; Pad — подкладки; Perc — перкуссия), а затем и сам луп. Зазвучало? Теперь можно попробовать изменить панораму (PAN) и громкость (VOL). Слишком медленно? Добавьте темпа движком BPM.

Как только вы добьетесь интересного звучания, нажмите кнопку MARK, и ваш микс сохранится в виде пронумерованного кружка на линии GROOVE (в левой части окна). Когда таких кружков будет много, из них можно составить последовательность на линии SEQ. Эту секвенцию можно воспроизвести (PLAY) или сохранить на диске (MIX) в виде WAV-файла. Позже в любом

многодорожечном звуковом редакторе на этот трек можно будет наложить вокал, выполнить мастеринг и, наконец, записать собственный диск.

Есть и еще несколько полезных функций.

Если активизирована кнопка SYNC, то начало воспроизведения новых лупов будет приходиться только на начало очередного такта, а не на тот момент, когда вы его выбрали (полезно для работы на публике в реальном времени).

Кнопкой ARP вызывается арпеджиратор — средство для превращения аккорда в последовательность звуков.

Можно сказать, что в вашем распоряжении имеется некий виртуальный синтезатор. Обозначьте мышью аккорд на его клавиатуре, выберите тембр (SYNTH), тип арпеджо (Up — вверх, Dn — вниз, Rnd — случайным образом, U/D — вверх/вниз), количество шагов и номер программы, реализующей арпеджо (PROG). Результатом будет ваш собственный грув. Понравилось? Сохраните на диске (MIX), чтобы потом можно было загрузить вновь (OPEN). Напомню: MARK — занести грув в секвенсор, PAN — панорама, VOL — громкость. Не понравилось — арпеджиратор закрывается повторным нажатием ARP.

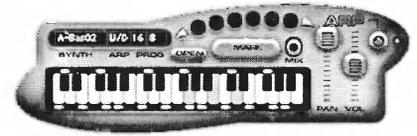
Автопилот

Диджеи — тоже люди. Если вдруг на вечеринке срочно захотите отлучиться, а ваши друзья требуют продолжения танцев и отпускать вас не собираются, нажмите кнопку V-MIX.

V-MIX представляет собой авто-



Автопилот V-MIX



Арпеджиратор

пилот, который будет микшировать лупы без вашей помощи. Теперь уже в окне V-MIX нажмите кнопку V-MIX. Ну вот, наконец-то можно улизнуть. Вашего отсутствия никто не заметит. Возвратившись на свое место, отключите автопилот кнопкой GROOVE.

В Groove Maker есть еще и функция импортирования лупов из WAV-файлов (инструмент LOOPMAKER, который вызывается кнопкой IMPORT LOOP). Но это уже для очень далеко продвинутых пользователей Groove Maker.



Инструмент для импорта лупов из WAV-файлов

Вот, в принципе, и все, что я хотел рассказать о виртуальном диджействе и программе Groove Maker. Конечно, подобные программы похожи скорее на игрушку, но Groove Maker — игрушка достаточно серьезная, для детей старшего школьного возраста (идеальный выбор для школьной дискотеки) и даже, в принципе, для более взрослых и серьезных людей.

Полезные ссылки:

<http://www.djsound.ru/> — если хотите стать настоящим DJ, то вам сюда;

<http://www.groovemaker.com/> — сайт, посвященный исключительно Groove Maker;

<http://www.ikmultimedia.com/> — сайт фирмы IK Multimedia;

<http://musicalpc.chat.ru> — наша страничка, Юрия и Романа Петелиных.



Игорь
Ананченко

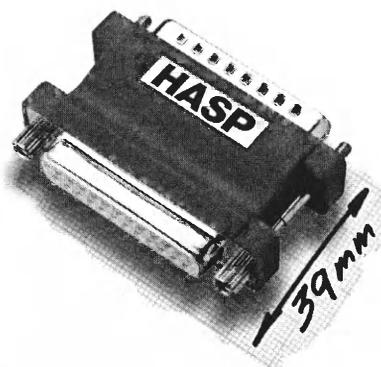
HASP, HARDLOCK и другие. Что новенького?

Борьба разработчиков ПО с компьютерными пиратами ведется непрерывно. Чисто технические способы противодействия, особенно в нашей стране, зачастую оказываются намного более эффективными, чем законодательные. Поэтому вполне понятен повышенный интерес разработчиков ПО к высокоэффективным средствам защиты, построенным с использованием электронных ключей. Однако использование заказных микросхем, применяемых для изготовления электронных ключей на аппаратной базе, делает такую систему защиты более дорогой по сравнению с менее стойкой чисто программной. Ее использование становится экономически целесообразным для относительно дорогих программ, от \$100. Очевидно, что чем сложнее технические средства и изощренней криптоалгоритмы, тем дороже система защиты, а значит, и защищенная ею программа, предлагаемая конечному пользователю. Это обязательно надо учитывать при рассмотрении предлагаемых разработчиками решений. Не стоит говорить о том, что одна система защиты хуже другой, сравнивая системы, ориентированные на защиту ПО разных ценовых групп. Критерием выбора защиты становится максимальная

эффективность и минимальная цена в пределах группы. Это понимают и создатели защит, предлагая свои системы, ориентированные на различных потребителей.

Соперничество разработчиков хитроумных защит с не менее изобретательными хакерами, применяющими для взлома чужих программ и раскрытия защищенной информации самые современные технические средства, подобно извечному соперничеству создателей брони и снарядов. На новое техническое решение одной стороны противник достаточно быстро находит адекватный

здесь такого привлекательного для многих разбора новых защит с позиции специализирующихся на их взломе хакеров. И дело не в том, что найдено нечто принципиально новое и недоступное зубам наших самых хакеристских хакеров. Просто новое решение требует новых подходов, а на это нужно время. Конечно, хотелось бы пожелать создателям защит, чтобы оно было максимально большим, а их разработки максимально стойкими, но опыт показывает, что не следует быть большим оптимистом в вопросах надежности защит. Поэтому, читая о надежных средствах и элегантных решениях, помните, что все это останется таковым до тех пор, пока... Пока не будет сломано!



ответ, и очередной оборот спирали фактически возвращает соперников на исходные позиции.

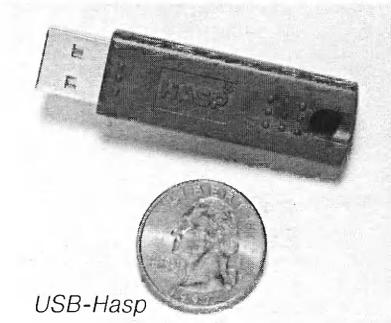
Эта статья содержит рассказ о новинках, предлагаемых разработчиками систем защит. Вы не найдете

Новый Хасп лучше старого

Идея использовать аппаратный ключ для защиты информации на персональном компьютере не нова и достаточно давно реализована разработчиками ПО. Данная тема в журнале уже рассматривалась ("Магия ПК" №1/1999, с.38—41), поэтому подробно останавливаться на общих принципах организации программно-аппаратной защиты я не буду.

Средства защиты, реализованные с использованием аппаратных электронных ключей, выпускаются несколькими фирмами, например,

NOVEX Software. Ее дилером в Санкт-Петербурге и Северо-Западном регионе является ООО "Джет Технолоджи" (<http://novex.jettech.ru>).



USB-Hasp

Однако наиболее известны пользователям, например, по продуктам от 1С, защиты, выполненные с использованием электронных ключей типа HASP. С такими ключами работает ПО от ALADDIN Software Security R.D. (<http://www.aladdin.ru>). Отмечу, что мое достаточно скептическое отношение к защита, выполненным с использованием HASP, изменилось в лучшую сторону после посещения выставки "Инвекон-2000", когда я на стенде ознакомился с информацией о ключах нового поколения HASP4. Вообще говоря, защиты, основанные на использовании HASP, сильно критиковали, и не без основания, за слишком простой драйвер ключа, взломать который не составляло особого труда для хакера средней квалификации, да и техническая реализация самого ключа оставляла желать лучшего. Поэтому было приятно узнать, что появилось не только нечто новое и более совершенное среди защит, построенных на основе HASP, но и возможность использовать для защиты дорогостоящих программ и особо важной информации более совершенные ключи класса HARDLOCK.

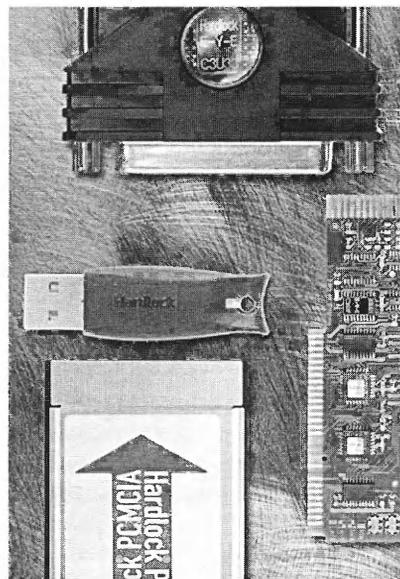
Знакомство с новинками начнем с HASP4. В этом ключе воплощен 15-летний опыт разработчиков в области защиты, микроэлектроники и криптографии. Встроенный криптопроцессор позволяет "на лету" шифровать код программы или данные, в том числе и передаваемые по сети. Ключ шифрования никогда не покидает ASIC-чип (Application Specific Integrated Circuit), поэтому его

нельзя перехватить. Кодированный обмен между всеми компонентами системы (ключ, драйвер, API) позволяет надежно защитить систему от эмуляторов. Ключи HASP4 могут использоваться для защиты программ, баз данных, электронных справочников, цифровых карт, веб-страниц и пр. При этом защищаемая программа "привязывается" к ключу, а данные надежно шифруются.

Ключи HASP теперь выпускаются в различном исполнении и работают практически во всех средах, платформах и сетях, на любых типах компьютеров: PC, Apple Macintosh, iMac, Quadra, PowerMac, NEC, Fujitsu и др.

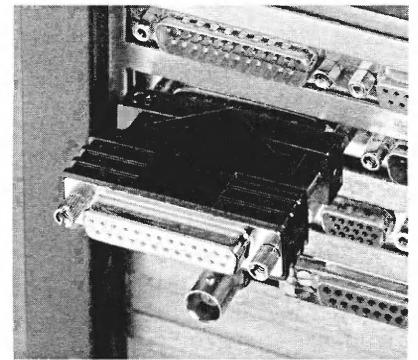
Отмечу, что HASP4 — общее название целой серии ключей, из которой разработчик ПО может выбрать ключ, наиболее подходящий для решения его задач. Например, ключи HASP4 Time содержат встроенные часы с автономным питанием и 512 байт памяти. Использование ключей этого типа позволяет организовать продажу программ в рассрочку (лизинг), сдачу в аренду, ограничение по времени использования и т.д.

Для уникальных разработок рекомендуется использовать более серьезную защиту, базирующуюся на электронных ключах Hardlock, воспроизвести которые с использованием существующих технических средств практически невозможно. Ключи сложны в изготовлении, по-



Варианты ключей Hardlock

этому и стоимость защиты на их основе значительно выше. Сердцем ключей Hardlock является заказной ASIC-чип с встроенной EEPROM-памятью. Чип имеет сложную внутреннюю организацию и нетривиальные алгоритмы работы. Логику работы чипа практически невозможно реализовать с помощью стандартных наборов микросхем PAL, GAL или PEEL, его практически невозможно воспроизвести, а содержащийся в его памяти микрокод — считать, расшифровать или смодулировать. Чип программируется только с ис-



Hardlock, подсоединенный к порту

пользованием специальной платы Crypto Programmer Card. Данные можно шифровать блоками по 64 бит. Напряжение питания ASIC-чипа — всего 1.5 В, поэтому он устойчиво работает на всех портах в любых режимах, а малый ток потребления позволяет каскадировать практически любое количество ключей.

Появились также новинки, ориентированные не на разработчиков ПО, а на рядовых пользователей компьютеров, заинтересованных в защите конфиденциальной информации. Например, система Secret Disk для защиты конфиденциальной информации пользователя и Secret Disk Server для защиты корпоративных баз данных.

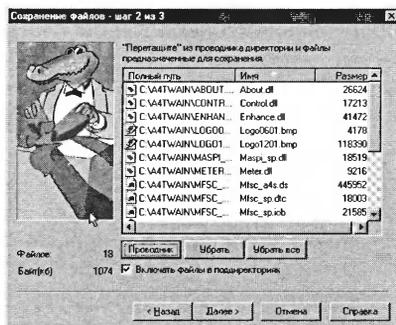
Защита в локальной сети и Интернете

Многие пользователи уверены, что их корпоративные базы данных надежно защищены после установки парольной защиты. Но пароль лишь ограничивает доступ, а для того чтобы открыть информацию, достаточ-

но просто переместить диск на другой компьютер. В случае использования парольной защиты человеческого фактор оказывается критическим. Пароли могут быть утеряны или система парольной защиты сконфигурирована с ошибками.

Для защиты информации в сети можно воспользоваться готовой разработкой, такой как Secret Disk Server. Система автоматически зашифровывает данные при записи на сервер и расшифровывает при чтении. Ключ шифрования вводится при загрузке сервера со смарткарты, защищенной PIN-кодом. После загрузки для дальнейшей работы смарткарта не требуется, и ее можно убрать в надежное место. Дальнейшая работа с сервером ничем не отличается от традиционной, однако после перезагрузки сервера без предъявления смарткарты или попытки чтения дисков на другом компьютере защищенные разделы будут "видны" как неформатированные области, прочитать которые нельзя.

Как правило, сервер организа-



Интерфейс Secret Disk

ции работает круглосуточно, оставаясь в ночное время без присмотра обслуживающего персонала. Выбрав такой момент, в закрытое помещение могут проникнуть злоумышленники. При возникновении опасности Secret Disk Server мгновенно "уничтожит" информацию, сделав защищенные разделы "невидимыми". Сигнал "тревога" для блокирования информации может быть подан серверу разными способами: с клавиатуры любой станции сети, от "красной кнопки" (спрятанной под столом), от охранной сигнализации (датчиков открывания дверей, окон, движения и пр.), от кодового замка

(при входе в помещение под принуждением). Очевидно, что в этом случае, получив физический доступ к серверу, злоумышленники не смогут завладеть защищенной информацией. Процедура восстановления доступа к информации предельно проста — достаточно перезагрузить сервер со смарткартой. Помимо встроенного алгоритма преобразования данных с длиной ключа 128 бит к системе можно подключать внешние модули криптографической защиты, например, входящий в Windows RC-4 или эмулятор известной платы "Криптон", реализующей мощнейший российский алгоритм шифрования ГОСТ 28147-89 с длиной ключа 256 бит.

Систему защиты не только информации, но и работающего в сети ПО, можно создать, воспользовавшись специальным комплектом разработчика. В частности, при работе в Сети приходится защищать информацию на web-и ftp-серверах. Для защиты веб-ресурсов можно использовать шифрование канала передачи данных, реализованное на основе электронных ключей серии Hardlock. Защита методом "прозрачного" преобразования позволяет одновременно обеспечить персонализацию доступа к охраняемой информации. Модифицированный файл, закодированный владельцем, помещается на сервер, после чего открыть его может только обладатель электронного ключа Hardlock и специального модуля расширения, сопряженного с браузером. Модуль является стандартным для всех вариантов ключей. Если требуется исключительно высокая степень защищенности, электронными ключами могут быть снабжены и ПО сервера, и приложение-клиент, установленное на компьютере пользователя.

Новый ключ eToken R2

Это полнофункциональный аналог смарткарты, выполненный в виде брелока, напрямую подключаемого к компьютеру через USB-порт. Работа с ним не требует наличия карт-ридеров и других дополнительных устройств. Созданный ключ может ис-

пользоваться для защиты электронной почты (цифровая подпись и шифрование), систем клиент—банк, электронной торговли, для защиты сетей VPN (Virtual Private Network, виртуальная частная сеть).



eToken со смарткартой

eToken R2 позволяет использовать до 64 Кб энергонезависимой памяти и аппаратно реализованный алгоритм шифрования DES-X со 120-битным ключом. Структуры данных в eToken либо повторяют файловую систему смарткарт (ISO 7816-3,4), либо строятся как объектно-ориентированная среда в соответствии с PKCS#11. Архитектура ключа дает три уровня полномочий доступа: гость (ограниченный доступ), пользователь (доступ к секретным данным с PIN-кодом) и администратор (конфигурирование и персонализация для пользователя с PIN'ом администратора). Протокол обмена информацией eToken с компьютером шифруется, а используемый для этого цифровой ключ никогда не покидает чип, поэтому его нельзя перехватить, даже физически вскрыв eToken. Злоумышленник использовать вместо одного устройства другое из этой же серии также не сможет, поскольку каждый eToken имеет 64-битный уникальный номер, доступный только для чтения.

Беглое знакомство с новинками завершено, и, я надеюсь, оно не показалось вам скучным. При подготовке статьи использованы технические описания и фотографии, представленные на веб-сайте <http://www.aladdin.ru/>.



Валентин Холмогоров

Поиск информации в Интернете

Информация — это еще одна форма материи. Если бы из информации можно было шить мужские сорочки, через несколько лет человечество обязательно придумало бы для нее отбеливатель...

Если вы полагаете, что для поиска нужной информации вам требуется только подключенный ко Всемирной Сети персональный компьютер и полчаса времени, это означает, что вы никогда не пытались разыскать там нужные вам позарез данные. На практике обычно оказывается, что найти в этой глобальной информационной системе то, из чего она, собственно, и состоит — информацию, — задача отнюдь не простая. То, что Интернет представляет собой гигантское хранилище, содержащее сотни тысяч гигабайт разнообразных электронных документов, только усложняет дело. Действительно, при обилии документов, так или иначе касающихся интересующего вас вопроса, поиск конкретной информации зачастую напоминает попытку найти иголку в стоге сена.

Подбор ключевых слов

Прежде чем обращаться к ресурсам поисковой машины в надежде найти ту или иную опубликованную в Интернете информацию, необходимо четко определить, что именно вы собираетесь искать. Для этого су-

ществует достаточно простой алгоритм, который позволит вам сэкономить массу времени и организовать работу с поисковой службой более эффективно.

В первую очередь постарайтесь дать краткую, но четкую и исчерпывающую характеристику предмету ваших поисков одним-двумя словами. Определив предмет своих поисков, попытайтесь придумать для него три—пять других, менее точных, но все же соответствующих его сути определений.

Далее необходимо придумать для каждого из записанных вами слов хотя бы один синоним. Вы будете крайне удивлены, узнав, сколько схожих понятий может иметь одно, казалось бы совершенно обычное и хорошо знакомое вам слово. В подборе синонимов вам может помочь текстовый редактор MS Word: для этого необходимо набрать в окне редактора исходное слово, установить на него курсор и, щелкнув на нем правой кнопкой мыши, выбрать в появившемся меню пункт "синонимы".

Получившийся таким образом список и является перечнем ключевых слов, по которым следует вести поиск. Данный список может ока-

заться внушительным, однако бояться этого не следует: чем больше информации вам удастся найти, тем проще впоследствии будет выбрать именно то, что вам необходимо.

Наибольший интерес будут представлять для вас ссылки, содержащиеся в файлах отчета, сгенерированных сервером в результате поиска по разным ключевым словам.

Национальные особенности

Достаточно нетривиальным является вопрос, к ресурсам каких поисковых систем следует обращаться при поиске информации на различных языках. На первый взгляд, использование "общеинтернетовских" поисковых машин выглядит более привлекательным, поскольку они поддерживают множество национальных языков и индексируют документы во всем пространстве Всемирной Сети. Тем не менее, я рекомендую искать данные с применением "национальных" служб, то есть русскоязычные документы — в российских поисковых системах, англоязычные — в системах, преимущественно работающих с английским языком. Во-первых, "национальные" поисковые машины гораздо лучше оптимизированы для работы с языком, на котором говорят ее создатели, они умеют более или менее правильно оценивать синтаксис и семантику только "приоритетного" языка. Во-вторых, разработчики веб-сайтов в первую очередь стараются зарегистрировать свои ресурсы в национальных поисковых системах и каталогах, а уж потом — в международных. Практика показывает, что, например, российская поисковая система "Яндекс" по любому запросу находит в два с лишним раза больше документов на русском языке, чем международный сервер AltaVista. С английским же языком ситуация, как правило, обратная.

Если вы обнаружили сайт, содержащий интересующую вас информацию, советую поискать на нем раздел "Ссылки". Вполне возможно, что там вы найдете адреса сайтов, до которых еще "не добрались" поисковые машины.

Язык поисковых машин

Одним из самых эффективных способов повысить результативность поиска информации в поисковых системах является грамотное использование так называемого языка запросов.

Язык запросов — это специальный набор служебных команд, позволяющих максимально конкретизировать ваш запрос поисковой системе путем использования в нем простейших логических функций объединения или исключения слов. Набор команд, составляющих язык запросов, в разных поисковых машинах может несколько различаться, поэтому ниже приведена некая "усредненная" информация о командах, поддерживаемых практически всеми современными поисковыми машинами.

Одна из самых распространенных команд для большинства поисковых систем — поиск по маске. Она обозначается символами "*" или "?". Знак "?" позволяет заменить в ключевом слове запроса один символ, на место которого может быть подставлена любая другая буква, а знак "*" — последовательность символов. Если, например, вы забыли, как пишется слово "математика", можно не рыться в орфографическом словаре. Вполне достаточно указать в тексте запроса символ "?" вместо "спорной" буквы, допустим, так:

высшая мат?мат?ка

Поисковый сервер сам расставит недостающие буквы "и" и "е" по своим местам. Если же вам необходимо найти документы, содержащие всевозможные производные от слова "контакт", можно использовать вместо его окончания знак "**":

контакт*

Тогда в файл отчета попадут слова "контакт", "контактный", "контактировать", "контактор" и т.д., то есть все словоформы, содержащие корень "контакт" и произвольную последовательность символов, автоматически подставляемую вместо знака "**".

Команды "+" и "-" позволяют принудительно добавлять или исключать какие-либо слова из текста запроса. Слово, помеченное знаком "+", должно обязательно присутство-

вать в искомом документе, а помеченное символом "-", — обязательно отсутствовать. Например, если вы хотите найти объявления о продаже недорогих автомобилей марки "жигули", ваш запрос может быть сформулирован так:

продам жигули +недорого

Поисковая система будет искать все сообщения о продаже автомобилей этой марки, в тексте которых встречается слово "недорого".



Допустим, у вас возникла необходимость разыскать в Интернете информацию о процессорах Intel Pentium IV, но вы хотели бы исключить из результатов поиска сообщения об их продаже и рекламу компьютерных магазинов. В этом случае запрос можно сформулировать так:

процессор Intel Pentium IV —компьютерный —магазин —продажа

Оператор "логическое И", обозначаемый знаком амперсанд (&), позволяет перечислять слова, которые должны встречаться в пределах одного предложения искомого документа. Например, запрос

коммерческое & предложение

заставит поисковую систему искать документы с предложениями, содержащими оба этих слова.

Оператор "логическое ИЛИ", обозначаемый символом "|", позволяет искать документы, в тексте которых содержится только одно из перечисленных слов. Например, по запросу

рисунок | изображение | иллюстрация

будут найдены файлы, в которых

встречается либо слово "рисунок", либо слово "изображение", либо слово "иллюстрация".

Символ тильды (~), как правило, выполняет то же действие, что и знак "-", то есть исключает из искомого документа отмеченные им слова. Иными словами, в ответ на запрос

программное обеспечение ~цена

поисковая система выдаст список документов, в которых встречаются слова "программное обеспечение", но нет слова "цена".

Удвоение какой-либо команды означает, что данное условие должно быть применено не к одному предложению, а ко всему документу в целом. Например, запрос

столица Франции ~~ турфирма

означает, что пользователю необходимы файлы, в которых встречаются слова "столица" и "Франция", но отсутствует слово "турфирма" в пределах всего документа, а не только в пределах предложения, где были обнаружены эти слова. Запрос

строительные && материалы

указывает поисковой системе на то, что в файл отчета необходимо включить все документы, в тексте которых встречаются слова "строительные" и "материалы", независимо от того, расположены ли они в одном предложении или разбросаны по всему тексту в произвольном порядке.

Логические операторы языка запросов можно комбинировать, для этого служат символы открывающей и закрывающей скобки. Например, запрос

музыка & (beatles | битлз)

означает, что пользователь ищет документы, содержащие либо слова "музыка" и "beatles", либо слова "музыка" и "битлз". В некоторых случаях комбинировать операторы можно и без использования скобок. Например, фраза

легковые & автомобили && Mercedes ~~ запчасти

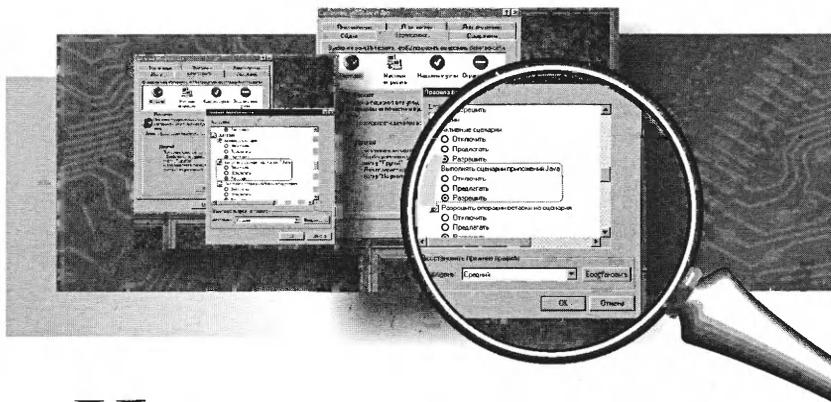
означает, что пользователю необходимы документы, в которых встречаются слова "легковые" и "автомобили" в пределах одного предложения, слово "Mercedes" в преде-

лах всего текста и нигде не встречается слово "запчасти".

Некоторые поисковые системы воспринимают кавычки как служебный оператор, означающий, что заключенную в них фразу следует искать только целиком. Другими словами, если ваш запрос поисковой машине будет выглядеть как "поющие гитары", то поисковая система включит в файл отчета веб-страницы только с указанным словосочетанием, а фразы вроде "после того, как они взяли в руки гитары, люди, поющие у костра, мгновенно смолкли" будут проигнорированы.

Большинство поисковых машин не различают регистр, в котором пользователь вводит свой запрос, то есть словосочетания "декоративные покрытия", "Декоративные Покрытия" и "дЕКОРАТИВНЫЕ ПокРЫТИЯ" будут восприниматься ими одинаково. Тем не менее, ряд поисковых систем, например, "Яндексо", учитывает регистр, и поэтому, во избежание ошибок, рекомендуется всегда вводить запрос только строчными буквами, используя заглавные лишь для имен собственных и географических названий.

Описанные здесь команды языка запросов не исчерпывают все возможности поисковых систем. Например, тот же "Яндекс" поддерживает еще порядка десяти операторов, позволяющих искать словосочетания, элементы которых разделены строго определенным количеством слов, а также искать информацию только в заголовках веб-страниц или в подписях к иллюстрациям. "Рамблер" с использованием специальных команд допускает гибкое включение или отключение функции автоматической генерации роботом производных словоформ из ключевых слов. Поэтому советую сначала внимательно прочитать инструкции по использованию языка запросов, публикуемые обычно в справочном разделе каждого поискового сервера. Потратив несколько минут на изучение документации, вы сэкономите гораздо больше времени, научившись составлять грамотные и четкие запросы, а в конечном счете будете без труда получать именно то, что вам нужно.



Индексирование веб-документов

Алексей Петюшкин, Валентин Холмогоров

Как известно, наиболее эффективный и результативный путь для раскрутки веб-сайтов — это их регистрация в каталогах интернет-ресурсов и индексирование поисковыми системами. Для того чтобы эти инструменты работали правильно в руках пользователей Всемирной Сети, автор веб-страницы должен хорошо знать их устройство и принципы работы.

Поисковые системы

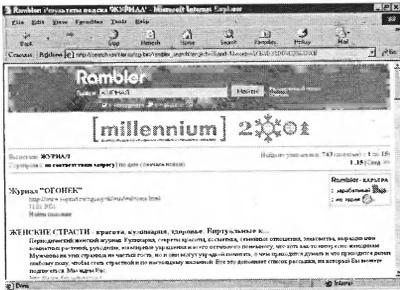
Поисковые системы (их называют также поисковыми машинами) — это полностью автоматизированные интерактивные службы поиска информации в Интернете по введенному пользователем текстовому запросу.

Создав и разместив в Интернете веб-сайт, разработчик регистрирует его адрес в поисковой системе, то есть вносит URL стартовой странички в соответствующее поле на поисковом сервере. Вслед за этим специальная программа поисковой системы, называемая роботом, или спайдером (от англ. spider — паук) просматривает данный документ, проходит по всем обнаруженным на нем гиперссылкам и вносит информацию об обнаруженных страницах в базу данных поисковой системы, которая носит название индекса. Этот

процесс называется индексацией веб-страниц. Как правило, индекс поискового сервера содержит только адреса веб-страниц и краткую аннотацию (70—150 символов), поясняющую их содержание и почерпнутую из текста самих индексированных документов. Спустя некоторое время робот автоматически проводит переиндексацию, в ходе которой удаляет из базы данных устаревшие и более не действительные ссылки.

При обращении к серверу робот проверяет наличие файла robots.txt в корневой директории каждого сайта. В этом файле с помощью специального макроязыка могут быть вручную установлены правила индексирования тех или иных документов текущего узла. Однако не все роботы "умеют" читать указанную в этом файле информацию: часть из них не поддерживает специальный стандарт описания правил индексирования файлов Standard for Robot Exclusion. К сожалению, сегодня не все веб-мастера и ведущие интернет-проекты имеют представление об этом стандарте, к тому же не каждый системный администратор, настраивающий веб-сервер, уделяет должное внимание применению правил описания документов для поисковых роботов.

Пользователь, обращаясь к поисковой машине, просто вводит в



Файл отчета в "Рамблере" по слову "журнал"

вить веб-документ для индексирования поисковыми роботами.

Релевантность документов

Когда пользователь вводит в строке запроса поисковой системы какое-то слово или словосочетание, робот обращается к базе данных поисковой системы, в которой хранятся ссылки на проиндексированные страницы с краткими аннотациями. Количество полученных ссылок может достигать десятков и даже сотен тысяч. Робот выводит результаты по критерию наибольшего соответствия содержания документов запросу пользователя, то есть в порядке убывания релевантности. В результате самые подходящие по его мнению страницы будут размещены в начале файла отчета. На релевантность влияют разные факторы.

Количество ключевых слов

Под количеством ключевых слов (keywords) подразумевается их частота присутствия в тексте веб-документа. То есть страница, на которой робот 15 раз обнаружит запрошенное пользователем слово, будет более релевантна, чем та, которая содержит это слово всего 3 раза. Логично допустить, что если страница называется "Лучшие бесплатные программы для Unix", в ее тексте вряд ли можно встретить слова "эротика", "макияж" или "трубопровод". Зато "система" или "Unix" могут встретиться десятки раз. Однако некоторые недобросовестные веб-мастера, стремясь искусственным путем повысить релевантность своих документов, идут на неэтичный шаг: указывают в документе ключевые слова, совершенно не имеющие от-

ношения к теме сайта. Делается это обычно очень мелким шрифтом в самом низу страницы или цветом, совпадающим с фоном. Иногда одно и то же ключевое слово указывается большое количество раз.

Густота ключевых слов

Под густотой (плотностью) ключевых слов принято понимать степень отношения количества ключевых слов к остальным словам в пределах документа. Поисковые роботы считают более релевантной страницу с конкретным словосочетанием, нежели документ, в котором есть помимо этого словосочетания другие слова и фразы. Иными словами, степень релевантности с точки зрения поискового механизма тем выше, чем ближе стоят искомые слова друг к другу в обнаруженной фразе и чем меньше между ними встречается других слов. Например, в файле отчета, сгенерированном поисковой системой на соответствующий запрос, документ, содержащий словосочетание "коммерческое предложение", будет идти в результирующем списке впереди страницы, содержащей фразу типа "оценив коммерческое состояние своей семьи и размер приданого, он сделал ей предложение".

Расположение ключевых слов

Поместив ключевые слова в особые места веб-страницы, можно надеяться на более высокую релевантность в индексах поисковых систем. Прежде всего это самое начало документа: текст, содержащий ключевые слова в верхней части страницы, заметно повысит релевантность. Ключевые слова также часто включают между парными тегами <TITLE></TITLE>. В результате многие авторы веб-страниц допускают ошибку, загромождая заголовок ключевыми словами. Такой подход имеет по меньшей мере три больших минуса: во-первых, исходный размер HTML-файла заметно увеличивается, во-вторых, вывод на экран данных, указанных в заголовке (<TITLE></TITLE>), делает название найденных документов малопривлекательным, в-третьих, пользователю

будет довольно сложно поместить страницу с таким длинным заголовком в папку Favorites/Bookmarks.

Кроме этого, на релевантность документов влияет наличие выделенных фрагментов текста (теги форматирования , , , <U>, <I>, <H1>, <H2> и др.) и внесение ключевых слов в комментарии, поле альтернативного текста <ALT> и мета-теги (им будет посвящена отдельная часть статьи). Комментарии обособляются тегами <!-- и --> и могут включать любую информацию (она в окне браузера не отображается), например:

```
<!--Далее в статье идет текст, посвященный мета-тегам-->
```

Когда речь заходит о релевантности страниц, поле альтернативного текста <ALT>, предназначенное для подсказок к графическим элементам при наведении на них курсора, тоже имеет большой вес. Приведу следующий пример: на странице абстрактного предприятия "Фирма" размещена кнопка about.gif размером 5 Кб со ссылкой, ведущей в раздел, рассказывающий об истории возникновения предприятия. Вот три варианта использования поля <ALT>:

```
<IMG SRC=about.gif ALT="about.gif, 5 Кб">
```

```
<IMG SRC=about.gif ALT="Об истории">
```

```
<IMG SRC=about.gif ALT="История возникновения ООО Фирма">
```

Очевидно, что релевантность страниц здесь будет возрастать от первого варианта к последнему.

Мета-теги

Мета-теги — специальные инструкции, указываемые в заголовках веб-документов между тегами <HEAD> и </HEAD>. По сути они имеют два основных атрибута: HTTP-EQUIV и NAME. Нас больше интересует атрибут NAME, а точнее — его значения "keywords", "description" и "robots" как непосредственно влияющие на индексирование поисковыми роботами.

```
META NAME="keywords" CONTENT="."
```

Здесь в значении атрибута CONTENT следует указывать через запятую ключевые слова и словосо-

четания. Сюда можно добавить слова, которые не встречаются в документе или встречаются крайне редко, но имеют отношение к тематике сайта, что повысит релевантность. Не стоит избегать указания словосочетаний: во-первых, многие пользовательские запросы состоят из нескольких слов, а во-вторых, документ с указанием в качестве keywords целой фразы будет иметь большую релевантность, чем страница, где вместо ключевой фразы определены ее отдельные слова (например, "электронные магазины" или "электронные" и "магазины"). В среднем в качестве ключевых слов допускается указывать до 150—200 символов.

META NAME="description" CONTENT=":"

Значение атрибута CONTENT уместно использовать, если в самом документе нет (или очень мало) тематического описания вашего сайта. Если робот не обнаружит данного значения, он приступит к проверке страницы с целью найти текст, характеризующий документ, чтобы затем вывести его аннотацию в файле отчета. Оптимальное для вас решение — использовать и значения ключевых слов, и описание (можно дублировать ключевые слова в конструкции META NAME="description" CONTENT="...").

META NAME="robots" CONTENT=":"

Эта конструкция предназначена для того, чтобы указать роботу, индексирующему ваш сайт, что надо добавлять в индексы, а что нет (во многих случаях просто необходимо запретить индексирование: чаты, служебные страницы с сообщениями об ошибке и пр.). Есть шесть возможных значений атрибута CONTENT для данной конструкции:

ALL (разрешение индексации документа со всеми присутствующими в нем гиперсвязями);

NONE (запрет индексации документа со всеми присутствующими в нем гиперсвязями);

INDEX (разрешение индексации документа);

NOINDEX (запрет индексации документа);

FOLLOW (разрешение индексации присутствующих гиперсвязей);

NOFOLLOW (запрет индексации присутствующих гиперсвязей);

Если этот мета-тег пропущен или не указано значение атрибута CONTENT, то поисковый робот понимает их отсутствие как конструкцию CONTENT="INDEX, FOLLOW" (эти два значения аналогичны ALL).

Проблемы индексирования страниц

При подготовке страниц к индексированию роботами вы можете столкнуться с рядом затруднений.

Так, в Сети множество сайтов создано с применением многооконной структуры (фреймов), а большинство поисковых систем не умеют с ними работать. Что же делать пользователю, который хочет, чтобы его ресурс был проиндексирован? Выход есть. Если вы когда-нибудь работали с фреймами, то знаете, что некоторые браузеры их не поддерживают. На этот случай в документе, определяющем адреса загрузки составных страниц, ставится парный тег <NOFRAMES> и </NOFRAMES>. В нем обычно пишут что-то вроде "Ваш браузер не поддерживает фреймы", дают ссылку на последнюю версию MSIE или NN, а в лучшем случае — линк на бесфреймовый вариант сайта.

Информация, заключенная в данный тег, не отображается браузерами с поддержкой фреймовых структур. Значит, туда можно поместить подробное описание вашего сайта, указать ключевые слова и словосочетания.

Причин отказа в индексировании может быть несколько: некоторые поисковые роботы индексируют страницы, расположенные только в зоне .RU и соответствующих ей городских зонах, и если ваш сайт размещен на зарубежном сервере, в индексы он не попадет. Некоторые роботы имеют ограничение на занесение в индексы (не более какого-то числа ресурсов с одного домена). В конце концов, возможно, что вы случайно вместо значения INDEX мета-тега указали NOINDEX или сделали синтаксическую ошибку в конструкции тегов мета-данных, влияющих на индексирование.

Основной работой в Интернете является, безусловно, получение информации. Типичное соотношение объема переданной и полученной информации — 1:10 (если, конечно, вы не передаете большие файлы через электронную почту). Поэтому все описанные в этой статье инструменты призваны оптимизировать, в первую очередь, входящий поток информации.

Так выглядит левый верхний угол экрана моего ПК во время работы в сети. Рассмотрим подробнее представленные здесь программы.



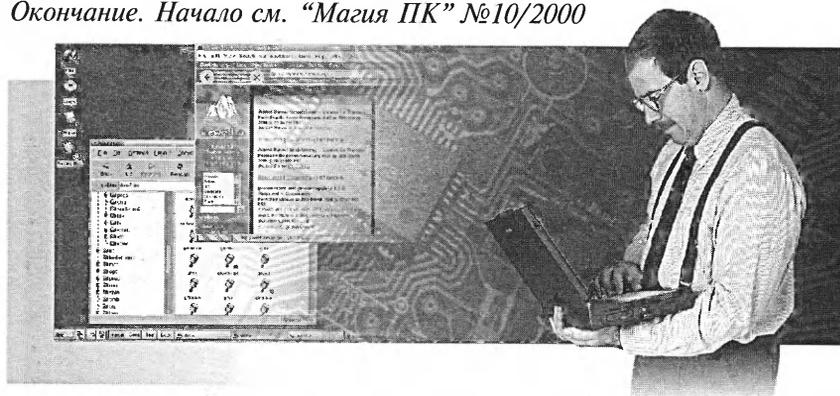
Веб-акселераторы

Ни один сеанс связи со Всемирной Сетью, даже с использованием прокси-сервера провайдера, не в состоянии обеспечить такую эффективность, какой можно добиться с применением различного рода ускорителей загрузки данных — так называемых веб-акселераторов. Существует большое количество подобных программных средств, которые по-разному подходят к решению этой проблемы, но в основе их, как правило, лежат два различных метода увеличения скорости связи:

1. Метод упреждающей загрузки страниц. В этом случае после открытия какой-либо веб-страницы акселератор в фоновом режиме начинает чтение следующих страниц по обнаруженным в документе ссылкам. Понятно, что при этом можно получить большой объем ненужной информации, особенно в том случае, если вы не собираетесь путешествовать по этому сайту далее, однако при этом простой модема практически сводится к минимуму, что положительно сказывается на скорости и устойчивости всего соединения в целом.

2. Метод "интеллектуального" кэширования, в основе которого лежит замена стандартного кэша браузера аналогичным механизмом ускорителя. Последний дополнительно ведет учет частоты обращения к различным веб-страницам и на основании этой информации старается прогнозиро-

Окончание. Начало см. "Магия ПК" №10/2000



Эффективная работа в Интернете

Олег Аверин, фирма "Компьютерный мир"

вать ваши дальнейшие действия и предоставлять дополнительный сервис, например, путешествие по заранее известным маршрутам в Сети.

Практически все современные акселераторы используют оба эти метода. Я рекомендую наиболее продвинутую, с моей точки зрения, программу такого типа — Naviscope ver.8.7 <http://www.naviscope.com/>. Сразу отмечу, что это далеко не самая простая в освоении программа, однако потратив некоторое время на изучение ее настроек и основных возможностей, вы получите эффективный, а главное — надежный с точки зрения безопасности инструмент, который увеличит скорость загрузки веб-страниц на 30—40%. Несколько облегчить положение вещей поможет программа русификации (ищите на сайтах <http://rus.agava.ru> или <http://nord-h.da.ru>).

Вот основные возможности Naviscope:

- мощный мониторинг просмотра веб-страниц;
- ускорение просмотра с использованием упреждающего чтения, кэширования DNS-адресов;
- возможность блокирования

рекламы, cookie и JavaScript, музыки и фона страницы;

- Ping & traceroute (функции для оценки эффективности соединения);
- работа в паре с прокси-провайдером и автоматическая настройка для Netscape, IE и Opera;
- история просмотра последних 40 страниц и возможность обновления их через Интернет;
- загружаемый размер — менее 600 Кб в памяти.

К этому стоит добавить, что Naviscope ничего не сделает без явного на то указания с вашей стороны, тем самым полностью соблюдая privacy (сохранность личной информации) пользователя.

FTP — получение файлов из Интернета

Поскольку наиболее емкая часть информации (в том числе и программы, которые мы используем) представлена в Сети, как правило, в виде архивов, перейдем к вопросу об оптимизации процесса загрузки файлов.

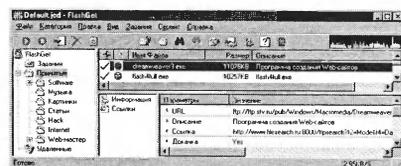
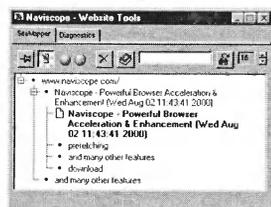
Как я уже упоминал, перед тем как щелкнуть на ссылке, следует просто навести на нее мышью и посмотреть в рабочую строку браузера. Если в начале адреса будет стоять ftp:// или в конце после точки расширение Eхе, Zip, Rar, Gz и т.п., то переход по этой

ссылке инициирует начало загрузки файла (download). Обычно после этого автоматически стартует сервис IE5, отвечающий за работу браузера по протоколу FTP. Однако ему присущи некоторые недостатки:

- при доступе к медленным серверам возникает ощутимая задержка, которая блокирует работу IE до тех пор, пока он не получит ответ от запрашиваемого сервера;
- отсутствует возможность докачки файла при разрыве соединения;
- диагностика о ходе процесса загрузки желает оставлять лучшего.

Все перечисленные трудности браузеров породили целый класс программ, называемых ftp-клиентами или, точнее, менеджерами загрузки, которые призваны существенно повысить эффективность процессов получения файлов из Сети. Основными представителями их являются ReGet, Go!Zilla, NetVampir и полноценный ftp-клиент CuteFtp. При желании загрузить их и получить более подробную информацию можно по адресу <http://www.listsoft.ru/95/down.htm>. Остановлюсь подробнее на небольшой программке FlashGet (ранее известной как JetCar, последняя версия доступна по адресу <http://www.amazesoft.com>). Позволю себе перечислить основные достоинства этой утилиты:

- умеет разбивать файлы до 10 частей и одновременно выполнять до 8 заданий — это выглядит так, словно десять пользователей загружают один и тот же файл с восстановлением загрузки из разных мест;
- осуществляет поиск по зеркалам через FTP и загрузку с самых быстрых и доступных серверов.
- распределяет файлы по категориям, может сразу писать аннотации к файлам, автоматически сохраняет ссылки загрузки с последующим использованием их для обновления версии программы.



Теперь несколько практических советов по организации процесса загрузки файлов с помощью FlashGet:

1. Если загрузка файла долго не начинается, приостановите выполнение задания и запомните или скопируйте в буфер имя файла.

2. Перейдите на один из поисковых серверов (я использую www.ftcenter.ru).

3. Введите имя файла в строке поиска и получите список серверов, с которого он доступен.

4. Если файл имеет большой размер, следует определить наиболее скоростной сервер, с которого будет выполняться загрузка. Для этого дополнительно установите у себя программу *dipstick v.2.52* (www.klever.net) и перетащите мышью выбранные ссылки в окошко программы — через некоторое время она покажет адрес самого быстрого сервера. Именно по этой ссылке и следует загружать файл.

Интернет под микроскопом

Напоследок хочется упомянуть о программе *@guard ver.3.22* (www.atguard.com). Это персональный файрвол, или межсетевой экран. Журнал уже рассказывал о нем, и для запрещения загрузки рекламных баннеров и блокирования cookies я использую именно его. В алгоритме работы этой утилиты, наряду с общепринятым слежением за появлением в ссылках ключевых слов, указывающих на возможное их отношение к рекламе, реализован механизм "Корзины", куда можно просто перетащить ненужную ссылку прямо с баннера и получить полную гарантию, что данный графический образ больше никогда не появится на экране. Кроме того, *@guard* дает возможность избирательно обрабатывать cookies, что позволяет без ущерба работать с проверенными системами, использующими этот сервис. Дополнительную информацию о *@guard* можно найти по адресу <http://www.inetcomm.ru/atguard.html>.

Все дополнения, замечания и комментарии направляйте по адресу javs2000@mail.ru.



Игорь Ананченко

Прокси-сервер. Обратная сторона медали

Подключаясь к сети Интернет, пользователь получает от провайдера подробную инструкцию. В ней сообщается телефонный номер модемного пула, логин и пароль, а также, как правило, номер порта и имя прокси-сервера провайдера. Для большинства пользователей необходимость прописывать прокси, указывая в настройках браузера нечто типа *cache.peterlink.ru 3128*, воспринимается как некоторое непонятное действие, но на деле секрет прост: обычно, если прокси-сервер в настройках не указан, увидеть другие сайты Сети, за исключением сайта своего провайдера, невозможно.

О том, что такое прокси-сервер, журнал уже рассказывал в статье "О портах, прокси-серверах и безопасности" (№10/2000), поэтому остановлюсь коротко на самом главном. Прокси-сервер — это специальная программа, установленная на компьютере провайдера. Ее основная задача — ускорять доставку документов (веб-страниц или файлов) пользователю Интернета с удаленного веб-сервера. Прокси выступает промежуточным звеном между браузером и веб-сервером. Запрос пользователя поступает на прокси-сервер, который обращает-

ся по указанному адресу и доставляет требуемый документ. При этом прокси сохраняет копию документа в своей памяти (кэширует на диске). Теперь, если через некоторое время пользователь (этот или другой, но использующий тот же прокси-сервер) запросит данный документ, прокси-сервер сразу извлечет его из своего кэша. Для пользователя данная операция неотличима от работы с удаленным сервером, но при этом скорость обычно на порядок выше, чем в случае повторного обращения за документом к веб-серверу. Поэтому чем больше людей используют прокси-сервер, запрашивая одну и ту же информацию, тем более существенной становится его помощь. Согласно статистике (<http://proxyccheck.spylog.ru/about.html>), количество обращений пользователей к одним и тем же документам в Интернете составляет около 60%. Естественно, что провайдеру, который тоже оплачивает трафик по арендованным им каналам, в целях экономики выгодно использовать прокси-сервер.

Для работы с этой программой необходимо указать имя компьютера, на котором она запущена, и номер порта, за которым эта программа закреплена.

один человек может несколько раз вызвать страницы своего сайта, но система учет эти обращения как визиты разных посетителей. Менять вручную адрес прокси-сервера утомительно, поэтому в Сети можно найти специальные программы, автоматизирующие эту операцию. На сайте <http://bgm.narod.ru/proxy.htm> предлагаются списки прокси для программ hiprom, хруМер, выполняющих операции автоматизированной смены прокси. Впрочем, алгоритм перебора настолько тривиален, что вполне может быть реализован на Delphi самое большее за час.



Вот и все, что касается теории. Но маленькая практика, как известно, стоит большой теории. Разумеется, речь не пойдет о технологии взлома банка, но рассмотренный ниже прием позволит вам в ряде случаев получать деньги практически "из воздуха". В качестве примера (и это только один из множества вариантов!) предлагаю игру на деньги, вернее, на их электронные эквиваленты — ЭРТы, которые можно затем вывести из системы электронных платежей PayCash (www.paycash.ru) на свой банковский счет. Особенно удобно переводить их на дебетную карту банка, под патронажем которого развивается PayCash.

Игра называется "Простейшая" (<http://games.paycash.ru/Scripts/ATMDLL.DLL/Game?ID=2>), так как выиграть в ней действительно очень легко. Для этого нужно внимательно прочесть правила о работе с платежной системой, выбрать правильный ответ и... перевести выигрыш на свой счет. Для того чтобы один игрок не взял себе все выигрыши, после ответа на вопрос IP-адрес игрока блокируется, и сыграть повторно можно только через определенное время

(как показывает практика, не более двух раз в сутки). Очевидно, что использование списка анонимных или искажающих прокси-серверов позволяет играть многократно, подставляя для каждой игры новый прокси. Другой прокси, другой IP, а значит, с позиции компьютера, и новый игрок.

Если учесть, что программу-анализатор, выбирающую правильный вариант ответа и подставляющую очередной прокси из списка, без особых затруднений способен написать студент-первокурсник, становится понятно, к каким неприятностям могут приводить порой оплошности в реализации игровой стратегии. Для борьбы с такими хитрыми игроками следовало бы ввести правило, по которому новую игру можно начать не сразу, а только после некоторой паузы, обязательно следующей после очередного выигрыша, а на обеспечение игрового процесса в течение суток выделять строго определенную сумму, которая не может быть превышена. Тогда хакер сможет выудить, например, все 100 рублей, разыгрываемые по частям, но никак не больше этого предела.

О работе с прокси-сервером с позиции пользователя Интернета сказано почти все. Как пользователь вы можете выбрать себе наиболее удобный прокси-сервер из списка возможных, но конфигурировать его вы не сможете — это прерогатива системного администратора. Однако если вам интересно взглянуть на прокси-сервер как на программу и есть желание установить его на свой ПК для экспериментов, из многочисленных реализаций рекомендую прежде всего ознакомиться с программой Eserv (<http://www.eserv.ru/eserv/>), которая может работать по протоколам HTTP (веб-сервер) и FTP (файловый сервер), а также в качестве прокси-сервера. Если в вашем распоряжении находится небольшая локальная сеть из 2—5 компьютеров, один из которых оснащен модемом и ПО для выхода в Интернет, то Eserv, установленный в качестве прокси-сервера на этом ПК, обеспечит выход в Интернет пользователям других компьютеров локальной сети.

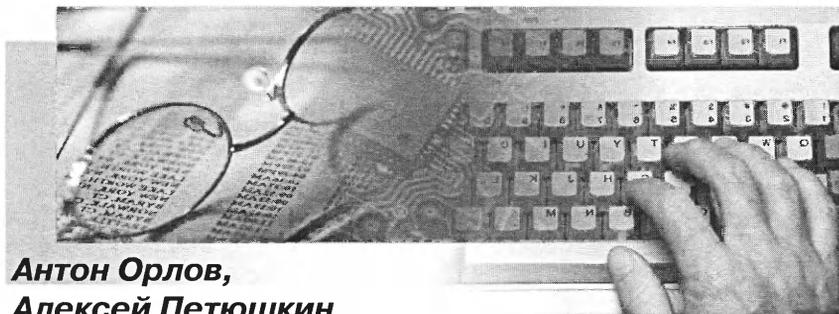
Зачем нужны скрипты, да и вообще, что это такое? Такой вопрос рано или поздно наверняка возникнет у каждого пользователя, решившего создать свой собственный веб-сайт. Неужели стандартного HTML, который и без того сложно изучить, недостаточно?

Действительно, язык разметки гипертекста HTML является основой всех размещенных в Интернете электронных документов. Но это всего лишь фундамент, на базе которого реализуются другие, более гибкие и функциональные сетевые программные технологии, а они позволяют получать поистину фантастические результаты. И такими технологиями можно по праву считать так называемые скрипты, или сценарии. С помощью некоторых скриптов можно реализовать такие интерактивные компоненты веб-страницы, как гостевая книга, голосование, чат, форум и многое другое.

Скрипты — это специальные программы, предназначенные для работы с браузером. Все скрипты можно разделить на две большие группы: исполняющиеся на том компьютере, который загружает данные из Интернета (то есть на клиентском компьютере), и на компьютере, который эти данные предоставляет, — на сервере.

Скрипты, выполняемые на клиентском компьютере

Такие скрипты (Client-side scripts) выполняются на компьютере пользователя посредством вызова определенных команд, которые обозначены в фрагментах кода обычной веб-страницы. Для того чтобы скрипты могли быть выполнены, браузер должен уметь их интерпретировать и выполнять, то есть поддерживать язык, на котором они написаны. В настоящее время существуют два языка для написания таких скриптов — JavaScript и VBScript, причем первый из них поддерживается практически всеми современными браузерами, а второй — только Microsoft Internet Explorer. Поэтому использо-



**Антон Орлов,
Алексей Петюшкин**

Какие бывают скрипты

*Quod scripsi, scripsi (я написал то, что написал)
Понтий Пилат*

вание JavaScript предпочтительнее, но этот язык несколько сложнее по своему синтаксису и не позволяет реализовывать некоторые возможности, которые доступны VBScript.

Синтаксис языков JavaScript и VBScript подробно описан в специальной литературе. Существуют сайты, посвященные этим языкам, библиотеки скриптов (наиболее известная и большая — www.scripts.ru). Пособия по программированию в электронном виде можно найти на сайтах <http://infocity.kiev.ua>, <http://rusdoc.df.ru> и др. Неплохой метод изучения JavaScript и VBScript — исследование страниц со скриптами, коих сейчас в Интернете великое множество.

Основные понятия JavaScript и VBScript таковы. Скрипт, располагаемый на веб-странице, заключается в теги `<script>` и `</script>` — без этого он будет интерпретирован как простой текст и отображен таким в браузере. Для того чтобы скрыть текст скрипта от браузеров, не понимающих тег `<script>`, описание самого сценария обособляется тегами комментария, открывающим `<!--` и закрывающим `-->`. Тогда браузер, поддерживающий скрипты, этот скрипт выполнит, а не поддерживающий — просто проигнорирует.

Если скрипт написан на языке VBScript, то в его теге должно стоять указание на это вида `<script`

`id="clientEventHandlersVBS" language="vbscript">`. То же рекомендуется делать и для скриптов на JavaScript: `<script language="JavaScript">`.

Текст скрипта может находиться не на странице, а в отдельном файле с расширением `.js` (для скриптов на JavaScript) или `.vbs` (для скриптов на VBScript), что облегчит труд по обновлению сайта, если обновляемой частью являются именно скрипты. К сожалению, увеличить скорость загрузки страницы путем вынесения скриптов в отдельные файлы не удастся — они все равно будут загружены не при вызове, а при загрузке страницы. Для включения текста скрипта из отдельного файла следует использовать теги `<SCRIPT language = JavaScript src = "Имя_файла.js">` `</SCRIPT>` или `<SCRIPT language = VBScript src = "Имя_файла.vbs">` `</SCRIPT>`.

В приведенных примерах указан путь до сценария, написанного на языке JavaScript (расширение `.js`) и на VBScript (расширение `.vbs`).

Скрипты на JavaScript и VBScript могут исполняться как автоматически во время загрузки страницы, так и при каких-либо действиях пользователя. Конкретный способ выполнения зависит от текста скрипта. Скрипт, выполняемый автоматически, состоит из простого набора ко-

манд, без каких-либо заголовков и окончаний. Например, скрипт

```
<script language="JavaScript">
if (navigator.appName == "Microsoft
Internet Explorer") document.write ("

```

будет выполнен сразу после загрузки, и в результате его выполнения в загружаемом документе в том месте страницы, где располагается текст этого скрипта, появится строка "Вы используете Microsoft Internet Explorer". Если необходимо, чтобы скрипт был выполнен до загрузки страницы, рекомендуется поместить его в раздел `<head>`, однако тогда функции работы с текстом страницы скрипту будут недоступны.

Для того чтобы скрипт выполнялся при наступлении определенных условий или событий (нажатие пользователя на кнопку, окончание загрузки страницы и др.), необходимо оформить его как функцию. Это значит, что тогда у скрипта должен появиться заголовок и окончание. Для JavaScript заголовком является оператор `function имя_скрипта()`, а весь текст скрипта заключается в фигурные скобки:

```
<script language="JavaScript">
function script()
{...
Команды скрипта
...}
</script>
```

В VBScript заголовок — слово `Sub имя_скрипта()`, а после текста скрипта должна следовать команда `End Sub`:

```
<script id = "clientEventHandlersVBS"
language = "vbscript"> Sub runsrncr()
...Команды скрипта...
End Sub
</script>
```

Скрипт, оформленный таким образом, имеет имя — то, что стоит после слов `function` или `Sub`. По этому имени его можно вызывать из любого места страницы. Вызов скрипта — это просто команда его выполнить. Например, при нажатии кнопки `<input type="button" value = "Готово" onClick = "startscript()">` будет выполнен скрипт с именем `startscript`. Вызываемый скрипт мо-

жет находиться в любом месте страницы, однако настоятельно рекомендуется, чтобы он располагался выше места его вызова. Иначе при неполностью загруженной странице может возникнуть неприятная ситуация: при вызове незагруженного скрипта браузер выдаст ошибку.

Скрипты, выполняемые на сервере

К этой категории (Server-side scripts) относятся сценарии, написанные на таких скриптовых языках, как Perl (CGI), ASP, PHP и JSP.

CGI/Perl

Принцип действия таких скриптов: при их выполнении берутся данные от браузера пользователя и обрабатываются на сервере. В этом смысле CGI (Common Gateway Interface) является шлюзом для передачи данных, полученных от клиента, программам-обработчикам информации — базам данных, электронным таблицам и пр.

Чаще всего CGI-скрипты пишут на языке Perl, хотя встречаются написанные на C++, Pascal и др. Чтобы CGI-программа запустилась на сервере, необходимо знать наименование и путь интерпретатора того языка, на котором она написана. Путь к Perl-интерпретатору чаще всего выглядит следующим образом:

```
#!/usr/bin/perl
или
#!/usr/local/bin/perl
```

Символом # обозначаются комментарии к тем или иным фрагментам кода CGI-программы. Например:

```
#!/usr/local/bin/perl
#Содержимое файла test.pl
print "Это моя CGI-программа\n";
```

Сохранив данный текст в файле под именем test.pl и запустив его на сервере (чаще всего запуск CGI-сценариев происходит в директории CGI-BIN), мы увидим в окне браузера надпись "Это моя CGI-программа".

При написании CGI-скриптов учтите следующее. Во-первых, прежде чем производить отладку программы, необходимо выяснить у администратора сервера, на котором раз-

мещен ваш сайт, разрешается ли исполнение CGI-сценариев. Во-вторых, обязательно уточните путь к интерпретатору языка, на котором написан ваш скрипт. В-третьих, не забывайте, что к исполняемым файлам (.pl, .cgi, .fcgi и др.) необходимо применить команду CHMOD с разрешением допуска, обычно это 755 (rwx-rx-rx). И, наконец, помните, что файлы сценариев следует загружать на веб-сервер в ASCII-режиме.

ASP

Язык ASP (Active Server Pages) был разработан компанией Microsoft и потому требует установки специального ПО — Microsoft Internet Information Server. По большому счету ASP представляет собой совокупность средств программирования с использованием HTML, чтение и запись в базу данных посредством ODBC (Open DataBase Connection), а также массу других возможностей, весьма похожих на такие технологии, как CGI, JavaScript, Perl, ActiveX и ISAPI.

Активные серверные страницы не требуют каких-либо специфических браузеров или иного ПО со стороны клиента. Все скрипты запускаются и выполняются на сервере, а браузер получает только результирующие HTML-файлы и отображает их на компьютере пользователя. ASP-сценарий начинается и заканчивается символами <% и %>, например:

```
<% if now()=1/1/01 %> Это новый,
2001 год.<p>
<% else %> Это просто обычный
день.<p>
<% end if %>
```

PHP

PHP (Personal Home Page) — язык серверных скриптов, которые встраиваются в HTML-документы и, исполняясь на сервере, выдают результат обработки данных в окне браузера. Именно поэтому PHP называют пре-процессором HTML (об этом гласит и надпись на входе на



официальный сайт этого языка <http://www.php.net>).

Основное отличие PHP, например, от Perl, заключается в том, что в процессе написания CGI-сценария программист сам определяет HTML-код выводимого в результате обработки документа, а используя PHP, можно встраивать в любую веб-страницу готовую программу, выделенную специальными тегами:

```
<html> <body>
<?php
echo "Это моя PHP-программа";
?>
</body> </html>
```

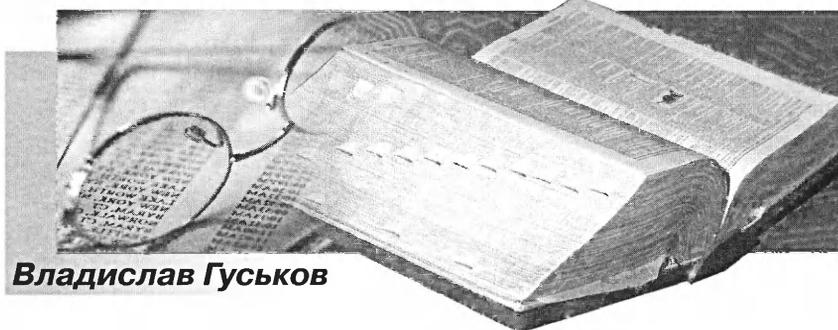
Сохранив этот текст в файле под любым именем с расширением .php3 или .html и запустив его на сервере, можно увидеть, что браузер выведет надпись "Это моя PHP-программа". Причем, если просмотреть источник HTML-кода результирующего документа, то от самого PHP-сценария не останется и следа.

PHP — очень гибкий язык, он поддерживает MySQL, NNTP, POP3, IMAP, Oracle, PostgreSQL, dBase, ODBC, Informix и многие другие технологии. С его помощью можно реализовать все то же, что позволяет сделать CGI-программа, отличие лишь в том, что PHP более прост для понимания.

JSP

Технология JSP (Java Server Pages) стала ответом компании Sun Microsystems на создание языка ASP от Microsoft и представляет собой совокупность элементов и шаблонов данных, описывающих процесс обработки запросов. JSP дает веб-программистам две основные возможности: доступ к удаленным блокам данных и вычленение Java-кода из гипертекстовых фрагментов. В основе механизма действия JSP лежит обработка XML-данных и небольших сценариев, написанных на языке Java, для последующей генерации содержимого конечного документа. Синтаксис JSP довольно сложен, но подробно описан в специальной литературе.

Вот и все главное о скриптах и скриптовых языках. Решайте, что изучать, а что нет, что размещать на своем веб-сайте, а что не стоит.



Владислав Гуськов

Создание CGI-скриптов

Многие мои знакомые, изучив HTML и создав свою первую домашнюю страничку, уже считают себя настоящими веб-дизайнерами. Конечно, подготовить простенький веб-сайт, написать немного текста и добавить картинок — дело не сложное, поэтому такие странички в Интернете исчисляются миллионами. Более профессиональными считаются ресурсы, содержащие так называемые интерактивные элементы: счетчики посещений, гостевые книги или веб-форумы. Все упомянутое выше реализуется с использованием технологии CGI.

Три магические буквы

CGI (Common Gateway Interface — стандартный шлюзовой интерфейс) — это набор спецификаций, определяющих механизм взаимодействия интерактивных сценариев (также называемых скриптами) и веб-серверов. То есть CGI формирует рабочую среду (набор переменных) и протоколы для обмена информацией между сервером и скриптом, а скрипты — это уже есть программы, выполняющие определенные действия. Скрипты могут быть написаны практически на любом языке программирования, например, на C, Perl, или на Delphi,

главное, чтобы данный язык поддерживал необходимые переменные среды окружения. Сложно, не правда ли? Чтобы стало немного понятнее, давайте рассмотрим основные этапы работы CGI-скрипта:

1. Броузер пользователя отображает форму (например, два поля: имя и отчество).

2. Пользователь вводит данные и щелкает на кнопку "Отправить".

3. Броузер соединяется с сервером и запрашивает файл (в нашем случае CGI-сценарий, обрабатывающий форму), указанный в атрибуте Action тега "Form", одновременно передавая серверу данные, введенные пользователем в поля формы.

4. Сервер находит необходимый сценарий и формирует среду переменных, которая включает сведения о броузере, самом сервере и, конечно, переданную информацию.

5. Сервер запускает сценарий, обрабатывающий все полученные переменные.

6. Сценарий возвращает серверу обработанную информацию, далее она встраивается в какой-либо файл HTML, отправляется электронной почтой, либо изменяет один из элементов размещенного на веб-странице графического изображения, если речь идет о счетчике посещений.

Конечно, описанная выше схема работы CGI-скрипта несколько уп-

рощена, однако она дает определенное представление об алгоритмах действия этой технологии.

Язык программирования Perl

Как я уже упоминал, создавать скрипты можно на любом современном языке программирования, главное, чтобы он мог "общаться" с CGI, то есть был способен получать от него переменные. Такими возможностями обладает любой современный язык программирования, но так сложилось, что наибольшую популярность приобрели языки C и Perl. Синтаксис последнего, на мой взгляд, несколько проще, поэтому далее речь пойдет именно о нем.

Немного истории

Perl разработал в 1986 году Ларри Уолл, системный администратор одного из UNIX-серверов. Основная цель, которую он преследовал — создание языка программирования, способного обрабатывать файлы и генерировать отчеты с возможностью системного администрирования. Новый язык быстро набирал популярность среди UNIX-программистов, но широкой публике он стал известен только в последние годы, после выхода Perl-интерпретатора для платформ Windows. Perl — интерпретируемый язык, то есть код программы хранится в текстовом формате (в виде скрипта), а для запуска этого скрипта используется интерпретатор — программа, которая переводит (транслирует) скрипт в байт-код и затем его выполняет.

В отличие от многих других языков программирования, в Perl обозначения переменных не требуется вообще, а тип обрабатываемых данных он определяет сам, в зависимости от содержащейся в программе информации. Представьте себе, какой рутинной работы мы можем благодаря этому избежать! Например, в любом месте программы можно написать:

```
$s="test";
```

```
$b=$s+4;
```

Переменной *s* присваиваем значение "test" (автоматически под нее выделяется динамическая память), а затем другой переменной *b* присваиваем значение *s* и прибавляем к 4. В результате *b* будет равен "test4", вот такие чудеса. Какая от этого польза? А помните, как вы мучились в Delphi при высчитывании количества элементов в массиве? Кстати, вас, наверное, заинтересовал знак \$ перед каждой переменной, он означает ее тип — скалярная. А еще бывают массивы и хэши (ассоциативные массивы). Хэши представляют собой пару ключ—значение, например:

```
%hash = ("Фамилия", "Иванов", "Имя", "Петр");
```

```
$hash = {"Фамилия"};
```

В первой строке хэшу (перед ним должен стоять знак процента) присваивается пара ключей—значений: Фамилия—Иванов, Имя—Петров. Далее при вызове ключа "Фамилия" выводится его значение, то есть "Иванов". Хэши широко используются CGI-программистами при обработке форм.

Продолжать выделение особенностей синтаксиса Perl и его возможностей можно до бесконечности, но мне хотелось бы напоследок рассказать о главной гордости CGI-программистов — "регулярных вы-

ражениях". Для этого в Perl используют возможности регулярных выражений. Например, нам необходимо найти последовательность "bbb" в переменной:

```
$s="aabbbcd";
print "Последовательность обнаружена" if $s =~ /b{3}/;
```

Присваиваем переменной символы aabbbcd. Теперь нужно узнать, содержит ли она символы bbb. С помощью оператора условия составляем блок: если переменная *\$s* включает последовательность из 3 символов *b*, то выводим сообщение "Последовательность обнаружена". Хочу заметить, что в Perl синтаксис операторов несколько упрощен по сравнению, например, с Delphi. В частности, оператор условия здесь можно записать так:

ВЫРАЖЕНИЕ if УСЛОВИЕ

Первый скрипт

Первым моим практическим применением Perl стало создание счетчика посетителей страницы. Скрипт счетчика занимает меньше десятка строк, но он добросовестно выполняет свое задание — считает посетителей. Счетчик состоит из двух файлов: первый — сам скрипт, а второй — count.txt, в котором хранится число, обозначающее количество

пришедших к вам на сайт посетителей. Схема работы этой программы такова. При заходе к вам на сайт посетителя запускается скрипт, открывающий файл count.txt и считывающий записанное там число. Затем CGI-сценарий увеличивает это число на единицу и полученные данные опять записывает в файл. И, наконец, скрипт выводит в окне браузера надпись: "Вы наш (указывает число) посетитель".

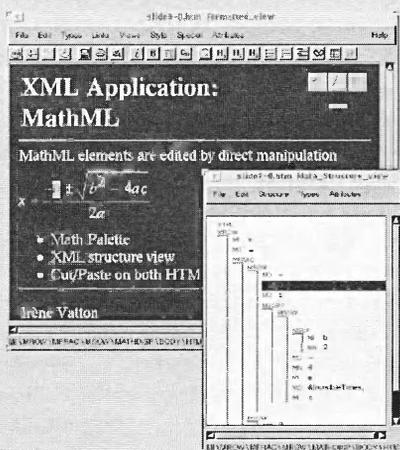
```
#!/usr/local/bin/perl
open(myfile,"+<count.txt");
$ff=<myfile;
$ff++;
seek myfile, 0, 0;
print myfile "$ff\n";
print "Content-Type: text/html\n\n";
print "Вы наш $ff-й посетитель\n";
```

А теперь разберемся в каждой строчке. Любой скрипт должен начинаться со строки, указывающей расположение Perl на сервере, на котором расположено ваш сайт (ее вы можете узнать у администратора сервера). Во второй строке обозначен файл, который необходимо открыть (count.txt) и имя дескриптора (любое символическое имя для представления файла в программе), которое будет соответствовать этому файлу. Заметим, что знак "+" перед именем файла определяет, в каком ре-

Математика + HTML = MathML

Не так давно, 8 января 2001 года, Консорциум W3C объявил о выходе второй версии еще одного языка разметки — MathML (Mathematical Markup Language). Этот язык, описывая совокупность XML-функций, позволяет осуществлять в WWW многочисленные математические операции. Более точная формулировка возможностей MathML звучит так: MathML предназначен для облегчения использования и повторного использования математического и научного наполнения Сети, а также для различных приложений типа компьютерных алгебраических систем, типографского набора и голосового синтеза. MathML может применяться с целью кодирования и представления математического содержания для последующей качественной ви-

зуальной интерпретации для приложений, в которых основную роль играет семантика.



Язык MathML, по спецификации Консорциума W3C, является ничем

иным, как XML-приложением. В совокупности с таблицами стилей CSS, новый язык позволит браузерам обрабатывать математические выражения. Уже сейчас некоторые производители могут предложить специальные апплеты и надстройки, с помощью которых конструкции MathML будут выполняться непосредственно в самом браузере. А в скором будущем появится программное обеспечение — переводчики и редакторы уравнений, — способные генерировать HTML-код на основе представленных математических данных на MathML.

Появление MathML обусловлено тем, что HTML, хотя и содержит довольно богатый набор тегов, не способен обрабатывать математические выражения. MathML состоит из множества XML-тегов, предназначенных для разметки уравнений в соответствии с их условиями. Математичес-

жиме он будет открыт. В нашем случае это режим чтение/запись.

В строке `$ff=<myfile>` мы присваиваем переменной `$ff` содержимое файла `count.txt`. В следующей строке увеличиваем значение переменной на единицу. Для того чтобы записать вместо предыдущего числа новое, функция `seek` переводит курсор в начало файла.

Обычно функция `print print myfile "$ff\n"` выводит какое-либо сообщение или переменную, но в нашем случае дескриптор, записанный после команды `print`, указывает на то, что вывод переменной `$ff` идет в файл `count.txt`.

При обработке какого-либо текста и отправке его в файл необходимо использовать директиву "Content-type", она обозначает тип выводимой информации. В нашем случае следует указать тип `html`, поскольку результат работы скрипта помещается в веб-страницу. И, наконец, последняя строка отображает саму надпись счетчика и содержимое переменной `$ff`.

SSI — помощник веб-дизайнера

Теперь необходимо вызвать скрипт на исполнение. Сделать это

можно разными способами, но лучше и проще всего использовать SSI (Server Side Includes, вложения на стороне сервера).

SSI — это набор специальных тегов, добавляемых в веб-страничку. Они позволяют вставлять в файл фрагмент другого документа, подставлять результаты работы скриптов, выводить информацию о файлах. В первую очередь SSI облегчает нелегкий труд дизайнеров, работающих над большим веб-проектом. При создании новой страницы им приходится копировать HTML-код заголовка, повторяющийся на каждой страничке. А если будет необходимо сменить картинку в заголовке? Кошмар! С помощью SSI смена картинки не представляет особого труда. Просто вы создаете два файла: в одном содержится статический заголовок (`zagolovok.html`), а в другом само тело страницы (`telo.html`). Теперь добавьте в начало второго файла следующий тег:

```
<!—#include virtual = "zagolovok.html"—>
```

Результат не заставит себя долго ждать: при просмотре через браузер файла `telo.html` вначале он выведет статический заголовок, а затем само тело файла. Таким образом, вместо рутинного набора текста на каждой

странице, вы вставляете всего один тег, а все остальное делает сервер. Если вы решили сменить дизайн — не беда: сделайте нужные поправки в файле `zagolovok.html`, и все странички, содержащие тег SSI, примут новый вид.

Чтобы вывести результаты работы скрипта, можно использовать следующую вставку:

```
<!—#exec cgi ="/cgi-bin/test.cgi"?
```

При просмотре страницы в браузере вместо этой строки будет выведен результат работы сценария. Так, если вставить ссылку на скрипт счетчика посетителей, то появится надпись: "Вы наш 11-й посетитель".

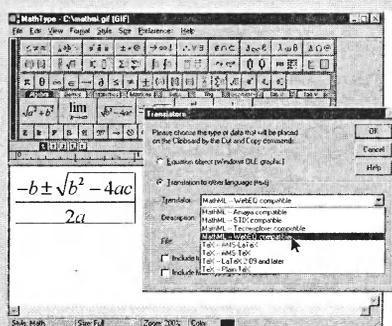
Заключение

Впоследствии, когда вы детально разберетесь в операторах языка Perl, вам не составит труда добавить на страничку новые возможности, такие как определение IP-адресов посетителей, вычисление пика активности пользователей и т.д. Но все же главным применением CGI остается обработка форм, требующая не только знания основ HTML, но и понимания работы CGI. Тему работы с формами, создания гостевых книг, опросов, анкет мы продолжим в следующих статьях.

кий язык разметки MathML направлен скорее на фиксирование исходящих значений, нежели на фокусирование того, что в результате решения будет отображено на экране. XML имеет много общего с HTML и обладает подобным, однако не идентичным синтаксисом. Одно из отличий в том, что в XML нельзя пропускать закрывающие теги. Кроме того, теги элементов, не несущих никакого содержания, отмечаются наклонной чертой (слэш) перед последней угловой скобкой закрывающего тега.

MathML — стандарт, предназначенный для математического описания как основы коммуникативной цепочки "машина—машина". Он не будет использоваться для ручного редактирования, а будет служить как инструмент управления специализированными средствами работы с авторским материалом (например, ре-

дакторы уравнений) и взаимодействия между различными математическими пакетами.



Чтобы представить, как с помощью MathML обрабатываются математические выражения, приведу простой пример:

$$x^2 + 4x + 4 = 0$$

Есть два способа решения этого уравнения. Первый использует теги представления (`presentational tags`),

которые обычно начинаются с "m", затем идет "o" для оператора "i" для идентификатора "n" и т.д. Тег "mrow" указывает на горизонтальную групповую иерархию:

```
<mrow> <mrow>
  <msup> <mi>x</mi> <mn>2</mn> </msup> <mo>+</mo>
</mrow>
<mrow>
  <mn>4</mn>
</mrow>
<mo>&invisibletimes;</mo>
<mi>x</mi>
</mrow>
<mo>+</mo>
<mn>4</mn>
</mrow>
<mo>=</mo>
<mn>0</mn>
</mrow>
```

Другой вариант — применение семантических тегов:

```
<apply> <plus/>
<apply>
```

Тема веб-дизайна сегодня является, пожалуй, одной из наиболее популярных. Общественность желает быть в курсе всех событий и получать наиболее полезную и достоверную информацию для самосовершенствования и самообучения.

Цель данной статьи, которая открывает новый в 2001 году цикл материалов, — отнюдь не пересказ неких прописных истин, которые описаны в многочисленной специализированной литературе. Прежде всего статьи этого цикла будут интересны начинающим веб-мастерам, желающим узнать новые возможности, казалось бы, известных аспектов сайтостроения. Планируемый рассказ о различных приемах и "примочках" веб-программирования не подразумевает того, что авторы журнала и его редакция призывают вас, дорогие читатели, к тому, чтобы, узнав пару-тройку интересных скриптов, вы остановились на достигнутом. Наоборот, этим циклом статей мы стремимся донести до вас глубину возможностей и функционального размаха различных языков и стандартов, сподвигнуть вас к более детальному изучению предмета. А публикуемые примеры советуем складывать в отдельную "копилочку". Разумеется, все публикуемые в дан-



Алексей Петюшкин

Копилка веб-мастера

ном цикле приемы и листинги не претендуют на звание новых и неизвестных для всех читателей. Возможно, кто-то уже знает и давно использует в своей практике некоторое из того, что мы предложим.

Куда пойти, куда податься?...

Горячие споры о несовместимости двух популярных браузеров, Microsoft Internet Explorer (далее MSIE) и Netscape Navigator (далее

NN) в части отображения веб-документов начался довольно давно, и до сих пор сторонники ни того, ни другого браузера никак не могут прийти к единому мнению. Настоящий веб-мастер должен понимать, что нельзя терять и доли потенциальной аудитории посетителей сайта из-за различия моделей браузеров. Поэтому придется более обстоятельно и скрупулезно подойти к HTML-верстке (и, естественно, разбираться при этом во всех особенностях обоих браузеров), чтобы страницы и там, и там одинаково смотрелись, либо делать две версии сайта.

Последний вариант иногда оказывается наиболее подходящим, например, когда вы хотите обязательно использовать некоторые элементы, не поддерживаемые одним из браузеров, или ваш сайт не очень велик по размеру.

Итак, нам нужно, чтобы после загрузки начальной страницы браузер посетителя, в зависимости от модели и версии, перешел на определенную страницу, предназначенную именно для него. Для этого необходимо вставить в код первой страницы вашего сайта следующий скрипт (между тегами <HEAD> и </HEAD>):

```
<script language="JavaScript">
<!--
```

```
</power/>
<ci>x</ci>
<cn>2</cn>
</apply>
<apply>
<times/>
<cn>4</cn>
<ci>x</ci>
</apply>
<cn>4</cn>
</apply>
```

Presentation MathML

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

Content MathML

$$x = \pm (b, \sqrt{b^2 - 4ac}) / 2a$$

В качестве примера ПО, поддерживающего MathML

2.0, можно назвать W3C Amaya browser (рис. 1), DSSSL stylesheets for MathML (рис.2), IBM techexplorer, MathType (рис.3) и другие (полный перечень ПО для MathML можно найти по адресу <http://www.w3.org/Math/implementations.html>).

Более подробную информацию о языке MathML и ссылки на интернет-ресурсы, посвященные этому стандарту, можно найти на сервере W3C (<http://www.w3c.org>).

Алексей Петюшкин

```

if((navigator.appName ==
"Netscape") &&
(parseInt(navigator.appVersion) >= 4 ))
  window.location.href = "nn4.html";
else if((navigator.appName ==
"Netscape") &&
(parseInt(navigator.appVersion) == 3 ))
  window.location.href = "nn3.html";
else if((navigator.appName ==
"Microsoft Internet Explorer") &&
(parseInt(navigator.appVersion) >=
4 ))
  window.location.href =
"msie4.html";
else if((navigator.appName ==
"Microsoft Internet Explorer") &&
(parseInt(navigator.appVersion) ==
3 ))
  window.location.href =
"msie3.html";
else
  window.location.href =
"unknown.html";
// —>
</script>

```

Как видно из листинга, сначала определяется модель браузера, далее его версия и, в зависимости от нее (3.x или 4.x), посетитель переадресуется на соответствующий документ. Например, если у пользователя стоит Netscape Navigator 4.7, то сценарий перенаправит его браузер на страницу nn4.html. Если же у пользователя обнаружен браузер, не являющийся ни MSIE, ни NN, сценарий откроет файл с именем unknown.html. Можно ограничиться только последними версиями или наоборот — добавить строки для еще более ранних версий (как как вы понимаете, на подготовку сайта по каждой из версий потребуется уйма времени).

Семь пятниц на неделе

Если вы хотите сделать свой сайт еще более привлекательным, то должны понимать, что без интерактивных элементов количество посетителей вряд ли будет стабильно высоким (хотя в ряде случаев динамическое "поведение" документа не нужно и даже противопоказано). Следующий сценарий, написанный на языке JavaScript, позволяет выводить разный текст в зависимости от дня

недели: в понедельник один, во вторник — другой, в среду — третий и т.д.

```

<script language="JavaScript">
nowDate=new Date();
theDay=nowDate.getDay();
text=new Array(30);
text[0]="Сегодня воскресенье и
мы пойдем гулять";
text[1]="Сегодня на работу";
text[2]="И снова на работу..";
text[3]="Как? Опять на работу?";
text[4]="Только не говорите, что
сегодня тоже на работу!";
text[5]="Дурацкая работа!!!";
text[6]="Сегодня самый хороший
день, потому что завтра — воскре-
сенье!";
document.write(text[theDay]);
</script>

```

Следует помнить, что указание текста начинается с воскресенья (text[0]), и помещать данный код следует в разделе BODY. Остается добавить, что данный прием часто используется на многофункциональных интернет-порталах, электронных биржах и магазинах, презентационных ресурсах и, само собой разумеется, на домашних страницах.

Время пролетит незаметно

Если вы считаете себя настоящим путешественником по Сети (web-surfer), то наверняка бывали на сайтах, времянахождение на которых ограничено. Режим наложения временного ограничения на нахождение конкретного посетителя на странице может быть использован в таких случаях, как онлайн-тестирование (тот же самый TecMetrics, а ныне — BrainBench), бесплатный кратковременный демонстрационный показ товара или услуги и т.д. Да мало ли ситуаций, когда долгое пребывание на определенной веб-странице нежелательно?

Вот, например, самый что есть бытовой случай: на своей домашней страничке вы заводите фотоальбом. Он состоит из двух разделов: общедоступные фотографии, на которых вы читаете Шекспира в подлиннике, сидите в театре или переводите старушку через дорогу; и фото, разрешенные для просмотра лишь узкому

кругу лиц. Вы спросите, а зачем здесь ограничитель времени? А вы подумайте. За 5 секунд ваши друзья успеют только вдоволь посмеяться над фотографией, где вы мирно спите в костюме Адама с соленым огурцом в руке под новогодней елкой, а за минуту—две они могут успокоиться и скопировать смешную картинку к себе на компьютер, и пойдет она гулять по Интернету в платных эротических рассылках. Чего это вы там прячете? Да не нужны мне ваши новогодние фотографии! Вы лучше код скрипта запишите:

```

<body onLoad="begintimer()">
<script>
<!--
var limit="1:00"
if (document.images){
var parselimit=limit.split(":");
parselimit=parselimit[0]*60+parselimit[1]*1
}
function begintimer(){
if (!document.images)
return
if (parselimit==1)
window.location = "http://
www.page.ru/timeout.html"
else{
parselimit-=1
curmin=Math.floor(parselimit/60)
cursec=parselimit%60
if (curmin!=0)
curtime=curmin+" минут(ы) и
"+cursec+" секунд(ы) осталось"
else
curtime=cursec+" секунд(ы)"
window.status=curtime
setTimeout("begintimer()",1000)
}
}
//—>
</script>

```

В данном примере лимит пребывания на странице — одна минута (var limit="1:00"), после чего браузер перенаправит посетителя по адресу http://www.page.ru/timeout.html. Причем в случае попытки нажать кнопку браузера "Назад" скрипт блокирует переход, и браузер снова загружает указанную страницу. Код размещается в разделе BODY. Необходимо помнить, что диапазон указания секунд до редиректа должен быть от 0 до 59.

Куда поедет, барин?

Следующий пример позволяет автоматически перенаправлять посетителя вашего сайта на любой другой адрес, указанный в строке запроса. Вот, собственно, сам скрипт:

```
<script language="JavaScript">
<!--
function GoToURL(j)
{
window.status=('Попытка соеди-
нения...')
var URLis;
        URLis           =
document.URLframe.Destination.value
if (URLis == "" || URLis.length <=
8)
{
j.value = "Попробуйте снова"
alert('\Опаньки!\n\Нет свя-
зи');
window.status=('Ох уж эти
провайдеры!...')
}
else
{
j.value = "Идет попытка соедине-
ния с http://" + URLis
var location=("http://" + URLis);
this.location.href = location;
window.status=('Идет попытка со-
единения с ' + URLis + ' Извольте
обождать, барин...');
}
}
//-->
</script>
```

Поскольку посетитель сам будет вводить адрес, необходимо на стра-

ные значения (о них чуть позже). Вот примерный листинг для формы ввода адреса:

```
<form name="URLframe">
<strong>&nbsp;&nbsp;&nbsp;<input type="text"
name="Destination" size="45"
maxlength="50">&nbsp;&nbsp;&nbsp;
<input type="button" value="Пе-
рейти" onclick="GoToURL(this)">
</form>
```

Обращаю особое внимание на значение "URLframe" атрибута NAME тега FORM, значение "Destination" атрибута NAME первого тега INPUT (строка запроса адреса) и значение onclick="GoToURL(this)" второго тега INPUT (кнопка), инициирующие запуск скрипта.

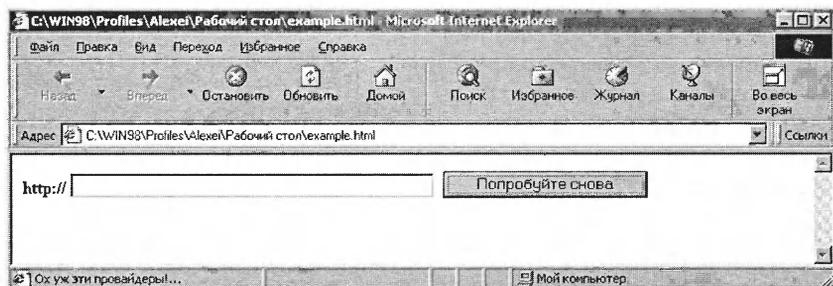
Если у вас нет соединения с Интернетом (или на момент запуска скрипта оно было нарушено), то браузер выдаст сообщение об ошибке (в данном примере это "Опаньки! Нет связи"), после чего надпись на кнопке подтверждения "Перейти" будет заменена другой — "Попробуйте снова", а в статусной строке можно увидеть "Ох уж эти провайдеры!...". Если же связь с Интернетом есть, статусная строка выдаст "Идет попытка соединения с ..." (указанный адрес). Извольте обожждать, барин...". Разумеется, все эти значения могут быть легко изменены на другие, по вашему мнению, более подходящие.

Покажите ваши документы!

Если на вашем сайте есть какая-нибудь форма для ввода пользова-

тинга, у вас нет. Что же делать? Оказывается, выход есть, и теперь, пока посетитель не укажет все требуемые данные, кнопка отправки просто-напросто не покажется! Реализуется этот прием при помощи Dynamic HTML и JavaScript. Применение первой технологии знающих людей должно было насторожить: известно, в каких отношениях динамический HTML и Netscape Navigator. И действительно: к сожалению, трюк не срабатывает даже в последних версиях этого браузера. Тем не менее, он может заинтересовать тех, кто делает по две копии сайта, для разных браузеров. Код скрипта выглядит так (размещается в разделе HEAD):

```
<script>
<!--
var iscomplete=false
function checkform()
{
iscomplete=true
for
(i=0;i<=document.myform.elements.length-
1;i++) {
i f
(document.myform.firstname.value=="")
{iscomplete=false}
if
(document.myform.lastname.value=="")
{iscomplete=false}
if
(document.myform.email.value=="")
{iscomplete=false}
if
(document.myform.homepage.value=="")
{iscomplete=false}
if
(document.myform.city.value=="")
{iscomplete=false}
}
if (!iscomplete) {
window.status="Пожалуйста, за-
полните все пункты"
if (document.all) {
submitButton.style.visibility="HIDDEN"
iscomplete=false
}
if (document.layers) {
document.myform.submitbut.value="Еще
нет"
iscomplete=false
}
}
if (iscomplete) {
```



ницу (тоже в раздел BODY, как и код сценария) поместить специальную форму, взаимосвязанную с кодом скрипта. Вы вправе задавать абсолютно любые значения атрибутов тега FORM, однако есть и обязатель-

тельской информации, бывает нужно проконтролировать полное заполнение всех граф. А доступа на запуск CGI- или PHP-скриптов на сервере, предоставляющем вам хо-

```

window.status="Вы заполнили
все графы и теперь можете отпра-
вить свою информацию"
if (document.all) {
submitbutton.style.visibility="VISIBLE"
iscomplete=true
}
if (document.layers) {
document.myform.submitbut.value
="Теперь"
iscomplete=true
}
}
var timer=
setTimeout("checkform()",200)
}
function oksubmit() {
return iscomplete
}
</script>

```

Далее необходимо в тег BODY вписать `onload="checkform()",` чтобы скрипт запускался при загрузке документа. Как видно из листинга, скрипт контролирует занесение данных в 5 пунктов: имя, фамилия, e-mail, адрес сайта (WWW) и город:

```

if (document.myform.
firstname.value=="")
{iscomplete= false}
и т.д.

```

Аналогичным способом можно вписывать сколько угодно строк, по разным параметрам. Атрибуты формы настраиваются в индивидуальном порядке, однако следует помнить о значениях атрибута NAME тегов INPUT, которые соответствуют условиям выполнения скрипта, событию `onsubmit` и обязательному указанию свойства `HIDDEN` (скрытый) для блока, в котором после заполнения всех граф должна будет появиться кнопка подтверждения (`Submit`).

Полностью коды всех описанных примеров можно найти и загрузить на сайте автора по адресу <http://www.alpet.spb.ru> в разделе "Статьи".

Надеемся, что вас заинтересовали предложенные в этом выпуске примеры. Если вы хотите узнать, как реализовать тот или иной прием, или желаете высказать свое мнение по поводу начатого цикла и нашего журнала вообще — милости просим: высказывайтесь в форуме "Магии ПК" (<http://www.magicpc.spb.ru/forum>).

Тормоза электронной коммерции

Судя по последним событиям, торговля в Сети несколько притормаживает, впрочем, как и рекламная деятельность...

Рекламность падает

В 2001 году сократится рост поступлений от рекламы в европейском Интернете, сообщает Yahoo! Actulaites. Аналитики полагают, что он составит всего 20%, в то время как в 2000 году достиг 100%. Это связано со снижением цен на рекламу, сокращением темпов роста "новой экономики", а также сомнениями рекламодателей в эффективности интернет-рекламы. По мнению Томаса Рашке, аналитика компании IDC, в 2001 году объем рекламы составит около 1 млрд евро, что будет означать рост в размере 20% по сравнению с прошлым годом. Интересно отметить, что еще в октябре прошлого года тот же г-н Рашке прогнозировал на 2001 год объем рекламы в 1,4 млрд евро. Сотрудник Jupiter MMXI Стаффан Энгдегард сообщил, что прошлой осенью стоимость рекламной кампании в Интернете снизилась в среднем на 30%.

Рекламодатели все больше сомневаются в эффективности рекламных баннеров и отдают предпочтение спонсорству и электронной почтовой рассылке. Баннеры составляют в настоящий момент 65% всей рекламы, но эта цифра, по мнению экспертов, будет сокращаться. В скандинавских странах она уже не превышает 56%, а в США — 46%.

Обсчитались

Как показало исследование, проведенное Pegasus Research International, около трети интернет-компаний могут разориться до конца 2001 года. Исследование было проведено среди 335 предприятий, акции которых котируются на NASDAQ.

В третьем квартале 2000 года затраты интернет-предприятий достигли 2 млрд долларов. Результаты исследования показали, что превышение затрат над доходами станет основной трудностью, выпавшей на долю электронных предприятий.

Кому верить?

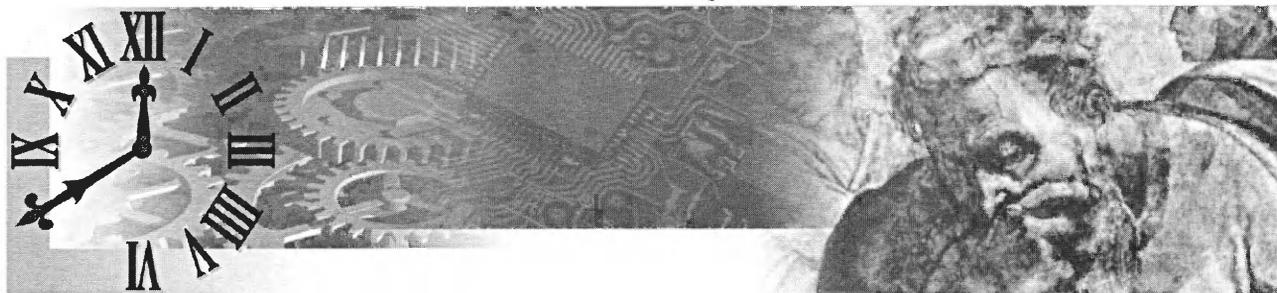
Онлайновые инвесторы заявляют о своей неудовлетворенности качеством финансовой информации, предоставляемой различными сайтами, сообщает CNET. Многие пользователи финансовых сайтов высказывают мнение о том, что при наличии альтернативы они поменяли бы источник информации. При этом большую часть финансовой информации сайты получают от онлайновых брокеров, таких как E*Trade и Ameritrade. Многие мелкие инвесторы разорились, полагаясь исключительно на информацию, предоставляемую в Сети. Более крупные инвесторы в своей деятельности ориентируются не только на Интернет, они используют также аналитические и обзорные отчеты.

Волков бояться...

Автомобильные компании не спешат заняться коммерцией в Сети, — таковы результаты исследования, проведенного консалтинговой компанией KPMG. Согласно отчету, не только автомобильные компании, но и поставщики комплектующих пока не решаются участвовать в интернет-коммерции, так как это неизбежно потребует от них раскрытия конкурентам определенной информации. Согласно отчету, наиболее важной причиной, удерживающей компании от выхода в Интернет, по-прежнему остается безопасность онлайн-делов.

Дмитрий Добрый

Продолжение. Начало см. "Магия ПК" №1/2000



Сто выдающихся деятелей IT ушедшего столетия

Бернерс-Ли Тим (Berners-Lee, Tim р. 1956)

Выпускник Оксфордского университета (Англия), работая в Европейской лаборатории ядерной физики (CERN) в Женеве, в марте 1989 г. предложил руководству концепцию новой распределенной

информационной системы, построенной на принципах гипертекста. Эта система должна была объединить все информационные ресурсы CERN (базы данных, отчеты, компьютерная документация, списки почтовых адресов, реферативная система и т. п.), а гипертекстовая технология — позволить легко "перепрыгивать" из одного документа в другой. В 1990 г. эти предложения были приняты, и проект стартовал. В том же году под его руководством с использованием других его инноваций — URL, HTML и HTTP, — был разработан прототип для World Wide Web (WWW). Журнал "PC Magazine" по итогам 1998 г. назвал его обладателем приза "Выдающиеся достижения" как человека, внесшего наибольший вклад в развитие IT. В настоящее время он директор организации World Wide Web Consortium (W3C), которая стремится реализовать полный потенциал Интернета, гарантировать его стабильность и способность развиваться.



Берри Клиффорд (Berry, Clifford Edward, 1918—1963)



Уроженец Нью-Йорка, будучи аспирантом Университета штата Айова, вместе с Джоном Атанасовым работал над созданием первой ламповой ЭВМ.

Бонч-Бруевич Михаил Александрович (1888—1940)

Российский ученый, один из пионеров радиотехники, член-корреспондент АН СССР (1931). В 1918 г. изобрел ламповое реле и организовал первое отечественное производство электронных ламп. В 1919 г. независимо от него такой же прибор изобрели англичане У. Икклз и Ф. Джордан и назвали его триггер. Устройство сыграло большую роль в развитии компьютерной техники.



Бриклин Дэн (Bricklin, Dan, р. 1951)

Выпускник Массачусетского технологического института, работая в компании Software Arts, вместе с Бобом Френкстоном в 1979 г. разработал первый пакет деловых программ VisiCalc (Visible Calculator, видимый калькулятор) для пер-



вых персональных компьютеров IBM PC и Apple II. Бриклин разработал интерфейс, а Френкстон написал код. Пакет, содержащий также электронные таблицы, во многом способствовал распространению домашних ПК, поскольку не требовал от пользователя специальной подготовки. Общество Computer Society института IEEE присудило Бриклину награду Computer Entrepreneur Award (За предпринимательство в компьютерной отрасли). В настоящее время он занимает пост директора по технологии компании Trellix.

Брук Исаак Семенович (1902—1974)

Разработчик ряда первых малых и средних отечественных ЭВМ, автор концепции создания малогабаритных специализированных и управляющих машин. Первая из них М-1 создана в 1952 г. (работы начаты в 1950 г.). Содержала 730 электронных ламп, рулонный телетайп. Впервые имела двухадресную систему команд. Производительность 15—20 операций в секунду. Память 256 слов. М-1 положила начало серии машин М-2, М-3, М-4(М) и др. С 1957 по 1964 год Брук — директор Института электронных управляющих машин АН СССР (ИНЭУМ). Его проблемная записка, опубликованная АН СССР в



1958 г., дала толчок к организации в стране в конце 50-х годов целого ряда НИИ и КБ по разработке универсальных и специализированных ЭВМ. Опубликовал более 100 научных работ, получил более 50 авторских свидетельств на изобретения.

Брусенцов Николай Петрович (р. 1925)



Разработчик отечественной ЭВМ "Сетунь" (1959), использующей троичную систему счисления. Окончил радиотехнический факультет МЭИ (1952). Работу над ЭВМ начал в середине 50-х годов, а первый экземпляр

был готов к концу 1958 года. Своей простотой и практичностью "Сетунь" обязана представлению чисел и команд в симметричном коде: -1, 0, 1. По сути ЭВМ представляет собой первый RISC-компьютер: длина машинного слова 9 тритов (трит — трехстабильный элемент памяти), всего 24 команды. Магнитный барабан был связан с быстрой оперативной памятью постраничным обменом, что дало своего рода кэш, повысивший производительность ЭВМ. "Сетунь" выпускалась серийно в Казани, по 15—20 машин в год. За пять лет было выпущено 50 машин. Модель "Сетунь-70" закончена в 1968 году. Убежденный в том, что "истинный RISC может быть только троичным", в этой ЭВМ Брусенцов объединил принципы эффективной архитектуры на минимальном наборе команд, трехзначную логику, троичный код и идеи структурного программирования. Однако после завершения работ лаборатории Брусенцова была вынуждена прекратить разработки ЭВМ.

Бурцев Всеволод Сергеевич (р. 1927)



Руководитель и разработчик суперкомпьютеров и вычислительных комплексов для систем управления реального времени. Выпускник МЭИ (1951), доктор технических наук

(1962), член-корр. АН СССР, с 1998 г. — советник президента Российской АН. При создании ЭВМ М-40 впервые предложил аппаратные принципы распараллеливания вычислений: все основные устройства машины (арифметическое, управления, ОЗУ, управления внешней памятью и др.) имели автономные системы управления и работали параллельно. Впервые был использован принцип мультиплексного канала, благодаря чему без замедления вычислений удалось осуществить прием и выдачу информации с десяти асинхронно работающих направлений общей пропускной способностью 1 млн бит/с. В ходе создания МК "Эльбрус-2" по его инициативе созданы новые быстродействующие интегральные схемы, высокочастотные групповые разъемы, многокристалльные и большие интегральные схемы, микрокабели, прецизионные многослойные печатные платы. В последнее время предложил структуру суперЭВМ, основанную на новом, не фон-неймановском принципе. Эта архитектура обеспечивает существенное распараллеливание вычислений на аппаратном уровне, использует новейшие принципы оптической обработки информации, обладает высокой регулярностью структуры и позволяет достичь высочайшей производительности.

Буш Вэннивер (Bush, Vannevar, 1890—1974)



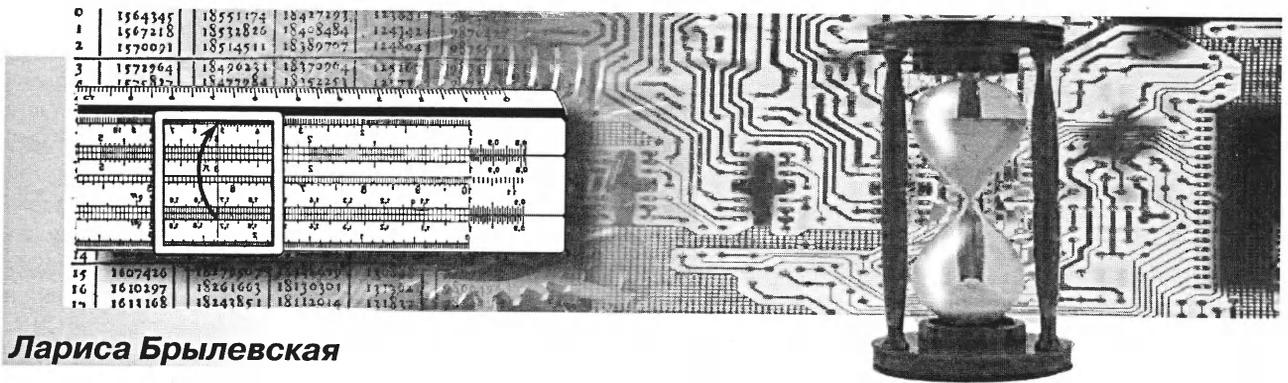
Американский инженер и математик. По окончании колледжа Тафтса (1913), получив степени бакалавра и магистра, работал в General Electric в отделе тестирования электрооборудования. В 1914—1915 гг. преподавал математику в колледже Тафтса. В 1916—1917 гг. получил дипломы инженера в Гарвардском университете и в Массачусетском технологическом институте (MIT), стал доцентом электротехники в колледже Тафтса, а в 1919 г. вернулся в MIT в качестве доцента по курсу "Передача электроэнергии". Во время первой мировой войны служил в ВМС США. С 1923 г. профессор, а с

1932 г. вице-президент MIT и декан его Технической школы. В 1938 г. избран президентом Института Карнеги. В 1940 г. назначен председателем Государственного комитета оборонных исследований при президенте США. В 1939—1941 гг. возглавлял Национальный наблюдательный совет по авионавигации, а в 1941—1947 гг. — бюро научных исследований и разработок при правительстве США. В 1942 г. под руководством Буша разработан один из первых электронно-механических анализаторов. В октябре 1945 в статье "As We May Think" ("Пока мы мыслим", журнал "The Atlantic Monthly") впервые сформулировал идею гипертекста. С 1946 г. председатель Объединенного совета по исследованиям и разработкам военного и морского ведомств. В 1947—1948 гг. — председатель совета по разработкам Национального управления военными учреждениями, одновременно возглавлял корпорацию AT&T, а в 1948 г. — Merck & Co. В 1950 г. организовал и возглавил Национальный фонд науки США (NSF, совмещает функции академии наук и министерства науки и технологии). С 1957 г. президент, а с 1959 г. — почетный президент MIT.

Бэкус Джон (Backus, John W., р. 1924)



Математик из Филадельфии, по окончании университета (1949) — программист компании IBM. Разработал для ЭВМ IBM 701 технологию "ускоренного кодирования", позволившую заметно упростить написание программ. В 1954—1957 гг. возглавлял исследовательскую группу IBM, создававшую первый машинно-независимый язык программирования Фортран. Первая версия компилятора состояла из 25000 строк машинного кода. В 1959 г. разработал правила для описания синтаксиса языков высокого уровня — форму Бэкуса-Наура. Спустя год участвовал в разработке другого языка высокого уровня — Алгол-60. Награжден Национальной медалью за вклад в науку (1976).



Лариса Брылевская

Джон Непер: 450 лет В истории науки

На рубеже тысячелетий мы отмечаем 500-летие со дня рождения выдающегося шотландского математика Джона Непера (1550—1617), оставившего значительный след в истории развития вычислительных средств.

Непер принадлежал к древнему и знатному шотландскому роду Ленноксов. Король, восхищенный воинской доблестью Дональда Леннокса, одного из предков Непера, воскликнул: "Donald — на реег", то есть "Дональд — несравненный". Впоследствии это прозвище стало фамилией. Однако Джон Непер не пошел по стопам предков, его больше привлекали наука и изобретательство.

Джон получил хорошее религиозное воспитание, которым был обязан своему дяде, епископу Оркнейскому. В 1563 году он поступил в колледж св. Сальватора университета Сент-Эндрюса, затем продолжил образование во Франции, Италии и Нидерландах, где познакомился с астрономическими и тригонометрическими трудами европейских ученых.

Круг его интересов был довольно широк: от изобретений различных механических приборов до тонких вопросов богословия. В 1593 году ученый опубликовал богословский труд "Простое объяснение всего откровения Святого Иоанна", в котором пытался проникнуть в тайну про-

рочеств Даниила, основываясь на числовой мистике. Это сочинение было весьма популярно и не раз переводилось на европейские языки.



Джон Непер

Невероятная, по мнению современников, ученость Непера, его замкнутость и странность некоторых привычек вызвали подозрения окружающих. Многие считали его чернокнижником. Зная об этом, Непер и сам иногда подогревал такие домыслы, устраивая "мистические" розыгрыши. Далеко не всякий мог позволить себе такие шутки в то время. От несправедливых обвинений Непера спасала лишь известность его богословских трудов и репутация добропорядочного протестанта.

Заинтересовавшись тригонометрией, он вывел несколько формул для решения сферических треугольников. Но больше его привлекли вопросы совершенствования способов вычислений и построения таблиц. В начале XVII века это был один из наиболее актуальных вопросов. Развитие мореплавания требовало более точных и полных астрономических таблиц, государственные служащие были вынуждены проводить долгое время за рутинными вычислениями, используя таблицы умножения, процентов, сложных процентов и т.д. Впоследствии Непер писал: "Я всегда старался, насколько позволяли мои силы и способности, избавиться от трудностей и скуки вычислений, докучливости которых обыкновенно отпугивает очень многих от изучения математики". Эти слова можно считать девизом всей его жизни.

Изобретение логарифмов

Непер известен прежде всего как изобретатель логарифмов, доставивших столько горя школярям всех времен. Идея логарифма возникла на основе сопоставления арифметической и геометрической прогрессий:

0, 1, 2, 3, 4, 5, 6, ...

1, a, a², a³, a⁴, a⁵, a⁶, ...

Еще до Непера было известно, что сложение членов арифметичес-

кой прогрессии соответствует умножению членов геометрической прогрессии, вычитание в арифметической прогрессии — делению в геометрической. На основе этого принципа можно было более сложные операции умножения и деления сводить к простому сложению и вычитанию. Непер образовал термин "логарифм" от двух греческих слов: "логос" — соотношение, соответствие и "аритмос" — количество, число.

Этой работе Непер посвятил около 20 лет своей жизни. Таблица Непера была опубликована в 1614 году в Эдинбурге. Ученый назвал свою работу "Описание удивительной таблицы логарифмов". Неперовы логарифмы были действительно удивительны, они имели огромное значение для астрономии, прикладной математики и механики. Его таблицы были не только более точными и полными по сравнению с таблицами шестнадцативенников, в их устройстве нашла выражение идея непрерывности, послужившая основой развития исчисления бесконечно малых величин, из которого выросло дифференциальное и интегральное исчисление. На основе логарифмического принципа позднее был создан ряд вычислительных приборов — логарифмических кругов и линеек (см. "Магия ПК" №11/1998).

Непер, много лет жизни отдавший созданию таблиц логарифмов, хорошо понимал достоинства и недостатки этого средства вычислений. Наряду

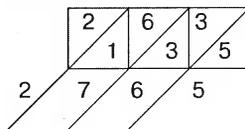
с логарифмическими таблицами он разработал несколько более доступных простому обывателю приборов. Это, прежде всего, различные виды палочек для умножения, впоследствии получившие название палочек Непера, и доска для вычислений в двоичной системе счисления.

Палочки Непера

Описание этого довольно простого прибора для умножения, деления, извлечения квадратного корня Непер дал в своей последней прижизненной публикации "Rabdologia seu Numerationis" в 1617 году. Едва ли он придавал большое значение своему изобретению. В предисловии к этой книге он отметил, что опубликовал ее лишь по настоянию друзей.

Термин "рабдология" образован от греческого "рабдос" — палка, прут. Так Непер назвал счет с помощью палочек.

Прибор представлял собой таблицу умножения, записанную в виде таблицы Пифагора на десяти подвижных линейках. Каждая клетка для записи произведения разделена диагональю на две части. В верхней записывают десятки, а в нижней — единицы. Прибор основан на способе умножения решеткой, которым пользовались при умножении на бумаге. Умножим 395 на 7. Для этого переместим палочки так, чтобы по горизонтали в нулевой строке получить число 395. По 7 строке, в соответствующих клетках, имеем числа: 21, 63, 35. Чтобы получить результат, мысленно продолжим диагонали квадратов и запишем суммы цифр, стоящих между двумя соседними диагоналями. Получаем 2765.



Идея палочек Непера оказалась очень плодотворной и неоднократно использовалась в конструкциях не только множительных, но и суммирующих машин. Усовершенствованием и развитием этой идеи занимались многие. Своего рода бум эта конструкция пережила в XIX веке. Ею зани-

мались французы Женай, Люка, Прюво де Гюе, а также наши соотечественники — Слонимский, Иоффе, Дьяков и др.

0	1	2	3	4	5	6	7	8	9	%
1	0/1	0/2	0/3	0/4	0/5	0/6	0/7	0/8	0/9	%
2	0/2	0/4	0/6	0/8	1/0	1/2	1/4	1/6	1/8	%
3	0/3	0/6	0/9	1/2	1/5	1/8	2/1	2/4	2/7	%
4	0/4	0/8	1/2	1/6	2/0	2/4	2/8	3/2	3/6	%
5	0/5	1/0	1/5	2/0	2/5	3/0	3/5	4/0	4/5	%
6	0/6	1/2	1/8	2/4	3/0	3/6	4/2	4/8	5/4	%
7	0/7	1/4	2/1	2/8	3/5	4/2	4/9	5/6	6/3	%
8	0/8	1/6	2/4	3/2	4/0	4/6	5/6	6/4	7/2	%
9	0/9	1/8	2/7	3/6	4/5	5/4	6/3	7/2	8/1	%

Палочки Непера

Счетная доска

В одном из приложений к "Rabdologia seu Numerationis" Непер описал конструкцию удивительной счетной доски, предназначенной для умножения, деления и извлечения квадратного корня в двоичной системе счисления. Понятно, что вряд ли кто-либо в XVII веке пользовался этим прибором на практике. Для двоичной системы счисления Непер разработал особый "алфавит", обозначив каждую степень 2 определенной буквой. Числа сначала нужно было перевести в двоичную систему, выполнить необходимые выкладки с помощью доски, затем перевести в десятичную систему. Выполнение арифметических действий с натуральными числами приводило к очень громоздким операциям, но поразительно то, что в те далекие времена Непер смог понять все преимущества двоичной системы счисления и впервые предложил использовать ее в практике вычислений.

Вычислительные приборы Непера очень просты по конструкции, они не несут в себе никаких революционных новшеств для своего времени. Ученому, казалось, просто удавалось реализовать идеи, уже носившиеся в воздухе, но эта реализация была столь талантлива, что идеи Непера сохраняли свою актуальность около 400 лет!

Gr.	9				
mm	Sinus	Logarithmi	Differentia	Logarithmi	Sinus
0	1564346	18511724	18127203	143381	9874383
1	1567115	18512826	18408484	124342	9876427
2	1570001	18514511	18589207	114804	9878527
3	1572964	18496231	18370064	125167	9875514
4	1575837	18477984	18152253	125731	9875056
5	1578709	18459722	18133576	126196	9874599
6	1581581	18441594	18114933	126661	9874137
7	1584451	18423511	18096347	127127	9873679
8	1587322	18405341	18077727	127594	9873216
9	1590197	18387265	18059103	128062	9872754
10	1593069	18369223	18040692	128531	9872291
11	1595941	18351214	18022213	129001	9871827
12	1598812	18333237	18003765	129472	9871362
13	1601684	18315294	18085351	129943	9870897
14	1604555	18297351	18066959	130415	9870431
15	1607426	18279507	18048610	130888	9869964
16	1610297	18261663	18030301	131362	9869496
17	1613168	18243851	18012014	131837	9869027
18	1616038	18226071	18003758	132313	9868557
19	1618909	18208323	18005533	132790	9868087
20	1621779	18190606	18007328	133268	9867616
21	1624649	18172924	18009127	133747	9867144
22	1627519	18155271	18010927	134226	9866671
23	1630389	18137654	18002948	134706	9866197
24	1633259	18120067	18004930	135187	9865722
25	1636129	18102511	17996842	135669	9865246
26	1638999	18084987	17998835	136152	9864770
27	1641868	18067495	17990859	136636	9864293
28	1644738	18050034	17982913	137121	9863815
29	1647607	18032604	17974997	137607	9863336
30	1650476	18015207	17967114	138093	9862856

Часть страницы из таблицы Непера

Сигнал будильника застал меня врасплох. Пытаюсь разомкнуть веки и снова проваливаюсь в сон. Но уже включилось сознание, которое решительно пресекает порочную слабость.

К счастью, головной боли нет: на этот раз я всего-навсего переработал, желая помочь давнему заказчику и отнюдь не забывая о немалой премии за сверхурочные. Ремонт и реанимация компьютеров уже много лет обеспечивают меня хлебом с маслом. Под душем я окончательно просыпаюсь. У меня еще достаточно времени, чтобы приготовить себе чашку свежего кофе.

Просматривая, уже на выходе, сообщения информариума, я встречаю напоминание: "Не забыть про ежегодный апгрейд". Если вы подумали, что подтверждение сертификата профессиональной пригодности для меня — вопрос жизни, то ошиблись. Вряд ли кому из моих клиентов вздумается потребовать от меня эту казенную бумагу, чтобы решить, можно ли со мной работать. Просто вопрос самоуважения. Значит, сегодня надо идти в Центр информационных технологий. Стационар там экстра-класса, многие из знакомых ищут блат, чтобы там апгрейдиться.

Едва не спотыкаясь о системный блок Юркиного компьютера — соседский сын вчера снова угробил свой винчестер. Дождавшись, когда я вернусь, поздно вечером он приволок ко мне железный ящик. Я обещал восстановить и после его ухода сделал за полчаса. Этот парень — уникал, хотя в свои пятнадцать лет успел всем попортить крови... Год назад сорванец был оставлен на полдня под моим присмотром. Пока я следил, как бы он не залез в мой компьютер, паршивец обнаружил на другом столе пару надписанных, но не заклеенных конвертов с письмами в редакции двух журналов. Как только я на миг отвернулся, он поменял местами содержимое конвертов. В результате редактор отнюдь не технического издания прочел статью о некоторых тонкостях ремонта компьютера, а второе письмо получила благовоспитанная особа, редактор ком-



Сергей Вардин

Профессиональная пригодность

(заметки из 2005 года)

пьютерного журнала. Я с ужасом представлял, как эта чопорная уса-тая дама с бюстом, подобным башне главного калибра линкора "Айова", утопая в сигаретном дыму, читает фривольную любовную историю из тех, что мне очень неплохо оплачивали, — ну, вам ясно, где именно!

Однако ребенок подрост и проявил незаурядные способности к программированию. Его импульсивная натура мешала получить сколько-нибудь систематизированные знания, но и того, что он вычитал в книгах и случайно подвернувшейся документации, хватило с лихвой. Его отец занимает солидную должность в московском руководстве крупнейшей американской софтверной корпорации. И когда ее глава, известный на весь мир миллиардер, приехал в Россию на празднование юбилея своего детища, его ожидал сюрприз. Наш вундеркинд, подглядывая у отца пароль для входа в корпоративную сеть, влез кое-куда со своего домашнего компьютера и сумел перепрограммировать графические процессоры. Но как перепрограммировал! Когда Великий Глава с бокалом в руке открывал торжество, через корпоративную сеть прямая трансляция шла на весь мир. И в этот момент его лицо на экранах вдруг дернулось, исказилось и пре-

вратилось в физиономию знаменитого чернокожего баскетболиста, которая продолжала говорить с сохранением мимики и даже артикуляции оригинала. Представьте, какой получился скандал! Но, к чести Великого, он вышел из конфуза, ловко использовав запись эпизода для дополнительной рекламы. А отцу пообещал, что как только чадо научится управлять своими порывами, возьмет его в исследовательский отдел корпорации...

Клиника ЦИТа действительно впечатляет. Ничего похожего на ту обшарпанную тесную лабораторию, в которой мне приходилось апгрейдиться раньше. Правда, тамошний персонал — мои хорошие знакомые, несчетное число раз я реанимировал их старенькие "пентиумы". А здесь — "Делл" последней модели с процессором в два гигагерца и базовой памятью три гигабайта. Этот "сундук" как раз выплевывает только что прочитанную карточку с моими психофизиологическими параметрами. Я возвращаю карточку в бумажник и ожидаю, когда медсестра с миленьким личиком отсканирует заполненную мною анкету. Она усаживает меня в кресло, похожее на зубо-врачебное, и берет из лотка выбранный компьютером коробку с лазерным диском, соответствующим

указанным в анкете коду профессии и дате последнего апгрейда. Сигнал телефона отвлекает сестру.

Незаметно наблюдая за ней, я вижу, что разговор переходит на выяснение отношений. Она готова расплакаться. Я деликатно перевожу взгляд на коробку, лежащую на столике, читаю этикетку и второй раз за этот день вспоминаю соседского шалопая Юрку. После того разговора с шефом Юркин отец бросился искать психотерапевта или просто воспитателя, но такого, что смог бы совладать с неуправляемым вундеркиндом. И тогда один из моих клиентов пообещал устроить парня к профессору Забловскому.

Профессор был педагогом мирового уровня, работал исключительно с "трудными" детьми, возвращая обществу заядлых хулиганов, наркоманов или потенциальных убийц, причем не знал неудач, используя собственную методику и начисто игнорируя все чудеса информационной технологии, обходясь только словами. Увы, накануне профессор скоропостижно скончался, никому не успев передать свою чудодейственную "систему". Он просто-напросто не нашел никого, кто пожелал бы изучать методы и приемы воспитания и психотерапии, в которых вообще не требуется компьютер. И вот теперь я вижу, что Забловский, оказывается, успел записать мнемोगрамму своего метода. Так что есть надежда: лет эдак через сто, когда человечество наиграется с компьютерами, может, и появится такой чудак, что захочет лечить и исправлять людей только с помощью слов.

Медсестра, наконец, закончила разговор и направилась к зеркалу, положив мой диск на столик. Ликвидация последствий бурного объяснения заняла еще минут десять. Наконец девушка завершила аварийно-восстановительные работы, лишь припухлость век выдавала личную драму. Она заученным движением опустила мнемощлем на мою голову, взяла диск и вставила его в лоток психоинформационного модуля. Электросон, которым сопровождается эта процедура, действует на меня очень благотворно...

Отперев дверь своей квартиры, я почувствовал смутную тревогу. Что-то неуловимо изменилось. Нет, в прихожей и в комнатах все на своих местах, я не заметил каких-либо перемен. Привычно заурчал, пробудившись при моем возвращении, компьютер, информариум выложил лист с отчетом за время моего отсутствия. Я сажусь за стол, и вдруг до меня доходит: я не знаю, что должен делать. Я понятия не имею, для чего у меня компьютер и куча непонятого оборудования на полках. Наконец, я просто не знаю, кем работаю!

Вот так. Вы уже наверняка поняли, в чем дело. Я же честно признаюсь, что затратил на это гораздо больше тех секунд, что потребовались вам. И немало дней, чтобы хоть как-то справиться, смириться со своей бедой. Ведь практически весь накопленный десятилетиями багаж моих профессиональных знаний, опыта и навыков оказался "затерт" школой реабилитации асоциальных детей и подростков профессора Забловского.

Ничего не поделаешь, надо жить дальше. Да и окружающие проявили такт, понимание и поддержку. Даже разбойник Юрка, узнав про мою беду, заглянул на минутку, чтобы, сочувственно посопев, предложить помощь. Я привычно подавил вздох, как-то по-новому разглядывая парнишку.

"Лабильный тип, возбудимая акцентуация характера, умеренно инфантилен...". В голове вдруг воцарились спокойствие и уверенность. Я поманил его в кабинет, взял с полки кубик Рубика — игрушку, когда-то очень популярную и незаслуженно, на мой взгляд, забытую, — привел в полный беспорядок его грани и протянул Юрке:

— Сможешь собрать?

Он презрительно хмыкнул:

— Подумаешь!

После двухчасовых мучений он смог, наконец, привести в порядок одну грань. Я приказал, как собрать вторую, и снова перемешал кубик.

В общем, вышел он из моего кабинета через четыре месяца. Нет, разумеется, он уходил домой обедать и ночевал тоже дома, а все ос-

тальное время проводил у меня. Еще через месяц он был принят на должность инженера-стажера в корпорацию — Великий и Могучий выполнил обещание. А счастливый отец двадцать пять часов в сутки всем и везде рассказывал, какое чудо я сотворил с его неуправляемым оболтусом.

Уже через неделю у меня набралась группа из десяти подростков, а вскоре очередь в мои группы растянулась на многие месяцы. Немалую часть составляли дети состоятельных и очень богатых родителей. Но я работаю не только за астрономические гонорары. Больше половины детей проходят мой курс бесплатно: ведь природа не выбирает, где появиться настоящему таланту — у хваткого миллионера или у туповатого недотепы-работяги. Вот только если упустить время, ребенок "из низов" сломается или погибнет в грязи и бурьяне, а сынок миллионера станет беситься с жиру и тоже плохо кончит.

Однажды у подъезда меня окликнул женский голос. Заливаясь краской, и оттого еще миловидней, ко мне обращается та самая медсестра из ЦИТ-овской клиники.

— Я так перед вами виновата! Простите, я слышала, чего вам стоила моя ошибка. — Я вспоминаю ту минуту, когда осознал происшедшую катастрофу, но вдруг замечаю новенькое обручальное кольцо на ее пальце и бравого молодца, который с деланно-безучастным видом маячит неподалеку. Ее счастливый вид также не очень соответствует чувству вины. И мне становится весело.

— Ничего, все оказалось гораздо лучше, чем могло быть. Но мне жаль, что вас уволили.

— Да, уволили, но зато я вышла замуж.

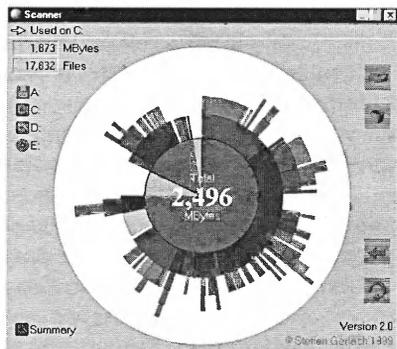
Я вдруг понимаю, что совершенно искренне желаю ей счастья. Светила психиатрии и нейроинформатики уверяют меня, что "базовые навыки", относящиеся к моей прежней профессии, со временем обязательно вернутся. Хотел бы я знать, какие из моих навыков "базовые" и сколько пройдет времени. А пока...

— Простите, вы не подскажете, каким боком вставлять вот эту, как ее, дискету?

Существуют специальные программы, основное назначение которых — помочь пользователю оценить реальный размер файлов и папок согласно количеству занимаемых ими кластеров, а не размер содержащихся в них данных. Кроме того, эти программы часто имеют еще и дополнительные функции наблюдения за диском, поиска файлов и др. Расскажу о некоторых из них.

Сканер диска

Программа Scanner, написанная Стефаном Герлачем, отображает в виде круговой диаграммы содержимое вашего жесткого диска, оценивая при этом реально занимаемое файлами место. Так, папка с 300 файлами по 10 Кб будет отображаться как имеющая размер 9600 Кб (если у вас кластер занимает 32 Кб), то есть занимающая почти 10 Мб. Можно оценить, какие файлы у вас занимают больше места, чтобы при недостатке свободного пространства на диске архивировать или удалять именно их.



Программа Scanner

Программа сделана очень удобно. Подведя курсор к любому сектору на диаграмме, можно узнать, какую папку он изображает. Щелкнув два раза на секторе, изображающем папку, вы перейдете к такой же диаграмме, построенной специально для этой папки и отображающей только ее содержимое. Для того, чтобы не загромождать рисунок, отображаются только файлы и папки, размер которых превышает определенную величину. Загрузить программу можно с адреса <http://www.steffengerlach.de/freeware/scn2.zip> или с сервера

Окончание. Начало см. "Магия ПК" №12/2000



Антон Орлов

Секреты файловой системы

www.freeware.ru по ссылке <ftp://ftp.freeware.ru/pub/mycomputer/system/scn2.zip>. Домашняя страница автора расположена на www.steffengerlach.de/freeware. Программа бесплатная.

Folder Size Shell Extensions

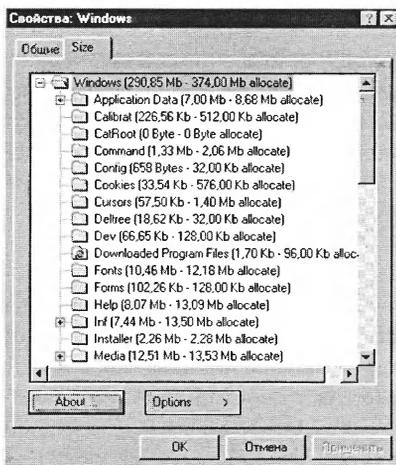
Программа Folder Size интегрируется в стандартное окно "Свойства папки" как дополнительная вкладка и при вызове выводит отчет о реальном месте на диске, занимаемом этой папкой и вложенными в нее папками с учетом всех занятых кластеров. Программа допускает различное представление информации о занимаемом папками месте —

от байт до гигабайт. Отчет программы можно вывести в отдельный файл, и тогда для каждой папки будет указан процент "потерянного места" — какую долю от объема реальных данных в папке занимает совокупный объем незаполненных фрагментов кластеров. Создана эта программа французом Лоренцом Бергером, а загрузить ее можно с адреса <http://filetools.nm.ru/dfolder.rar>. Программа бесплатная.

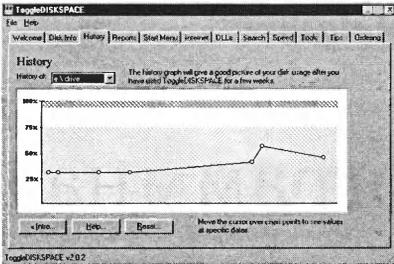
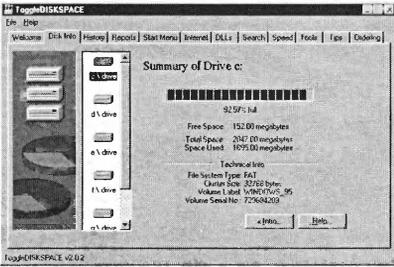
Toggle DiskSpace

Эта многофункциональная программа — универсальное средство для отслеживания изменений свободного места на жестком диске и выработки оптимальных способов размещения файлов на нем. Ее окно состоит из множества вкладок, и с каждой из них можно запустить встроенные в программу полезные функции.

Одна из таких функций, присущая только этой программе, — "История" свободного места на жестком диске. В день своего первого запуска программа записывает количество свободного места на каждом из логических дисков, а затем при каждом запуске отмечает точку на графике его изменения. Еще одна интересная особенность — это "Отчеты" о файлах на диске. Программа сканирует логический диск, а затем может выдать пользователю список, напри-



Удобно: вся папка — как на ладони

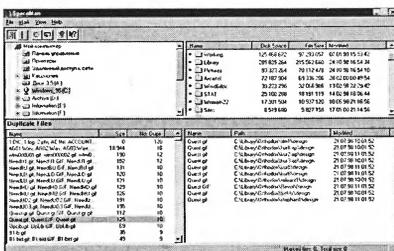


Основной экран и "История" Toggle DiskSpace

мер, самых больших или самых маленьких файлов на диске, самых новых или самых старых, всех картинок или архивов. Загрузить Toggle DiskSpace можно с сайта программы www.toggle.com или с адреса <http://filetools.nm.ru/tdiskspc.rar>. Распространяется по принципу Shareware.

SpaceMan

Эта программа тоже принадлежит к классу утилит для анализа дискового пространства и умеет рассчитывать объем реальных данных и объем занимаемого места на диске для папок и файлов. Но в отличие от других аналогов в ней имеется уникальная возможность поиска одинаковых файлов, причем не по названию, а по содержанию. Просканировав содержимое диска, SpaceMan ищет файлы, дублирующие друг друга, и отображает в окне их названия. Безусловно, отнюдь не все дубликаты файлов надо удалять. Многие из них — резервные копии файлов, и их,

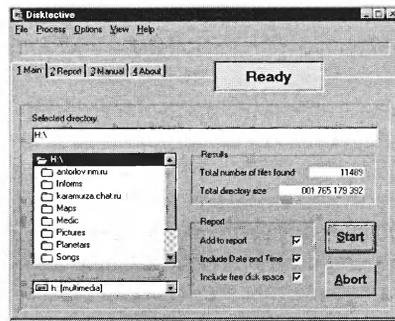


Поиск одинаковых файлов в SpaceMan — редчайшая функция!

наоборот, надо хранить особенно тщательно. Но все-таки при нехватке свободного места на диске результаты такого анализа могут дать полезную информацию. Загрузить SpaceMan можно с сайта программы <http://www.extrabit.com> или с адреса <http://filetools.nm.ru/spsman.rar>. Программа создана Марком Уильямсом из Хьюстона и распространяется по принципу Shareware.

Disktective

Бесплатная программа Disktective просто сканирует диск или отдельную директорию и выдает отчет о реальном месте, занимаемом файлами и папками. Ее преимущества — небольшой размер, отсутствие необходимости инсталляции и удобное строковое представление информации. По набору функций она похожа на Scanner Стефана Герлача, но представляет информацию в текстовом виде, в то время как Scanner — в графическом. Выберите, что вам удобнее. Создана эта программа датчанином Хагедорном из Амстердама, загрузить ее можно с адресов <http://www.simplex.nl/~hjh> и <http://filetools.nm.ru/diskdet.rar>.



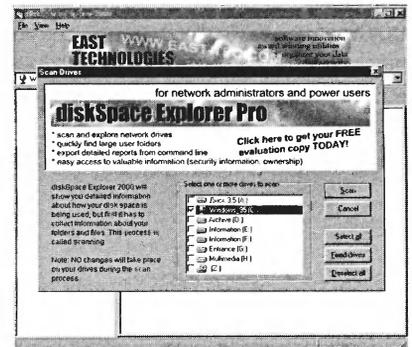
Программа Disktective

Disk Space Explorer

Это пожалуй, наиболее универсальная утилита из всех описанных в данном обзоре. Ее назначение — помочь вам оценить потери места на диске из-за несоответствия размеров файлов размерам кластеров и на основании такой оценки определить необходимость либо архивирования большого количества мелких файлов, либо преобразования диска в FAT32, если это даст необходимое увеличение свободного про-

странства на диске, либо попросту удаления тех файлов, которые реально занимают больше места.

Программа позволяет поэкспериментировать с размером кластера и посмотреть, сколько места та или иная папка будет занимать при том или ином размере кластера. В общем можно сказать, что она незаменима, когда места на жестком диске, как всегда, не хватает, и необходимо максимально использовать имеющиеся ресурсы, в частности, пустое место в кластерах, не полностью занятых файлами. Disk Space Explorer даст вам возможность оценить, где конкретно происходит "утечка" свободного места.



Disk Space Explorer

Написана программа в Румынии программистами Хоратиу Танеску и Евгением Малитой. Загрузить ее можно с адресов <http://www.easttec.com> и <http://filetools.nm.ru/dskspex.rar>. Программа бесплатная, но в рабочем окне постоянно показывает баннеры спонсоров проекта.

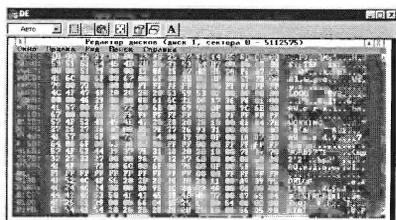
DiskEditor

Рассказ об утилитах для работы с жестким диском был бы неполным без упоминания о программе DiskEditor. Это редактор содержимого диска, разработанный нашими соотечественниками — компанией ФизТехСофт. Он позволяет просматривать содержимое диска не средствами операционной системы, а "на физическом уровне", то есть смотреть, что записано в каждой электронной ячейке диска, в каждом его секторе.

Основное назначение программы — помощь в восстановлении данных на жестком диске в случае поражения его вирусом или повреждения

служебной информации. Поэтому в ней возможны два варианта представления информации — шестнадцатеричный и текстовый. В случае печальной необходимости восстановления винчестера можно запустить DiskEditor с дискеты. Можно создать загрузочную дискету специально для DiskEditor, для чего в комплекте редактора есть специальная программа, устанавливающая на дискету сокращенную версию операционной системы PTS-DOS, которая позволяет загрузиться с этой дискеты и работать с DiskEditor. Шестнадцатеричное представление информации удобно использовать для восстановления программного кода загрузочных секторов диска, а текстовое поможет вам все-таки вытащить нужную информацию с диска... Правда, только представленную в формате простого текста или HTML и только за один прием из одного кластера (второй кластер по всему диску искать придется), но все же хоть что-то...

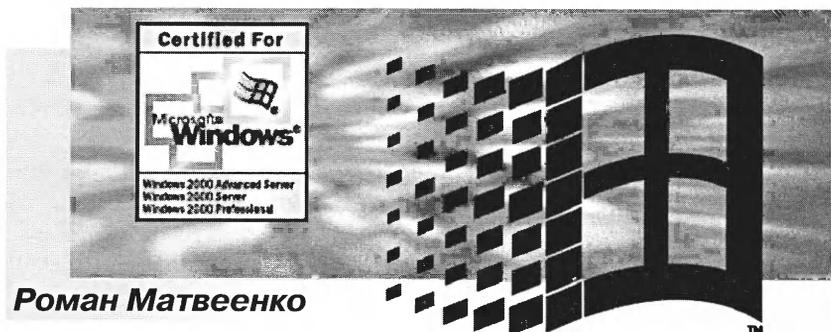
Скачать DiskEditor можно с сайта компании-производителя <http://www2.PhysTechSoft.com> или с адреса <http://filetools.nm.ru/discedit.rar>.



Это — корневой каталог диска, отображенный в DiskEditor, на уровне дорожек, секторов и битов

Заключение

В этой серии статей приведены только самые основы такой огромной темы, как принципы размещения информации на магнитных носителях, рассмотрена лишь не самая сложная и не самая удачная, но наиболее распространенная файловая система FAT. Существуют и другие файловые системы, значительно превосходящие FAT по сложности и возможностям. Однако для рассказа о них будет мало и приличной по размерам книги. И все же весьма полезно знать, как работает то, с чем сталкиваешься каждый день...



Роман Матвеев

Win2k. Программная совместимость

Операционная система Windows 2000 является перспективной и выигрышной почти во всех отношениях. Вопрос перехода на эту ОС сейчас, когда между словами компьютер и Интернет можно поставить знак равенства, становится очень актуальным. Сеть предъявляет особые требования и накладывает особые ограничения на использование той или иной операционной системы. Наконец, существует просто элементарная потребность в стабильной и надежной работе компьютера, когда не нужно думать о том "что будет, если я запущу то-то или то-то".

Эта статья для тех, кто еще только собирается ставить на свой компьютер эту ОС или уже сделал это и теперь хочет понять, почему некоторые из старых программ отказываются нормально работать в ее среде и как устранить обнаруженные конфликты.

Проблема совместимости

Причины, вызывающие эту проблему, могут иметь различную природу. Например, уже на этапе ее установки программа может вести себя некорректно, пытаться получить нерегламентированный (запрещенный) доступ к различным системным установкам. Но даже если программа ус-

тановилась и система "приняла" ее, возможен другой вариант конфликта, когда уже подготовленная к использованию программа после запуска пытается выполнить опять-таки нерегламентированный системный вызов. В этих случаях возможность использования данной программы в среде Windows 2000 ставится под большое сомнение. Какие же причины вызывают несовместимость той или иной программы? Рассмотрим несколько конкретных примеров.

Многие программы подразумевают, что системная папка Windows называется не иначе как C:\Windows. В Windows 9x дело обстоит именно так, но вот в Windows 2000 данная папка имеет другое название, C:\WinNT или близкое к нему. Не подготовленная должным образом программа вполне способна выдать сообщение об ошибке, не определив истинное расположение системной папки.

Другой пример. В процессе установки программа может проверить номер версии ОС Windows, установленной на компьютере. Если этот номер будет отличен от требуемого, то программа может тут же прекратить работу. В данном случае весьма вероятно, что программа в принципе может запуститься в среде Windows 2000, если подкорректировать соответствующим образом ее исполняемый код.

В Windows 2000 имеется функция защиты системных файлов Windows File Protection. Она позволяет защитить фундаментальные системные файлы от нежелательных изменений и, соответственно, возможных повреждений. Такая защита может стать проблемой для некоторых программ. Дело в том, что Windows 2000 позволит этой программе установить необходимые ей системные файлы как в своей собственной директории, так и в директории Windows. Но как только Windows 2000 определит наличие изменений в своих системных файлах, она "перематывает" все обратно и вернет всем измененным системным файлам их первоначальный вид. Это приведет к проблеме, если устанавливаемая программа для успешной работы требует как раз необратимой модификации и никак не рассчитывает на "обратную перемотку".

Наконец, Windows 2000 (как и NT4.0) не разрешает прямой доступ к аппаратным средствам компьютера. Это означает, что большинство 16-битных приложений, в том числе и игр, не будут запускаться под этой ОС. Звуковая карта, мышь или видеокарта надежно защищены от прямого обращения к ним со стороны запускаемого ПО, но это не означает, что к ним совсем нельзя обращаться. Естественно, можно, но любое такое обращение будет предварительно просматриваться и контролироваться ОС. В этом смысле Windows 2000 гораздо менее гибкая, чем Windows 98, и при работе "под нее", разработчики ПО должны быть особенно внимательны. Если в Windows 9x еще можно было получать доступ к аппаратным устройствам как через API, так и напрямую, то в Windows NT проделать подобный трюк уже не удастся. Поэтому любые некорректные или устаревшие процедуры должны быть просто исключены из программного кода и заменены другими, совместимыми с Windows 2000.

Несмотря на очевидные неудобства, вызванные таким отношением к используемому ПО, именно такая архитектура операционной системы как раз и обеспечивает ей надеж-

ность и стабильность. Вспомните, под Windows 9x сбой одного приложения мог привести к краху всей системы. С помощью Диспетчера задач иногда удавалось "спасти" компьютер от сбоя, последовательно закрывая все подозрительные или зависшие приложения. Здесь же все процессы, связанные с проблемным приложением, закрываются автоматически. Такой способ выполнения программы, при котором она помещается в специальную оболочку, называется инкапсуляцией. Делается это с целью минимизировать ущерб, который эта программа может принести ОС в случае неправильного выполнения.

Учитывая все сказанное, можно сделать довольно простой вывод: в Windows 2000 совместимость принесена в жертву надежности. Еще в процессе разработки этой ОС в Microsoft очень часто говорили о том, что первоочередная и наиболее важная цель — это стабильность и надежность создаваемой платформы, а вопросы совместимости будут рассматриваться уже во вторую очередь.

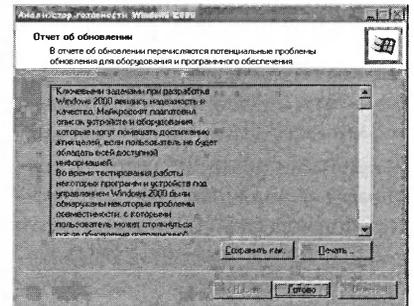
Проверьте ваши приложения

Рассмотрим три способа проверки совместимости той или иной программы с Windows 2000.

1. Посещение веб-сайта Microsoft и проверка официального сертификата программы.

На веб-сайте Microsoft можно обнаружить список программных продуктов, сертифицированных к использованию в среде Windows 2000. Самостоятельно посмотреть на этот список вы можете, посетив адрес <http://www.microsoft.com/windows2000/upgrade/compat/certified.asp>. Для получения подобного сертификата программа должна пройти серию тестов, которые проводят специалисты Microsoft. На сегодня количество официально сертифицированных программных продуктов невелико. Гораздо большее число приложений просто имеют пометку, сообщающую об их совместимости с Windows 2000. Наличие у программы подобной метки

говорит о том, что разработчики тестировали ее в среде Windows 2000, но это не означает, что программа после этого автоматически получает статус официально сертифицированного программного продукта.



2. Использование специальной программной утилиты Readiness Analyser.

Программа Readiness Analyser сообщит вам, с какого рода проблемами вы столкнетесь при установке Windows 2000. После завершения всех тестов она подскажет, какое оборудование и какие приложения на вашем компьютере полностью несовместимы с Windows 2000. Тесты этой программы выполняются также и в процессе установки Windows 2000. Но если вы захотите узнать, с какими программными и аппаратными неполадками столкнетесь при установке Windows 2000, то можете заранее загрузить и использовать эту программу. Ее официальная страничка расположена по адресу <http://www.microsoft.com/WINDOWS2000/downloads/deployment/readiness/default.asp>. Размер русской версии — примерно 2.6 Мб.

3. Посещение веб-сайта авторов программы.

Большинство серьезных разработчиков ПО публикуют на своих веб-сайтах полный список программных и аппаратных платформ, совместимых с их программными продуктами. Возможно, в результате посещения такого сайта вам не удастся найти специально адаптированную под Windows 2000 версию нужной программы. В этом случае стоит поискать на том же сайте соответствующий патч к уже существующей версии программы.

Игры

Многие любители игр именно из-за боязни несовместимости не хотят ставить на свои компьютеры Windows 2000. Тут уж действительно как повезет. Одна игра будет запускаться и выполняться абсолютно нормально, а другая либо совсем не будет работать, либо будет вести себя, скажем так, не совсем обычно. Особая статья — это старые игры, созданные для MS-DOS. Из-за запрета на прямой доступ к оборудованию эти программы в подавляющем большинстве под Windows 2000 не работают.

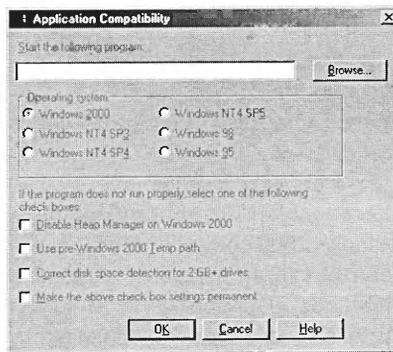
До появления Windows 2000 наиболее подходящей игровой платформой была Windows 98. Windows 2000 впитала в себя многие мультимедийные возможности Windows 98, дополнив их своей достаточно жесткой архитектурой. В результате вопросы совместимости той и или иной игры теперь, скорее всего, придется решать в индивидуальном порядке: единого и универсального способа, подходящего для всех игр, к сожалению, нет. Некоторые программы удастся "подружить" с Windows 2000, используя фирменные патчи от Microsoft, а в остальных случаях придется обращаться за поддержкой на веб-сайты авторов игры или на специализированные веб-сайты, такие как <http://www.win2kgames.com> или <http://www.ntcompatible.com>.

Заставьте ваши приложения работать

Что можно сделать для минимизации или полного разрешения проблем совместимости? Для начала стоит установить наиболее свежие пакеты обновлений с Windows Update site (<http://windowsupdate.microsoft.com>). Это надо проделать для всех приложений, установленных на вашем компьютере. Пакеты обновлений перерабатываются с появлением новых проблем совместимости. Установив подобные обновления на своем компьютере, возможно, вы автоматически устраните множество проблем для той или иной программы.

Другой способ — использование

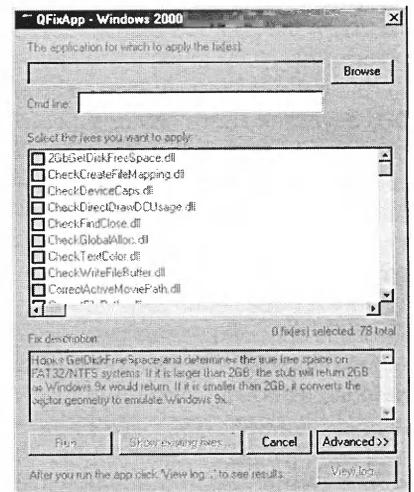
специализированных программ. Например, вы можете скачать пакет Application Compatibility Toolkit для устранения всякого рода нестыковок. Программы из этого пакета пытаются примирить приложения, созданные для более ранних версий Windows, и подстроить их под требования новой ОС. Пакет можно найти на веб-сайте Microsoft (<http://msdn.microsoft.com/compatibility>), его размер — около 6.5 Мб.



Первая программа из этого комплекта называется Application Compatibility (Arcompat.exe). Пользоваться ею довольно просто. Для начала вы должны выбрать то приложение (исполняемый файл), степень совместимости которого хотите определить или отрегулировать. Для большинства программ единственное, что необходимо указать, какой именно операционной системой должна "притвориться" Windows 2000. Если при запуске этой программы возникает ошибка, которой не было при запуске под более ранней ОС семейства Windows, начинайте устанавливать различные комбинации управляющих флажков. Выбрав галочкой 'Disable Heap Manager on Windows 2000', вы обойдете конфликты, связанные с управлением и распределением памяти. Установкой метки 'Use the pre-Windows 2000 Temp path' вы можете разрешить испытываемой программе использовать альтернативный временный каталог, если временный каталог по умолчанию занят или не доступен. Если при запуске вы получаете сообщения о нехватке дискового пространства, хотя точно знаете, что это не так, поставьте флажок в боксе 'Correct disk space detection for 2GB+ drives'.

Все проблемы совместимости Arcompat.exe, конечно, решить не сможет. Но если с помощью перечисленных опций вам все же удалось решить проблему для конкретной программы, вы можете закрепить за ней эти изменения, то есть сделать их постоянными. Для этого надо выбрать 'Make the above checkbox settings permanent'.

В добавление к Arcompat.exe, в комплекте Application Compatibility Toolkit можно также найти и еще одну, более полезную регулировочную утилиту QFixApp. Эта программка позволяет устранить почти восемь десятков известных на сегодня конфликтов. Пользоваться ею несколько сложнее, чем Arcompat.exe. Поэтому, прежде чем приступить к работе с ней, имеет смысл подробно изучить руководство. В награду за это вы получите весьма мощный регулировочный инструмент, способный с успехом устранить многие явные и скрытые программные конфликты.



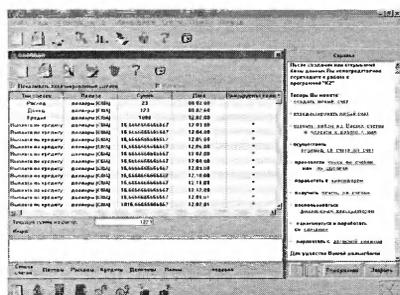
Service pack 1

Настоятельно рекомендую установить первый сервисный пакет SP1 для Windows 2000 (он был публично анонсирован в июле 2000 года). В пакет входит большое количество исправлений и добавлений как по всей системе в целом, так и по вопросам программной и аппаратной совместимости. Размер пакета — примерно 13 Мб, а загрузить его можно с веб-сайта Microsoft <http://www.microsoft.com/windows2000/downloads/recommended/SP1>.

K2-Финансы

Это уже самое что ни на есть "тяжеловесное" приложение для деловых людей. По принципу работы и интерфейсу программа во многом напоминает профессиональные бухгалтерские приложения. Фактически добавление каждого нового дохода или расхода представляет собой заполнение полноценной товарно-кассовой накладной. Идея создания данной программы не родилась на пустом месте: возможно, некоторым читателям знаком ее предшественник "Казначей". О популярности этого бесплатного приложения свидетельствует число пользователей — более 70 000.

Приложение используется для формирования произвольной структуры счетов в нескольких валютах, при помощи которых будут учтены все финансовые средства. Итоговая информация может быть представлена как по одному из виртуальных счетов, заданных пользователям, так и по всем счетам вместе. Таблицы данных могут быть преобразованы для дальнейшей работы и дополнительного анализа в формат Microsoft Excel.



Примечательные особенности K2 — календарь планирования сделок, представление графической информации о структуре и динамике доходов и расходов, изменении уровня благосостояния, а также возможность распределения сумм по счетам. Среди прочих возможностей программы особого внимания заслуживает редактор бланков, позволяющий создавать любые бланки. Разработчики даже позаботились о том, чтобы наиболее часто используемые шаблоны (бланки оплаты телефона, электроэнергии, квартплаты, платежи в ГАИ и т.д.)

были под рукой у пользователя сразу после инсталляции K2. К тому же в K2 реализован такой интересный, правда, довольно специфический и редкий инструмент, как финансовый калькулятор, который служит для расчетов и моделирования различных финансовых состояний. Он содержит следующие разделы:

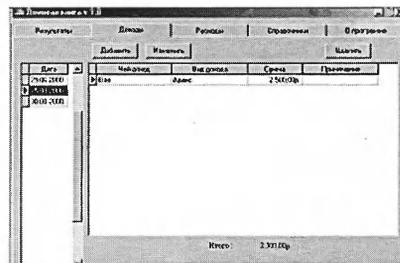
- обмен валют;
 - расчет суммы долга при получении кредита;
 - расчет целевого накопления;
 - расчет сложного процента.
- Кроме того, в приложение интегрированы некоторые справочники:
- список статей доходов;
 - список статей расходов;
 - список валют;
 - данные о курсах валют, используемых в счетах пользователя.

Вы имеете возможность редактировать данные в каждом из них, изменяя, удаляя и добавляя записи. Правда, за универсальность платить придется скоростью работы и несколько усложненным интерфейсом, к которому, впрочем, быстро привыкаешь.

Сайт разработчика, компании So\USE Technologies — <http://www.souse.com/>. Страница загрузки — http://www.souse.com/content/k2_download.html (3674 Кб)

Домовая книга

Эта программа в чем-то сродни "Кошельку", но имеет ряд существенных отличий. Начнем с того, что она многопользовательская, а значит, могут быть учтены и проанализированы доходы и траты каждого члена семьи. Кроме того, рабочая область приложения реализована как набор вкладок. Содержит справочники доходов и расходов. Справочник расходов "Домовой книги" двухуровневый, то есть статьи расходов вы располагаете в соответствующих группах, что дает дополнительные возможности для анализа и помогает более точно представлять себе свои траты. Для получения сводной статистики используется вкладка "Результаты". Статистических возможностей не так и много, зато изучение программы не займет и минуты, не больше времени уйдет и на ее

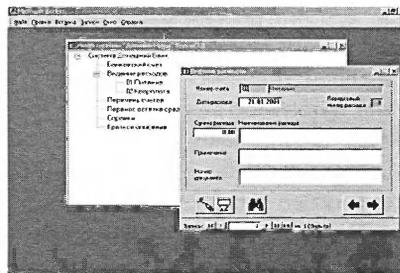


ежедневное пополнение. Вам останется только время от времени сверять содержимое своего кошелька с данными "Домовой книги" и по мере необходимости вносить коррективы. Это следует делать как можно чаще во избежание отрыва теории от практики.

Сайт программы "Домовая книга v 1.0" — <http://www.denrassk.narod.ru/budget.htm>. Скачать программу можно на <http://www.denrassk.narod.ru/download/houmebuh.exe>

Домашний Банк

За этим громким названием скрывается достаточно функциональная система учета домашних финансов, написанная в среде MS Access 2000, наличие которого на компьютере необходимо для работы программы.



Приложение не потребует инсталляции и после распаковки займет на винчестере 1,04 Мб. Идея, положенная разработчиком в основу системы, оригинальна. Весь семейный бюджет рассматривается как банковский счет, открываемый на один месяц, в течение которого на него поступают средства. Они сразу распределяются пользователем по статьям расходов, представленным как специальные счета, с которых снимаются средства для оплаты определенной группы трат. В

В 1983 году корпорация Microsoft объявила о своих планах дальнейшего развития программной индустрии, в которые входило создание оконной графической оболочки для платформы IBM PC. В то время существовали только две линейки персональных компьютеров, работающих под управлением операционных систем с графическим интерфейсом — Xerox Star ("Магия ПК" N7/2000) и Apple Lisa ("Магия ПК" N9/2000). Обе платформы были выпущены самими производителями аппаратного обеспечения этих ПК. Поскольку данные машины, в отличие от персоналок IBM с процессорами Intel, были очень дорогими, они не имели широкого распространения среди частных пользователей. Microsoft же задалась целью разработать поистине "народную" операционную систему для платформы PC.

Основное отличие Microsoft от других компаний, работающих в сфере компьютерной индустрии, заключается в том, что эта корпорация всегда сама создавала для себя рынок. В то время, как все остальные фирмы терпеливо "подстраивались" под производителей "железа", Microsoft всеми доступными средствами "разворачивала" их конвейеры в нужном для нее направлении. Немалую роль в этом сыграло личное обаяние и неисчерпаемый энтузиазм Билла Гейтса, лично возглавлявшего все переговоры с компьютерными гигантами. Именно ему удалось убедить совет директоров IBM, до этого тратившей 90% своих ресурсов на разработку огромных корпоративных компьютеров — мэйнфреймов, — перестроиться на производство персоналок, в перспективе и прибыльность выпуска которых никто в IBM не верил. И все потому, что сам Гейтс изначально сделал ставку на продажи операционной системы именно для этих компьютеров — MS DOS.

Теперь настало время убедить IBM в необходимости комплектовать эти компьютеры системой с оконным графическим интерфейсом, которую еще предстояло разработать.

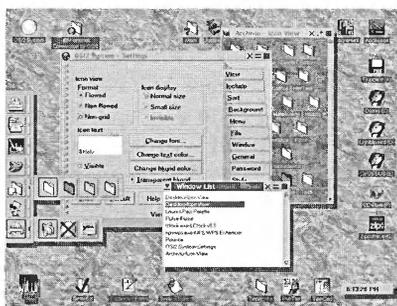


Валентин Холмогоров

"Полуось"

Антология операционных систем

Гейтс отлично понимал, что без тесного сотрудничества с производителями аппаратной части ЭВМ, силами одной только Microsoft, такой масштабный проект можно и не "поднять". В конце концов договор о намерениях был подписан, и две компании — IBM и Microsoft — взялись за создание новой операционной платформы, которая получила название OS/2. Согласно условиям договора, обе фирмы трудились над проектированием системы совместно, но каждая имела право вносить в структуру программ собственные изменения, а также независимо продавать систему другим производителем ПК.



Работа шла медленно, что было обусловлено излишним количеством специалистов, принимавших участие в продвижении проекта со стороны IBM. Политика этого компьютерного гиганта строилась на том, что представители всех отделов огромной корпорации могли выносить на суд разработчиков свои

рекомендации, которые незамедлительно рассматривались и либо принимались, либо отвергались. В результате программисты были завалены таким количеством замечаний и поправок, зачастую прямо противоречащих друг другу, что их обсуждение занимало едва ли не все рабочее время проектного отдела.

Более того, IBM планировала сделать OS/2 базовой системой для всех выпускаемых ею аппаратных платформ — от многотонных мэйнфреймов до настольных персоналок, — что дало бы ей неоспоримое преимущество на рынке перед конкурентами: корпоративные заказчики, до этого эксплуатировавшие большие IBM-овские машины, стали бы покупать новые поколения ЭВМ только от IBM. Попытка объединить в одной операционной системе поддержку компьютеров с разной архитектурой превратила OS/2 в настоящего "программного монстра", оснащенного множеством избыточных и ненужных модулей. OS/2 загружалась в память ПК порядка пяти минут и работала так медленно, что пользователей скорее устроила бы работа в консольной DOS, чем переход на красивую, интуитивно понятную, но ужасно неповоротливую и требовательную к ресурсам оконную систему.

Microsoft добивалась от IBM обеспечения совместимости OS/2 с разрабатываемой ею платформой

MS Windows. Такая совместимость обеспечила бы перенос программ с одной системы на другую без внесения в них глобальных изменений, однако IBM, все еще делавшая ставку на продажи больших вычислительных машин, с каждым днем добавляла в структуру OS/2 новые элементы, уводившие эту систему все дальше от Windows и превращавшие ее в платформу для корпоративных вычислительных машин и рабочих станций.

В конце концов Microsoft, отчаявшись сделать из OS/2 доступную владельцам маломощных компьютеров операционную систему, рассчитанную на массовое распространение среди частных пользователей, вышла из проекта и сосредоточилась на производстве и совершенствовании Windows. С этого момента OS/2 стала стремительно сдавать свои позиции: чем более громоздкой и медленной она становилась, тем большим спросом пользовалась компактная, быстрая, простая в установке и изучении Windows.

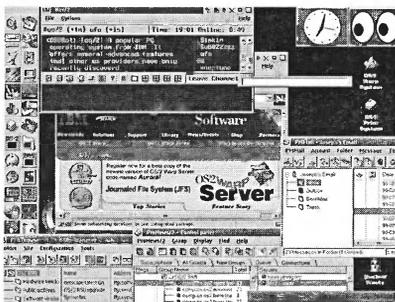
К 1990 году IBM наконец осознала свою стратегическую ошибку и подписала новый контракт с Microsoft — уже на распространение в комплекте поставки своих ПК системы MS



Windows версии 3.0. По подсчетам специалистов, к этому моменту затраты IBM на создание OS/2 составили порядка 2 млрд долларов, в то время как сама эта система достаточно вяло использовалась лишь крупными корпоративными потребителями, для которых Windows была слишком слабой и малофункциональной ОС. В 1992 году сотрудничество между Microsoft и IBM по дальнейшему развитию OS/2 прекратилось, а появившаяся вскоре Windows NT начала быстро теснить OS/2 и с корпоративного сектора рынка.

Однако дальнейшая эволюция OS/2 на этом закончилась: IBM продолжала развивать систему самостоятельно — не то из "врожденного" упрямства этой корпорации, не то в силу огромных средств, уже вложенных в проект. Отказавшись от поддержки мэйнфреймов, IBM постаралась оптимизировать OS/2 исключительно для использования на ПК в надежде на реванш. Однако реванша не получилось: к этому времени Windows уже стала общепризнанным стандартом, и динамичная, быстро реагирующая на изменения рынка Microsoft упорно держалась "на шаг вперед" от неповоротливого голубого гиганта.

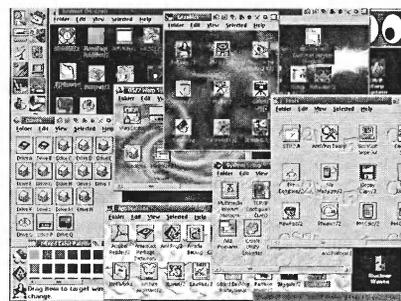
Одной из самых удачных современных модификаций данной ОС яв-



ляется OS/2-4 Warp, имеющая русскую локализацию, которую российские пользователи не замедлили окрестить "полуосью". Сегодняшняя OS/2 — это мощная многозадачная операционная система с оконным графическим интерфейсом и набором созданных специально для нее прикладных программ, ориентированная на рынок ПК и рабочих станций. Интерфейс OS/2 содержит все необходимые элементы современных ОС — рабочий стол и корзину, иконки и панель задач, программу просмотра содержимого дисков, часы и драйверы множества периферийных устройств, таких как порты USB или инфракрасный порт. Установка производится автоматически, причем OS/2 самостоятельно определяет оптимальную конфигурацию системы, исходя из быстродействия процессора и объема оперативной памяти (хотя пользователь может и самостоятельно указать комплект необходимых программ, исключив ненужные), тестирует оборудование

и настраивает все необходимые драйвера без участия оператора.

В комплект поставки входит пакет IBM Works, аналогичный MS Office и содержащий текстовый и табличный редактор. Имеется удобный веб-браузер WebExplorer и почтовый клиент NotesMail, система для создания анимации

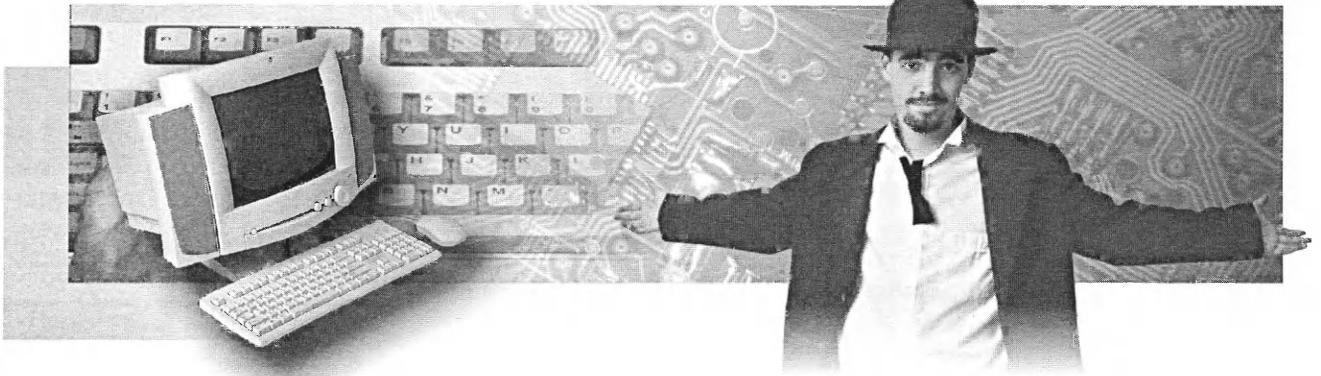


NeonGraphics, широкий выбор всевозможных бизнес-приложений и множество игр от Civilization и Quake III до Master Orion.

Есть и глобальные отличия OS/2 от MS Windows. Например, специальный самообучающийся программный пакет позволяет управлять системой с использованием голосовых команд, для чего в коробку с компакт-дисками разработчики вкладывают микрофон и наушники.

Однако, несмотря на поистине широчайшие возможности, высокую производительность и потрясающую надежность данной платформы, она не пользуется сейчас высоким спросом: на рынке доминирует более распространенная и дешевая MS Windows. Основная проблема, препятствующая развитию OS/2, заключается в ее несовместимости с программами производства Microsoft, посредством которых создается практически вся деловая документация и с которыми работает подавляющее большинство частных пользователей. В наследство от мэйнфреймов "полуоси" достались громоздкость и высокие требования к системным ресурсам, однако стабильность работы этой платформы, многократно превосходящая стабильность Windows, делает ее весьма привлекательной для специальных систем, требующих большой надежности, таких как банковские компьютеры и рабочие станции.

Клуб любителей программирования



Без If-ов и Switch-ей

Александр Хайт

Вниманию читателей предлагается не совсем традиционное решение одной часто встречающейся в практике программирования задачи. Оговорюсь, что данный способ может быть реализован на С или Паскале, еще эффективнее — на Ассемблере, хотя с Бейсиком возникли проблемы.

Задача

Нужно выполнить задание (фрагмент программы, функцию, процедуру), определяемую неким текстом, который вводит пользователь. Например, имеются два числа: 7 и 3. Вводится директива Add, если эти числа нужно сложить, Sub — вычесть, Multy — перемножить и Div — поделить, проверив заодно, не является ли делитель нулем. Если же пользователь ввел директиву, отличную от перечисленных, то на экран выдается диагностика.

Конечно, эту задачу легко решить, применяя условный оператор if, либо, что профессиональнее, конструкцию switch-case (C), Case (Паскаль). Предлагаемый этюдный вариант изящнее, легко модифицируется, дает возможность экономить и память, и время. На его основе можно решать и более сложные задачи. Вообще говоря, решение похоже на решение шахматной многоходовки. Впрочем, убедитесь сами.

Решение

Начальные опции, подключение библиотек

```
#include <stdio.h> /* подключаем библиотеки и, в отличие от Pascal, без них в принципе не обойтись */
#include <string.h> Uses Crt; {Без этой библиотеки можно обойтись, но вдруг захочется что-то вывести на экран по-красивее}
```

Определение пользовательских типов (шаблонов) — важнейшая часть разработки программы.

```
typedef float TYPE
(*texfunc)(float a, float b); /*определили указатель на функцию, возвращающую число с плавающей точкой, от двух параметров — чисел с плавающей точкой */
typedef struct {char c[6]; texfunc myf;} tmyswit; /*определяем шаблон структуры из двух полей: строки ASCIIZ не более чем из 6 символов (директива) указателя на функцию обработки этой директивы */
```

```
tmyswit tarrsw = array[0..12] of tmyswit; {шаблон массива из 13 записей}
typedef tmyswit tarrsw = array[12]; /* подготовлен шаблон на 13 строк (массив из 13 элементов, типа заданной структуры */
```

Вот он, наш массив структур, который мы в головном модуле проинициализируем такими значениями:

Директива — текст не более чем из 6 символов	Адрес функции, реализующей директиву
add	fadd
sub	fsub
multy	fmul
div	fdiv
Введенная пользователем директива	ferr

(Если при сравнении введенной директивы с имеющимися добрались до последней строки, значит, введенная директива эталоном не предусмотрена)

Теперь пора определить сами функции, которые "обслужат" директивы. Имена функций говорят сами за себя. Изящнее это сделать отдельным модулем, "собрав" код в библиотеку, а в главном модуле только показать шаблон. Но не все сразу.

```
float fadd(float a, float b) { return a+b; } Function Fadd(a,b: Real): Real; Begin Fadd := a+b End;
float fsub(float a, float b) { return a-b; } Function Fsub(a,b: Real): Real; Begin Fsub := a-b End;
```

float fmul(float a, float b) { return a*b; }	Function Fmul(a,b: Real): Real; Begin Fmul := a*b End;
float fdiv(float a, float b) { if (b) return a/b; else {printf("\nДеление на 0!"); return 0; } }	Function Fdiv(a,b: Real): Real; Begin If b<>0 Then Fdiv := a/b Else begin WriteLn('Деление на 0!'); Fdiv := 0 end End;

Функция обработки ошибок такая же по структуре. Однако вынесем ее отдельно, потому что о необходимости такой функции хорошо помнят только профессионалы. К тому же написать правильную реакцию — дело естественное и понятное, а предусмотреть весь спектр пользовательских ошибок бывает как непросто!

float ferr(float a, float b) { printf("\nвведеное действие не описано"); return 0; }	Function Ferr(a,b: Real): Real; Begin WriteLn('Введеное действие не описано'); Ferr := 0 End;
--	---

Внимание! Один из ключевых моментов: ниже создается функция, возвращающая номер введенной директивы в таблице директив. Этот номер — ключ к ветвлению, ведь он соответствует номеру функции обработки! Обратите внимание на то, что сама директива, номер которой определяется, помещена в ту же таблицу, в ее k-ю строку. Это позволяет осуществить переход, обратившись ко второму столбцу строки с тем же номером, то есть к полю указателя на искомую функцию в той же структуре.

int fobr(tarrsw d, int k) {int i; for (i=0; strcmpi(d[i].c, d[k].c); i++); /* По сути, это цикл While, но возможности C позволяют использовать for. Другой вопрос, хорошо ли это в стилистическом отношении. */ Return i; }	Function Fobr(d: tarrsw; k: byte): integer; Const i: 0..12 = 0; {Инициализируем счетчик} Begin While d[i].c <> d[k].c do {Сравниваем строки с последней, пока не совпадут} Inc(i); Fobr := i End;
---	---

Проверка здесь достаточно проста. За счет размещения сравниваемой строки можно обойтись только анализом на совпадение строк. Не позднее, чем на k-м шаге совпадение неизбежно!

Подготовка закончена. Как всегда, она занимает существенно больше места, времени и сил, чем собственно головной модуль. Теперь осталось выполнить инициализацию, а затем одним оператором вызвать и функцию сравнения, и функцию обработки директивы, причем этот оператор сразу выводит результат на экран, что называется "одним махом".

void main() { tarrsw t; /* Выделяется память для переменной, которая на самом деле массив структур! */ Strcpy(t[0].c, "add"); /* Для ASCIIZ-строк лучше использовать стандартные функции. Мы описываем свои директивы и указываем */ t[0].myf=fadd; /*на функции, их обрабатывающие.*/ Strcpy(t[1].c, "sub"); t[1].myf=fsub; strcpy(t[2].c, "multy"); t[2].myf=fmul; strcpy(t[3].c, "div"); t[3].myf=fdiv; t[4].myf=ferr; scanf("%n%s",&t[4].c); /* Вводим директиву прямо в последнюю строку массива структур! */	VAR t: tarrsw; {В Pascal описание переменных головного модуля предшествует коду и входит в самостоятельный раздел VAR} BEGIN t[0].c := 'add'; {Инициализируем массив записей. В отличие от C "в лоб", без функций str... } t[0].myf := Fadd; {Инициализируем функции директив} t[1].c := 'sub'; t[1].myf := Fsub; t[2].c := 'multy'; t[2].myf := Fmul; t[3].c := 'div'; t[3].myf := Fdiv; t[4].myf := Ferr; ReadLn(t[4].c); {Вводим директиву, причем прямо в поле последней записи!}
--	---

Выводим на экран результат, пользуясь сразу двумя функциями — одной в качестве индекса для функциональной переменной. Вся суть программы свелась к одной строке!

Вот и все. Разветвились почти без всяких условных операторов. Красивый пример, хотя и не новый. Если нужно обслужить больше директив, код не меняется, просто увеличивается размер таблицы, длиннее инициализация. Есть и другие резоны использовать процедурные переменные. Например, в примере с меню (прошлый выпуск "Магии ПК")

мы определили номер строки меню. Если заведен массив процедурных переменных (указателей на функции), то номер строки i — и есть индекс массива. Для обработки выбранной строки меню достаточно простого оператора: t[i](). В скобках перечисляются параметры, если они предусмотрены. Нередко задача требует, чтобы после выполнения какой-либо директивы другие директивы обрабатывались иначе, чем до выполнения. Для решения такой задачи порой составляют целое "личное дело" выполненных операций. Между тем, задачу просто решить, если написать все необходимые функции для предусмотренных случаев, а в функцию обработки директивы передавать еще один параметр — ссылку на массив структур. Тогда функция обработки очень просто переопределит последующие реакции на вызов директив. Например, чтобы директива add после первого исполнения (сложения) в дальнейшем работала как команда умножения, достаточно записать ее так:

```
Function Fadd (Var t: tarrsw; a,b: Real): Real; Begin Fadd:=a+b; t[0].myf := Fsub; End;
```

Функция Fsub должна быть определена заранее, и для этого есть способы, но о них — в другой раз. И еще пара замечаний. Во-первых, предложенные алгоритмы эффективно реализуются на Ассемблере и хороши для управляющих программ, записываемых в ПЗУ. Только при этом ввод директивы в структуру ПЗУ, разумеется, невозможен. Приходится усложнять условие функции fobr, но это не очень усложняет саму программу. Во-вторых, сравнив длину программ на C (28 Кб) и Паскале (7 Кб), можно подумать, что язык C намного хуже. На самом деле, если не употреблять библиотеку обработки строк, а выполнить данные операции "вручную" и удалить опции отладки, код существенно сократится. Владельцы "навороченных" ПК, возможно, посмеются, глядя на размер кода, но программистам часто приходится бороться за каждый байт, за малейшую экономию времени. В таком случае этот алгоритм — ваш.

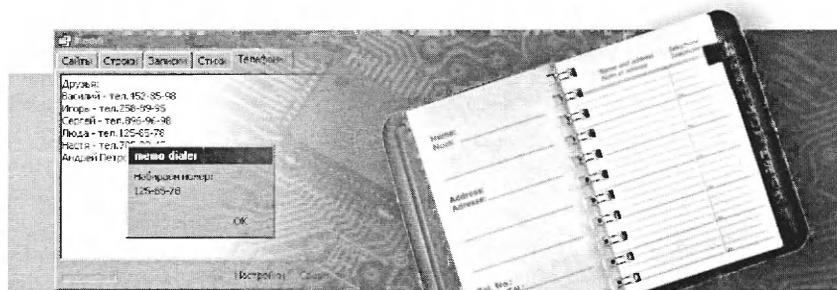
Наверняка у каждого из вас во время работы на компьютере возникали какие-то мысли, которые очень хотелось сразу записать — планы на будущее, творческие идеи, может быть, даже стихи. Иногда нужно срочно скопировать себе на диск адрес хорошего сайта с веб-странички, информацию о погоде, важную новость в Интернете.

Можно, конечно, открыть MS Word и копировать все нужное туда, но он и грузится долго, и сохраняет медленно, и ресурсов машины потребляет немало. А сохранять каждый раз веб-страничку на диск и потом читать ее, отключившись от Сети, конечно, можно, но через некоторое время ориентироваться среди них будет ой как непросто... Куда приятнее иметь что-то вроде блокнотика, в который можно быстро копировать нужную информацию и так же быстро ее доставать оттуда. И чтобы этот блокнотик всегда был под руками, но не требовал к себе особого внимания.

Эта статья рассказывает о небольших программах, назначение которых — быть "записной книжкой" пользователя.

В набор необходимых базовых функций таких программ входят:

- возможность создавать несколько страничек для записей и работы с ними с использованием буфера обмена;
- простота установки и использования;
- сохранение записей в отдельном файле или файлах;
- быстрота работы;
- наличие системы поиска по отдельным записям;
- сохранение элементарного форматирования вставленного текста (выделение, курсив, размер шрифта) и возможность изменять его в самой программе;
- удобный интерфейс или возможность его настройки, система навигации по страничкам, возможность быстро свернуть и развернуть программу, удобные кнопки управления и др.



Антон Орлов

Записки на манжетах

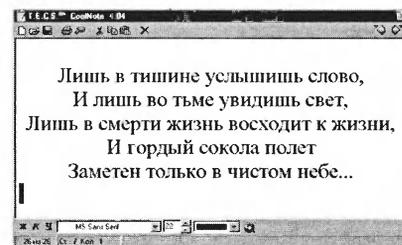
CoolNote

CoolNote можно назвать "классической" программой из класса "записных книжек". После ее запуска (его можно осуществлять вместе с запуском системы, скопировав в папку "Автозагрузка" главного меню ярлык исполняемого файла) в панели задач рядом с часами появляется иконка блокнотика, а по двойному щелчку мышкой открывается окно небольшого текстового редактора, имеющего одну замечательную особенность: он состоит из набора записей.

Нажимая на стрелки, можно перемещаться между отдельными записями, а нажав на изображение чистого листа — создать новую. Есть функция поиска нужной записи по содержащимся в ней словам, возможность использовать различные шрифты и выбирать цвет текста. Сами записи хранятся в каталоге с программой в отдельных файлах в формате Rtf.

Формат Rtf был создан для того, чтобы разные программы разных производителей могли обмениваться между собой данными. Его особенность в том, что в нем может сохраняться любое оформление текста, любые рисунки и таблицы. И при этом любая программа, поддерживающая этот формат, сможет с ним работать. Только вот видеть она в файле будет лишь то, что сможет. Но если файл в ней не редактировать, то все остальное содержимое сохранится. Вы можете сохранить в формате Rtf сложноформатированный документ в Word 2000, а потом открыть его в WordPad или даже в Lexicon для MS DOS. И текст вам все равно будет доступен! Те элементы оформления, которые Lexicon не поддерживает, просто не будут видны.

Теперь, найдя на страничке Интернета, в электронной книге или в письме друга хорошую фразу или полезную ссылку, достаточно открыть CoolNote и вставить ее туда. Всегда будет под рукой. Очень удобно, почти не занимает памяти и не потребляет ресурсов компьютера.

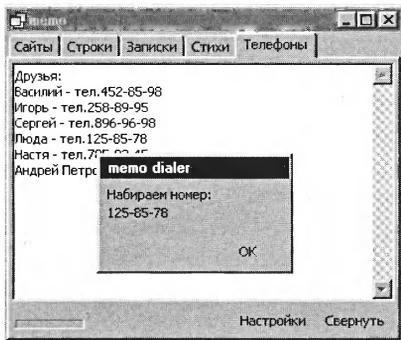


Программа CoolNote

Coolnote создали русские программисты Сергей Дурманов и Андрей Петров. Загрузить эту программу можно с адреса <http://coolnote.chat.ru/cool404.exe>.

Мемо

В этой программе тоже можно создать множество страничек и помещать туда свой текст. При вставке текста сохраняются параметры шрифта, а сами страницы сохраняются в отдельных файлах формата Rtf. Она умеет сворачиваться в трей, а вызвать ее оттуда можно "горячей клавишей" F12. Мемо можно использовать ее как самую настоящую телефонную книжку: если вы помещаете в нее номер телефона, то для того, чтобы позвонить по нему, достаточно просто щелкнуть мышью пару раз на номере. Естественно, для этого надо подключить к модему микрофон или наушники. Точно так же работают и гиперссылки, помещенные в окно программы. Причем не нужно их как-то преобразовывать — достаточно просто набрать. А нет необходимости в этой функции — спокойно ее отключите.



Программа Мемо.
Позвоните друзьям...

Мемо позволяет даже защищать свои записи паролем и шифровать им страницы с записями. Тогда файлы с вашими заметками нельзя будет прочитать без помощи Мемо. Программа работает даже быстрее, чем CoolNote и, как и та, имеет элементарные функции текстового редактора (выбор характеристик шрифта, выравнивания, вставки и копирования).

Недостатками данной программы можно считать отсутствие системы поиска фрагментов текста, а также сложность процесса добавления новых страниц (для этого приходится использовать отдельное диалоговое окно). Впрочем, если перета-

щить уже существующий текстовый файл в окно программы, автоматически создается новая запись, в которую окажется помещенным этот файл. Кроме того, если модем выключен или установлен на иной порт, нежели указанный в настройках Мемо, то при попытке набора номера из нее программа "зависает".

Создал Мемо М.Бондаренко из Омска, ее можно загрузить с адреса <http://maxnetzone.narod.ru/files/memo.zip>.

NoteBook

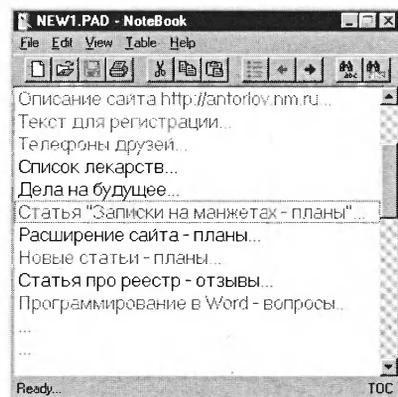
NoteBook американского программиста Грегори Брауна представляет собой некоторую комбинацию записной книжки и планировщика дел. Она может работать в двух режимах — отображения оглавления и редактирования страничек с записями.

При загрузке программы в ее окне отображается оглавление — список записей, состоящий из их первых строк (заголовков). Двойной щелчок на заголовке записи приведет к ее открытию. Перемещаться по записям можно как посредством выхода в оглавление, так и нажатиями на стрелки вверх окна. Есть возможность поиска фрагмента текста в записях, а также замены фрагментов. Текст в записях можно редактировать — вырезать, копировать и вставлять части. Вставить текущие дату и время можно одним нажатием клавиши, формат даты можно задать в настройках. Заголовки записей в оглавлении можно помечать разными цветами: красным и серым. К примеру, красный может означать пометку "Срочное", а серый — "Сделанное". Весь набор записей вместе с оглавлением можно сохранить в Книгу NoteBook. Создав несколько таких Книг с разными наборами записей, вы получите возможность загружать нужную тогда, когда это необходимо.

К сожалению, NoteBook не умеет сворачиваться в трей и не позволяет оформлять текст разными шрифтами или выделением. Соответственно, он и не сохраняет эти параметры текста при вставке в запись. К тому же окно программы имеет фиксированную ширину, что в иногда не

очень удобно. Программа не работает, если путь к ее исполняемому файлу содержит русские буквы.

NoteBook для тех, кому наличие удобного оглавления всех записей с

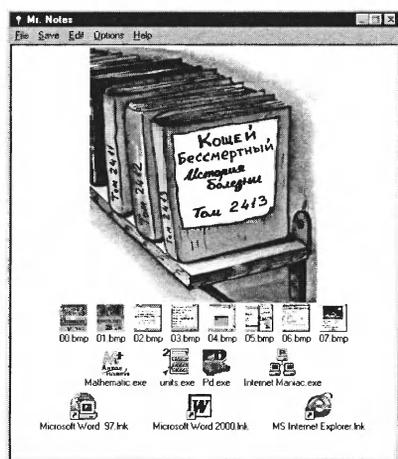


NoteBook — удобное средство для организации списка дел и проектов

возможностью пометки отдельных пунктов и функции замены фрагментов текста во всех записях важнее красивого оформления текста и наличия программы в трей. Загрузить NoteBook можно с адреса <http://www.execpc.com/~sbd/nbook.zip>.

Mr. Notes

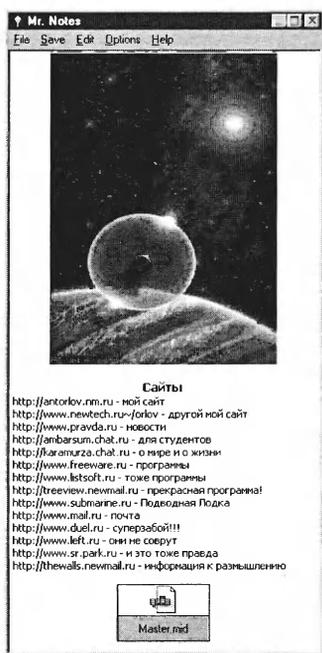
Простая и быстрая программа Mr.Notes Риана Мартинсена позволяет создать всего одну запись, но зато в нее можно вставлять не только текст, но и рисунки, звуки, другие файлы и даже ярлыки! Сделать это можно, просто перетащив иконку нужного объекта в окно программы или че-



"Центр управления компьютером" — Mr.Notes

рез буфер обмена. То, что вставлено в Mr.Notes через буфер, как данные в нем и остается, а то, что перетасовано мышью или скопировано как файл, вставляется как ссылка на файл.

Возможность вставить в окно Mr.Notes разные объекты дает большие преимущества. Так, если вы поместите туда ярлыки программ или файлов на вашем ПК (просто переносом иконки мышью), то затем сможете запускать их из этого окна. Получится этакий "пульт управления". Работать с ним будет куда удобнее, чем с меню "Пуск". К примеру, вы можете сделать сборник любимых мелодий и запускать их прямо из окна Mr.Notes. А искать ее долго не придется — сидит в тее. Содержимое окна Mr.Notes сохраняется в каталоге с программой как файл Rtf, читать его можно и другими приложениями.



Страница для записей в Mr.Notes одна, но вставить в нее можно все, что угодно

Mr.Notes можно загрузить с адреса <http://www.hotwww.com/ryan/notes.zip>. Для запуска программы требуются компоненты Visual Basic, которые вы также можете загрузить с сайта Риана. Однако если вы хотя бы раз устанавливали на свой компьютер программу, написанную на этом языке, то они у вас уже есть.

Продолжение следует



Владимир Буслаев

Паскаль

История программирования

Паскаль (Pascal) — один из наиболее распространенных языков программирования 70—90-х годов, поддерживающий самые распространенные методологии проектирования программ (нисходящее, модульное и структурное программирование). У него богатая история развития.

Начало этой истории положило объявление в 1965 году конкурса по созданию нового языка программирования — приемника Алгола-60. Участие в конкурсе принял швейцарский ученый Никлаус Вирт, работавший доцентом на факультете информатики Стэнфордского университета. Однако проект, предложенный им в 1967 году, был отвергнут комиссией.

Но Вирт не прекратил работу над созданием нового языка. Вернувшись на родину, в 1968 году он с сотрудниками Швейцарского федерального института технологии в Цюрихе разработал первую версию языка Паскаль (назван в честь великого французского механика и математика Блеза Паскаля, создавшего в 1642 году первую счетную машину).

Pascal был первым языком, имевшим ясную, логичную структуру, что делало его идеальным для обучения программированию. Звездный час Паскаля наступил в середине 70-х,

когда на рынке впервые появились персональные компьютеры Apple и Commodore. Тогда несчетное количество школ и вузов во всем мире установило Паскаль на свои машины.

В 1971 году Вирт выпустил описание языка, а в 1975 году появилось руководство для пользователей, положенное в 1982 году в основу стандарта ISO на язык Pascal.

Язык создавался, с одной стороны, как средство для обучения программированию, а с другой — для эффективного решения самых разнообразных задач на ЭВМ того времени. Вирт уделил большое внимание структурному программированию, благодаря чему конструкции Pascal позволяют писать надежные, легко проверяемые программы с ясной и четкой структурой. Кроме того, для структурированных программ характерны легкость отладки и корректировки, они достаточно просты в сопровождении и модификации.

Pascal обладает полным набором структурных типов данных, таких как простые переменные, массивы, файлы, множества, записи, ссылочные переменные. Особо следует отметить надежность Pascal-программ, которая достигается, во-первых, за счет простоты и естественности конструкций языка, которые соответствуют (или "стремятся"

соответствовать) логическому мышлению разработчика программ, а во-вторых, — за счет некоторой избыточности, например, обязательного описания всех переменных и соответствующих им типов.

Язык содержит ряд удачных изобретательных средств, таких как операторы case, repeat, if, while, обеспечивающих организацию ветвления в программе без использования операторов перехода, что также способствует пониманию программы.

Вторую жизнь в язык вдохнула фирма Borland, разработавшая на его базе в 80-х — начале 90-х годов семейство Pascal-систем, названных Turbo Pascal, включающих в первых версиях компилятор и интегрированную среду разработки и отладки. Компиляторы фирмы поддерживались многими распространенными операционными системами персональных компьютеров, такими как CP/M, MS DOS (причем стоимость его на тот момент составляла около \$80).

В последующие версии были включены компилятор, позволяющий создавать объектно-ориентированные программы (версия 5.5), встроенный ассемблер и объектно-ори-

ентированную оболочку для создания прикладных программ (версия 6.0), интегрированную среду, позволяющую редактировать, компилировать и отлаживать программы больших объемов (версия 7.0). Кроме того, пакет программирования Borland Pascal 7.0, выпущенный фирмой в 1992 году, обеспечивал возможность работы в трех режимах: в обычном и защищенном режимах работы операционной системы MS DOS и в среде Windows.

Примерно в это же время появились другие разновидности языка. Так, Пер Брич Хансен создал на его основе SuperPascal — язык параллельного программирования для описания вычислительных алгоритмов. В нем добавлены новые конструкции для создания детерминированных параллельных процессов и обмена синхронизирующими сообщениями. Язык позволяет создавать неограниченные комбинации рекурсивных процедур и параллельных конструкций.

В ходе дальнейшего развития вычислительной техники и технологии программирования фирма Borland разработала принципиально

новый программный продукт — Delphi — ориентированный на работу в Windows. Для представления программ в Delphi используется созданный фирмой Borland язык Object Pascal, в основе которого лежит все тот же Pascal. Но об этом — в следующий раз.



Никлаус Вирт родился в 1934 году в Швейцарии. Его детство прошло в маленьком городке Винтертуре, где преподавал географию его отец. Единственный ребенок в семье, Никлаус обрел свое призвание в отцовской библиотеке.

Всю жизнь он неуклонно шел своим собственным путем. Получив в в Федеральном цюрихском технологическом институте (ETH-Zurich) диплом инженера-электронщика, в 1959 году он покинул Швейцарию и прошел обучение по курсу разработки программного обеспечения в Калифорнийском университете в Беркли, после чего работал преподавателем в Стэнфордском университете.

После восьми лет, проведенных в США, Вирт получил предложение



организовать в ETH группу компьютерных исследований и вернулся с семьей в Швейцарию. Несмотря на триумф Паскаля, швейцарская почва оказалась не слишком благодатной для компьютерных исследований. Вирта это очень разочаровало.

В 1965 году он и К. Хоар предложили модифицированную версию Алгола, которая, по мнению Хоара, объединяла лучшие свойства Алгола-60 и не содержала его изъянов. Эта версия, получившая название Алгол-W (в честь Вирта), была опубликована и приобрела большую популярность в университетских городках. Но комитет по Алголу ее отверг.

В период с 1977 по 1980 год Вирт сконструировал компьютер "Лилит". Эта мощная рабочая стан-

ция была одной из первых машин, имевших мышь, монитор с высоким разрешением и графический интерфейс пользователя. По сравнению с ней компьютер Apple II, снабженный только клавиатурой, выглядел почти традиционным (за разработку "Лилит" Вирт был удостоен Тьюринговской премии, считающейся высшей наградой в области компьютерной науки).

В 1981 году Никлаус Вирт разработал язык программирования Модула-2.

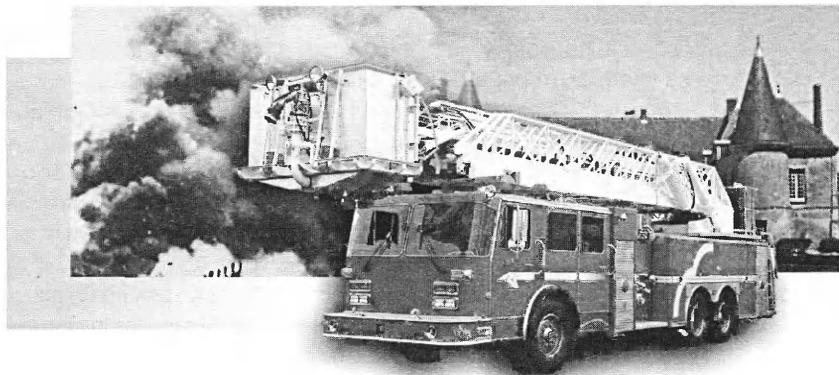
Следующей вехой его работы в 80-е годы стали компьютер "Церес" и операционная система "Оберон", позволявшая пользователю вести простой диалог с компьютером. Затем Вирт работал на переднем крае аппаратных и программных исследований, в области программируемых полевых полупроводников (FGPAs). Его карьера отмечена семью почетными докторскими степенями и множеством других наград.

Эта игра с самого начала не дала нам шанса даже на малейшую интригу, все было прекрасно известно заранее. Разработчики из корейской компании Phantagram ничего не скрывали от народа. Да, их Kingdom under Fire будет похож на Варкрафт, совмещенный с Диябло. Да, стратегические миссии будут чередоваться с ролевыми. Да, будет всего две противоборствующие стороны с однотипным набором юнитов. Так, конечно, неинтересно, но зато спокойно.

Наконец-то KUF появился и у нас в полной версии на двух компактках. Начнем, помолясь! Забудьте все то, что вы видели на скриншотах! "Вживую" KUF производит совершенно другое впечатление. Все, что казалось мрачным и корявым, неожиданно оживает, юниты оказываются на удивление подвижными и забавными, а тени и пиротехнические эффекты оставляют старину Starcraft далеко позади. Видимо, таковы особенности движка игры — статичная картинка производит отталкивающее впечатление, так как скрывает все те многочисленные движения юнитов, которые вы видите в процессе игры. Колоритные карлики-бомбардиры, например, периодически прочищают маленькими банниками свои мортиры, закрывают руками уши при выстреле и с кряхтением таскают тяжелые пушки на спине. Все остальные тоже по мере сил и возможностей развлекают игрока. Кстати, портретики с изображениями персонажей анимированы на манер Starcraft и, надо сказать, если не лучше, то, по крайней мере, не уступают художествам Blizzard.

В общем, графика на вид — забавливо перерисованный Warcraft с добавлением хорошо знакомых черт Starcraft.

Идем дальше. Панель управления и общий вид на игровое поле заставляют вспомнить другой хит упомянутой фирмы. Это было видно уже по скриншотам — Warcraft II, да и только. Другое дело, что это обстоятельство не вызывает раздражения, наоборот, радует. Видимо, соскучились мы по старым добрым



Королевство сгорело, короли разбежались...

временам и ордам огров! Еще один любопытный момент. Поскольку управление в KUF полностью скопировано со Starcraft, учиться играть практически не нужно. Все примочки "короля RTS" тут, никуда не делась, надо только привыкнуть к новой линии апгрейдов и немного другой схеме горячих клавиш...



Звуки му

Звук и музыка. Как ни странно, все просто отлично. О криках-воплях, издаваемых юнитами, вообще лучше не говорить. Это надо слышать! Вопли, стоящие на поле боя во время штурма базы Dark Forses, достойны занесения в анналы истории RTS. Теперь я знаю, кто умеет хорошо и качественно записывать вопли умирающих гоблинов. Это корейцы.

Однако давайте все же по порядку, сначала о стратегической части игры, потом о ролевой и, под завязку, немного о героях.

Стратегия по-корейски

Стратегическая часть KUF — это забавная смесь Warcraft и Starcraft. Можно до бесконечности мусолить частности, но первое впечатление — самое верное. В принципе, играя с компьютером в Custom, а не в сюжетные миссии, можно попробовать разные стратегические ходы — искусственный интеллект в этой части довольно неплох. Однако очень быстро надоедает: все решается или тотальным контролем над картой (у кого золота больше, тот и зарулил), или быстрой атакой на противника имеющимися юнитами.

В сюжетных миссиях все совсем по другому — тут правят бал скрипты и триггеры. Все действия противника заранее жестко прописаны, и вы можете даже не беспокоиться, атаки всегда следуют через определенные уровни сложности игры промежуток времени и направлены по одним и тем же линиям. Вы можете вообще не держать на базе оборону, отстраивая защитные башни только в тех местах, где вас атакует неприятель. Скрипт есть скрипт, и противник никогда не свернет с намеченной цели, пусть даже если вход на вашу незащищенную базу находится двумя пикселями левее. Кстати, непрошибаемая оборона по версии KUF выглядит крайне просто. Ставим 10 башен, в каждой по 2 лучницы со всеми

возможными апгрейдами, впереди строим баррикаду из ферм. Дырки между фермами затыкаем любыми юнитами (мечники, те же лучницы, рыцари, хоть пезанты, в конце концов). Неплохо иметь немного медиков-священников и мобильный отряд конницы — и все в ажуре!

Вариант при игре за Плохих абсолютно идентичен — выстраиваем в линию десяток Паучьих Башен, которые сильно тормозят движения противника, стреляя в него липкой паутиной, впереди также отгораживаемся фермами, а между башнями и фермами ставим несколько Темных Эльфов, которые от своих предшественников из Warcraft-2 отличаются только цветом кожи. Все те же доходяги, швыряющие топорики, только теперь они не зеленые, а фиолетовые. Одними лишь Темными Эльфами можно забороть Хороших на протяжении первой половины кампании — все, как в старые добрые времена. Так что, если играть грамотно, KUF не доставит вам никаких проблем. Теоретически, за день можно пройти всю кампанию за Хороших.

В сюжетных миссиях AI как своих, так и чужих юнитов просто отвратителен. Солдаты застревают среди деревьев, в "карманах" карты, среди своих собратьев и вообще везде, где только можно застрять. Перевести отряд в 50—60 человек через треть карты — занятие не для слабонервных. На мануальное (ручками и мышкой по 10 раз) возвращение заблудившихся лоботомированных всадников может уйти минут 5—10, а то и больше. Грустно, что и в третьем тысячелетии мы по-прежнему сталкиваемся с детскими болезнями RTS середины 90-х годов прошлого века. Впрочем, в Сети уже появился патч версии 1.07, который исправляет целых 19 серьезных ошибок и немного улучшает искусственный интеллект.

Пройдя пару этапов игры, доблестные герои оказываются нос к носу с ролевым заданием, и один из них, молодой Куриан (а при игре за Плохих это сладкая парочка вампир-Richter и огромный вояка Rick Blood), спускается в подземелье. Соответственно, игра переходит в RPG-модификацию.

Ролевая игра по-корейски

А это уже Diablo в исполнении компании Phantagram к нам в гости пожаловала! Так что, если вы хоть раз играли в Близзардовский хит, все узнаете сразу. На самом деле путешествие по данжеонам и заброшенным дворцам выглядит забавно. Огорчает ограничение по количеству слотов для барахла (много чего приходится выбрасывать), однообразие окружения и малое количество оружия. Но зато — много разнообразных монстров (в RPG-модификации мы можем встретить как юнитов из стратегической части игры, так и уникальных, подземных созданий), много веселых драчек и сильных заклинаний. Баланса просто нет — вы маг и бог, невменяемый монстр, и стоит вам поднакопить маны и понажимать в темпе вальса пару кнопок с заклинаниями, все твари и некромансеры выйдут из подземелья вперед ногами. Кого не удастся завалить мечом (например, толстяка Минотавра), того можно "запилить" до смерти магией.



На самом деле умереть в RPG-моде можно только в первые минуты миссии, дальше ваш герой/герои (их может быть до 3 штук) становятся настолько неприлично круты, что задание превращается в веселую прогулку с перманентным расшвыриванием по углам чудовищ. Весело, ненапряжно и под бодрые корейские песнопения. Это было бы скучно, если бы не волшебные очки опыта, которые сослужат вам добрую службу в следующей стратегической баталии. Кстати, использование лекарств, невидимости и прочих бонусов снижает опыт, поэтому не злоупотребляйте склянками с булькающим! Когда играешь за хороших, в подземельях сплошь одни уроды и орки, а когда за плохих, отку-

да ни возьмись появляются паладины и священники.

Песни о главном

Теперь о самом главном и сокровенном. О балансе. Удивительно, но баланс в демке тоже есть (разработчики долго играли в Starcraft и "просекли" его основную фишку). Толпа Templar Knights вовсе не самодостаточно и не может привести вас к победе. Нехорошие демоны "перечикают" всех с воздуха и останутся абсолютно безнаказанными. Так что храбрым рыцарям неплохо бы обеспечить подобающее прикрытие из лучников и "гром-птиц" (в отличие от Warcraft на птичках теперь летают не дубоподобные мужики, а миловидного вида дамочки). Компьютерный оппонент тоже не отстает — во время первой партии он сделал мне классический "зилот раш" (я еще не знал, что в KUF башни без посаженных внутрь двух лучников не стреляют). В качестве зилотов выступили какие-то злобные создания с дубинками, но эффект был очень разрушительным. Кстати, играет компьютер даже в демке хорошо. Например, он имеет склонность атаковать вашу базу в самый неподходящий момент — во время штурма его базы или обустройства вами дополнительной базы.

Заключение лечащего врача

В заключение хочу сказать следующее. Если вы с удовольствием играли в Warcraft I и II, Starcraft и его клоны, эта игра должна вам понравиться. Kingdom under Fire — это тот самый Warcraft III, который мы так ждали от Blizzard. Она нас обманула, и теперь мы получим гениальную, но СОВСЕМ ДРУГУЮ игру. А вот корейская Phantagram пошла по стопам "большого брата" и переделала древний хит, доведя игру до сегодняшних стандартов (графика, управление, звук и прочее). Это уже не клон, это, с вашего позволения, "ремейк". Похоже, центр разработки "классических" 2D RTS смещается на Восток.

Amicus humani generis

По материалам игрового портала The Daily Telegraph <http://www.dtf.ru>

Рукопись, найденная в ZIP архиве

Туман, белый и невесомый, словно одеяние призрака, поднимался над вересковыми пустошами. Во мраке таинственно перемигивались болотные огоньки, увлекающая неосторожного путника в самое сердце страшной Даблсайдской топи. Однако в этот час дороги были пустыни. Гости поместья Мидитауэр сидели в уютной гостиной. Яркий свет заливал комнату, в камине весело потрескивал огонь, и беседа гостей леди Агаты поначалу не касалась явлений мрачных и загадочных. Первым эту тему затронул полковник МакКиборд, недавно вернувшийся из Индии и теперь не упускавший случая поведать о чудесах Востока.

— Представьте себе, леди и джентльмены, вот стоит перед нами этот факир (полковник, как и многие колониальные офицеры, произносил слово "хакер" на туземный манер). — Ничего абсолютно у него нет, разводит голыми руками. Сюю дискету в дисковод — отформатирована начисто! И это несмотря на write protect!

— Ловкий трюк и ничего более, — заявил лорд Мортимер, третий баронет Хардэнсофт. Этот молодой человек, недавно окончивший Оксфорд, придерживался сугубо материалистических взглядов.

— А знаете, сэр Мортимер, — обратилась к нему леди Агата, вдовствующая баронесса Хардрайв, — не все так просто, как вы думаете, и чтобы убедиться в этом, необязательно ехать в Индию. Как вам известно, я унаследовала Мидитауэр от тетушки, графини Флопинг, которая увлекалась оккультизмом. Здесь даже имеется комната с кое-какой атрибутикой. Мы могли бы прямо сейчас провести спиритический сеанс.

— Отличная идея! — воскликнул Уильям Файлоу. — Возможно, духи помогут мне отыскать, наконец, баг в моей программе.

— И не стыдно вам приставать к духам с такой чепухой? — отозвался

МакКиборд. — Вот у меня действительно важный вопрос: как пройти 30-й этап DOOM2?

И хотя сэр Мортимер заявил, что знает это лучше любого духа, большинство поддержало идею леди Агаты. Баронесса позвонила, и вошел дворецкий.

— Багинс, — сказала леди Агата, — дайте ключ от черной комнаты.

— Но, миледи... — попытался возразить Багинс, однако, видя непреклонное лицо хозяйки, отцепил от связки нужную магнитную карточку.

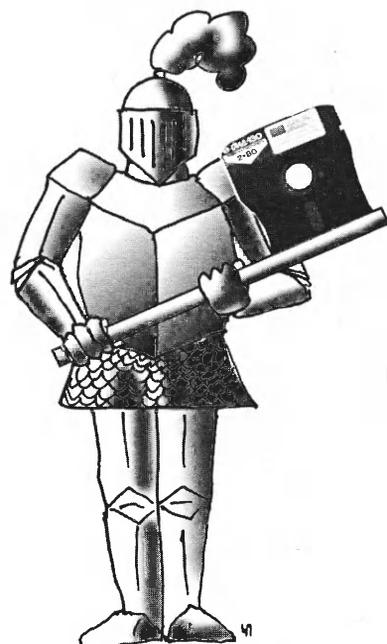
Общество во главе с баронессой Хардрайв спустилось по скрипучей лестнице в подвал и остановилось перед массивной дверью, судя по всему, давно не отпиравшейся. Скригнули ржавые петли; изнутри помещения дохнуло затхлостью. Леди Агата нащарила выключатель и зажгла свет.

Туслая лампочка озарила тесную комнату с заросшими паутиной углами. Повсюду виднелись атрибуты мистических культов: восьмидюймовые дискеты, радиолампа от компьютера ЭНИАК, восковая фигурка Билла Гейтса, пронзенная тремя проводами, а рядом, под стеклянным колпаком, — разноцветное надкусанное яблоко. На полках стояли книги по черной магии. Стерев пыль с одного из корешков, сэр Мортимер различил готическую надпись: "Windows'95 Installation Guide". В центре находился столик из черного дерева; на нем возвышался старинного вида компьютер с двумя пятидюймовыми дисковыми дисками.

— Трешка, SX25, — восхитился доктор Принтеринг, большой ценитель старины. — А то и вовсе 286-я.

Леди Агата включила питание. Зловеще вспыхнул красный огонек; непривычно медленно защелкал Memory Test. Затем на экране возникло изображение клавиатуры и поверх него — курсор.

— Все должны сесть вокруг и взяться за руки, — распорядилась леди Агата. В течение нескольких



минут ничего не происходило, и сэр Мортимер уже хотел произнести что-то насмешливое, как вдруг столик начал дрожать и покачиваться. Курсор шевельнулся.

— Здесь кто-нибудь есть? — спросила леди Агата.

Курсор встал на клавишу Y.

— Кто ты? — продолжала баронесса.

Курсор не двигался.

— Что-то мешает тебе говорить?

Курсор снова указал на Y.

— Что? О-Р-Т...

— Options! — догадался сэр Мортимер и нажал Alt-O. Появилось короткое меню. Курсор выжидательно замер возле строчки Sound On. Сэр Мортимер нажал Enter.

— Вот так гораздо лучше, — донесся из динамика хриплый и неприятный голос. — Но еще не совсем хорошо. У меня нет контакта с этой женщиной. Пусть ее место займет Мортимер.

Баронет усмехнулся, но поменялся местами с леди Агатой и сел напротив монитора.

— Теперь совсем хорошо, — констатировал дух. — Как долго я ждал этого! Мортимер, ты помнишь меня?

— Нет. Кто ты?

— Я дух Cyberdaemon'a из пер-

вого DOOM'a, которого ты убил девять лет назад!

Сэр Мортимер, до сих пор считающий происходящее розыгрышем, вдруг почувствовал страх. Никто из присутствующих не знал, когда он впервые прошел DOOM.

— Не сомневайся, это именно я. Я помню все твои результаты. Kills — 100%, items — 85...

— Но... как ты попал сюда? — пробормотал сэр Мортимер. — Я никогда не играл за этим компьютером!

— А Интернет на что? — удивился демон.

— Чего тебе от меня надо? — нервно спросил баронет.

— Ты еще спрашиваешь? Ты, всаживавший в меня заряд за зарядом? Отмщения!!!

— Но... какие могут быть счета... ты ведь тоже в меня стрелял!

— Я защищался! Ты явился в мой мир без приглашения и стал крушить всех и вся! Но теперь пришел час расплаты!

Изображение клавиатуры исчезло, и на экране возник жуткий паукообразный монстр. Сэр Мортимер попытался освободить руки, но державшие его соседи словно окаменели. Пот струился по их бледным лицам; они не могли пошевелиться.

Монстр поднял свое оружие. Ослепительный сгусток зеленой энергии вырвался из монитора и вонзился в грудь сэра Мортимера. Баронет закричал от адской боли. Он пытался произнести защитное заклинание "IDDQD", но язык не повиновался ему. Участники сеанса с ужасом наблюдали, как с каждым выстрелом демона меняется лицо баронета: оно опухало, глаза заплывали, из носа и рта полилась кровь.

— Знаешь, что будет, когда health упадет до нуля? — разглагольствовал монстр, не переставая стрелять. — Я заберу тебя к нам и сделаю одним из солдат. Тысячи игроков каждый день будут расстреливать тебя из винтовок и пулеметов, разносить на куски ракетами, сжигать плазмен-

ным оружием, пилить заживо бензопилой — о, они любят это делать!

Глова сэра Мортимера бессильно упала на грудь.

Внезапно раздался электрический треск, и экран монитора погас. С басовитым гудением остановился вентилятор. В комнате воцарилась темнота и тишина. Одеревеневшие пальцы людей разжались, и круг распался. Все тяжело дышали. Доктор Принтеринг нащупал пульс лежавшего без сознания сэра Мортимера.

— Кажется, все обошлось! — воскликнул он с облегчением.

Открылась дверь, и вошел Багинс со свечой в руке.

— Извините, миледи, перебои с электричеством. Уже второй раз за последние 20 лет!

— И все-таки нет ничего лучше британского консерватизма! — воскликнула леди Агата. — Иначе моя покойная тетушка давно купила бы UPS!

Юрий Нестеренко
(<http://yun.complife.net>)

Лифчик от Microsoft

О том, что бесконечно уважаемая, почитаемая и любимая пользователями персональных компьютеров корпорация Microsoft, судя по всему, задалась твердой целью довести до инфаркта едва ли не половину потребителей своих продуктов, догадываются многие. Однако жизнь, как всегда, вносит свои коррективы: далеко не все пируэты компании вызывают у очевидцев приступ острой сердечной недостаточности. Некоторые вполне способны привести и к более тяжелым последствиям.

Представьте себе весьма типичную ситуацию: устав от инсталляции и настройки MS Windows, падающей после третьей перезагрузки машины с регулярностью откушавшего полтора литра водки соседа по лестничной площадке, вы принимаете мудрое решение выключить компьютер, чтобы немного отдохнуть и развеяться. Купив для этой цели в близлежащем магазине аппетитный кремный торт,



вы приглашаете на чашечку кофе симпатичную знакомую девушку. И вот, между второй и третьей порцией горячего ароматного напитка, когда в комнате согласно общепринятой традиции становится нестерпимо душно, вы в порыве искренних дружеских чувств избавляете свою гостью от излишков верхней одежды и вдруг видите на нижнем белье вашей возлюбленной красивый лейбл Microsoft. Временная импотенция вследствие тяжелого нервного потрясения вам гарантирована.

На самом деле описанная картина — отнюдь не плод бурной фантазии автора этих строк. Такое, к сожалению, вполне возможно. И корпорация Microsoft здесь, как ни странно, совершенно ни при чем. Однако, обо всем по порядку.

По всей видимости, придумывая название для компании-производителя программного обеспечения, ее основатели полагали, что выпускаемая данной фирмой продукция — программы и операционные системы (soft) — будут достаточно компактными и небольшими по объему занимаемого ими дискового пространства (micro). Вполне возможно, что фигурирующий в данной торговой марке термин "soft" был призван символизировать гибкость в настройках программных пакетов фирмы Microsoft. В те далекие времена никто еще не мог и предположить, что в конце 90-х годов продукция Microsoft станет громоздкой как Эйфелева башня и "жесткой" как ватное одеяло, изготовленное нетрезвыми ударниками социалистического труда трижды краснознаменной фабрики "Ленинградский Комму-

Особенности интернационального сетевого секса

Ленка сидела перед компьютером, уставившись в экран, и с завидным аппетитом поглощала ужин, приготовленный, кстати, мною. Эту картину я наблюдала в течение последних нескольких дней с удручающей регулярностью. Собственно, с того момента, как я переехала в это гнездышко на правах второй квартиросъемщицы, мне ни разу не удалось занять теплое местечко у компьютера и со вкусом пообщаться с сетевыми друзьями. И когда моя подруга спит — не понимаю! Во всяком случае, мне ни разу не удалось ее подкараулить за проявлением естественных человеческих слабостей.

Но сегодня я созрела для бунта. Под угрозой перегрызания телефонного провода я потребовала реализации своих прав интернет-пользователя. Ленка решительно воспротивилась. Она здесь сидела, сидит и

сидеть будет. И вообще, она первая заняла это место. После получасового скандала консенсус был достигнут — мы устроились в одном кресле перед монитором и попытались действовать сообща. Беда была в том, что сферы интересов у нас оказались совершенно разные. А вот мышь одна. Но была бы добрая воля — компромиссы найдутся. Мы взяли Ленкину шляпу и насыпали туда кучу бумажек с названиями серверов, которые нам хотелось бы посетить, и стали тянуть жребий.

— Что это? — спросила я Ленку, вертя в руке бумажку с интригующей надписью Adult Friend Finder.

— Ах, это, — потупила взор моя подруга, — это клуб друзей. Ну, в общем, поиск партнера для сетевого секса.

— И что мы там будем делать? — осведомилась я.

— Разве непонятно? Найдем партнера и займемся виртуальным

нар". Как бы то ни было, вполне очевидно, что оба слова в логотипе корпорации можно трактовать только в переносном смысле. А после появления на рынке самой надежной, быстрой и компактной в мире операционной системы Windows 95 with Microsoft Internet Explorer, название которой продвинутые российские пользователи не замедлили перевести как "95 мелкомягких форточек с прибабасами", любые попытки каким-то образом истолковать название выпустившей ее фирмы утратили всякий смысл.

Вместе с тем, одна из британских компаний, специализирующаяся на производстве женского нижнего белья, в конце 1997 года анонсировала запуск в серийное производство новой линии очаровательных бюстгалтеров и трусиков, изготовленных по принципиально новой технологии. Новизна этой технологии заключается прежде всего в том, что данные предметы дамского туалета

изготавливаются с использованием материи, состоящей из тончайших синтетических микроволокон (micro), благодаря чему готовые изделия имеют на ощупь необычайно мягкую текстуру (soft). Не мудрствуя лукаво, английские промышленники назвали выпущенные с использованием этого материала изделия просто и незатейливо — microsoft, — и принялись злорадно потирать руки в предвкушении многомиллионных дивидендов.

Ни для кого не секрет, что руководство корпорации Microsoft охотно судится со всем миром по поводу и без повода, видимо, в попытке заставить своих высокооплачиваемых юристов честно отрабатывать свой кусок хлеба с толстым слоем кабачковой икры. Другими словами, подобного развития событий юристы Microsoft потерпеть никак не могли. Трезво рассудив, что если дело так пойдет и дальше, то в ближайшем будущем все, кому не лень, нач-



сексом. Возможно, даже групповым, — хихикнула Ленка, проворно набирая адрес в окне браузера.

Сайт оказался американским. Лично меня это смутило. Английский я знаю в пределах школьной программы. То есть могу сказать: "My name is Katya". Дальше перехожу на интернациональный язык жестов. Для того чтобы изъясняться жестами на тему секса, да еще по Интернету,

нужно выпускать презервативы под названием "Windows 98" и туалетную бумагу марки "Outlook Express", юристы Microsoft доели свой бутерброд с икрой и вежливо попросили заокеанских производителей трусиков прекратить это хулиганство. Те внимательно выслушали мнение означенных джентльменов и не менее вежливо послали их на фиг. В воздухе запахло паленым. Кажется, ситуация неизбежно движется к естественному апогею: громкому судебному процессу, в котором речь пойдет о факте присвоения англичанами чужой зарегистрированной торговой марки. А это, следует отметить, грозит проигравшей стороне многомиллионными убытками и жестокими штрафными санкциями.

Однако вскоре выяснилось, что майкрософтовские юристы кушают свой хлеб с продуктами метаболизма кабачков не даром. Прежде, чем передать дело в суд, они постарались досконально разобраться в

надо быть просто ассом. Я в себе таких возможностей не чувствовала. Но Ленка, которая, кстати, еще в школе прославилась стойким отвращением к английской грамматике, была невозмутима.

— Подыщем какого-нибудь симпатичного мальчика. У тебя есть конкретные пожелания?

— Пожалуй, я предпочла бы блондина. Остальное на твой вкус. А как мы будем с ним объясняться?

— Элементарно. Я запущу Stylus. Очень неплохой переводчик. Правда, у меня только какая-то древняя версия. Ну да не беда. А ты поищи на полке словарь.

Пока я рылась в книжном шкафу, Ленка острвенело щелкала мышью. Поиски словаря и партнера закончились одновременно.

— Вот, взгляни какой! Нравится? Швед. Смотри, какой зайчик!

Швед на фотографии был молод и красив, как греческий бог. Золотистые кудри, соболиные брови, бронзовый загар. Вообще, достоинств у него было много. Учитывая, что он был совершенно наг, ни одно из них не осталось скрытым.

— Не обольщайся особенно, —

обратилась она ко мне, поймав мой зачарованный взгляд, — я думаю, мальчик кое-что преувеличил с помощью Фотошопа. Но, в конце концов, это простительно. Я тоже обычно всем мужикам в сети вру, что мои параметры 90x60x90. Мне пустяк, а людям приятно.

После короткого диалога со шведом мы выясняли, что зовут его Клаус, ему 30 лет, он очень-очень крупный бизнесмен и всегда любил русских женщин. Stylus выполнял свои функции вполне сносно. Посоветовавшись, мы с Ленкой решили представиться одной девушкой. Итак, нас звали Карри, нам было 18 лет, а выглядели мы как Памела Андерсон, Клаудиа Шиффер и Наоми Кэмпбелл в одном лице. Еще минут пять мы перекидывались короткими ничего не значащими фразами типа: "How are you" и "I am fine, and how are you?". В конце концов, нам эти реверансы надоели. Видя, что швед не спешит переходить к главному, Ленка взяла инициативу в свои хрупкие руки. "Хотите знать, что на мне надето?" "О, да! Что Вы изнашиваетесь сегодня вечером?". Мы с Ленкой переглянулись.

— Он что, старушек предпочитает? Геронтофил?

— Вероятно, он имеет в виду "носите".

— И что же мы изнашиваем?

Наш ответ не блистал оригинальностью. "На мне легкий ночной халатик. Он светло-розовый".

То есть, по нашему замыслу, эта фраза должна была звучать именно так. В процессе ее создания нам пришла в голову интересная мысль — прогнать этот шедевр через переводчик туда и обратно. После этой процедуры фраза приобрела следующий вид: "Я в индикаторе ночь купальных костюмах, это — световое — розовая видимость".

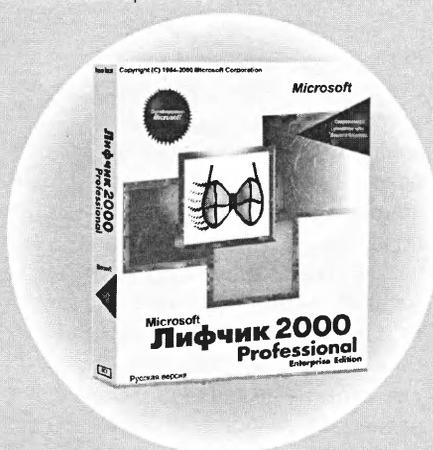
Нам подумалось, что, возможно, мы не единственные, кто пребывает в непонятках из-за ответов партнера. Однако швед не смутился. Похоже, "розовая видимость" его не обескуражила. Или наши ответы просто не имели значения. "А я нахожусь только в коротком замыкании" — радостно возвестил он. Хорошо, что слово "шорты" международное, а то лично у меня возникли неприятные ассоциации с электрическим стулом.

"Я был бы Вы касаться меня Ва-

сути проблемы и пришли к весьма неутешительному выводу: если эта неприятная история закончится судом, его результатом, скорее всего, станет жестокий и коварный облом.

Тонкость ситуации заключается в том, что, согласно международному законодательству и сложившейся процессуальной практике, торговая марка действует только в пределах какого-либо географического региона и в рамках определенного сектора рынка. Ведь владелец персонального компьютера при всем желании не сможет проинсталлировать на свой винчестер лифчик марки майкрософт, даже если предварительно выпьет три ящика пива "Навигатор", название которого, к слову, является частью торговой марки корпорации Netscape. Таким образом, суд вполне может решить дело и не в пользу истца. Так это уже случилось однажды с компанией Digital Equipment Corporation, представители которой неожиданно обнаружили в американ-

ских магазинах пылесос, название которого совпадало с наименованием выпускаемой их фирмой серии компьютеров VAX.



И Microsoft решила уладить скандал миром, де-факто признав право на существование трусиков и лифчиков с подобным наименованием. В ходе переговоров двух корпораций было решено, что торговая марка Microsoft, пишущаяся с заглавной

буквы, безраздельно принадлежит разработчикам программного обеспечения, а такая же марка, но с маленькой буквы — производителям ижнего белья.

Так что если за чашечкой чая вы еожиданно обнаружите, что ваша чаровательная подруга облачена в юстгалтер "microsoft", пугаться того, право же, не стоит: по свидетельствам очевидцев и пользователей, данное нижнее белье, в отличие т одноименных операционных систем, не требует установки драйвов, функционирует стабильно и без глюков. Иными словами, лифчики microsoft исправно расстегиваются, резинка на трусиках не лопаается самый ответственный момент, а сами трусики при попытке от них избавиться не вызывают недопустимую операцию в модуле VXD по адресу F28HC245 с последующей установкой активного процесса и снятием текущей задачи.

Валентин Холмогоров

шими руками..., я хочу штриховать Вашу амбразуру" — не унимался любвеобильный швед. С первой частью фразы все было более или менее понятно. А загадочное "штриховать вашу амбразуру" при ближайшем изучении оказалось совсем не тем, что мы сначала подумали. Выяснилось, что парень всего-навсего хотел ласкать нашу грудную клетку. Вполне законное желание.

"Я поцеловал бы ваши темные соски и ваши участки маршрута..." — изливался Клаус.

— Что это еще за "участки маршрута"? — удивилась я. Пришлось обращаться за помощью к словарю. Оказалось, загадочные "участки маршрута" — это всего лишь "ноги". Никогда бы не подумала.

Ничуть не растерявшаяся Ленка ответила Клаусу, что она, мол, будет думать о нем всю сегодняшнюю ночь. Это привело парня в полный восторг. Ленку было уже не остановить. Она принялась в красках описывать, как, куда и зачем мы будем целовать нашего виртуального любовника. В изложении переводчика это звучало следующим образом: "Я нежен — абсолютно слегка — я веду штырьком (имелось в виду, пальцем) на внутренней области бедренной кости..., слегка я буду касаться задней стороной лапы (тыльной стороной ладони). Здесь мой язык абсолютно номер (рядом). Вы чувствует мои ветры (дыхание)... языком я мягко касаюсь установок с натяжными роликами (уздечки) от различных направлений... передала вал" (непереводимая игра слов, мы так и не вспомнили потом, чем это было в оригинале). После короткой дискуссии это интригующее послание было отправлено несчастному шведу. Наступило молчание.

— Слушай, чего это он умолк? — поинтересовалась Ленка. — Надеюсь, я ласкала не труп?

Я подло хихикнула.

— Прочитав наши откровения по поводу твоего штырька на его натяжном ролике, он точно умер. Или от смеха, или от ужаса. Скорее всего, последнее. Швед молчал. Ленка недоуменно пялилась в экран. Там ниче-

го не происходило. Наконец, Клаус неожиданно разродился тирадой, полной междометий и восклицательных знаков. Мы скопировали фразу в Stylus и стали с нетерпением ждать результатов. Переводчик не обманул наших надежд. Высказывание было убойное: "Ах, я могу чувствовать ваши сырые сливные носки!!!!... Ооо...".

Сказать, что мы были потрясены, значит ничего не сказать. Минут пять мы стонали и икали от смеха. Пришлось даже сбежать на кухню за водой. После короткой консультации со словарем мы установили, что moist lips, переведенные как "сливные носки", на самом деле — влажные губы.

"Я поцелую вашу амбразуру... и Ваш желудок..." — нежно пообещал швед. Я поежилась.

— Далась ему эта амбразура... И, кстати, как он собрался добираться до нашего желудка? Будет делать вскрытие? Ленка успокаивала меня, как могла.

— Да, брось ты, не придирайся. Не забывай, что "амбразура" — это "грудь". Мы же уже выяснили. А "желудок"... наверное, это живот.

— Не учи ученого, — огрызнулась я. — Я и сама прекрасно догадалась.

Возможно, наша словесная перепалка кончилась бы дракой, но тут наш швед учудил кое-что новенькое. "Вы не имеете что-нибудь, с чем Вы могли бы fuck Ваш осел?"

Та-а-ак, теперь еще и скотоложество. Только этого нам не хватало. К счастью, в отличие от целомудренного Stylus, перевод слова fuck мы знали отлично. Но причем здесь осел? Мы схватились за словарь одновременно и снова чуть было не подрались.

— Боже мой, он что думает, у нас тут звероферма? — бормотала Ленка, листая толстую книжку. Мы все-таки нашли там слово ass. Вислоухая скотина оказалась банальной задницей. Ленка воспротивилась решительно. "Никаких ослов!" — заявила она мне. С Клаусом она была более деликатна: "Я предпочитаю мягкий секс". Ответ не заставил себя ждать: "О, да, Кэрри! Я люблю секс программного обеспечения

тоже!". А чего она еще ожидала? Если уж используешь слово "soft", используй его с умом. "Я хочу Вас". Видимо, Ленка решила быть осторожной и действовать наверняка. В такой фразе сложно ошибиться. Но наш швед почему-то замолк, всерьез и надолго. Ленка, обозленная тем, что нас бросили на самом интересном месте, снова отослала ему свое "I want you". Безрезультатно.

Клаус откликнулся только на третий призыв. Это сообщение несколько отличалось по стилю от предыдущих: "Мой отец вернулся домой. Как удалить копию адреса из памяти Internet Explorer? Отец убьет меня, если узнает, что я был на этом сайте".

— Сколько, ты говорила, ему лет? — поинтересовалась я у Ленки.

— Тридцать, — недоуменно ответила она.

— Может, мы перевели неправильно?

— Что мы там могли напереводить? Это же арабские цифры!

— Сейчас переспрошу.

Ответ был лаконичен: "Мне 14. Скорее. Помогите!" С каменным лицом Ленка дала соответствующую техническую консультацию. Я давилась от смеха в уголочке. Хотя, собственно, чего мы хотели? Нам ведь тоже не 18, как мы проанонсировали. Даже в среднем. А уж тем более, если сложить. Минуты через три, закрыв окно чата, Ленка уставилась на меня со странным выражением лица.

— У тебя не будет истерики, если я сообщу тебе еще кое-что?

Сил говорить у меня уже не было. Я только вяло дрыгала конечностями.

— Так вот, я выяснила еще одну подробность из жизни нашего плейбоя. Оказывается, он тоже пользуется Stylus. И зовут его не Клаус, а Коля. И он неплохо поразвлекался сегодня вечером. Я это вытянула из него под угрозой того, что не скажу, как удалить адрес.

Я застонала и убежала в ванную пить воду.

Говорят, с педофилией в Сети борется специальная служба. Интересно, нас посадят за растение несовершеннолетних?

Екатерина Емельянова



Компьютеры **ATLANT**

на базе процессоров
Intel® Pentium® 4
с тактовой частотой
1400MHz



ЧЕТВЁРТОЕ ИЗМЕРЕНИЕ на скорости **1400 MHz**

ATHLETE
на базе процессоров **Intel® Pentium® III**
с тактовыми частотами **450-1000MHz**
от **399 у.е.**

KIDDY
на базе процессоров **Intel® Celeron™**
с тактовыми частотами **366-700MHz**
от **309 у.е.**

mb (ATX), SDRAM от 32Mb,
SVGA от 4Mb, HDD от 4,3Gb UDMA66, FDD 1,44,
CD-ROM от 50-х, Sb 16, kb., mouse.

Настоящая двухлетняя гарантия
Любое изменение конфигураций
Бесплатная модернизация в течение гарантийного срока
Доставка и подключение

www.svegaplus.ru
E-mail: ag@svgaplus.ru



Авторизованный поставщик решений Intel®

ПРИЁМ ЗАКАЗОВ В МАГАЗИНАХ И СЕРВИС-ЦЕНТРАХ

без обеда с 10 до 19, в субботу и воскресенье с 11 до 18

наб.р. Фонтанки, 120
тел. 259-9109, 259-9107, 251-1872, 251-1892
в.о. 9-я линия, 56 тел. 327-4630

Поголовно Intel Inside и Pentium являются зарегистрированными торговыми знаками, а Svegaplus - торговым знаком Intel Corporation



Взгляд
из
Петербурга

LINUX.
Сага о пингвинах

ГИБЕЛЬ
компьютерной цивилизации



Оформить подписку на журнал "Магия ПК" с любого номера вы можете в редакции по адресу: С.-Петербург, пр. Стачек, 47, местный тел.: 93-59

Оформить подписку на I полугодие 2001 г. (86 руб.)

можно в любом почтовом отделении по каталогам "Прессинформ" и "Роспечать".

Подписной индекс журнала 29961.

Сайт журнала "Магия ПК" находится по адресу:

<http://www.magicpc.spb.ru>