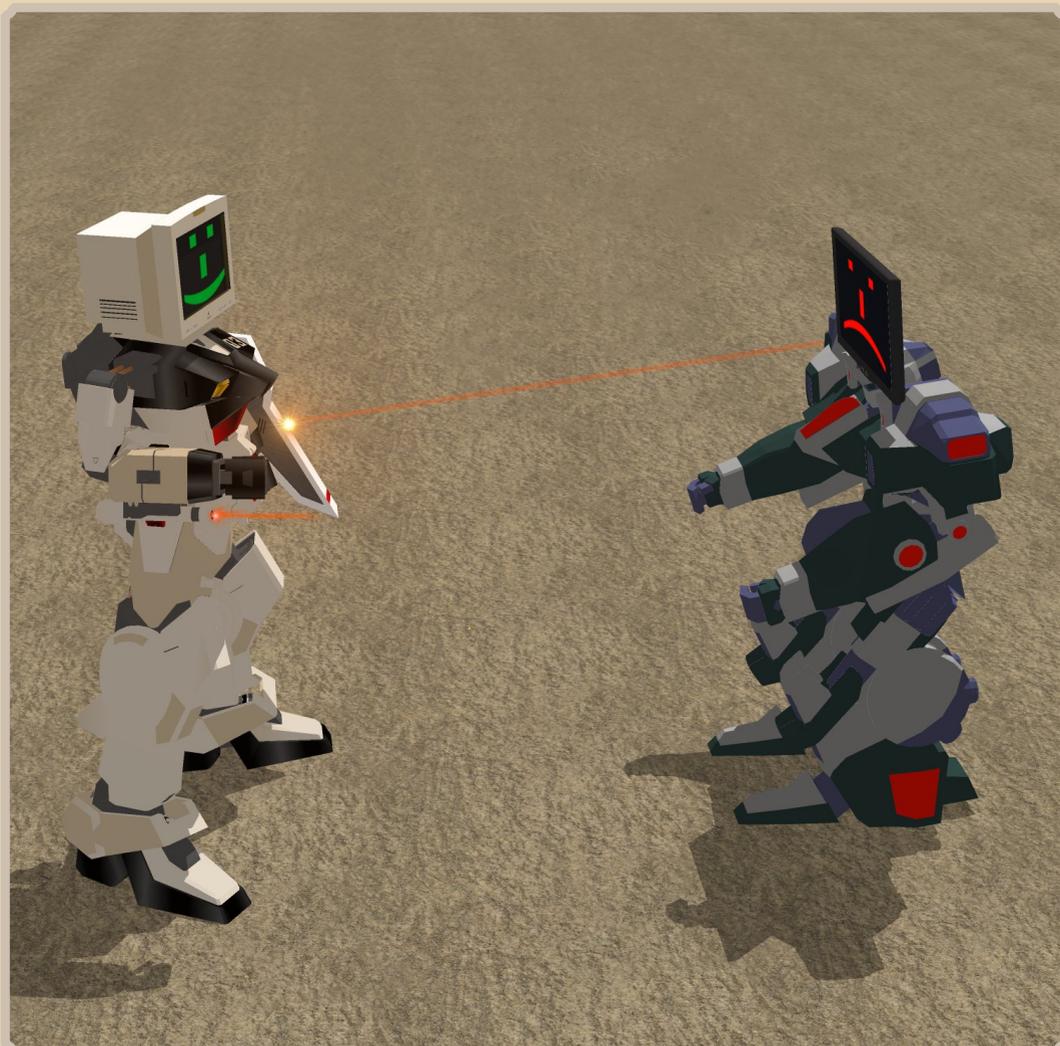


DOWNGRADE



027'2019



СОДЕРЖАНИЕ

● Обложка -----	1
● Содержание -----	2
● От редактора -----	3

ТЕОРИЯ DOWNGRADE

● Новости, события, комментарии (uav1606)-----	4
● Интервью с В.Ф. Фёдоровым (В.Ф.Фёдоров, uav1606)-----	6
● ОГАС, или Saga об электронном социализме (Олег Павлов)----	16

КНИЖНАЯ ПОЛКА

● BIOS: показать всё, что скрыто (А. Шаронов) -----	25
---	----

DOWNGRADE-ЖЕЛЕЗО

● Вверх дном (Андрей Шаронов aka Andrei88)-----	27
● Crystal - дешевле некуда (uav1606)-----	30
● «Реванш» бюджетной тройки (А. Шаронов)-----	33

DOWNGRADE-СОФТ

● Программы-тесты для Windows 3.1 (А. Шаронов)-----	38
---	----

ПРОГРАММИРОВАНИЕ

● Разработка программ для текстового редактора NewsMaster («Журналист»). Ч. 1 из 3: клавиатурный драйвер (SysTools) --	55
● Знакомство со SWEET 16 (Sh)-----	64

ИНТЕРНЕТ И СЕТИ

● Субъективное мнение о фидо (Андрей Тумилович)-----	70
● Календарь по мотивам истории Фидонет (В.Фёдоров) -----	73

СТАРЫЕ ИГРЫ

● Игры и обучение подрастающего поколения (П.Ярославцев) ---	81
--	----

РАЗНЫЙ ЮМОР

● Просто разный юмор -----	84
● Над номером работали -----	86

От редактора

Здравствуйтесь, уважаемые читатели.

Итак, перед Вами №27 журнала Downgrade. Тема номера – всё о старом компьютерном «железе». Правда, к сожалению, по теме номера статьи прислали только Андрей Шаронов и я... В последнее время меня вообще терзают смутные сомнения – стоит ли делать тематические номера, если в результате оказывается, что тема номера – несколько эфемерная вещь, так как статей по ней обычно половина или меньше? Да и ещё от одного-двух авторов. Ну да ладно, вернёмся к этому вопросу позже.

А так – номер получился, как мне кажется, неплохой, объём, я бы сказал, даже рекордный. Правда, больше за счёт нескольких больших статей, а общее их количество невелико.

В этом номере у нас новый автор – **SysTools**, тут опубликована его интересная статья про текстовый редактор «Журналист» и про создание для него вспомогательных утилит – советую почитать.

Есть пара статей про Fidonet, интервью (наконец-то) и много ещё чего (хотелось бы думать) интересного.

Как обычно, если у Вас есть статьи, предложения, замечания и т.п. – пишите мне на e-mail [uav16060 \[Gastropoda\] mail.ru](mailto:uav16060@Gastropoda.mail.ru)

uav1606

НОВОСТИ, СОБЫТИЯ, КОММЕНТАРИИ

20 лет ЖЖ

15 апреля исполняется ровно двадцать лет с того дня, когда американский программист Брэд Фицпатрик зарегистрировал домен livejournal.com и запустил свой сервис ведения блогов. Точнее, сам сервис начал работу ещё 18 марта, но тогда им пользовались только друзья Брэда, а 15 апреля 1999 года «Живой журнал» стал публичным.

Несмотря на засилье социальных сетей и уменьшение популярности блогов, «Живой журнал» продолжает развиваться, публикует интересные материалы, а его преданная аудитория желает ему и дальше оставаться «на плаву».



Мод для Doom от Джона Ромеро

Один из создателей Doom – Джон Ромеро – 31 мая выпустил долгожданный мод для этой игры. Скачать SIGIL Megawad совершенно бесплатно можно здесь:

<https://www.romerogames.ie/sigil>

Или прямая ссылка на скачивание:

https://www.romerogames.ie/s/SIGIL_V1_1.zip

Мод включает целый дополнительный эпизод с девятью новыми уровнями.

SIGIL Megawad требует наличия установленного Doom с оригинальным файлом **DOOM.WAD**. У меня этот мод с DOS-версией Ultimate Doom v1.9 работал нестабильно, пришлось скачивать неофициальное обновление **doom.exe** – Doom+ v1.92 ([ссылка](#)).

Впрочем, с современными адаптированными вариантами движка вроде GZDoom мод работает вполне нормально (именно из него сделан скриншот выше, использовался файл **SIGIL_COMPAT.wad**).

Обновление безопасности для старых ОС от MS

Корпорация Microsoft выпустила обновление безопасности, исправляющее критическую уязвимость CVE-2019-0708 в коде Remote Desktop Services. Необычность этого апдейта в том, что он выпущен, в том числе, и для снятых с поддержки операционных систем – Windows XP и Windows Server 2003.

Подробнее [здесь](#).

Скачать обновления для XP и 2003 можно здесь:

<https://www.catalog.update.microsoft.com/Search.aspx?q=4500331>

50 лет первому RFC

7 апреля 1969 года был опубликован Request for Comments: 1 – первый из множества ныне существующих документов RFC, которые описывают различные стандарты и технологии, применяющиеся в сети Интернет (а до этого – в ARPANET).

RFC-1 был создан Стивом Крокером, тогда – аспирантом Калтеха.

Ознакомиться с этим документом можно здесь:

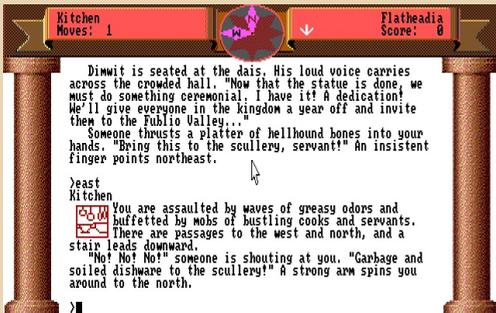
<https://tools.ietf.org/html/rfc1>





Закончилась поддержка Windows POSReady 2009

9 апреля 2019 года закончилась расширенная поддержка Windows Embedded POSReady 2009. Эта ОС является, по сути, разновидностью Windows XP для различных встраиваемых решений (банкоматы, платёжные терминалы и др.). После завершения поддержки «обычной» Windows XP только POSReady 2009 какое-то время продолжала получать обновления, причём существовал «хак», позволявший замаскировать Windows XP под POSReady, продлив таким образом возможность получать апдейты для XP. Сейчас этот «фокус» уже не сработает – фактически, POSReady 2009 была последней ОС семейства XP, которая ещё хоть как-то поддерживалась Microsoft.



Опубликованы исходники старых игр от Infocom

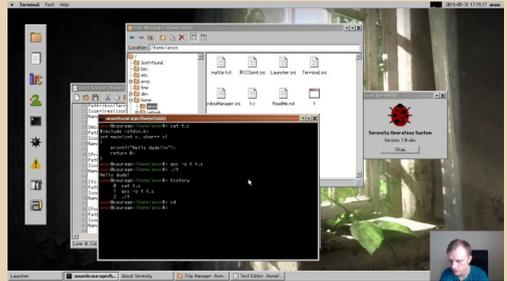
В число открытых исходных кодов попали Zork Zero, Zork I, Zork II, Zork III, Arthur, Shogun,

Sherlock, Witness, Wishbringer, Trinity и The Hitchhiker's Guide to the Galaxy. В основном это игры 70-х – 80-х годов в жанре текстовых квестов. Впрочем, последние части имели и графическое наполнение.

Код написан на специально созданном для этого языке ZIL (Zork Implementation Language).

Посмотреть исходные коды можно здесь:

<https://github.com/historicalsourc?tab=repositories>



SerenityOS

Andreas Kling с недавних пор занимается разработкой собственной Unix-подобной операционной системы – SerenityOS. Её отличительной особенностью является графический интерфейс в стиле 90-х. Система написана на C++ и распространяется (в виде исходников) под лицензией BSD. Познакомиться с ней поближе можно тут:

<https://github.com/SerenityOS/serenity>

Обзор подготовил iav1606



ИНТЕРВЬЮ С ВИКТОРОМ ФЕДОРОВИЧЕМ ФЕДОРОВЫМ



Вашему вниманию предлагается интервью с Виктором Фёдоровичем Фёдоровым – одним из разработчиков компьютера «Союз-Неон» (см., например, номер 17 журнала), также Виктор Фёдорович много лет был связан с IT-индустрией, поэтому может рассказать немало интересного.

uav1606: Расскажите немного о себе – как Вас зовут, сколько Вам лет, где родились и живёте?..

В. Фёдоров: Зовут меня Виктором Фёдоровичем Фёдоровым, мне 69 лет.

Родился 01.11.1949 в хуторе Лихом Красносулинского (в то время – Зверевского) района Ростовской области.



На фото – памятник на окраине хутора и мои земляки в день его открытия.

С 1980 г. живу в Зеленограде, куда попал в 1979 г. по распределению по окончании института.

Заканчивал я 2-й МОЛГМИ им. Н.И. Пирогова (ныне – ФГБОУ ВО РНИМУ им. Н.И. Пирогова Минздрава России – <http://rsmu.ru/>), кафедру медицинской кибернетики и информатики, в первом выпуске врачей-кибернетиков (http://rsmu.ru/mbf_cmci.html). Поскольку наша специальность ещё не была утверждена, то в дипломе значилась как «биофизика с присвоением квалификации врача-кибернетика».

Я – доктор медицинских наук по специальности «03.01.09 – Математическая биология, биоинформатика». Тема докторской – «Методология анализа динамики количественных параметров функциональной диагностики сердечно-сосудистой системы».

Начнём, пожалуй, в хронологическом порядке – как Вы познакомились с миром компьютеров? В какие годы это было, что это были за машины? Как развивалось Ваше знакомство с ЭВМ?

На кафедре математики факультета была ЭВМ «МИР-1», но студентов к ней не подпускали (разве что посмотреть). ;-)

В вычислительном центре института была машина «ЕС-1010» (советский аналог младшей модели IBM-360). Машина класса мэйнфрейм стояла в машзале с кондиционированием, и студентов туда не пускали. В ходе обучения программированию на кафедре мы изучали языки IBM System/360 Basic Assembler Language,



PL/1, COBOL и Fortran, однако учебные программы на специально размеченных листах сдавались в табуляторную, где их превращали в стопки перфокарт (не мы), а прогон шёл в ночное время без нас. В результате студент мог получить на руки листинг с комментарием, что программа «не прошла». Причиной могла быть ошибка табуляторщицы, потеря одной из перфокарт при передаче из табуляторной в машзал или при прогоне, а также ошибка самого студента. И в этом надо было разбираться.

Дипломную работу я выполнял в лаборатории биофизики ЦНИИ рефлексотерапии, который был создан в 1977 г. (см. http://www.tradmed.ru/n8_1.shtml). Темой дипломной работы была «Автоматизация пульсовой диагностики». В лаборатории была ЭВМ «Электроника-60» ([фото из Википедии](#) – аналог младшей модели PDP-11) с терминалом [VT-340](#).

К моменту окончания моей дипломной работы в ЦНИИР появился легендарный Plurimat S (см., напр., <https://alple.net/plurimat/art.htm>), у которого был 64-канальный АЦП, пульт микропрограммирования (например, кнопка «FFT»), монохромный векторный дисплей Tektronix и термопринтер. В работе было показано, что спектры суммарного сигнала, содержащего сфигмоартериограмму при измерении артериального давления и звуки Короткова, существенно различаются у здоровых лиц и пациентов, проходящих лечение по избавлению от табачной зависимости.

Работая в ЦНИИР над дипломом, я одновременно участвовал в ряде НИР по заданию ЦКБ «Дейтон». В результате меня пригласили работать в отдел медэлектроники этой организации, оформили мне московскую прописку (а в то время это было очень сложно!) и вскоре после окончания института предоставили жильё.

В ЦКБ «Дейтон» я вначале попал на «downgrade».

В лаборатории, в которой я начинал работу в «Дейтоне», стоял ИВУК на базе «Электроника-100И». Лаборатория в то время находилась в полуподвале 3-й горбольницы и

работала в тесном сотрудничестве с лечебным учреждением.

Увы, не помню, откуда у меня это фото. Гугл не находит источника.



Как видно на фото, слева стоит собственно ЭВМ с тремя дополнительными блоками памяти по 8 К на ферритовых кольцах (без них – 8 К собственной памяти из микросхем 155 серии, вероятно, K155PY2 – <https://microshemca.ru/RU2/>).

Ввод осуществлялся с терминала Consul или со считывателя перфолент, вывод – на тот же Consul или перфоратор.

«Электроника-100И» имела два языка – Ассемблер и Фокал. Была ещё возможность программирования непосредственно в машинных кодах. Но ими владела только одна сотрудница – Елена Саак (на снимке – сидит за «Консулом»). Большая часть программ писалась на Фокале Фёдором Борисовичем Генкиным.

Измерительно-вычислительный и управляющий комплекс, несмотря на его скромные возможности, использовался для разработки средств рефлексодиагностики на основе [метода Накатани](#) и лечения методами электропунктуры.

По мере набора статистики диагностических данных на пациентах 3-й горбольницы мы начали разработку алгоритмов автоматизированной диагностики на основе методов многомерной статистики. Естественно, это делалось на арендованных вычислительных средствах «на стороне» с помощью пакета программ SPSS



для ЭВМ системы «Ряд» (аналогов IBM-360). Так, дискриминантный анализ позволил нам на основании измерения проводимостей в точках акупунктуры ставить диагнозы ряда заболеваний с достоверностью выше 85%.

Вы в своё время много работали с серийой ДВК – можете рассказать об этом подробнее? Что это были за ЭВМ, где применялись, какая периферия использовалась и так далее.

Поскольку работа по созданию и применению средств рефлексодиагностики/рефлексотерапии шла успешно, начальство стало возить к нам на обследование и лечение всё более высокопоставленных пациентов. Дошла очередь и до Владислава Григорьевича Колесникова ([https://ru.wikipedia.org/wiki/Колесников, Владислав Григорьевич](https://ru.wikipedia.org/wiki/Колесников,_Владислав_Григорьевич)). В то время он был первым замом у министра электронной промышленности (МЭП) А.И. Шокина (в 1985 г. стал министром).

Электропунктура – процесс неспешный, временами пациенты лежали под электродами до 40 минут, курс лечения – 10 – 12 сеансов. Чтобы профессору, д.т.н. Колесникову не было скучно лежать без дела, я рассказывал ему о наших исследованиях (а кроме рефлексодиагностики/рефлексотерапии мы занимались разработками в области кардиополиграфии), сетовал на проблемы с обработкой информации. Вначале Владислав Григорьевич считал, что для наших задач достаточно того, что есть, но когда я ему стал рассказывать о необходимости вычисления в реальном времени кросскорреляционных функций 24 временных рядов, изменил свою точку зрения. Он сообщил мне, что сейчас (это был 1981 год) на предприятиях МЭП готовится к выпуску новое изделие – диалоговый вычислительный комплекс – ДВК (аналог «Электроники-60» с терминалом VT-340), и мы можем подать заявку на несколько первых машин, а он подпишет. Я запросил десяток машин, Владислав Григорьевич с улыбкой урезал мой аппетит до семи. Я сообщил об этом директору «Дейтона»

Б.В. Тарабрину, и он запустил процесс получения техники. Бумажная процедура прошла на удивление быстро, и наш завлаб Александр Павлович Михайлов с несколькими инженерами отправились на «Ангстрем» получать невдомые ДВК в минимальной комплектации, т.е. вообще без периферии, не считая дисководов 5" флоппи-дисков, с которых загружалась операционная система (RT-11, см., напр., <https://refdb.ru/look/3260131.html>) и пользовательские программы и на которые выводились результаты обработки.

Получили, привезли, радости не было предела, но была она недолгой. Из 7 полученных ДВК включились и заработали только 2. Как такое могли пропустить в ОТК?! Открыли паспорта, а там вместо штампа ОТК по диагонали стоял чернильный штамп «лабораторный образец». Позвонили на «Ангстрем», а там ответили, что других пока нет, но не выполнить указание замминистра они не решились. ;-)

Вскрыли корпус неработающего ДВК и внутри увидели бумажку с надписью «Монтаж вела группа ПТУ №..., дата, время, преподаватель...» и торчащие проводки, которые то там, то там не успели припаять.

Ещё раз созвонились с производителями и договорились совместными усилиями довести изделия до рабочего состояния, что и было вскоре сделано.

Контакты с высокопоставленными пациентами не прошли даром, и руководство «Дейтона» вскоре добилось расширения отдела медтехники до отделения с увеличением штатной численности более чем втрое.

Соответственно, закупили и более мощную вычислительную технику – «Электронику 100/25» (так же, как и «СМ-4» другого министерства, это был аналог ДЭЖовской PDP 11/40, см., напр., [здесь](#)). В нашем комплекте было ОЗУ на микросхемах на 256 К, привод магнитных лент на 12 МВ, дисковод сменных жёстких магнитных дисков (кажется, на 1,2 МВ), привод 8" флоппи-дисков (дисководы из Абовяна, дискеты – «Изот» из Болгарии). Армянские дисководы вначале не работали. Мы с инженером



Сашей Кириллиным посветили в щель медицинским лазером (красным, гелий-неоновым) и увидели, что головка выступает над уровнем дискеты миллиметра на 3! Разобрали, подкрутили, выставили по лазерному лучу, зафиксировали, дисководы заработали.

Потом появились и более поздние модели ДВК, уже с цветными дисплеями, двумя 5" дисковыми на 2 дискеты каждый и подсистемой ввода/вывода с других устройств. Дискеты были и отечественные, и «Изот» на 720 КБ.

Насколько я знаю, Вы участвовали в разработке и продвижении компьютера «Союз-Неон» – расскажите, пожалуйста, про это – как компьютер проектировался, как внедрялся, что за люди принимали участие в разработке, какие были сложности и т.п.

Разработка многорадиального компьютера «ПК 11/16» проходила многоэтапно, и к ней имели отношение специалисты-энтузиасты из разных структур.

Это начиналось в период «перестройки», когда ещё существовал Советский Союз, но уже появились новые формы хозяйствования, в частности, научно-производственные кооперативы. Поскольку постановлением Совмина СССР им было разрешено переводить наличные деньги в безналичные и обратно (что было категорически запрещено государственным предприятиям!), руководство Научного центра Зеленограда увидело в этом возможность привлечения молодёжных творческих коллективов для решения инновационных задач, не принимая их в штат (штатная численность НИИ была нормативно ограничена!), и решило создавать такие кооперативы, включающие в себя как сотрудников предприятий, так и «сторонних» специалистов. Одним из сотрудников НЦ, которому было поручено создание таких кооперативов, был Вячеслав Владимирович Бердник. Запустив такой процесс для нужд НЦ, он обратился ко мне (т.к. мы ранее были знакомы как врач и пациент) с предложением создать кооператив в сфере разработки методов и средств

неинвазивной диагностики и немедикаментозного лечения. Мы привлекли ещё ряд специалистов различных профилей, и в конце 1988 года создали кооператив «МЕДИОН», что расшифровывалось как «МЕДИко-Информационное Обслуживание Населения». Председателем, конечно, избрали Бердника.

На первых порах наш кооператив был вынужден заниматься зарабатыванием денег, т.к. разрабатывать что-либо бесплатно невозможно, а госфинансирования кооператорам не полагалось. В то время «челноки» уже начинали штучно завозить первые персональные компьютеры класса IBM XT с минимальной периферией (системный блок, монитор, клавиатура, мышь, принтер) и продавать их здесь за наличные в разы дороже, чем приобретали. Мы их перекупали и предлагали научным организациям по безналичному расчёту с хорошей наценкой. И всё равно это всем было выгодно. Дело было в том, что в плановой экономике ценообразование было нормативным, и цена ввозимых «персоналок» определялась путём сравнения их вычислительной мощности и объёма оперативной памяти с советской вычислительной техникой. Поскольку ввозимые IBM XT были несколько «мощнее» советских «СМ-4», а те стоили ~ 200–250 тысяч безналичных рублей за комплект (напомню: зарплата «среднего» инженера была на уровне 150-170 рублей, а автомобиль «Волга» стоил 10000 рублей), то покупка «персоналки» за ~ 100 тыс. у кооператоров радовала начальство институтов.

Получив на такой «спекуляции» начальный капитал, мы решили перейти к собственным разработкам.

Наш председатель кооператива В.В. Бердник был знаком со многими интересными людьми в Зеленограде, и он привёл к нам Александра Ивановича Тарана и Марата Ахмедовича Хаджихметова, работавших где-то в системе Научного центра. Александр Иванович предложил поддержать идею коллектива Вигдорчика Г.В., разработавшего проект персонального компьютера на базе отечественной комплектации (см. [здесь](#)). Видимо,



разработчики не смогли сами найти финансирования, и искали союза с «новыми буржуями» в лице кооператоров, и Таран предложил им помочь к обоюдной выгоде.



Марат Ахмедович
Хаджихахметов

Бизнес-идея была в том, что можно переработать исходный проект под низкопотребляющий микропроцессорный комплект K1806 (см. <http://155la3.ru/k1806.htm>) и начать массовый выпуск компьютера на заводе «Квант», у которого к тому времени возникли проблемы с загруженностью производства. Нужно заметить, что серия K1806, хотя и была функциональным аналогом микросхем фирмы DEC, не имела прямых зарубежных аналогов как по низкому уровню энергопотребления (исходно разрабатывалась для «бортовой» аппаратуры), так и по некоторым функциям. Учитывая, что в Зеленограде в это время производились «учебные» компьютеры в клавиатурном блоке (см., напр., [здесь](#)), предлагалось снизить затраты на конструирование и производство, договорившись с разработчиками корпуса о небольшой доработке, позволяющей не только вставить в этот корпус новую плату, но и разместить в нём дисководы 3,5".

Мы провели ряд встреч с Вигдорчиком (ведущим экспертом страны по операционным системам для ЭВМ фирмы DEC и аналогов) и Климковичем, взявшим на себя разработку

схемотехники нового компьютера. Согласие было достигнуто, и проект начался.

Основным координатором проекта стал А.И. Таран, решение вопросов производства взял на себя М.А. Хаджихахметов, а я координировал работу со стороны «МЕДИОНА» (участие в разработке ТЗ, контроль множества бумаг по разработке и подготовке к производству и т.п.). Исходное название проектируемого компьютера было «ИОН» (Информационное Обслуживание Населения).

События в стране в то время развивались бурно, и большая часть коллектива нашего кооператива оказалась одновременно сотрудниками ведущего предприятия энергетики СССР в области наладки оборудования «Союзтехэнерго», на базе которого мы, с одобрения и при поддержке замминистра энергетики по науке и новой технике (позже – министра Минтопэнерго РФ) Анатолия Фёдоровича Дьякова ([биография](#)), начали создание Межотраслевого медико-инженерного центра Минздрава и Минэнерго СССР (ММИЦ «Медэнерго»), призванного взять на себя заботу об оценке и восстановлении функционального состояния оперативного персонала всей энергетики. Тема была сверхактуальной после событий на Чернобыльской АЭС.

В Минэнерго разрабатывалась всесоюзная программа по созданию около 400 учебно-тренажёрных центров, в которых весь оперативный и диспетчерский персонал отрасли должен был проходить ежегодную переподготовку и контроль профессиональных знаний и навыков. Директор ММИЦ «Медэнерго» Анатолий Анатольевич Талалаев (одновременно – член нашего кооператива) договорился с А.Ф. Дьяковым о включении в состав этих центров подразделений, которые, одновременно с профессиональной переподготовкой, должны были проводить тщательное обследование физиологического и психологического состояния персонала и, при необходимости, его функциональную реабилитацию. Для этих целей нужно было разрабатывать автоматизированные методики диагностики и реабилитации, а также



технические и программные средства их реализации. Эти функции были возложены на ММИЦ «Медэнерго», где я, в должности замдиректора по науке, координировал разработки. Нами, с привлечением специалистов из других организаций, была разработана и утверждена на отраслевом уровне «Концепция обеспечения надёжности деятельности и сохранения здоровья работников топливно-энергетического комплекса России» и программа по её реализации.

Нам было понятно, что для оснащения всех 400 УТЦ понадобится большое количество персональных ЭВМ, и мы полагали, что наша разработка выполнит эту функцию.

По ходу работ стало очевидным, что средств кооператива может не хватить, и А.А. Талалаев обратился к руководству «Союзтехэнерго» за финансовой поддержкой, аргументируя это широкими перспективами внедрения. После ряда многосторонних переговоров согласие было получено, и ещё не созданный компьютер получил новое название – «СОЮЗ-ИОН», в котором отразились части названий финансирующих и координирующих структур.



Анатолий Анатольевич Талалаев

Для компьютера были необходимы новые программные средства, позволяющие опера-

тивно разрабатывать как собственно тренажёрные программы, так и программные средства оценки и коррекции состояния персонала энергопредприятий. А.И. Таран предложил привлечь коллектив молодых и талантливых программистов из Зеленограда во главе с Александром Гречишкиным, уже успевших проявить себя в ряде разработок, в т.ч. ПО для создававшегося в Зеленограде туннельного микроскопа. Так началась работа над программной средой «Аспект», которая должна была стать частью разрабатываемого проекта.

(Подробнее смотрите в №17 журнала Downgrade – прим. ред.)

Вскоре новые веяния в развитии государства привели к тому, что руководство «Союзтехэнерго» решило выделить «Медэнерго» в дочернее малое предприятие и передать ему проект нового компьютера для дальнейшего развития.

Однако отношения между «мамой» и «дочкой» на фоне новых принципов хозяйствования в стране стали постепенно портиться, и мог возникнуть спор о коммерческих правах на создаваемый компьютер. Чтобы минимизировать возможный ущерб для проекта от такого спора, организационная документация проекта была существенно изменена, и возникло (не без юмора) очередное название – «СОЮЗ-НЕОН», как бы отрицающее предыдущие работы и финансирования проекта.



ПК-11/16 «Союз-Неон» – фото с сайта Сергея Фролова <http://www.leningrad.su>

Именно под этим названием компьютер собирались выпустить на заводе «Квант», где уже была начата подготовка к производству и выпуск установочных партий.



Но, как гласит старая шутка: «Хочешь насмешишь Господа – расскажи о своих планах».

После развала СССР стали разваливаться отрасли с единым планированием и управлением. Программа создания 400 УТЦ была свёрнута ввиду отсутствия финансирования: внебюджетные фонды развития отраслей, в которые все предприятия отчисляли средства на прикладную отраслевую науку, были «экспропрированы» в бюджет правительством Гайдара, в задачи которого входила приватизация энергопредприятий и сетей, а «частникам» никакая наука не была нужна.

Вслед за этим начался массовый ввоз в страну IBM-подобных компьютеров, и руководство «Кванта» решило начать их сборку из комплектующих, ввозимых из Европы. «Кванту» стало не до нашего компьютера. И зря. На тот момент он был вполне конкурентоспособным, а ввоз дешёвых персоналок с Тайваня похоронил идею сборки на «Кванте», т.к. цены европейских комплектующих сделали производство нерентабельным.

По слухам, на «Кванте» ещё какое-то время «подпольно клепали» и продавали приверженцам идеологии DEC наши компьютеры, но я уже не имел к этому никакого отношения.

Возможно, Вы участвовали в разработке или внедрении ещё каких-то компьютеров советского или раннего постсоветского периода?

Нет, не участвовал.

Если не ошибаюсь, Вы имеете отношение к компьютерной медтехнике. Можете рассказать, как шёл процесс компьютеризации медицинской сферы, где и какие компьютерные системы внедрялись, для чего использовались, как это всё менялось со временем?

Ответить на этот вопрос коротко не получится. Это – тема для серьёзной диссертации. Если говорить совсем коротко, то в компьютеризации медицины исторически можно выделить три направления:

1) Информатизация управленческой деятельности (или организации здравоохранения). И вначале она велась на «больших» ЭВМ – аналогах IBM-360, а затем, по мере развития вычислительной техники, «мигрировала» на «малые» машины и на сети персональных ЭВМ.

2) Информатизация собственно врачебного труда (или медицинских технологий). В этом направлении существовали и множились сотни, если не тысячи разработок автоматизированных рабочих мест (АРМ врача) различной сложности и направленности. Простейшие из них смыкались с первым направлением и служили для минимизации рутинных операций в труде врача (заполнение и ведение различных документов), а были и настоящие шедевры хайтека на тогдашнем уровне. Например, уже на уровне программ для ДВК был разработан комплекс, позволявший рассчитать оптимальную программу многоосного гамма-облучения опухоли пациента сложной формы для минимизации поражения здоровых тканей и максимизации поражения раковых клеток.

3) Третье направление связано с «делокализацией интеллекта» при регистрации диагностической информации или при управлении лечебными процедурами. Вначале это были измерительные, управляющие или измерительно-управляющие комплексы на базе «малых» ЭВМ (например, «СМ-4» или Plurimat S), но по мере развития и совершенствования микропроцессоров они стали встраиваться непосредственно в диагностическую аппаратуру и выполнять функции управления измерениями и предварительной обработки информации. Наиболее яркими представителями этого направления являются томографы – рентгеновские (КТ), магнитно-резонансные (МРТ), позитронно-эмиссионные (ПЭТ), мультиспиральные компьютерные (МСКТ). Мощности вычислительных средств МСКТ, например, хватает для того, чтобы «увидеть», построить и отобразить на дисплее работу сердца и его сосудов в 3D-образах, что позволяет оценить риск инфаркта миокарда без опасной процедуры коронарографии.



Естественно, предложенное деление, как и любая классификация, условно. В современных медицинских информационных системах (МИС) эти направления переплетаются и сливаются. Если посмотреть описание сегодняшних МИС (см., напр., [здесь](#)), то мы увидим в них все названные составляющие.

В два последних десятилетия, в связи с появлением и развитием телемедицинских технологий, происходит слияние информационных, телекоммуникационных и собственно медицинских (т.е. диагностических и лечебных) технологий. Но это – предмет отдельного разговора.

Может быть, вспомните какие-то (связанные с компьютерами) забавные случаи из Вашей практики?

По этому поводу вспоминаются начальные стадии внедрения IBM-подобных персональных компьютеров в практику исследователей и разработчиков.

Понятно, что вместе с импортными компьютерами пришло и англоязычное программное обеспечение, а всем нужно было писать отчёты, статьи и т.п. на русском языке. Однако под MS-DOS никаких русскоязычных редакторов ещё не было. И вот в ВЦ Академии наук СССР началась лихорадочная работа над соответствующим ПО.

За первый русскоязычный текстовый редактор (его назвали «Лексикон») взялся Евгений Николаевич Веселов (<http://1803150.mya5.ru/>), тогда ещё просто Женья – интеллектуал-энтузиаст «с горящими глазами». Наш директор А.А. Талалаев жил с ним на одной лестничной клетке и постоянно общался, в т.ч. и по поводу новой разработки. В результате мы одними из первых получали «сырые» версии «прямо со сковороды», которым были несказанно рады – ещё бы, ведь мы могли печатать и хранить русскоязычные документы!

Но Женья никак не мог остановиться в увлечении своего продукта, в результате, каждые две недели нам приходилось перечисляться новому раскладу функциональных клавиш, что сильно сбивало с толку при напряжённой работе.

В то же время мы были в восторге от возможности «откатить» назад при каких-либо ошибках в тексте, или при необходимости сделать вставку в уже напечатанный текст: «Лексикон» автоматически сохранял предыдущие состояния работы (сейчас это кажется тривиальным, а тогда это было «революцией»). С ловушкой этой возможности мы столкнулись в самое неподходящее время, когда нужно было срочно сдавать большой отчёт.

Поскольку работать приходилось за разными машинами (техники ещё не хватало), то мы, обычно, держали свои тексты на дискетах и переходили с ними с одного рабочего места на другое. И вот, уже ближе к полуночи, один из наших сотрудников прибегает ко мне с «квадратными глазами» и говорит, что всё зависло и почти готовый отчёт он не может сохранить. Оказалось, что «Лексикон», по умолчанию, сохранял текст и его копии на тот диск, с которого начали работать с текстом (в данном случае – на дискету 3,5"), а если на нём не хватало места – просто зависал и подвешивал MS-DOS. Т.к. всё печаталось «из головы», пришлось сон «отложить на потом», существующие кусочки с разных дискет сохранить на жёстком диске и несколько часов восстанавливать утраченное.

Вспомнил ещё один забавный эпизод по поводу нашего компьютера. Когда мы тестировали первые экземпляры ПК «СОЮЗ-ИОН», один из членов «МЕДИОНа», работавший в ВЦСПС, предложил нам заключить контракт с фирмой Sony, которая в то время искала партнёров для открытия представительства в Москве. Мы («Медэнерго») тогда занимали второй этаж маленького кирпичного флигеля



во дворе «Союзтехэнерго», располагавшегося рядом в большом здании на Семёновской площади (на фото – красное и серое строения).



Сказано – сделано, и через несколько дней группа «японских товарищей» прибыла к нам для знакомства. Естественно, мы готовились основательно. У японцев был переводчик, хорошо владевший русским языком.

Для начала гости поинтересовались, какими разработками занимается наш коллектив (мы представлялись как «МЕДИОН»). Мы показали им кое-что из аппаратно-программных комплексов для тестирования физиологического и психологического состояния человека. Реакция была сдержанно-вежливой. Тогда Александр Иванович Таран включил наш компьютер и запустил демонстрационный ролик цветной 3D-графики. Заметим, что ничего подобного тогдашние IBM AT на базе 286-го процессора ещё не умели.

Гости заметно оживились и стали бурно обсуждать увиденное по-японски.

Потом переводчик спросил: «А где компьютер?»

Таран ответил: «Перед вами».

– Нет, это клавиатура и монитор, а где компьютер?

Таран ткнул пальцем в клавиатурный блок.

Опять бурное обсуждение на японском. Выражение лиц, несмотря на восточную сдержанность, читалось как «Зачем вы нас обманываете?»

Александр Иванович выключил компьютер, перевернул клавиатурный блок и монетой в 1 копейку начал откручивать винты (они были латунными, со шляпкой около 10 мм и большим шлицем). Через минуту он снял крышку и стал показывать изумлённым гостям плату, объясняя, что компьютер собран на советских комплектующих.

Японцы посоветовались и сказали, что согласны на сотрудничество.

Однако через день приехал их юрист для подготовки контракта и попросил документы на помещение. Узнав, что мы сидим на площадях, арендованных у «Союзтехэнерго», он расшаркался и заявил, что, к сожалению, фирма Sony не может рисковать, заключая контракт с партнёром, не имеющим собственных помещений.

Вопрос общего плана – какое Ваше мнение о развитии ИТ-индустрии в целом за прошедшие годы? То есть что Вам больше нравилось раньше, что нравится или не нравится сейчас, что существенно изменилось?

Если раньше ИТ-индустрия была делом профессионалов-интеллектуалов, то ориентация на массового пользователя (в пределе – на домохозяйку), сильно снизила планку к проработке продуктов. Достаточно сравнить различные процессоры разработки одного из военных НИИ (размер – 256 КБ) с MS Excel того же времени как по объёму кода, так и по требованиям к компьютеру, чтобы понять, что понятие «оптимизации кода» стало, мягко говоря, необязательным. А ведь их функционал был практически одинаковым! Естественно, это связано и с желанием производителей ПО и «железа» заставлять пользователей покупать всё более «навороченные» продукты.

Кроме того, переход к использованию коммерческих библиотек в программировании, наряду с упрощением процесса создания программ, привёл к невозможности для 99,99% программистов разобраться с тем «как оно работает».



Конечно, возможности современных компьютеров и ПО нам и присниться не могли во времена работы на «Электронике-100И». Сейчас я, при необходимости, за один вечер могу «слепить» презентацию для лекции или доклада на конференции, вставив туда несколько фото и/или видео из интернета, из собственных архивов и т.п., а раньше мне на это потребовался бы месяц.

Да и в быту я уже просто не могу себе представить жизни без фото-, видео- и аудиоредакторов, без общения с родственниками и друзьями, раскиданными по всей стране и далеко за её пределами.

Чем Вы занимаетесь сейчас, в какой сфере работаете?

Два последних десятилетия я занимаюсь, преимущественно, развитием телемедицины в России. Российская ассоциация телемедицины (РАТ), ведущим экспертом которой я являюсь, с конца девяностых годов прошлого века проводит международные школы по телемедицине, на которых проходят обучение как специалисты-медики (в основном – заведующие отделениями, главврачи и их замы), так и IT-шники из медучреждений. Мы обучаем их технологиям, организации процессов, экономическим аспектам внедрения телемедицины. В 2011 году в МГМСУ нашей ассоциацией была создана первая в стране кафедра телемедицины, однако, после смены ректора, перспективы её развития стали сомнительными и наш коллектив перешёл в Мединститут РУДН, где мы преобразовали существовавшую там кафедру медицинской информатики в кафедру медицинской информатики и телемедицины.

В настоящее время я работаю профессором (частичная занятость) сразу в трёх университетах: МФТИ (кафедры мультимедийных технологий и телекоммуникаций), РУДН (кафедры медицинской информатики и телемедицины Мединститута) и РНИМУ им. Н.И. Пирогова (кафедры медицинской кибернетики и информатики МБФ). Кроме этого,

руководжу малым научно-внедренческим предприятием, занимающимся исследованиями и разработками в области автоматизированных средств диагностики и коррекции функционального состояния человека.

Вы раньше сталкивались с понятием downgrade или, скажем, просто с любителями старых компьютеров, истории ЭВМ и т.п.? Возможно, участвовали в каких-то обсуждениях, посещали какие-то сайты или форумы? Бывает ли, что «ностальгируете» по «старым добрым временам»?

О downgrade впервые услышал несколько лет назад от Александра Гречишкина. В форумах не участвую – просто некогда, а что до ностальгии, то временами сожалею, что в нечестной конкурентной борьбе (это, опять-таки, тема отдельного разговора) платформа «Винтел» сумела уничтожить гораздо более перспективную и прогрессивную «ДЭК-Альфу», опережавшую на десятилетие всё существовавшее в то время.

Огромнейшее спасибо за познавательное и интересное интервью! Удачи и успехов в Вашей нужной и полезной работе.

Прим. ред.: пользуясь, случаем, хочу поблагодарить Александра Гречишкина за помощь в организации интервью.

В.Ф. Фёдоров
uav1606





ОГАС, ИЛИ САГА ОБ ЭЛЕКТРОННОМ СОЦИАЛИЗМЕ

Мы — система. А если система не развивается, то гибнет.

Виктор ГЛУШКОВ

Философия ОГАС

ОГАС («Общегосударственная автоматизированная система сбора и обработки информации для учёта, планирования и управления народным хозяйством в СССР») представляла себе как некую «умную» нейронную сеть, нервную систему советской экономики.

Такая кибернетическая аналогия между компьютерной сетью и мозгом наложила свой отпечаток и на другие инновации в теории вычислений.

Например, вместо так называемого бутылочного горлышка фон Неймана (которое ограничивает объём передаваемых данных на компьютере), команда Глушкова предложила поточную обработку данных по образцу и подобию одновременного возбуждения многих синопсов в мозгу у человека.

ОГАС дала бы возможность объединить информационную и телекоммуникационную структуру в стране в единую систему, позволявшую на научно-техническом уровне решать вопросы экономики, образования, здравоохранения, экологии, сделать доступными для всех интегральные банки данных и знаний по основным проблемам науки и техники.

Виктор ГЛУШКОВ: Россию в своё время связывали в единое целое железные дороги, затем электрические сети. При современных масштабах экономики, постоянном расширении связей между объектами хозяйственной деятельности функция интеграции страны выпадает на долю ОГАС.

В опорных вычислительных центрах сети планировалось сосредоточить основные мощности средств вычислительной техники, предназначенных для решения сложных экономических задач с большим объёмом вычислительных операций.

В низовых (кустовых) вычислительных центрах должны быть сосредоточены основные мощности средств вычислительной техники, предназначенных для обработки данных и решения информационных задач с относительно малым объёмом вычислительных операций.

ОГАС, по мысли Глушкова и Китова, должна была заниматься программно-целевым управлением народным хозяйством СССР. На принципах долгосрочного планирования (об этом в соответствующем разделе статьи).

С персональной ответственностью каждого за выполненную работу.

Вводилось понятие внутренней госприёмки на каждом из этапов работы, а также конкуренции при выполнении параллельных задач при разработке. Таким образом, можно было оптимизировать научно-технический прогресс на основе постоянно динамически изменяющейся системы полученной информации.

Объективные данные о производстве, энергетике, транспорте без искажений должны были быть доступны для принятия решений на любой ЭВМ любому пользователю системы.

Виктор ГЛУШКОВ. Машина — это огромный усилитель, и она усиливает ту информацию, те возможности, которые в неё закладывают... Основываясь на правдивой информации, можно получать действительно



усиление человеческого интеллекта с помощью машины — то, чего без машины сделать было бы нельзя. Но если заложить в машину заведомую ложь, то она эту ложь усилит и выдаст такую уже ложь, в которой разобраться окончательно будет невозможно... Главное условие успеха автоматизации — это правдивость, честность исходной информации.

Первый этап – Единая государственная сеть вычислительных центров

Вот выдержки ключевых разработок (по моему скромному мнению) предэскизного проекта ЕДИНОЙ ГОСУДАРСТВЕННОЙ СЕТИ ВЫЧИСЛИТЕЛЬНЫХ ЦЕНТРОВ СССР (ЕГСВЦ) – прямого прообраза ОГАС от 21 февраля 1964 г.



Реальная карта Единой Государственной Сети Вычислительных Центров (ЕГСВЦ) – 1964 г.
Взято из журнала «Кибертония» №1'2012 г.

Проект появился гораздо раньше Интернета.

1965 год – начать унификацию методов и средств автоматизации и механизации процессов сбора и обработки первичной информации.

1966 год – закончить преобразование (на основе типовых проектов низовых ВЦ) отдельных вычислительных центров, имеющих в предприятиях, в низовые вычислительные центры системы.

1967 год – закончить создание ВЦ отраслевых комитетов, провести эксперименты по управлению отраслью через низовые ВЦ Единой сети, которые к этому моменту должны начать функционировать.

1968 год – закончить разработку вычислительных машин, пригодных для использования в опорных центрах ЕГСВЦ, начать серийный их выпуск.

1969 год – укомплектовать опорные и низовые центры первой очереди вычислительными системами и организовать каналы связи между ними;

– закончить экспериментальную обработку экономико-математических методов, алгоритмов и программ работы опорных ВЦ.

1970 год – закончить преобразование Главного вычислительного центра Госплана СССР в головной вычислительный центр сети. Провести эксперименты по совместной работе головного центра в едином режиме с несколькими опорными центрами сети.

Фактически 1970 год должен быть стать годом рождения «красного интернета».

1971 год – закончить создание первой очереди опорных центров. Они должны быть соединены каналами связи между собой и головным центром сети и работать в едином режиме.

1972 год – закончить совершенствование и окончательно отработать модели динамического планирования в масштабе страны, провести соответствующие эксперименты.

1973 год – целесообразно закончить создание автоматизированной системы сбора и передачи первичной информации.

В 1974 году целесообразно:

– закончить создание ЕГСВЦ, включая все опорные и низовые центры;

– готовить окончательную отработку функционирования Единой автоматизированной системы планирования и управления народным хозяйством.

1975 год – окончание работ. Стоимость работ оценивалась в 5 млрд рублей на 10 лет.



Второй этап – Общегосударственная автоматизированная система сбора и обработки информации (ОГАС)

Цели ОГАС



- переход экономики СССР на пятый технологический уклад;
- создание информационного общества;
- электронный документооборот (безбумажное делопроизводство);
- электронные деньги.

Персональные электронные счета граждан СССР, отсекающие возможные спекуляции при получении дефицитных товаров по безналичному расчету. Тем самым уменьшая возможность организованной преступности получать ресурсы. Сначала он предлагал правильно организовать распределение с помощью денег, о чём можно узнать в книге-интервью В. Моева «Бразды управления»: «Давайте условимся, что на личные счета в банке будут приниматься перечисления только от официальных организаций, выплачивающих людям вознаграждение за труд. Снять со своего счёта наличные вы можете, но внести туда наличными нельзя... Если банк, будучи включён в систему автоматизированных безналичных расчетов, начнёт принимать деньги только от официальных организаций, где люди получают зарплату, в этот кругооборот никак не могут попасть заработки частные и сомнительные. Декретно, в один день и час истребить все так называемые «левые» операции с деньгами невозможно. Но после предложенной меры обращение их замкнётся в своём ограниченном кругу. Из первого «официального» круга, что ли, круга об-

ращения во второй деньги переходить могут – достаточно вам снять часть своего заработка с банковского счёта, а вот из второго круга в первый они уже никогда не вернуться.

- электронные очереди;
- онлайн-общение – сначала предприятий, а потом и граждан;
- динамическое планирование народного хозяйства;
- обмен научно-технической информацией;
- база данных нерешённых проблем на предприятиях.

В. М. Глушков в 1982 г. утверждал: «Построение такой сети позволило бы собирать и оптимальным образом использовать экономическую, научно-техническую и любую другую информацию, а также обмениваться ею в интересах потребителей, что очень важно в наше время для перехода к информационному обществу»

Структура ОГАС

ОГАС должна была состоять из 200 вычислительных центров, 2,5 тыс. кустовых и 22,5 тыс. индивидуальных ВЦ и покрывать территорию всего Советского Союза.

Модель у этой сети была иерархическая, соответствуя трёхуровневой структуре государства и его экономики. Один головной компьютерный центр в Москве должен был подключиться к 200 компьютерным центрам среднего уровня в крупных городах, а те в свою очередь должны были подключиться к 20 тысячам компьютерных терминалов, распределённых по ключевым производствам народного хозяйства. Характерным было наличие распределённого банка данных и возможность безадресного доступа из любой точки этой системы к любой информации после автоматической проверки полномочий запрашивающего лица. Был разработан ряд вопросов, связанных с защитой информации, широкополосный доступ.



Глушков считал, что данная сеть должна была стать преднамеренно децентрализованной. То есть Москва могла указывать, кто какие разрешения получает, а уполномоченный пользователь имел возможность связываться с любым другим пользователем локальной сети. При этом ему не надо было получать разрешение от вышестоящего центра. При этом данные о материальных потоках были бы прозрачны для центра. То есть возникала иерархическая двухконтурная система, состоящая из централизованного управления в первом контуре, плюс обмен информацией и материалами на уровне производства во втором контуре.

Этот проект ни много ни мало должен был положить начало эпохе «электронного социализма».

История ОГАС в этапах

1 этап – 50-63 годы XX века

Первая в мире идея создать общенациональную компьютерную сеть для населения принадлежала военному исследователю Анатолию Ивановичу Китову.

В 1952 году Китов познакомился с капитальным трудом Норберта Винера «Кибернетика» (1948 год). В 1959 году, будучи директором секретного вычислительного центра Министерства обороны, Китов начал заниматься другими вопросами, обратив внимание на «неограниченное количество надёжной вычислительной мощности», которая должна была обеспечить оптимальное планирование в советской экономике.

Китов написал письмо на имя Хрущева, в котором поделился своими мыслями на эту тему (они получили название проект «Красная книга»). Он предложил разрешить гражданским организациям использовать военные компьютерные «комплексы» для экономического планирования в ночное время, когда основная часть военных спала. Он думал, что экономические органы планирования сумеют освоить вычислительные мощности военных для решения проблем в режиме реального времени.

Резкая критика этого проекта со стороны Министерства обороны и работников аппарата ЦК КПСС привела к тому, что Китов был исключен из коммунистической партии и его уволили с работы. Так было покончено с самым первым предложением о создании публичной общенациональной компьютерной сети.

А. И. Берг, А. И. Китов, А. А. Ляпунов в ноябре 1959 году опубликовали доклад, озвученный на секции кибернетики всесоюзного совещания по вычислительной математике и вычислительной технике: «О возможностях автоматизации управления народным хозяйством» о комплексном введении ЭВМ в народное хозяйство СССР.

1961 год – проект Автоматической системы управления предприятием разработан в Институте кибернетики УССР.

В США вводится служба автоматизированного снабжения армии.

2 этап – 63-65 год

Основные принципы создания Единой сети связи (ЕСС) впервые в мире выдвинул известный советский учёный, академик АН СССР А. А. Харкевич в статье «Информация и техника» (журнал «Коммунист», 1962, № 12). Первый проект ЕАСС был завершён в 1965 г. под руководством С. А. Аджомова.

Задача построения общегосударственной автоматизированной системы управления (ОГАС) экономикой была поставлена Глушкову первым заместителем Председателя Совета Министров (тогда А.Н. Косыгиным) в ноябре 1962 года.

В сентябре 1963 г. был создан Междудомственный научный Совет по внедрению математических методов и вычислительной техники в народное хозяйство при Госкомитете по координации научно-исследовательских работ СССР.

В.М. Глушков предлагает систему ОГАС в 1964 году, намереваясь сделать из неё национальную компьютерную сеть в режиме реального времени с удалённым доступом на базе действующей и новой телефонной сети.



В своей самой амбициозной версии данная сеть должна была охватить большую часть евразийского континента, став своеобразной нервной системой, проникающей на каждое предприятие плановой экономики.

Были и альтернативные предложения, которые в конечном итоге привели к катастрофе страны в конце 80-х – начале 90-х годов.

Здесь надо сказать пару слов о реформе Косыгина-Либермана.

Начиная с 1964 года (времени появления проекта ОГАС) против Глушкова стали открыто выступать учёные-экономисты Либерман, Белкин, Бирман. Все трое впоследствии эмигрировали в Израиль. По их словам, реформа стоила столько, сколько постановление, напечатанное на бумаге.

Суть реформы Бирмана-Либермана – в получении прибыли предприятий при перевыполнении плановых показателей. Последние предлагали ориентировать экономику на прибыль от себестоимости.

В сталинские времена, напротив, ориентировались на снижение себестоимости (отсюда — знаменитые понижения цен), не связывая жёстко её и прибыль. В результате роль стоимостных показателей снижалась.

Теперь прибыль жёстко привязывалась к себестоимости выпускаемой продукции. Снижать себестоимость стало невыгодно, ибо это снижение уменьшало прибыль.

Фактически, при внешне плановых статистических успехах и социалистической форме собственности, был перекокс в сторону узковедомственных и шкурных интересов отдельных министерств и предприятий. А также в сторону увеличения ничем не обеспеченных денег на счетах населения и предприятий, в конечном итоге приведших страну к всеобщему дефициту и недовольству населения, разрушивших СССР из-за несбалансированности экономики.

В июле 1965 г. состоялось заседание Президиума Совета Министров СССР, на котором проект был отвергнут и отправлен на доработку в ЦСУ СССР и в другие госведомства. Была принята либермановская реформа.

Ещё одним противником В.М. Глушкова был министр финансов Василий Гарбузов.

Суть возражений Гарбузова, по словам Глушкова: Вот, говорит, что нам надо делать: сначала все птицефермы в Советском Союзе автоматизировать, а потом уже думать про всякие глупости вроде общегосударственной системы. Было вынесено контрпредложение, которое всё снижало на порядок: вместо Госкомупра – Главное Управление по вычислительной технике при ГКНТ, вместо научного центра – ВНИИПОУ.

Под конец выступает Сулов и говорит: «Товарищи, может быть, мы совершаем сейчас ошибку, не принимая проект в полной мере, но это настолько революционное преобразование, что нам трудно сейчас его осуществить. Давайте пока попробуем вот так, а потом будет видно, как быть». И спрашивает не Кириллина, а меня: «Как вы думаете?». А я говорю: «Михаил Андреевич, я могу вам только одно сказать: если мы сейчас этого не сделаем, то во второй половине 70-х годов советская экономика столкнётся с такими трудностями, что всё равно к этому вопросу придётся вернуться». Но с моим мнением не посчитались, приняли контрпредложение.

Вместо цельной системы решение переведилось на уровень подсистем.

Сторонником ОГАС был министр обороны Д.Ф. Устинов, который отсутствовал на том историческом заседании.

Гарбузов убедил Политбюро в том, что проект ОГАС, с его амбициозным планом оптимального моделирования и управления информационными потоками в плановой экономике, является поспешным и чрезмерным.

3-й этап – 66-70 годы

Поддержанная на словах система была растащена на уровне министерств, которые стали создавать свои ВЦ, несовместимые с другими ВЦ. Проект был отложен, и на его месте появились, десятки и сотни изолированных и функционально несовместимых локальных



систем управления на заводах и фабриках. Единые фонды, ресурсы и мощности были растащены по крысиным норам разных ведомств.

В 1969 г. в связи с увеличением сложности управления Комиссия при Государственном комитете по науке и технике приняла решение вынести концепцию ОГАС в планы IX пятилетки (1971-1975 гг.).

За рубежом

Сеть с пакетной коммутацией была введена в Национальной физической лаборатории в Великобритании в 1966 г.

В США в 1968 г., благодаря отличному финансированию и обстановке сотрудничества в научной среде, была создана система, ставшая прообразом глобальной сети ARPANET, ставшей прообразом Интернета. Заказчиком системы выступил Пентагон. Были связаны крупные научные центры, ведомства и военные базы. ARPANET стал американским ответом на планы создания советского ОГАС-информатория.

Так тень Советского Союза стала повивальной бабкой Интернета.

Более подробно вы об этом можете узнать из книги М. Максимовича «Беседы с академиком В. Глушковым» (например, [здесь](#)).

4-й этап – 70-80 годы

В.М. Глушков и А.И. Китов занялись решением более узкоспециализированных, частных вопросов, параллельно продолжая продвигать идею интегрирования в систему ОГАС всех автоматизированных систем управления народным хозяйством.

1973-75 годы – технико-экономическое обоснование и техпроект государственной системы вычислительных центров.

В более позднее время (1977 г.) против создания ОГАС выступил Старовский, начальник ЦСУ. Он убедил Косыгина, что информации, предоставляемой его ведомством, достаточно для управления СССР.

В конце 70-х годов началось строительство единой сети, связывающей ВЦ страны.

В декабре 1978-го года был осуществлён обмен данными между ВЦ, расположенными в Москве, Риге, Киеве, Ташкенте и Томске.

В УССР в 1973-м году и позднее проводились работы по построению республиканского уровня ОГАС (РАСУ). Что стало немалым вкладом в тогдашние экономические успехи Украины.

Следующим шагом должны были стать отраслевые АСУ, потом общесоюзные. В общих чертах частично было реализовано к 1987 году, потом работы были свёрнуты Горбачёвым.

За рубежом

В Болгарии и Чили начинается разработка собственных систем, родственных ОГАС.

Болгария ведёт работы по созданию Единой Системы Социальной Информации (ЕССИ)

Киберсин в Чили. Разработанный Стаффордом Биром и несколькими его учениками.

В момент запуска и отработки происходит переворот Пиночета.

Об этих разработках будет отдельная статья. В 80-е годы лучшие ученики Бира погибли насильственной смертью.

5-й этап – 80-87 годы

20 декабря 1980 г. проект ОГАС был подпisan его разработчиками. В начале 1980-х гг. на основе проекта ОГАС составлялась Долгосрочная целевая программа, согласно которой должна была осуществляться его реализация. Вместо 5 миллиардов рублей надо было потратить 40 миллиардов. Такова стала цена ошибки пленума 1965 года.

С 1966 г. по июнь 1984 г. было создано:

3300 АСУ на отдельных предприятиях;

3200 ведомственных АСУ.

Благодаря ОГАС даже в таком урезанном виде, рост экономики в ВВП был 1-2% в год, и не было перебоев в выплате заработной платы, пенсий и соцпособий по всей территории СССР.

В единой сети было более 2000 предприятий народного хозяйства. Главный компьютер в сети находился в Москве в ВНИИ проблем организации и управления (ВНИИПОУ).



В декабре 1981 г., согласно проекту ОГАС, был составлен документ «Основные положения разработки Долгосрочной целевой программы создания ОГАС».

Реализация проекта ОГАС остановилась в период «перестройки» в ходе «радикальной экономической реформы», провозглашённой на Пленуме ЦК КПСС в январе 1987 г.

Страна находилась в системном кризисе, одним из спусковых крючков стала косыгинская реформа. С массой необеспеченных денег у населения, дефицитом и трескотнёй о перестройке.

Запуск системы был запланирован на 1990 год. Большинство работ свёрнуто в середине 1987 года.

Увы, но зашоренность в идеологических постулатах людей, принимавших решения, не дала довести разработку этой идеи до конца, что в конечном итоге привело к распаду страны.

Сообщество и ссылки

Сайт Леонида Отоцкого, посвящённый памяти

Стаффорда Бира (Веб-архив):

http://ototsky.mgn.ru/it/beer_menu.html

Зеркало:

http://home.mgn.ru/~ototsky/it/beer_menu.html

Сообщество кибернетического управления в России:

http://community.livejournal.com/ru_cybersyn/

Сообщество кибернетики менеджмента:

<http://cybermanagement.ning.com/>

The Metaphorum Group Developing Stafford

Beer's Legacy in Management Cybernetics:

<http://www.metaphorum.org/>

ОГАС. Опередившая время:

<http://vestnikburi.com/>

ogas-operedivshaya-vremya/

Видеообзор ОГАС:

<https://www.youtube.com/watch?v=O9CwVEJun6Y>

Сайт об ОГАС:

<http://ogas.kiev.ua/>

Краткая биографическая справка

Виктор Михайлович Глушков



1923. Родился 24 августа 1923 г. в Ростове-на-Дону.

1941. Окончил среднюю школу в г. Шахты.

1948. Окончил четыре курса Новочеркасского индустриального института.

1949. Окончил Ростовский государственный университет.

1951. Присуждена учёная степень кандидата физико-математических наук.

1956. Присуждена учёная степень доктора физико-математических наук. Заведующий лабораторией ВТ Института математики АН Украины.

1957. Директор Вычислительного центра АН Украины в Киеве.

1958. Избран членом-корреспондентом АН Украины.

1961. Избран академиком АН Украины.

1962. Директор Института кибернетики АН Украины.

1964. Избран академиком АН СССР, вице-президентом НАН Украины.

1964. Присуждена Ленинская премия за цикл работ по теории автоматов.

1967. Награждён орденом Ленина за достигнутые успехи в развитии советской науки и внедрение научных достижений в народное хозяйство. Присуждена премия имени Н. Н. Крылова за цикл работ по теоретической кибернетике, посвящённых формальным методам проектирования электронных вычислительных машин.



1968. Государственная премия СССР (в составе авторского коллектива) за разработку новых принципов построения структур малых машин для инженерных расчетов и математического обеспечения для них, внедрённых в вычислительных машинах серии «МИР».

1969. Присвоено звание Героя Социалистического Труда за большие заслуги в развитии советской науки.

1970. Избран в члены немецкой академии «Леопольдина».

1970. Присуждена Государственная премия Украины (в составе авторского коллектива) за разработку и внедрение автоматизированной системы управления радиотехническим предприятием массового производства.

1973. Награждён орденом Октябрьской Революции за заслуги в развитии кибернетики и вычислительной техники. Награждён орденом «Народная республика Болгария» I степени за большие заслуги в развитии болгарской науки.

1974. Избран иностранным членом Академии наук Болгарии.

1975. Награждён орденом Ленина за заслуги в развитии советской науки и в связи с 250-летием АН СССР. Избран почётным доктором Дрезденского университета, почётным членом Польского кибернетического общества, иностранным членом Академии наук ГДР.

1976. Награждён орденом «Знамя Труда» ГДР за выдающийся вклад в сотрудничество специалистов СССР и ГДР в разработку прогноза развития технических средств электронных информационных систем.

1977. Присуждена Государственная премия СССР (в составе авторского коллектива) за цикл трудов по теории дискретных преобразователей и методам автоматизации проектирования ЭВМ, которые нашли применение в действующих системах. Избран иностранным членом Польской Академии наук.

1978. Присвоено звание заслуженного деятеля науки Украины.

1979. Присуждена премия имени С. А. Лебедева за цикл работ по теории перспективных ЭВМ и создание высокопроизводительных

средств вычислительной техники и систем управления.

1980. Присуждена премия имени А. Н. Крылова за цикл работ по методам оптимизации в планировании и управлении.

1981. Присуждена премия Совета Министров СССР за разработку и внедрение в народное хозяйство комплекса программно-технических средств по созданию автоматизированных систем сбора, передачи и обработки данных. Присуждена Государственная премия Украины (в составе авторского коллектива) за разработку и внедрение в народное хозяйство базовой АСУ ТП на магистральных нефтепроводах.

1982. 30 января 1982 года умер в Киеве.

Анатолий Иванович Китов



1920. Родился 9 августа 1920 г. в г. Самара.

1939. В 1939 г. А. И. Китов окончил среднюю школу с золотой медалью и поступил на физико-математический факультет Ташкентского государственного университета, где проучился всего три месяца, так как был призван рядовым в Красную армию.

1940. А. И. Китов по личному распоряжению маршала К. Е. Ворошилова был зачислен курсантом Ленинградского артиллерийского училища.

1941. Был досрочно признан окончившим училище и направлен на Южный фронт командиром артиллерийского взвода.

1945. Зачислен в Артиллерийскую инженерную академию им. Ф. Э. Дзержинского (ныне имени Петра Великого), он, в том же месяце,



когда были вступительные экзамены, сдал все экзамены за первый курс и был принят сразу на второй.

Параллельно с учёбой в артиллерийской академии посещал лекции на мехмате МГУ, в частности, семинар академика А.Н. Колмогорова.

1950. Академия закончена с золотой медалью.

1950-1952 гг. работал научным референтом в Министерстве обороны СССР

1952. Защитил кандидатскую диссертацию на тему «Программирование задач внешней баллистики ракет дальнего действия».

1952-1954 гг. Начальник отдела вычислительных машин Артиллерийской инженерной академии им. Ф. Э. Дзержинского.

1954. Руководит созданием на базе возглавляемого им в военной академии отдела ЭВМ головного вычислительного центра Министерства обороны СССР ВЦ-1 МО СССР (в/ч 01168).

1956. В начале 1956 года выпущена монография «Электронные цифровые машины» – первая в нашей стране книга в данной области.

В том же году – А. И. Китов, Н. А. Криницкий и П. Н. Комолов. «Элементы программирования».

1959. А. И. Китов, Н. А. Криницкий. «Электронные цифровые машины и программирование».

Первый в СССР учебник программирования. Переведён в странах восточной Европы и Китае.

В январе А.И.Китов готовит обстоятельный доклад (известный как «Письмо Н.С. Хрущёву»). Приложением к данному Письму была его брошюра «Электронные вычислительные машины».

Ноябрь – И. Берг, А. И. Китов, А. А. Ляпунов. «О возможностях автоматизации управления народным хозяйством».

Осень 1959 года – И. Китов работал над докладом для ЦК КПСС (так называемая «Красная книга»). В проекте резко критиковалось сложившееся положение дел во внедрении ЭВМ в народное хозяйство.

Китов был исключен из партии и снят с работы.

Также 1959-м годом датируется авторское свидетельство «Метод четырехкратного совмещения этапов такта машинных команд». Разработанные на основе изобретения автоматы наводили зенитные ракеты по самолётам противника.

1961. Статья А.И. Китова «Кибернетика и управление народным хозяйством». В сборнике «Кибернетика – на службу коммунизму» под редакцией А. И. Берга.

1963. Доктор технических наук, защищена диссертация «Применение ЭВМ для решения задач противовоздушной обороны».

1967. Монография «Программирование информационно-логических задач».

В **1965-1972** гг. А. И. Китов был главным конструктором отраслевой автоматизированной системы управления (ОАСУ) Министерства радиопромышленности СССР и директором Главного вычислительного центра этого министерства.

1970. Книга под его редакцией «Система автоматизации программирования АЛГЭМ».

1971. Монография «Программирование экономических и управленческих задач».

В **1970-75** гг. А. И. Китов был главным конструктором АСУ Здравоохранение.

1976. Монография «Автоматизация процессов обработки информации и управления в здравоохранении».

1977. В издательстве «Медицина» вышла книга А. И. Китова «Введение в медицинскую кибернетику».

1983. Монография «Медицинская кибернетика».

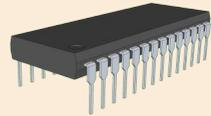
1987. Письмо Китова Горбачёву о неудовлетворительном состоянии дел в информатизации СССР. Было проигнорировано.

Свою научную карьеру А.И. Китов завершил в должности заведующего кафедрой «Вычислительная техника и программирование» и профессора этой кафедры Российской экономической академии им. Г. В. Плеханова, проработав там с **1980** г. по **1997** г.

2005. Умер 14 октября 2005 года в Москве.



BIOS: ПОКАЗАТЬ ВСЕ, ЧТО СКРЫТО

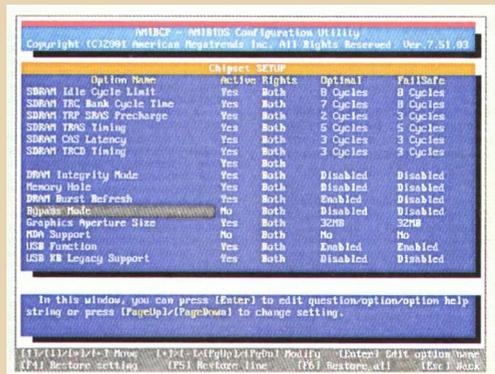


3 адумывались ли Вы, уважаемый читатель, что-нибудь сотворить с BIOS своего старого компьютера? Нет, ну понятно, что когда мы собираем «новую» машину, то обязательно откроем настройки CMOS и, как минимум, настроим параметры жёсткого диска, дисководов, очередности загрузки, время с датой не забудем и т.д. А вот такие операции, как обновить BIOS, обработать его биос-патчем или произвести какую тонкую настройку? Автор как-то не касался – разве что перешивал один раз BIOS, чтобы включить на плате поддержку AMD K6-2. Однако, как показывают обсуждения на форумах, владельцы старых компьютеров балуются с BIOS достаточно активно – зашивают последнюю версию, чтобы включить поддержку новых процессоров, направляют биос-патчер, чтобы заставить материнскую плату увидеть жёсткий диск большого размера или же снова поддержать неподдерживаемые официально процы.

А вот ещё один интересный вариант «издевательства» над BIOS – разблокировка скрытых настроек. Так уж получилось, что на сегодняшний день ни создатели чипсетов, ни создатели материнских плат не занимаются самостоятельно разработкой BIOS для своих творений – разработку ведут отдельные фирмы – те самые AWARD/Phoenix и AMI. Делается, видимо, какая-то «болванка» BIOS под определённый чипсет, а уже производитель материнской платы, купивший такую болванку, подгоняет её под себя – под реализованные в схематехнике платы возможности чипсета, под нужды и психологию потенци-

ального потребителя. Например, на компанию Gigabyte жаловались, что она максимально урезает настройки, способные поднять производительность материнской платы – пользователь не сумеет материнскую плату убить кривыми ручками, но и разогнать или просто что-то немного оптимизировать, если захочется, тоже не сможет. Однако из кода прошивки эти функции не изымаются и, как выяснилось, могут быть включены снова. Как именно – рассказывает, например, статья А. Горячкина «BIOS: полный контроль или как разблокировать скрытые функции», опубликованная в «Радио» №1 2004 года:

<http://www.chipinfo.ru/literature/radio/200401/p24-25.html>



Статья носит скорее обзорный характер – автор рассказывает саму идею разблокировки, приводит список необходимого софта – прошивальщик и, собственно, разблокировщик. Предполагается, что экспериментатор



будет модифицировать версию BIOS, заштиту, на данный момент, в материнскую плату. Поэтому первым этапом предлагается считать BIOS. После чего можно приступать к разблокировке. Для BIOS фирмы AWARD предлагается использовать программу **Modbin**, для AMI – **AMIBCP**. Увы, с тех времён прошло уже 15 лет (страшно думать, если честно ☹) и, как выяснилось, приведённые в статье ссылки – битые. Несколько версий ModBin удалось найти здесь:

<https://www.ixbt.com/mainboard/biostools.html>

(См. внизу страницы.)

В принципе, все три ссылки работают. Версию AMIBCP, использованную автором статьи, удалось найти здесь:

<http://k7jo.de/tools/ami/amibcp75103.zip>

Скачать файл получилось без проблем.

Как уже понял читатель, в статье больше внимания уделено модификации BIOS от AMI. Увы, всё равно не слишком много. Автор показал скриншот программы-модификатора и в двух словах описал способ открытия файла с BIOS и указал, как производится разблокировка. Наиболее интересным, по мнению автора, является раздел **Chipset setup** – именно он наиболее богат опциями, которые могут быть полезны при разгоне и просто оптимизации системы. После того, как модификация файла BIOS закончена, необходимо выйти с сохранением – по нажатию клавиши **F10**. При этом, как я понял, возможно изменение и фирменных версий BIOS, например, от ASUS, хоть и зашивать придётся фирменным прошивальщиком.

Под конец автор приводит собственные примеры разблокировки различных скрытых настроек, например, включение возможности регулировки напряжения слота AGP и DDR с

шагом 0,1 В, увеличивать напряжение ядра процессора, разблокировать режим S.M.A.R.T. жёсткого диска. Однако мне бы хотелось привести другой пример – пользователь форума «Полигон призраков» **i8088** использовал AMIBCP для замены оконного интерфейса BIOS материнской платы SuperMicro P6DLs на текстовый (см. статью <http://sannata.org/konkurs/2017/kt1701.shtml> – в конце) – так что, разблокировщик BIOS может оказаться полезен не только для разгона, но и для достаточно косметических операций.

Напоследок, случайно наткнулся на статью по редактированию BIOS от AWARD <https://www.rom.by/biosedit.htm> – куда более содержательную, но и написанную, думается, уже не в конце 2003-го года. ☺

Андрей Шаронов (Andreii88)





ачало эпопеи, в принципе, не имеет никакого отношения к описываемому компьютеру, однако, возможно, что-то объяснит в мотивах поступков автора, так что, думается, лучше начать с этого. Осень 2017-го года. Автор благополучно купил диск, на котором нашёл «Русский офис», о котором было написано в 21-м номере и... ему стало очень грустно – мечта, с которой он бегал одиннадцатая лет, осуществилась. Сперва казалось, что куплен несколько другой комплект, что в искомом должны быть FineReader 2.0 и «Сократ 1.0», но перед сном память лишний раз убедила – нет, тот самый, и долгая мечта, к которой уже привык, как к почти неосуществимой, сбылась. Оттого и стало грустно. Однако дистрибутив офиса дал и надежду – есть куда более полный комплект, и его предстоит найти.

Но под стать этому офису должен быть и компьютер. Какой? UMC55 33 МГц, на котором был испытан имеющийся «Русский офис»? А, может быть, другой? Какой именно? Оглядев то, что есть в закромах, было решено, что это будет машинка на 486-ой материнке с шиной VLB – с VLB-шной видеокартой и такой же мультикартой – никаких SCSI-контроллеров и компромиссов с ISA-мультиками. Я так хочу! До этого эксперименты с VLB-материнками проходили плохо – то не опознавался жёсткий диск, то вообще не было картинки, то флоппик не грузит. А тут повезло – собранная материнка с VLB-мультиком и видеокартой завелась. Опознала жёсткий диск, получилось поставить и DOS с Windows 3.1. Что ещё остается? Малость – корпус. Каким он должен быть? Desktop? Почему-то

не захотелось, обычный Tower? Но тут мне на глаза попала обложка диска «Золотой софт» (см. рис. 1).



Рис. 1. Обложка диска, с которой всё началось

Первое, что бросается в глаза, будто перед нами «перевернутый» корпус tower, но приглядитесь – не знаю, как у вас, а у автора создалось впечатление, что это стоящий на ребре desktop или же корпус, способный занимать и вертикальное, и горизонтальное положение. Скажете, что такое чаще практикуют сейчас – устанавливают небольшие десктопчики вертикально? Не-е, как минимум, в 90-е годы такими вещами баловался Acer. В [статье](#) про AcerMate 450s говорилось, что у него даже ножки есть специальные, чтоб вертикально ставить. Да и «КОМПА» на 286-м процессоре имел примерно такой же дизайн – всё время, когда на него смотрю, возникает желание положить его горизонтально. Так почему бы и не быть такому корпусу под стандартную материнскую плату? Вот и захотелось собрать машину именно в таком корпусе – конечно, когда такой корпус попадётся. ☺

Пока же шло время, шли эксперименты – не получилось на жёстком диске поставить утилиты Нортон – поставил PCTOOLS для Windows. Потом был Новый год – ваш покорный слуга уехал на каникулы к родителям. Там отдыхал и никого не трогал ☺, но случайно узрел на «Полигоне призраков» тему по продаже удивительного компьютера – с дисководом снизу.



Стильный корпус, скругленный сверху, отсеки для дисководов снизу. На дне – ножки для установки в вертикальное положение, для горизонтального, увы, таких не было. Но что поразило ещё? Материнская плата – не помню, была ли тогда фотография в объявлении или же увидел позже, когда получил машину, но внутри стояла материнка стандарта LPX! Да, та самая, которая с «ёлкой». Пройти мимо такого не мог – тем более, что внутри стоял 486-й процессор. Долго ли думал – не вспомню, но уже до вечера написал, что компьютер возьму с пересылом. Продавец согласился отправить из Екатеринбурга в Пермь транспортной компанией, повременить с отправкой (или нет – уже не вспомню), автор отправил ему деньги, и усе.

Через какое-то время компьютер прибыл в Пермь. И вот в институтской лаборатории можно его установить и рассмотреть получше. На рис. 2 можно увидеть внешний вид системного блока спереди. Как уже было сказано выше, отсеки накопителей и дисководы располагаются снизу.



Рис. 2. Системный блок – вид спереди

Если взглянуть сзади – мы увидим панель разъёмов материнской платы стандарта LPX и заглушки для плат расширения. Думается, все читатели видели подобное, но на лежащих кор-

пусах. Тут же всё то же самое, только установлено вертикально ☺ (см. рис. 3). Интересно, что блок питания расположен внизу, а не сверху, как у обычного tower'a.



Рис. 3. Вид сзади – собственно, всё то же самое, что мы привыкли видеть на брендовых системных блоках LPX, но повернуто на 90 градусов



Рис. 4. Внутреннее содержание системника – «ёлка» со слотами расширения установлена горизонтально

В качестве процессора установлен Cyrix 486DX2 50 МГц, но куда более интересен встроенный видеоадаптер – обычно в таких платах используется Cirrus Logic, однако здесь



применён Trident TVGA8900 – лично автору такое попало в первый раз.



Рис. 5. Распаянный Trident TVGA8900 – на отдельных-то платах такой не часто встретишь, а уж встроенный...

Одной из проблем конструкции системного блока оказалось то, что платы расширения лучше устанавливать в наиболее дальние от материнской платы слоты – плата, вставленная в слот ближе к материнке, влезала только наполовину.

Также была ещё одна проблема, нестрашная на первый взгляд, но заставившая немного напрячь мозги – севшая батарейка. Батарейка находилась в стандартном гнезде, так что до дербана Dallas’а дело не дошло. Но когда батарейку заменил – сбросились настройки. «А чего хотел?» – спросит читатель, и будет совершенно прав. Но в большинстве компьютеров это никакая не проблема – залез в настройки BIOS, выставил всё как надо... Правильно, вот здесь и засела неприятность – дисководы выставить удалось без проблем, а вот с диском возникла неприятность – компьютер обнаруживал только 500 с чем-то мегабайт и отказывался грузить установленную операционку. Жёсткий же был несколько большего объема. Видимо, дополнительными настройками BIOS как-то заставляли его видеть, но сейчас с насюка этого не сделать. Проще отформатировать как есть и поставить то, что хотелось.

Подключил, сперва, жёсткий диск к новому компьютеру – оказалось, что на 486-ой установлена Windows 95. Сохранив, на всякий случай,

содержимое диска, вернул его на компьютер и, загрузившись с дискеты, форматнул его. После этого водрузил связку из MS-DOS 5.0 и русской версии Windows 3.11, после чего установил Word и Excel. Компьютер пока что не подключил к локальной сети, однако ему нашлось интересное применение – он стал демонстрационным стендом для привлечения потенциальных студентов (см. рис. 6).



Рис. 6. Стенд для привлечения потенциальных студентов

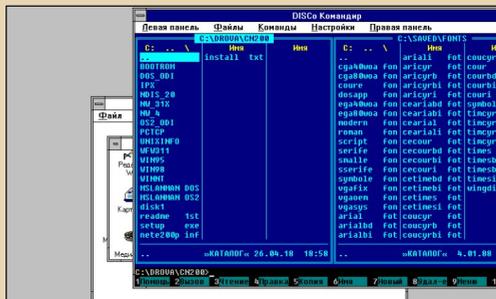


Рис. 7. Рабочее пространство Windows 3.11

К нему был подключен принтер Robotron K6314, о котором автор писал в 23-м номере журнала. Правда, стенд больше привлекает не молодёжь, а преподавателей, которые с удивлением узнают и слоты ISA, и память SIMM 72, и 486-ой процессор. Правда, печатающий из третьей винды ГДРовский принтер, наверное, больше всего радости приносит именно автору статьи – уж больно давно хотелось это сделать – и получилось.

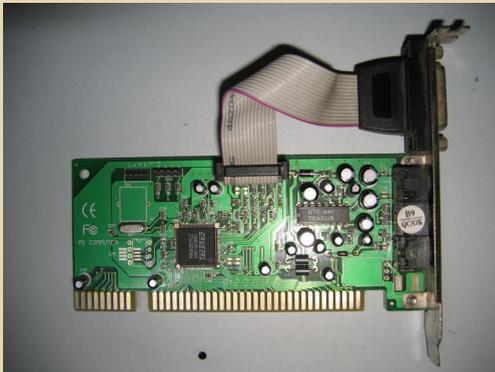




CRYSTAL – ДЕШЕВЛЕ НЕКУДА

В разных старых компьютерных журналах мне часто попадались обзоры каких-то отдельных устройств – видеокарт, жёстких дисков, звуковых карт и т.п. А вот в нашем журнале почему-то такой жанр – редкость. Так что я решил подать пример (надеюсь, не дурной :-), и написать небольшой обзор бюджетной звуковой карты на чипе Crystal CX4235-XQ3.

Вот, собственно, она сама:



«Фу!» – скажут ретроаудиофилы, нежно глядя на свою коллекцию из GUS'ов, оригинальных Sound Blaster'ов и всяких Turtle Beach. :-)

Но что поделаешь, в нашем захолустье купить хорошую звуковую – большая проблема, а эта карта достаточно распространённая и досталось мне задёшево (уже точно не вспомню цену, но не дороже 100 рублей).

Точно модель и производителя установить не удалось – на карте есть надписи **PG COMPUTER**, **GT (или OT) 2004** и **MODEL NO: AT-PRO PG012601634**. Что-то по таким запросам ничего не науглилось.

Да и по самому чипу информации – кот наплакал. Вот сам чип, кстати:



Когда-то даташит и краткая характеристика были на сайте philscomputerlab.com, но на момент написания статьи он почему-то не открывался, а на Web-Archive осталось совсем немного.

К счастью, даташит удалось найти в другом месте, я его даже на всякий случай выложил на сайте журнала:

<http://dgmag.in/N27/crystal/CS4235.pdf>

Из даташита и нескольких упоминаний подобных карт на разных сайтах удалось выяснить, что карта совместима с Sound Blaster Pro (но не SB16) и с Windows Sound System (WSS) – а это довольно необычно для таких бюджетных



карт (та же ESS не имеет такой «фичи»). Есть FM-синтезатор, совместимый с AdLib.

В целом комплектация карты довольно скудная. Нет гнезда для подключения wave-table-дочки или чего-то подобного. Всего три разъёма под мини-джек – микрофон, линейный вход и выход на динамики. Есть порт джойстика и вход для CD-аудио (в двух вариантах). Вот и всё. Видно, что на карте куча площадок под нераспаянные детали, которые, возможно, использовались в более дорогих моделях.

Кстати, в даташите выше описывается чип Cirrus Logic CS4235, а не Crystal CX4235-XQ3, но характеристики и принципы программирования у них явно одинаковые (см. ниже). В одном месте удалось найти информацию, что Crystal Semiconductor – подразделение Cirrus Logic. Даташит на CX4235-XQ3 всё ещё в розыске, упоминание об этом чипе удалось найти только [здесь](#).

В интернете очень много вариаций карт на таком чипе, но, в общем и целом, информации немного. К счастью, найти драйвера было не сложно, к примеру, когда-то Андрей Шаронов выкладывал их на old-dos.ru (за что ему большое спасибо):

<http://old-dos.ru/index.php?page=files&mode=files&do=show&id=7167>

Меня в первую очередь интересовали драйвера для DOS, отдельно их там нет, но можно вытащить файлы **CWDINIT.EXE** (инициализатор) и **CWDMIX.EXE** (микшер) из, к примеру, комплекта для Windows 95. Ну и из набора для Windows 3.1 тоже (я как раз их использовал), но нужно сначала распаковать пакет с помощью **SETUP.EXE**.

Основные эксперименты я проводил на компьютере на базе Pentium 133 MHz/16 MB RAM.

Честно говоря, ожидал каких-нибудь проблем, но карта оказалась на редкость покладистой. Разве что пришлось переставить её в другой ISA-слот – сразу не заработала, но это мелочи – может, контакты в том слоте грязные были и т.п. А дальше всё пошло как по маслу – запускаем в DOS'e **CWDINIT.EXE**, он всё инициализирует – собственно, больше ничего делать не надо, можно работать со звуком. Я проверил работоспособность карты несколькими программами:

1. **QuickView Pro v2.60** – проверил воспроизведение MP3 и WAV – всё работает отлично. Кроме того, в этой программе решил проверить эмуляцию WSS – с помощью ключа **-W3,534,5,1** можно заставить QuickView Pro использовать не SB, а эту систему – всё тоже работало нормально.

2. **QuickView 1.03b** – урезанная версия QuickView Pro для работы на старых процессорах в реальном режиме. Тоже вполне неплохо воспроизводит WAV-файлы через эту карту.

3. **Plany** – миниатюрный плеер, тоже всё воспроизводит нормально.

4. **AdLib MIDI Player** – использовался для проверки работы FM-синтезатора, тоже никаких проблем.

5. Ну и, наконец, игры – куда же без них. Попробовал три игры – **Ultimate Doom**, **Duke Nukem 3D** и **Warcraft** (первая часть) – везде нормально работает и звук, и музыка.

Ещё меня ждал приятный сюрприз от Windows 98 – она сама нашла драйвера на карту, пару перезагрузок – и у нас есть звук под Windows!

В целом карта показал себя вполне неплохо, единственный минус – она оказалась довольно «шумной», т.е. я всё время слышал в наушниках посторонние шумы, особенно когда на компьютере что-то происходило – обращение к диску, вывод на экран и т.п. Наверное, это можно поправить перепайкой электролитов, но я пока не решился.



Всё работало как надо, но это на Pentium'е – это PnP-система, таким картам здесь, как говорится, сам бог велел работать нормально.

А ну-ка, попробуем запустить эту звуковую на XT с процессором KP1810BM88. «Постойте, – спросит внимательный читатель, – но ведь у XT шина ISA 8 bit, а карта на фото – 16-битная?»

Да, это так, но существует много разных ISA-карт, которые, несмотря на свою 16-разрядность, более-менее нормально работают и на XT. Просто карта вставляется так, что её 16-битный «хвост» висит в воздухе.

Но это ещё не все проблемы. Вставив карту в XT, я попробовал запустить **CWDINIT.EXE** – он завис сразу после старта. Поискал другие версии – не помогло. Нашёл даже более старый инициализатор для более ранней модели чипа – файл **CS4232C.EXE** – но и он не захотел работать, тоже завис.

Судя по дате выпуска драйверов, такие карты выпускались где-то в середине 90-х (а эта, возможно, даже в 99-м), а тогда XT уже были неактуальны.

Жаль, конечно, но и мы не лыком шиты! В даташите подробно расписана процедура инициализации карты в обход PnP. Суть примерно следующая – сначала в порт **279h** посылается 32-байтный ключ – насколько я понял, это нужно, чтобы карта поняла, что обращаются именно к ней. Затем в этот же порт посылаются специальные команды – выбрать логическое устройство (SB, FM, джойстик и т.д.), потом – установить адрес порта, номер прерывания и канала DMA. В принципе, в даташите есть довольно подробный пример, который я и использовал, чтобы написать свой инициализатор. То есть я написал на ассемблере программку, которая отправляет в карту нужную последовательность байт (ключ и команды

настройки ресурсов). Единственное, пришлось поменять DMA1 – в примере из даташита устанавливались DMA0=1 и DMA1=3, но в таком варианте у меня XT-шка зависала сразу после старта инициализатора – при этом постоянно горел светодиод жёсткого диска. Возможно, контроллер HDD использует DMA 3, ну или ещё какой конфликт. Поэтому я сделал оба DMA одинаковыми, т.е. у меня DMA0=DMA1=1, используется только первый канал.

Вот эта моя программа с исходниками:

<http://dgmag.in/N27/crystal/CSINIT.zip>

(Целый 81 байт. :-)

Она ничего не проверяет, ничего не выводит на экран, просто инициализирует карту с параметрами: порт 220h, IRQ 5, DMA 1 (переменную **BLASTER** она тоже не устанавливает – это вы должны сделать сами). Может быть, если будет время, доделаю интерфейс и прочее.

С этой программой звук на XT появился! Проверено с помощью тех же вышеупомянутых Planу и AdLib MIDI Player. На самом деле, программ, поддерживающих звуковые карты на XT, довольно мало, поэтому и пользуюсь чаще всего этими.

Подытоживая всё вышесказанное – эта звуковая карта показала себя вполне неплохо, и если у вас нет возможности приобрести что-то получше, то она вполне подойдёт для различных downgrade-экспериментов.

uav1606





втор несколько раз писал, что первым Downgrade-компьютером стала машина на основе материнской платы с впаянным процессором 386SX 40 МГц. Собственно, именно на этой машине проводились первые эксперименты с DOS (PTS и MS ☺) и Windows 3.1x. Именно сидя за этой машиной автор давился завистью, поглядывая на 486-е и строя планы по апгрейду своего стального коня. Увы, ни один из планов не сбылся тогда, но волею судьбы вспомнилась снова материнская, и было решено провести несколько экспериментов, чему и будет посвящена настоящая статья.



Рис. 1. Собственно, «виновница торжества» – M396F – бюджетная материнская плата с распаянным пластиковым 386SX на 40 МГц

Что было сделано тогда?

Когда тогда? Понятно, что в начале даунгрейдерской деятельности автора, но в каком примерно году? Отвечаю – с лета 2005-го до лета 2006-го. Собственно, автору машина (именно компьютер, а не материнка или комплект комплектующих) попала, примерно, в следующей конфигурации:

- материнская плата с впаянным процессором Am386SX 40МГц, 4 слота для SIMM30 и

слоты ISA16 (укороченных ISA8 не было, хоть и на других материнках попадались);

- 1 МБ ОЗУ, набранный четырьмя планками SIMM30;
- FDD 5,25”;
- жёсткий диск IDE 333 МБ (изначально стоял IBMовский винт на 120 МБ, но его ещё поменял товарищ, который отдал мне компьютер);
- видеокарта HMC на 256 КБ (производитель узнал позже, всё время думал, что у меня был воткнут Realtek);
- сетевая карта под коаксиальный кабель.

Первый апгрейд был сделан уже тогда:

- ОЗУ была расширена до 4 МБ – заменил все планки; ☺
- видеокарточка была заменена на Trident TVGA9000i 512 КБ;
- был установлен FDD 3,5” (жёсткий диск покинул единственный отсек 3,5” и был криво приделан в пятидюймовом отсеке корпуса);
- была установлена звуковая карточка AWE64;
- был установлен CD-ROM.

Казалось бы, апгрейд был совершён достаточно серьёзный, но осталось достаточно проблем и хотелок, какие тогда никаким образом не реализовались. Так как драйвера для TVGA 9000i были установлены криво, то удалось только добиться 256 цветов на разрешении 640x480. Жёсткий диск, хоть и не был полностью забит, но хотелось его всё же поделить на два раздела, и в одном, как делаю сейчас, хранить дистрибутивы и резервные копии Windows и **autoexec.bat** с **config.sys** – лучше несколько копий всех трёх ☺. Сидюк, хоть и был установлен, не завёлся нормально. Тогда всё упиралось в достаточно бедный



набор комплектухи, что сейчас не составляет проблемы – звук, видео, планки памяти до одного мегабайта каждая, сидюки и жёсткие диски лежат по коробкам и ждут своего часа. Так почему бы не откопать материнку и не попробовать всего этого богатства?

Кроме всего прочего, куда больше интересова видеокарточка – больше всего запомнился ряд перемычек (штук девять), которые переключали объём памяти с 256 на 512 килобайт. Увы, без пары дополнительных микросхем такое переключение приводило к мусору на экране. Но если поставить микросхемы, можно добиться 256 цветов на разрешении 800x600 (на «Трайденте», в конце концов, гораздо позже это сделал ☺, и на «Реалтеке» тоже, о чём писал в статье «Не самый крутой компьютер» в 15-ом номере журнала). Заодно было интересно, насколько хорошо материнка переварит винты объёмом около гига.

Ну а раз есть желание и есть необходимое, почему бы не сделать? Так что, начнём.

Подключаем жёсткий диск

Началось всё достаточно просто и прозаично. Жёсткий диск размером 1,2 ГБ был разбит на разделы и отформатирован ещё на 486-м компьютере. Там же была установлена DOS. Когда подключил через мультикарту к материнской плате с 386-м, жёсткий определился примерно в том же объёме, что и на 486-м – 1 ГБ. Ничего не предвещало беды, но из десяти попыток DOS загрузилась раза два. Было решено переустановить, но дисковод отказался читать дискеты. Так как дискет чистых и целых не было больше, а замена дисковода ничего не дала, решено было попробовать поменять мультикарту. Дурацкое решение дало, однако, неплохой результат – на третьей мультикарте DOS загрузилась и грузилась стабильно. Но тут незадача – если системный раздел – партиция на 500 МБ – определялся нормально, и DOS грузилась, то второй раздел не определялся вообще.

Пришлось запустить FDISK, удалить второй раздел и создать его заново. Увы, было опознано только 300 МБ вместо 500-600 метров, но и то хлеб.

После создания нового логического диска и его форматирования, в системе появился диск **D:**, куда можно спокойно складывать установочные комплекты, а также сохранённые копии папки Windows и **autoexec.bat** с **config.sys** на случай отката после неудачного эксперимента. Прибегать к Ontrack Disk Manager как-то не хотелось – решил удовлетвориться таким результатом. Тем более, что на фоне 333-х метров предыдущего жёсткого диска такой объём выглядит просто шикарно – на 486DX2, что стоял у товарища, был ровно гигабайт винт. На этом же жёстком диске прекрасно уместился и дистрибутив Windows 3.1, и сохранённая копия – на диске **D:**, и установленная пятая MS-DOS и Windows 3.1 на диске **C:**. Конечно же, софт в таком виде существовать не будет, и будет постепенно расширяться ☺.

Эпопея видеокарты

С жёстким диском всё обошлось – можно сказать, отделался легким испугом. Увы, с видеокартой было куда сложнее. Для начала стоит сказать, что ваш покорный слуга, как выяснилось, напрочь забыл, какая видеокарточка была воткнута до установки «Трайдента». Всё время в памяти всплывало название Realtek и вспоминалось, что на плате для переключения с 256 на 512 КБ памяти было восемь или девять перемычек. Интернет показывал несколько видеокарт на ISA с реалтековским чипом, но всё было не то. Продавцы тоже разводили руками – видеокарта, где объём устанавливается перемычкой – не редкость, но чтобы столько перемычек и одинакового назначения? Нет, не было. Но однажды повезло – на «Полигоне призраков» была выставлена видеокарточка и как раз имела ряд перемычек переключения объёма. Только производитель чипа был не Realtek, а HMC.





Рис. 2. Видеокарта в изначальном виде

Карточка была куплена, получена, но пролежала без дела полтора-два года. Наконец была извлечена, в пустые кроватки были установлены микросхемы памяти от убитого «Трайдента» – две микросхемы уже были использованы для апгрейда Realtek'a ☺. Перемычки, как тринадцать лет назад, снова были переставлены в положение 512 КБ (см. рис. 3).



Рис. 3. Видеокарта после «апгрейда»

И вот, момент истины – вставляем видеокарту в материнскую плату и запускаем. И да, на экране появляется строчка, извещающая, что видеокарта действительно теперь имеет на борту 512 КБ оперативной памяти. Что ж, остаётся только установить драйвера и поменять разрешение.

А вот тут, как выяснилось, лежал подводный камень. Видеокарта собрана на чипсете NM86304. Может быть, я плохо искал, но удалось только найти парочку тем на «Вогонсе» с просьбой помочь найти драйвер, [страничку](#), которая, такое ощущение, переадресует на DriverGuide, и больше ничего. Забегая вперёд, скажу, [обсуждение](#) на «Полигоне» обнаружил, только когда уже писал статью. Не знаю, почему не выпало в первый раз. Сам взял драйвер от NMC86314 по [ссылке](#). Хотя и боялся, но драйвер установился без проблем. В принципе, 256 цветов и разрешение 800x600 появились ☺ (см. рис. 4).

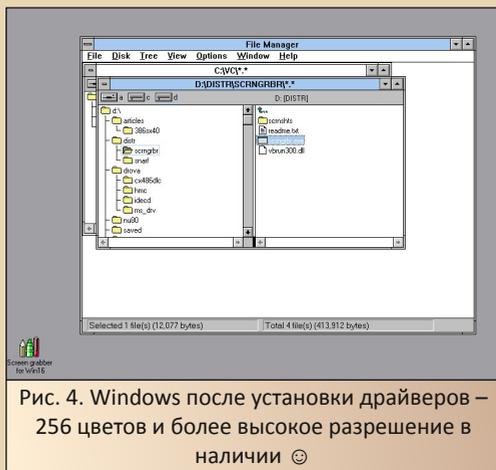


Рис. 4. Windows после установки драйверов – 256 цветов и более высокое разрешение в наличии ☺

Так получилось, что во время возни с этой материнской платой автор сумел добиться только 256 цветов при разрешении 640x480. Если бы удалось тогда повысить разрешение даже на Trident, автор бы выпивал грудь колесом. Увы, тогда получилось выжать из достаточно неплохой видеокарточки только минимум. Творение НМС было вообще заброшено. Сейчас же даже оно было немного «прокачено» и демонстрирует достаточно неплохие результаты. Правда, прорисовывает всё несколько медленней – на плате распаяны микросхемы на 100 нс, что, увы, заметно, но всё равно даже это не особо омрачает радость от полученного результата.



вила попотеть в поисках драйверов, но не разочаровала. Хотя, даже на «Трайденде» автор, большей частью, наблюдал картинку с разрешением 640x480 в шестнадцать цветов – и ничего. ☺

За кадром также остались такие апдейты, как установка сопроцессора и 4-метровых планок. Увы, к установке сопроцессора автор относится с некоторой прохладцей. Когда-то спрашивал, насколько полезна будет такая установка, но из ответов понял, что наибольшая польза, лично для меня, не пользующегося САПРами и графическими пакетами, будет в том, что кроватка для сопроцессора не будет больше пустовать. Касательно третьей винды, создалось впечатление, что радикально быстрее она работать не станет, хоть и существует патч для поддержки сопроцессора. Установка же 16 МБ ОЗУ показалось настолько фантастичной в своё время, что автор так и не приобрёл такие планки – показалось куда более интересным найти материнки, содержащие или восемь слотов SIMM 30-pin, или же хотя бы с одним слотом для 72-пиновых, какой стала, например, материнская плата с UMC U5S.

Что можно было ещё добавить? Конечно же, призвать более внимательно присмотреться к валяющимся под ногами подобным материнским платам. Звёзд они с неба не хватают, но собрать, в принципе, рабочую систему с Windows 3.x на относительно ёмком жёстком диске с CD-приводом на ней можно – и с удовольствием поиграться. Но автор собрал конфигурацию несколько по другому поводу.

Так получилось, что зимой была куплена примерно такая же материнская плата, но с процессором 486SLC 33 МГц (см. рис. 6). Практически такая же плата, с таким же чипсетом, но процессор!!! Кто-то считает, что с четвёрками роднит его только цифра четыре в названии, но всё равно создаётся впечатление, что перед нами не «тройка», а хотя бы 3,5-86. ☺

Но, подключив к материнской плате жёсткий диск, автор вспомнил 386-ю материнку – возможно, и она может показать не меньший класс, что и послужило поводом к написанию статьи.



Рис. 6. Материнская плата с 486SLC – как раз на неё было решено водрузить и видеокарту, и, возможно, данный жёсткий диск

В заключение же приведу показатели теста процессоров двух материнских плат – измерялось, правда, на скорую руку утилитой Sysinfo из комплекта Norton Utilities 8.0, но результат, думается, достаточно интересный:

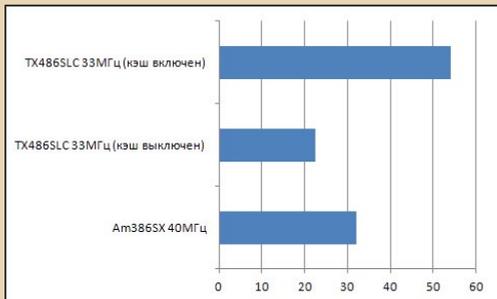
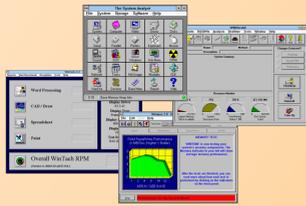


Рис. 7. Сравнительный тест 386SX 40 МГц и 486SLC 33 МГц – с выключенным и включенным кэшем

Увы, с некоторыми программами на 486SLC возникли проблемы – например, установленная под 386-м Windows 3.1 отказалась запускаться, сообщив об ошибке контроллера прерываний. Делалось это с наскока, так что, думается, доведению данной платы будет посвящена ещё одна статья. ☺ Как и, думается, сборке компьютера на её основе.





ПРОГРАММЫ-ТЕСТЫ для Windows 3.1



Получить информацию об установленном на компьютере железе, а также оценить возможности этого железа, уровень его использования (в первую очередь, объём используемой памяти) – весьма интересно, а порой просто необходимо, когда нужно разыскать драйвер видеокартки, звука или чипсета материнской платы. А ещё интересно порой поглядеть, насколько крут собранный компьютер – кого обгоняет, кому наоборот уступает, либо провести соревнование среди процессоров, чипсетов или видеокарт. А как понять, насколько эффективной оказалась та или иная оптимизация или разгон? Для всех этих случаев используют программы-тесты.

Казалось бы, ну показывают пользователю на экране, какой установлен процессор, какого объёма жёсткий диск, оперативная память и т.д., ну покажут диаграмму сравнения производительности с несколькими сферическими, аки конь в вакууме, компьютерами на разных процессорах – чего тут такого? Ан нет, всё равно интересно. Да так интересно, что такие тестовые программы, пусть и не самые умелые и разборчивые в железе, включали в наборы утилит, такие как Norton Utilities и PC-TOOLS. К тому же System Information включалась в комплект нортонских утилит вплоть до NU 2007 включительно, если верить Википедии – сам пользовался максимум NU 2003.

И уж, тем более, существуют отдельные тестовые программы – куда более знающие и умелые – определяют установленное железо и протестируют производительность, причём, сравнив по достаточно объёмной базе различных конфигураций, а то и покажут, насколько производителен каждый элемент системы – судя по постам на «Полигоне призраков», CheckIt может дать исчерпывающую информацию по этому вопросу. Настолько исчерпывающую, что автор сам бы с удовольствием почитал о тонкостях толкования результатов измерений производительности, проведённых этой программой. Большинство тестовых программ разработано либо для DOS, либо для Win32. Однако присутствуют тестовые программы и для Windows 3.x. Так что ничего не мешает фанату третьей винды скачать несколько таких программ для своей любимицы и позапустить их на радость себе и окружающим. Даже если полученные сведения будут не слишком корректны, можно просто полюбоваться на дизайн программы. Ведь, например, System Consultant из пакета PC-TOOLS произвела большее впечатление именно внешним видом, а не умелостью.

Для обзора автор нашёл следующие программы:

- **WinTach**
- **WinBench 96**
- **WinTune 2.0**
- **SAW 1.10**
- **WinCheckIt 4.0**

Интересно, что первые две программы использовались А. Фрунзе при тестировании различныхборок в цикле статей «Модернизирuem IBM-совместимый ПК», публиковавшемся в журналах «Радио», так что эти программы можно, думается, уравнивать в авторитетности, как минимум, с Norton SysInfo. ☺

Итак, приступим.



WinTach – чисто офисный тест

С программой WinTach вышла загвоздка – Фрунзе почему-то указал в качестве автора великую и ужасную Microsoft, в диалоге «О программе» указывается Texas Instruments. Поверим, думается, диалогу ☺ – создателям программы всё же виднее, где они работают ☺. Что же тестирует программа? Собственно, производительность системы, но, в первую очередь, ориентируется на измерение скорости выполнения офисных задач – открытие документа текстовым редактором или электронной таблицей, работы с САПР и графическим редактором.

Автор, в принципе, практически и не использует Windows для каких-то других целей – только эксперименты с офисными пакетами, так что тест, проверяющий выполнение именно таких задач, очень даже интересен.

Взять программу можно на old-dos.ru: <http://old-dos.ru/dl.php?id=7948>. Программа не содержит инсталлятора и может быть запущена сразу после распаковки. Внешний вид программы показан на рис. 1. После запуска окно программы разворачивается на весь экран, но может быть легко ужато в размерах (см. рис. 2).

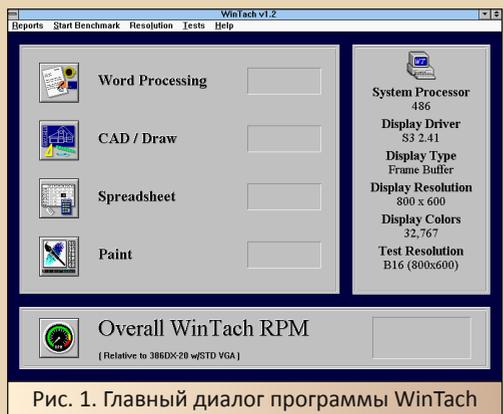


Рис. 1. Главный диалог программы WinTach



Рис. 2. Окно программы после ужатия в размерах – ужать удалось не очень много, чтобы сохранить информативность, но работать с программой куда удобнее

Программа тестирует выполнение компьютером четырёх типовых офисных задач:

- работа в текстовом редакторе;
- работа в САПР;
- работа в табличном процессоре;
- работа в графическом редакторе.

Каждый из тестов запускается отдельной кнопкой из главного диалога программы. Как видно из подписи внизу окна, производительность измеряется относительно компьютера с 386-м процессором частотой 20 МГц, снабжённого стандартным VGA-адаптером.

Ну, запускаем тест...

Увы, программа оказалась с норовом. При запуске теста с открытыми фоновыми приложениями – скриншотер, диспетчер файлов и др. – автор получил предупреждение (см. рис. 3).

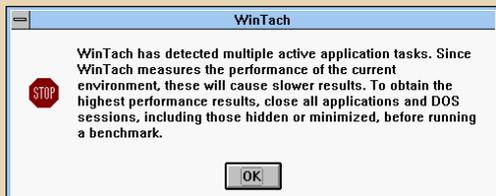


Рис. 3. По мнению программы, запущенные, кроме теста, приложения могут повлиять на измерение производительности, посему, их необходимо закрыть



Поэтому закрываем все приложения, кроме теста и диспетчера программ, и запускаем. При запуске теста открывается окно текстового редактора с открытым файлом и к нему последовательно применяются различные процедуры форматирования (см. рис. 4). В частности, автор заметил изменение разметки и изменение кегля.

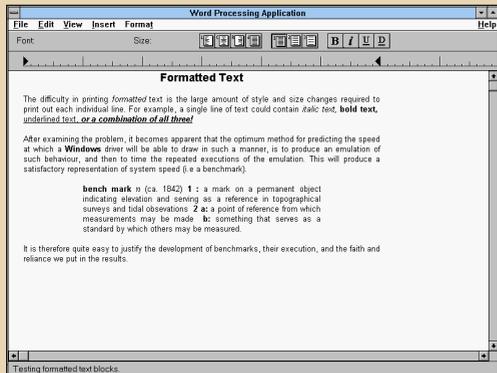


Рис. 4. Процесс тестирования работы с текстовым редактором

По завершении теста выводится диалог с результатом (см. рис. 5). Интересно, что диалог блокирует переключение окон по **Alt-Tab**, и перед тем, как свернуть программу теста, придётся щелчком мыши закрыть этот диалог. Поэтому, чтобы сохранить скриншот процесса тестирования и результата, автору пришлось прогонять тест дважды. Однако, как потом выяснилось, данный диалог можно вызвать щелчком левой кнопки мыши по числовому значению.

Word Processing RPM: 13.12 / B16

Performance Comparison
(Longer Bar Indicates Better Performance)

	13.12 / B16
--	--------------------

This System: 486/S3 2.41 (S3FLAT.DRV)

Рис. 5. Диалог результатов теста

Аналогичным образом выглядело тестирование работы с САПР, таблицей и графическим редактором (см. рис. 6 – рис. 8 соответственно). На экране мы видим плату расширения компьютера, но, судя по элементам управления (увы, на скриншот не попали), САПР скорее обычная – механическая, скажем так ☺. Достаточно интересно, использовались ли в качестве тестов элементы реальных программ или же их муляжи с типовыми функциями.

Кнопка **Overall WinTach RPM** запустит последовательное выполнение всех четырёх тестов.

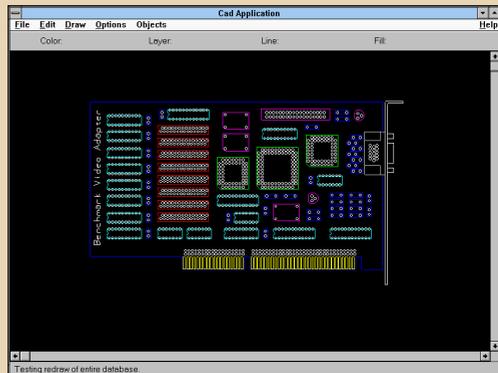


Рис. 6. Процесс тестирования работы с САПР

		Jan	Feb	Mar	Apr	May	June	July	Aug
4	Cog Sales	87.23	93.45	106.65	156.57	98.38	135.48	187.00	206.45
5	Sprocket Sales	208.30	174.29	151.20	101.23	79.81	63.02	31.87	83.24
6	Cog Advertising	23.04	28.90	35.78	43.27	34.16	32.36	31.58	29.76
7	Sprocket Advertising	12.93	17.89	11.69	10.97	11.95	12.45	13.23	16.76
8	Cog Cost	52.31	56.07	63.99	93.94	69.02	87.74	83.59	103.22
9	Sprocket Cost	123.78	104.57	83.16	55.72	39.90	31.51	14.34	37.46
11	Cog Profit	11.88	6.48	6.86	13.36	5.00	35.30	81.94	73.46
12	Sprocket Profit	70.45	57.74	56.54	34.48	27.86	19.08	4.30	27.93
14	Total Profit	82.35	66.22	63.42	53.86	33.16	54.44	86.24	100.46
20									
21									
22	Cog Sales	87.23	93.45	106.65	156.57	98.38	135.48	187.00	206.45

Рис. 7. Процесс тестирования работы с табличным процессором



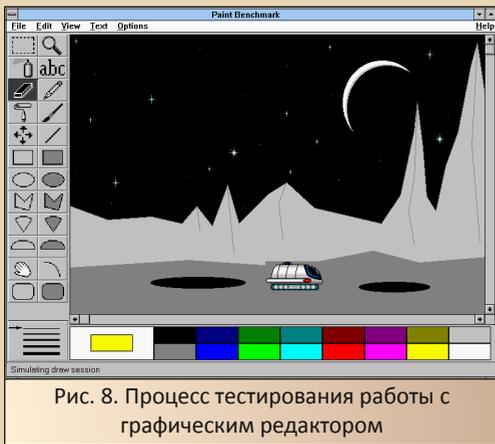


Рис. 8. Процесс тестирования работы с графическим редактором

Результаты выполнения тестов в виде цифровых значений сохраняются в соответствующих полях главного диалога (см. рис. 9).

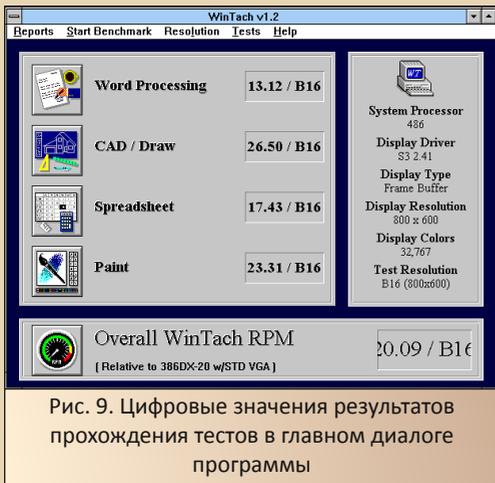


Рис. 9. Цифровые значения результатов прохождения тестов в главном диалоге программы

Некоторые дополнительные функции доступны через выпадающее меню. Например, присутствует возможность распечатки или копирования результатов тестирования. Меню **Start Benchmark** хоть и дублирует кнопки главного диалога программы, но содержит также пункт непрерывного прогона тестов. Меню **Resolution** позволяет задать разрешение меньше, чем установленное для Windows.

В целом, достаточно симпатичная и лёгкая программulina. Кроме того, приятно, что тестируется производительность системы именно для офисных задач, хотя, тут скорее лучше оперировать результатами тестов в сравнении для различных машин, основываясь на впечатлениях от реальной работы в том же ворде, экселе или файнридере, но разве не так же интерпретируют любые результаты тестов?

WinBench 96

Ещё одна тестовая программа, использованная А. Фрунзе в своём цикле. Данная программа была найдена автором в пакете тестов сайта vogons.org. Отдельно архив программы был загружен на old-dos.ru: <http://old-dos.ru/dl.php?id=19487>. Данная программа уже содержит инсталлятор и, после установки, создаёт свою программную группу с ярлыком деинсталлятора. Из этой программной группы и запускаем программу щелчком по значку WinBench 96. Главное окно программы показано на рис. 10. Окно чем-то напоминает диспетчер программ или же окно «Евфрата» от Cognitive Technologies – наверное, из-за пиктограмм, открывающих результаты тестов, внизу окна. Однако управление осуществляется с панели, находящейся в левом верхнем углу.

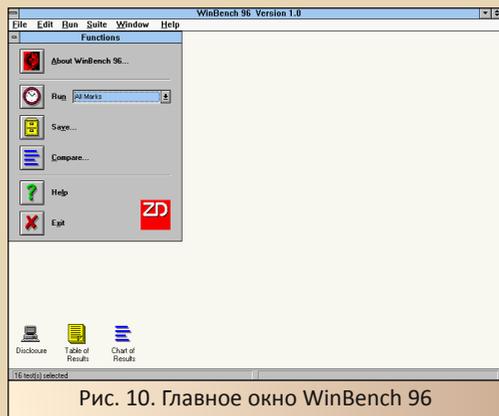


Рис. 10. Главное окно WinBench 96



Программа также предлагает не один тест, а целый набор тестов, проверяющих самые разные подсистемы, скажем так, компьютера:

- тесты процессора – работа в 16-ти и 32-х разрядном режиме;
- тесты видео;
- тесты дисковой подсистемы.

Как выяснилось, для выполнения теста процессора в 32-х разрядном режиме необходима установленная Win32s, в противном случае, при запуске получаем сообщение о невозможности запуска соответствующей тестовой программы:

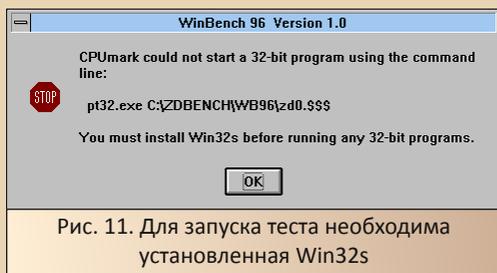


Рис. 11. Для запуска теста необходима установленная Win32s

Запускается выполнение тестов нажатием кнопки **Run**. Рядом с ней располагается выпадающее меню, позволяющее выбрать те или иные тесты. По умолчанию запускается выполнение всех тестов.

Перед запуском тестов программа выводит диалог с некоторыми параметрами системы – объём оперативной памяти, наличие 32-разрядного доступа к диску и файлам (см. рис. 12).

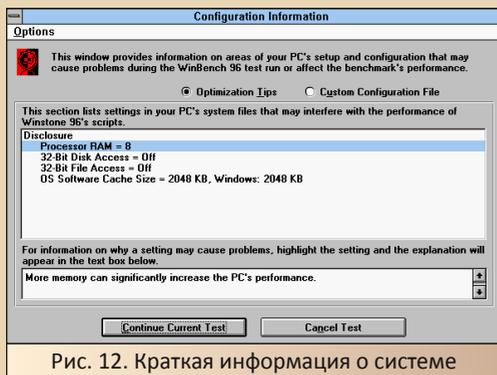


Рис. 12. Краткая информация о системе

После нажатия кнопки запуска теста открывается тестовое окно. Большую часть времени оно представляет собой белый фон со значком ZD посередине. Однако в какой-то момент окно начинает забиваться какой-то абракадаброй – надписи, элементы то ли окон, то ли следы глюков (см. рис. 13).

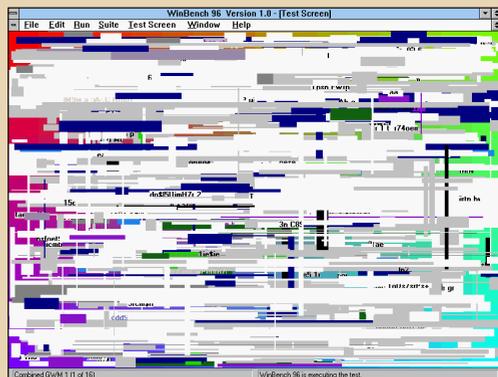


Рис. 13 – А вы бы не испугались, увидев такое в первый раз?

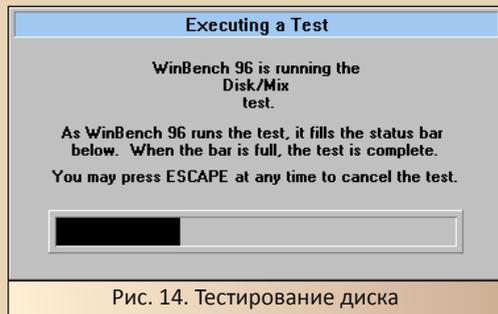


Рис. 14. Тестирование диска

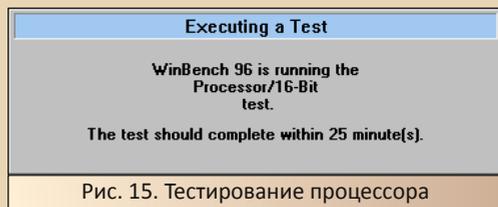


Рис. 15. Тестирование процессора

Когда-то автор запускал одну из версий WinBench на одном из компьютеров. Забитый надписями, как на рис. 13, экран конкретно тогда напугал – уж не повисла ли система. Оказалось, что всё же нет.



Интересно, что тесты выполняются куда дольше, чем у WinTach, но, такое ощущение, что и тестируется система подробнее. Например, один из тестов, такое ощущение, проверяет, насколько быстро отрисовывается шрифт Times New Roman. Увы, долго смотреть на то, что пишется на экране «безобразие» скоро надоедает, и начинаешь просто ждать окончания тестирования. Кстати, тесты можно прервать нажатием левой клавиши мыши и ответив «Yes» в появившемся диалоге прерывания тестирования (см. рис. 16).

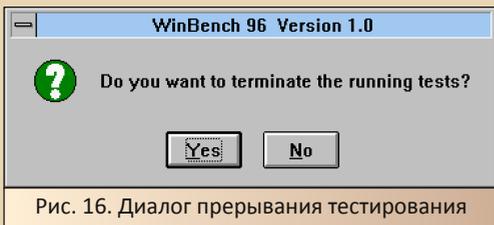


Рис. 16. Диалог прерывания тестирования

После окончания тестов откроется диалоговое окно результатов (см. рис. 17) – **Chart of results**. В первую очередь, будут показаны диаграммы производительности отрисовки тех или иных элементов интерфейса Windows и программ, но далее (можно перейти с помощью прокрутки) последуют результаты тестов производительности дисковой подсистемы (см. рис. 18) и тестов производительности процессора (см. рис. 19).

После закрытия, данное окно можно будет вызвать двойным щелчком на пиктограмме **Chart of results**.



Рис. 17. Диалоговое окно результатов тестов

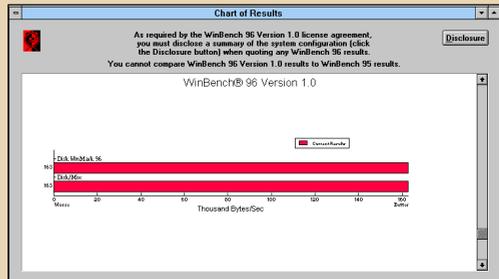


Рис. 18. Результаты тестов производительности дисковой подсистемы

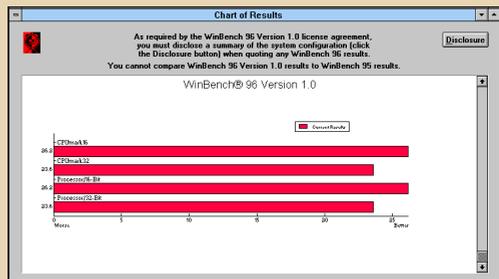


Рис. 19. Результаты тестов производительности процессора

В виде таблицы с цифровыми значениями результаты можно увидеть в окне, открываемом по щелчку на пиктограмме **Table of results** (см. рис. 20).

WEIGHTED SUITE	Current Results	UNITS
Combined GMM 1	1.21 ¹	Milion Pixels/Sec
Combined GMM 2	1.26 ¹	Milion Pixels/Sec
Combined GMM 3	0.776 ¹	Milion Pixels/Sec
Combined GMM 4	1.3 ¹	Milion Pixels/Sec
Combined GMM 5	1.26 ¹	Milion Pixels/Sec
Combined GMM 6	1.56 ¹	Milion Pixels/Sec
Combined GMM 7	0.862 ¹	Milion Pixels/Sec
Combined GMM 8	4.18 ¹	Milion Pixels/Sec
Combined GMM 9	0.954 ¹	Milion Pixels/Sec
Combined GMM 10	0.862 ¹	Milion Pixels/Sec
Combined GMM 11	0.0336 ¹	Milion Pixels/Sec

Рис. 20. Результаты тестов в цифровой форме

Казалось бы, всё – тесты проведены, цифровые значения получены, как и в предыдущей программе, но нет – есть ещё возможность сравнить с показателями тестов из библиотеки программы.



Нажимаем кнопку **Compare**, а в открывшемся окне (см. рис. 21) – кнопку «**Add from Database**».

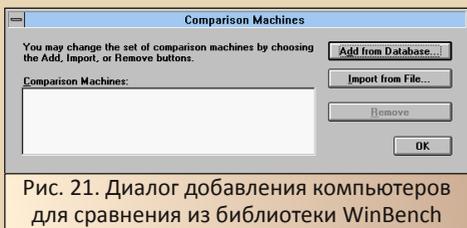


Рис. 21. Диалог добавления компьютеров для сравнения из библиотеки WinBench

Откроемся ещё один диалог со списком компьютеров, содержащихся в библиотеке WinBench (см. рис. 23). Внизу окна находятся строчки для ввода параметров для фильтрации (например, автор ввёл в строке **Processor** значение 486), по которым можно выявить нужные компьютеры. Увы, компьютеров только четыре, и автор добавил их все – двойным щелчком по строке с названием. Увы, это одна база данных, которая есть в составе программы.

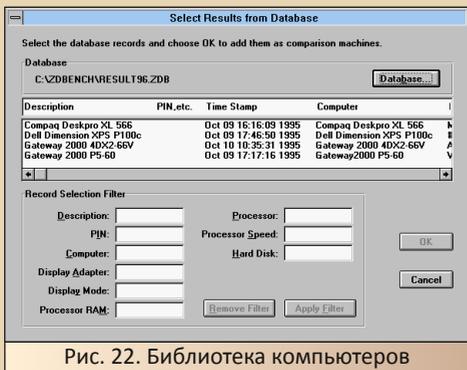


Рис. 22. Библиотека компьютеров

После добавления всех компьютеров, диалоговое окно из рис. 21 приобрело следующий вид:

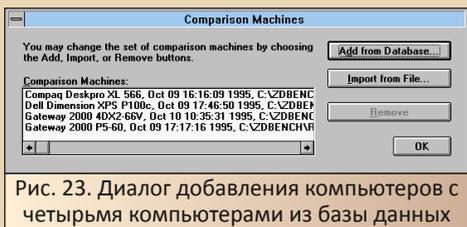


Рис. 23. Диалог добавления компьютеров с четырьмя компьютерами из базы данных

Нажимаем кнопку **OK**, и нашему вниманию снова предстаёт диалог **Chart of results**, но уже с пятью разноцветными столбиками (компьютер автора – красный столбик).

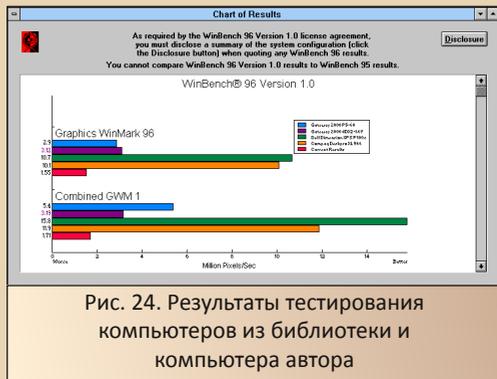


Рис. 24. Результаты тестирования компьютеров из библиотеки и компьютера автора

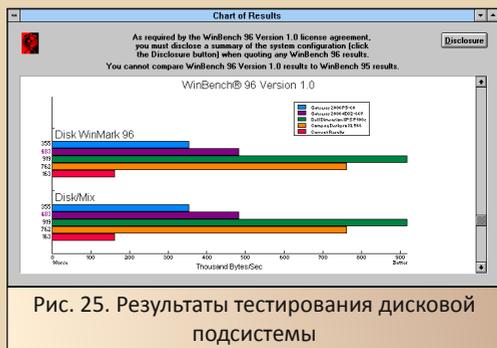


Рис. 25. Результаты тестирования дисковой подсистемы



Рис. 26. Результаты тестирования процессоров

Программа имеет достаточно гибкие настройки по запуску тестов – есть возможность запустить только тесты графики либо только тесты жёсткого диска, только процессоров, а по команде **Run->Select and run tests** откроется



диалог, где можно самостоятельно сформировать пакет тестов.

Можно сказать, что перед нами достаточно умелая, хоть и проигрывающая в эффективности тестирования программе WinTach, утилита. А благодаря встроенной библиотеке результатов тестирования показывающая более наглядно производительность системы. Как видно, компьютер автора, в понимании программы, далёк от совершенства. ☹

WinTune

На эту программу автор наткнулся абсолютно случайно в одном из постов vogens'a, где один из посетителей форума показывал результаты тестирования собранной 486-ой, вроде, машины. Программу можно скачать по адресу: <http://files.mpoli.fi/software/WIN16/UTILS/MISC/WT20.ZIP>.

Программа тестирует следующие параметры:

- производительность процессора;
- производительность оперативной памяти;
- производительность видеосистемы;
- производительность жёсткого диска.

Внешний вид программы показан на рис. 27. Первое, что бросилось в глаза – программа не разворачивается на весь экран. Лично автору такие размеры нравятся больше.

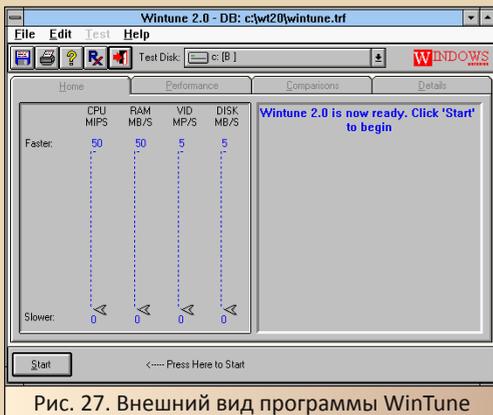


Рис. 27. Внешний вид программы WinTune

Тестирование запускается нажатием кнопки **Start** в левом нижнем углу, к чему пользователя призывает анимированная надпись рядом.

Наиболее интересное в дизайне программы, что бросается в глаза – это трёхмерные диаграммы и графики – например, такой график рисуется при тестировании оперативной памяти:

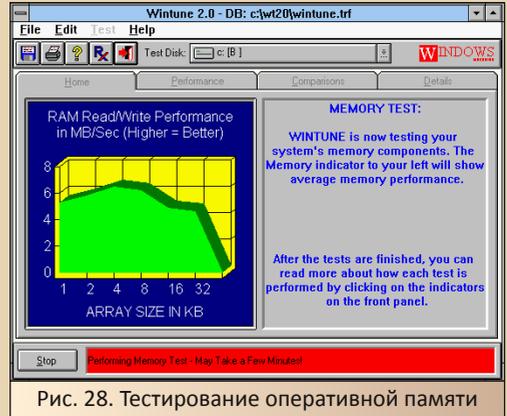


Рис. 28. Тестирование оперативной памяти

После окончания теста в главной вкладке указываются на шкалах результаты тестирования. Кроме того, справа программа выводит советы по улучшению производительности системы (см. рис. 29).

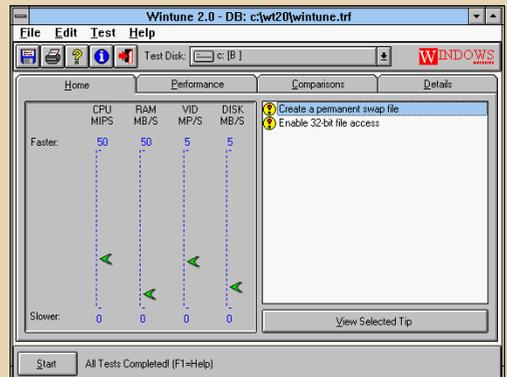


Рис. 29. Результаты тестирования на движковых диаграммах и советы по увеличению производительности компьютера – например, организовать 32-разрядный доступ к диску



Но что можно ещё увидеть у программы? Давайте откроем другие вкладки. Открываем вкладку **Performance** – дизайн уж точно не разочаровал – нашим глазам предстаёт трёхмерная столбиковая диаграмма, показывающая результаты тестирования производительности пяти машин (см. рис. 30) – те же самые параметры, что проверялись у нашего компьютера. Программа для сравнения представляет четырёх конкурентов:

- 386-й компьютер;
- Ноут IBM 720C с процессором 486SLC2 50МГц;
- 486DX33;
- Pentium 90МГц.

Однако список куда шире, и нажав на любую из «конкурирующих» машин вы увидите достаточно длинный выпадающий список, где можете выбрать другую модель – от 386-х до 90-х пеньков.

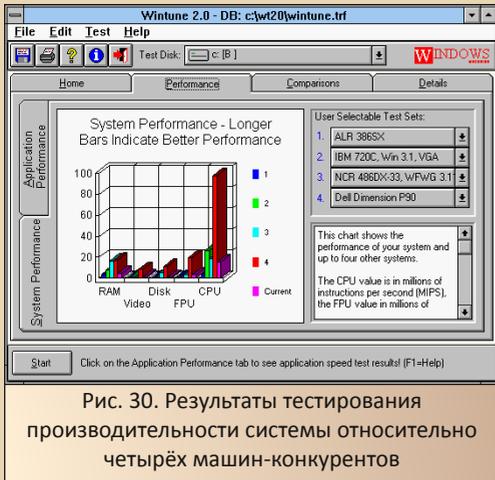


Рис. 30. Результаты тестирования производительности системы относительно четырёх машин-конкурентов

Однако в данной вкладке есть две вертикальные вкладки – **System Performance**, которая открывается по умолчанию и показана на рис. 30, и **Application Performance** (см. рис. 31), где показано, как я понял, время выполнения тестов системами.

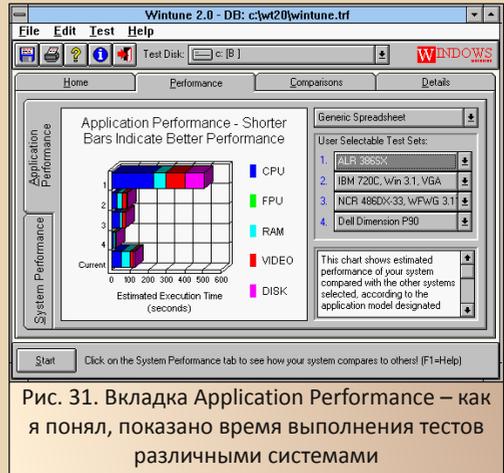


Рис. 31. Вкладка Application Performance – как я понял, показано время выполнения тестов различными системами

Следующая вкладка – **Comparisons** – показывает в крупной диаграмме – кто же быстрее в каком из состязаний. Есть пять вкладок:

- тест процессора;
- тест математического сопроцессора;
- тест дисковой подсистемы;
- тест видео;
- тест оперативной памяти.

На рис. 32 показана диаграмма сравнения результатов теста оперативки.

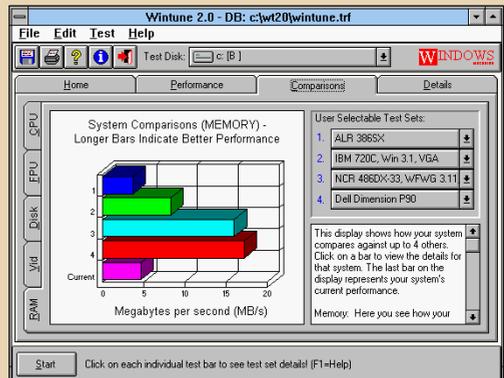


Рис. 32. Крупная трёхмерная диаграмма результатов теста оперативной памяти

И, наконец, вкладка **Details** содержит достаточно подробный отчёт о системе – показатели производительности каждого компонента, объём памяти, объём занятой памяти,



например, для жёсткого диска и т.д. Отчёт выводится в виде обычного текста с прокруткой – см. рис. 33.

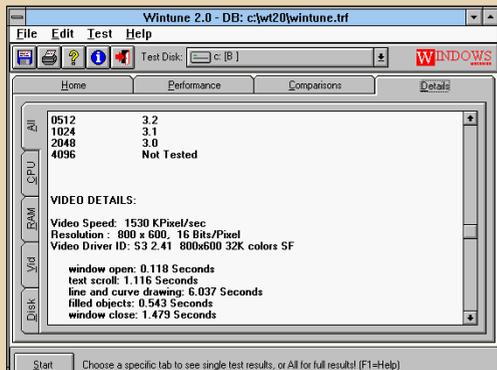


Рис. 33. Детальная информация о системе – просто текст, но достаточно длинный и подробный

Какое впечатление оставила эта программа? Пожалуй, очень приятное – именно такой виделась тестовая утилита – маленькое окошко с кучей столбиковых диаграмм и большой базой компьютеров. Конечно, не хит-парад, как в AIDA32, но четырёх конкурентов можно выбрать из очень большого списка.

SAW – я буду Сандрой

Все описанные ранее программы выполняли задачу тестирования производительности – всей системы, её элементов – просто производительность или применительно к тем или иным задачам. Данная же программа является тестом другого плана – тестом информационным, т.е. показывающим, какое железо установлено на компьютере, насколько занят жёсткий диск и оперативная память и т.д. Обычно для этих целей используется или AIDA32/Everest/AIDA64, или ASTRA (может быть, кто-то использует и AIDA16 под DOS?). В данном же случае мы будем пробовать утилиту от автора теста SiSoft Sandra. Увы, сама «Сандра» была разработана для Windows 95 и NT, но зато SAW может быть запущен и на третьей винде.

Программу можно скачать по адресу: <ftp://ftp.mpoli.fi/pub/software/WIN16/UTILS/MIS/C/SAW110.ZIP>. Также в интернете нашлось описание на английском языке со скриншотами:

<https://www.danielsays.com/ssg-win1x2x3xsw-tsa110.html>

Программа удивила уже на этапе установки – в открывшемся окне необходимо было задать каталог установки, но ещё и выбрать группу в диспетчере программ. По умолчанию программа почему-то просилась в **Accessories** – группу стандартных программ. При этом присутствовали такие настройки, как размещение INI-файлов в каталоге Windows и, куда более безобидная – запуск программы сразу после установки – см. рис. 34.

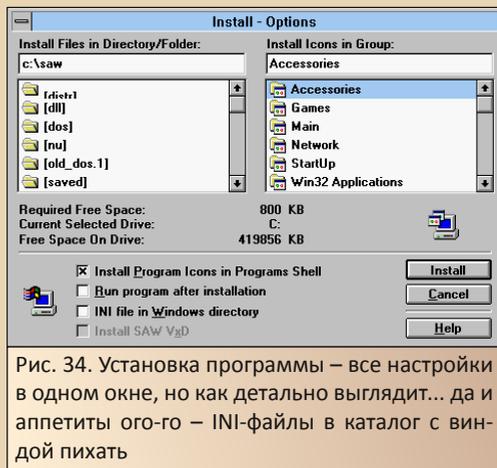


Рис. 34. Установка программы – все настройки в одном окне, но как детально выглядит... да и аппетиты ого-го – INI-файлы в каталог с виндой пихать

Автор решил, что писать INI-файлы в каталог Windows – это перебор, но, если программа хочет непременно в группу стандартных – что ж, можно и так. После окончания установки (программа ещё хотела похозяйничать – удалить группу автозапуска по причине отсутствия там ярлыков) в группе **Accessories** образовался ещё один ярлык для запуска SAW и пара ярлыков для текстовых файлов. Что ж, запускаем программу.

Главное окно показано на рис. 35. Все элементы компьютера разделены на группы и за каждую из групп отвечает свой значок-плитка.



Кстати, путешествовать между разделами можно не только через главное окно, но и нажатием кнопок «<<<» и «>>>» в окне уже открытого раздела – в таком случае, после нажатия вы перейдёте к предыдущему или следующему разделу соответственно.



Рис. 35. Главное окно SAW

Наиболее общие сведения присутствуют в разделах **System** и **Computer** (см. рис. 36 и 37 соответственно). Содержимое разделов практически идентично: процессор, память, клавиатура – разве что, в разделе **System** вы сможете увидеть сведения о видеокарте, портах и версии DOS, а в разделе **Computer** – сведения о шинах и BIOS.

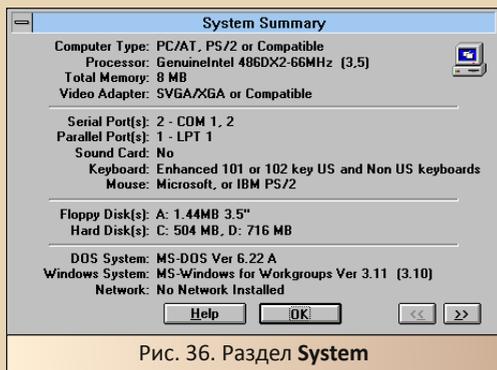


Рис. 36. Раздел **System**

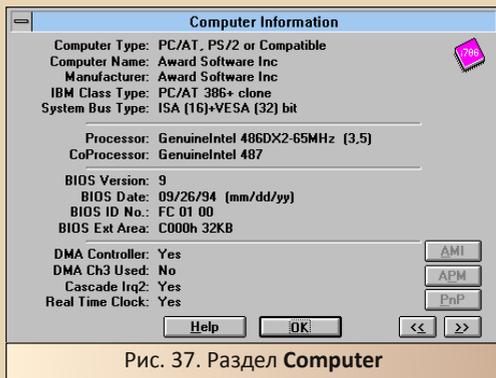


Рис. 37. Раздел **Computer**

Однако это только начало. В общих сведениях мало что сказано о видеокарте, но ей посвящён отдельный раздел – **Video** (см. рис. 38).

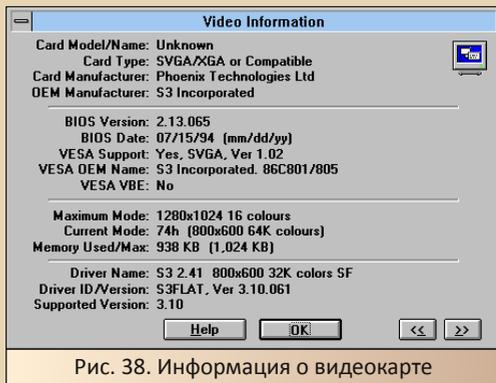


Рис. 38. Информация о видеокарте

Вот тут уж действительно – детальней не придумать – чипсет видеокарты, производитель (где только нашли), объём памяти (правда, в четвёртой строчке снизу ☺), версия BIOS (а это, наоборот, пятая строчка сверху ☺) – текущее разрешение и глубина цвета, максимально возможное для данной видеокарты. Что ж, с последними ориентирами и действительно не очень-то нужен объём памяти – программа и так покажет, на какие показатели равняться.

Кстати, у автора с данным пунктом вышла заминка – в первый раз почему-то отказались нормально отображаться символы. Решилась



проблема нажатием кнопки **Help** и вызовом помощи. Помощь открылась, и тут же буквы постепенно стали различимы.

Следующий раздел – **Drives** – посвящён логическим дискам (см. рис. 39). Здесь в виде симпатичной круговой диаграммы показано, насколько занят каждый из логических дисков, доступных в системе. Не обойдён вниманием и дисковод.

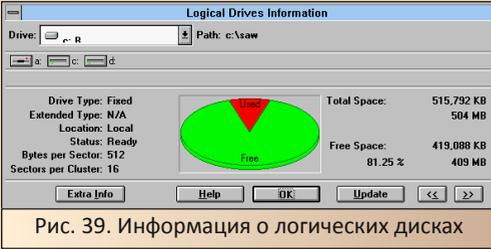


Рис. 39. Информация о логических дисках

Физическим же носителям посвящён раздел **Disks** (см. рис. 40). Информация тут более скудная – тип, объём, количество цилиндров и головок – есть ощущение, что данные читаются из BIOS, но всё равно неплохо.

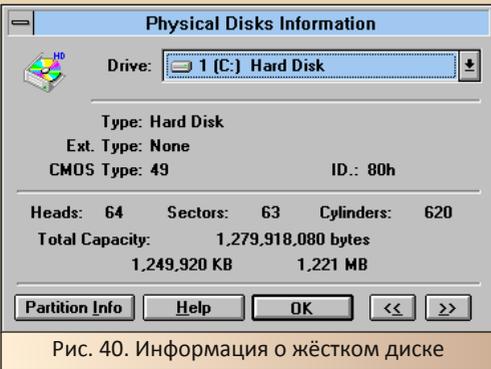


Рис. 40. Информация о жёстком диске

Кроме того, присутствуют разделы, посвящённые портам, клавиатуре, мыши, звуковой карте, принтеру, распределению оперативной памяти, DOS, Windows, драйверам и т.д. Кроме того, есть раздел **Report**, где можно сформировать отчёт и либо его распечатать, либо сохранить в виде текстового файла. Диалог формирования отчёта показан на рис. 41.

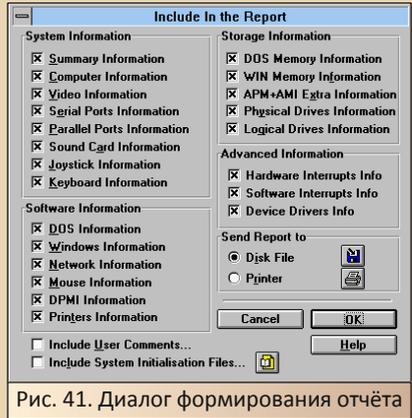


Рис. 41. Диалог формирования отчёта

Что можно сказать о программе? Конечно, едва ли читатель будет использовать её по назначению – для идентификации неизвестного железа, подключенного к компьютеру, для чего предназначена эта программа – скорее воспользуется ASTR'ой, но всё равно очень приятно, что программы такого класса существовали и для Windows 3.x. В принципе, программа показала всё верно, хоть автор и не пробовал узнать с помощью программы модель звуковой карты (обычно это наиболее насущный вопрос, за которым обращаются к таким программам), но всё равно остался доволен результатами.

CheckIt – ещё и для Windows

Программа CheckIt для DOS заслужила в глазах автора звание весьма подробной, дошной, но очень мудрёной по строению интерфейса утилиты, выдающей при этом достаточно много полезной и интересной информации о железе. DOSовская версия CheckIt стала настоящим анализатором, позволяющим измерить производительность практически каждой подсистемы компьютера с такой детальностью, какая помянутым выше программам и не снилась. Это показывали сто раз и пользователи форума «Полигон Призраков», и пользователи форума Vogons с помощью этой программы измеряли и эффективность кэша – первого уровня, второго, а то и третьего, если



аппаратно получалось, оценивали скорость работы с оперативной памятью, и, вроде бы, даже с жёстким диском. Но вашему покорному слуге проще было читать толкование самого автора поста касательно полученных результатов, чем самостоятельно изучать скриншот.

Сперва думалось, что авторы CheckIt для Windows просто перенесли интерфейс ДОС-овской программы в виндозные реалии, как это было, например, с играми от Gamos, и на выходе автор сделает просто один скриншот главного окна – смотрите, собственно, имеем примерно то же самое – награждение графиков, линейку производительности с метками-процессорами и некие численные данные, но в виндозном окне – на этом всё и закончится. Но программа явила абсолютно другой интерфейс, посему, автор решил остановиться на программе также подробно, как и на предыдущих.

Для опытов была взята четвёртая версия, которая доступна на [Old-DOS](#). Первое, что предлагает программа при запуске – добавить текущий компьютер в коллекцию. Собственно, можно отказаться – автор так и сделал, но программа покажет минимальные сведения о забитости памяти, дискового пространства, системных ресурсов, а половина разделов будет недоступна (см. рис. 42).

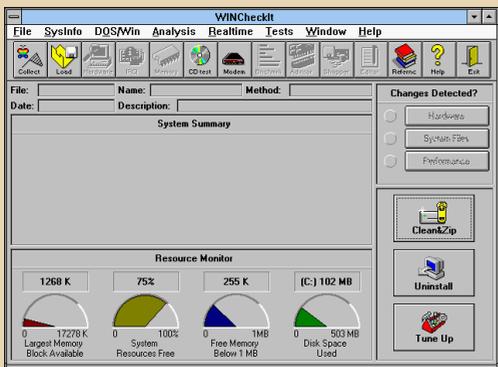


Рис. 42. Компьютер не «разделен в коллекцию», посему половина разделов вообще не активна, так что, по большому счету, пользователю остаётся лицезреть диаграммы использования памяти, жёсткого диска и системных ресурсов

Однако не стоит горевать – достаточно нажать кнопку **Collect** на панели инструментов. Появится диалог добавления компьютера в «коллекцию» программы (см. рис. 43), и после нажатия кнопки «**Collect Now**» программа приметесь тестировать вашу систему (см. рис. 44 и 45).

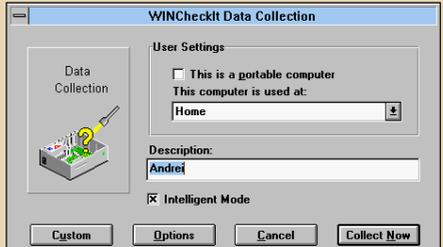


Рис. 43. Диалог добавления компьютера в «коллекцию»

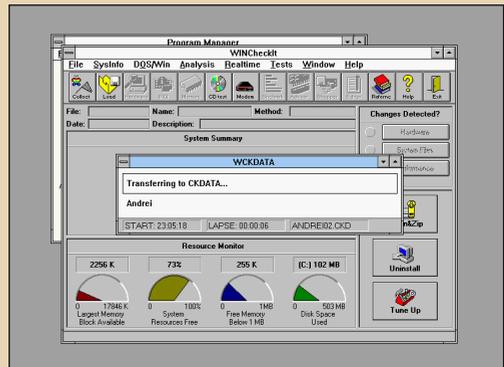


Рис. 44. Сканирование системы

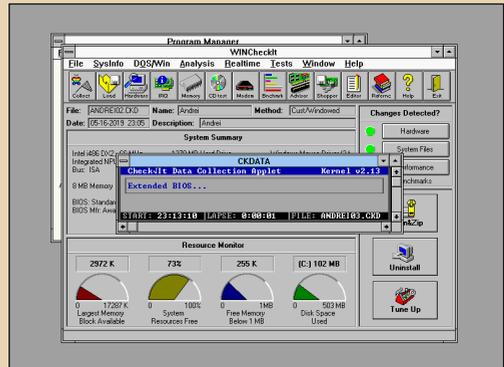


Рис. 45. Даже какая-то досовская утилита запускается...



В конце сканирования системы программа предложит провести дополнительный тест мультимедийных возможностей. Так как у автора ничего мультимедийного на компьютере нет – даже звуковой карты и CD-привода, то он от такого предложения отказался.

Зато после окончания сканирования все разделы программы стали активны (см. рис. 46).

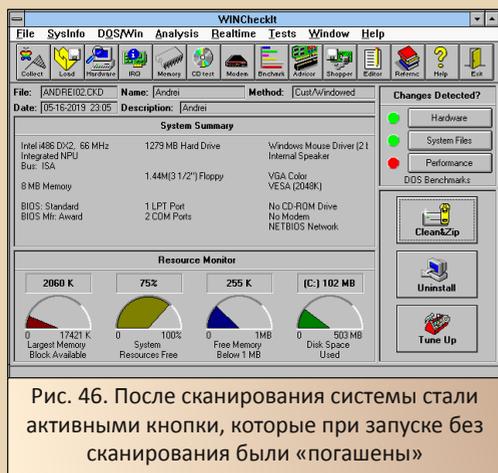


Рис. 46. После сканирования системы стали активными кнопки, которые при запуске без сканирования были «погашены»

Читатель может заметить в правой стороне окна раздел «Changes detected». Автор не обратил на этот раздел внимания при первом запуске, но сейчас решил посмотреть его поближе. Как выяснилось, программа анализирует изменения результатов тестов в сравнении с предыдущими сканированиями и, как я понял, выделяет красным, если изменения были значительными, зелёным – если практически ничего не изменилось. Итак, по нажатию кнопки «Hardware» появилось окно со списком, возможно, всех характеристик компьютера (см. рис. 47). Изменившиеся были отмечены красным цветом. Оказалось, что программа сочла изменениями, например, смену даты и времени и увеличение использования буфера сопроцессора.

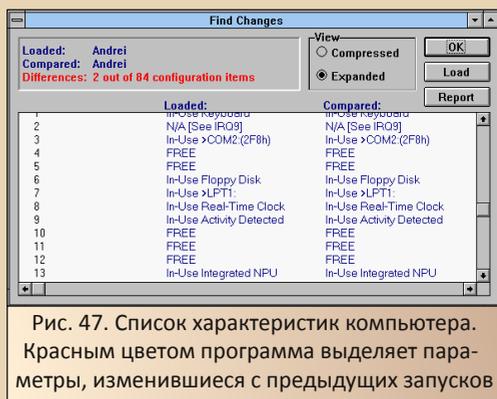


Рис. 47. Список характеристик компьютера. Красным цветом программа выделяет параметры, изменившиеся с предыдущих запусков

По кнопке «System Files» программа выдаёт отчёт о состоянии пяти системных файлов:

- autoexec.bat
- config.sys
- protocol.ini
- system.ini
- win.ini

Наиболее же интересным оказался раздел «Performance» – здесь программа выводит результаты тестов производительности системы. Основное окно представляет результаты тестов (текущие и предыдущие) в числовых значениях (см. рис. 48), но по нажатию кнопки «View Graph» откроется окно со столбиковыми диаграммами результатов каждого теста (см. рис. 49).

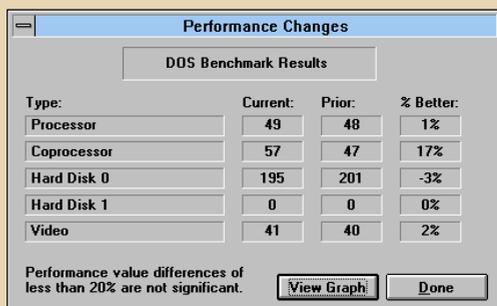


Рис. 48. Окно с числовыми данными результатов тестирования



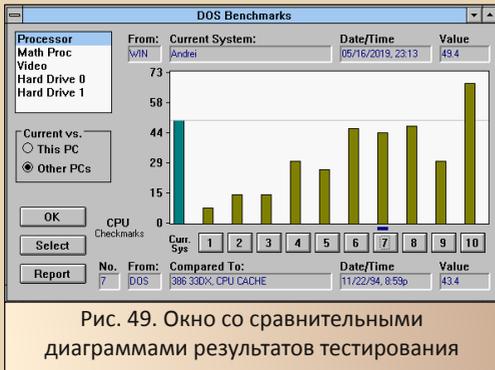


Рис. 49. Окно со сравнительными диаграммами результатов тестирования

Производительность компьютера сравнивается с десятью другими машинами – по умолчанию, в случае автора были взяты машины с процессорами от 286-го до 486DLC, однако по нажатию кнопки «Select» откроется меню, где можно самостоятельно указать «конкурентов» для сравнительного анализа. В библиотеке программы доступны результаты тестирования машин вплоть до Pentium 100 МГц (см. рис. 50).

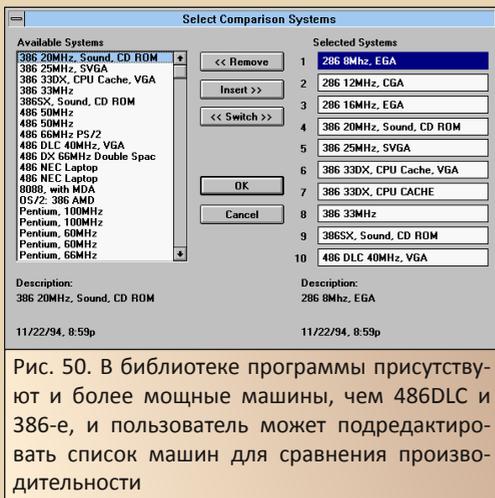


Рис. 50. В библиотеке программы присутствуют и более мощные машины, чем 486DLC и 386-е, и пользователь может подредктировать список машин для сравнения производительности

Кроме того, программа позволяет сравнивать текущие результаты с результатами предыдущих сканирований.

Мы же вернёмся снова к главному окну программы. По нажатию кнопки «Hardware»

программа выдаст структурированную подробную информацию об аппаратном обеспечении компьютера, сопоставимую с данными «Сандры» (см. рис. 51), однако объём оперативной памяти видеокарты у автора определился как 2 МБ (максимально доступный), а не 1 МБ, как установлено на самом деле. А вот частоту процессора, оперативную память, объём логических дисков программа определила правильно. В отличие от «Сандры», здесь присутствует также раздел модема.

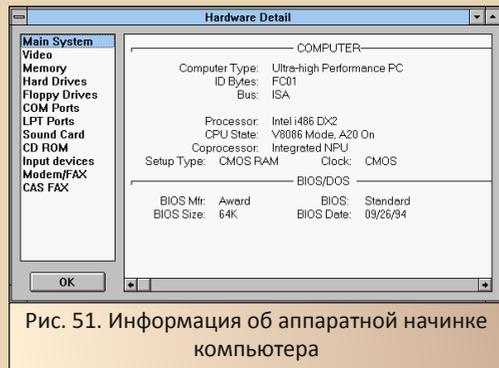


Рис. 51. Информация об аппаратной начинке компьютера

В разделе «IRQ» пользователь может получить сведения об использовании прерываний устройствами компьютера, а также программами.

В разделе «Memory» можно увидеть распределение памяти между различными устройствами компьютера, а также элементами Windows (см. рис. 52). Доступны для «изучения»:

- нижний диапазон – словотуые 640 КБ и UMB-память (кнопка «Base/Upper»);
- EMS-память (кнопка «Extended»);
- XMS-память (кнопка «Expanded»).

По нажатию кнопки «Windows» можно увидеть распределение ресурсов винды в пространстве оперативной памяти и файла подкачки. Чтобы увидеть на диаграмме, кто какое место занимает – необходимо кликать указателем мыши на элементы списков, расположенных в левой нижней стороне окна.



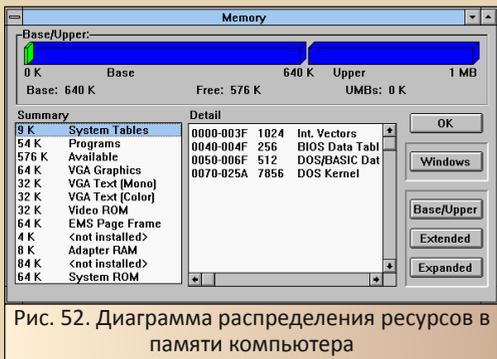


Рис. 52. Диаграмма распределения ресурсов в памяти компьютера

Кнопки «CD test» и «Modem» – должны, видимо, запустить тестирование или выдачу результатов тестирования сидьюка и модема соответственно. Увы, у автора не установлено ни того, ни другого, посему, программа выдала только предупреждения об отсутствии устройств.

Кнопка «Benchmark» таки запустила тест мультимедийных возможностей компьютера – тестировались не только отсутствующие сидюк и звуковуха, а вернее, они-то и не тестировались ☹, но и процессор с сопроцессором, видеоподсистема, накопители.

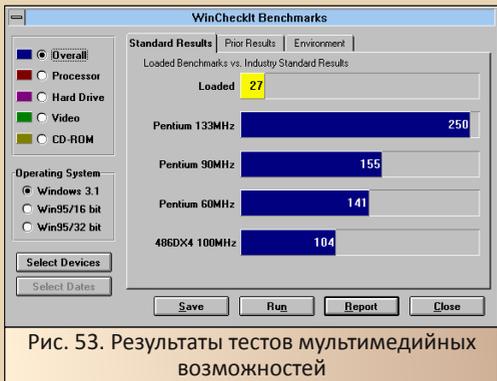


Рис. 53. Результаты тестов мультимедийных возможностей

По кнопке «Advisor» открывается такая библиотека-диагност (см. рис. 54) – уж не знаю, как лучше назвать. Чем-то напоминает диалог установки драйверов Windows, но при этом имеет более обширный список устройств и... драйвера не устанавливает, а просто проверяет – установлено ли устройство и с какими

параметрами. Последняя процедура выполняется по нажатию кнопки «Analyze».

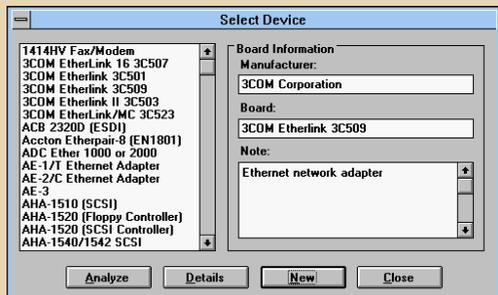


Рис. 54. «Анализатор», вызываемый по нажатию кнопки «Advisor»

Следующая кнопка – «Shopper» – достаточно интересный раздел – консультант по выбору софта (см. рис. 55) – в левом списке достаточно внушительная библиотека САПров, графических редакторов, утилит и других софтин. Выбрав ту или иную программу, пользователь может увидеть её системные требования, а кнопкой «Analyze» проверить, хватит ли мощности компьютера на интересующую софтинку. Кроме того, если что, программа укажет, что нужно доустановить. Например, для использования программы BitFax автору нужно установить модем – предсказуемо, но всё равно сам был удивлён, узнав, что компьютер нуждается в апгрейде для установки факсимильной программы.

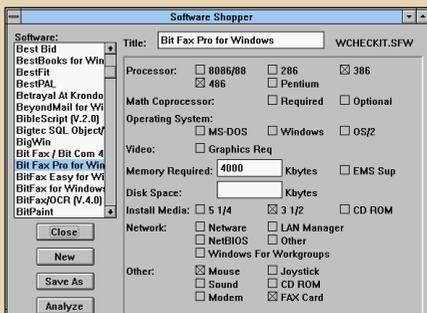


Рис. 55. Система-консультант – покажет системные требования интересующей вас программы, а также проверит, потянет ли её ваш компьютер – если что, сообщит, что нужно установить



Кроме того, присутствует редактор системных файлов, вызываемый по кнопке «**Editor**», и гипертекстовая справочная система по компонентам компьютера (кнопка «**Reference**»).

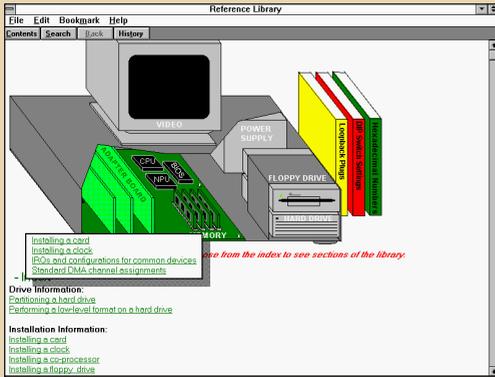


Рис. 56. Гипертекстовая справочная система по компонентам компьютера – обычный файл помощи, но всё равно выглядит эффектно

Также у автора вызвала интерес кнопка «**Tune Up**» – думалось, что после нажатия выскочит какой-нибудь мастер увеличения производительности или советчик, который укажет узкие места в системе – типа того, что мы видели в WinTune, однако ничего не произошло – возможно, что немного высвободилось памяти на диаграммах занятости – только и всего.

Кроме того, Tune Up имеет собственный ярлык для запуска. По щелчку на ярлычке открывается окно, показанное на рис. 57.

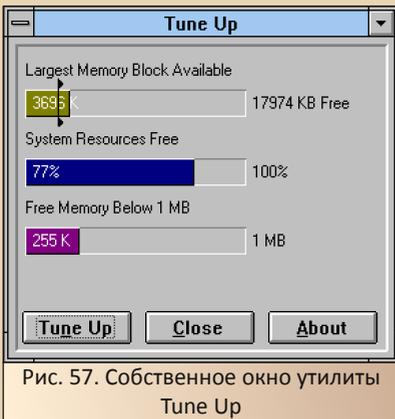


Рис. 57. Собственное окно утилиты Tune Up

После нажатия на кнопку **Tune Up** тоже ничего не произошло – видимо, действительно какой-то оптимизатор использования памяти.

Заключение, или Да будет цвет

Вот и закончился, наконец, экскурс по тестовым программам для Windows 3.x. За кадром осталась, например, программа Checkit Pro Analyst, версия Nokia Monitor Test для Windows 3.1 (вроде бы такая была) и, наверное, другие тестовые утилиты, о которых автор даже не знает. Однако и так получилось достаточно много объёма и очень много картинок.

Увы, у автора, видимо, осталась обида на чёрно-белые иллюстрации в книгах Фигурнова и на страницах журнала «Радио» – да и не особо разбирали интерфейс программ – есть программа, и всё тут. А так хотелось посмотреть, как там у неё всё устроено – какие диалоги, какие настройки. И ещё в цвете. И автор очень рад, что сейчас есть возможность в Downgrade-журнале увидеть и самостоятельно поместить красочные иллюстрации, скриншоты и цветные фотографии старых программ и старого железа. Так что, да здравствует журнал! Да здравствует цвет... в иллюстрациях. ☺

Андрей Шаронов (Andreii88)



РАЗРАБОТКА ПРОГРАММ ДЛЯ ТЕКСТОВОГО РЕДАКТОРА NewsMaster («Журналист») Часть 1 из 3: клавиатурный драйвер



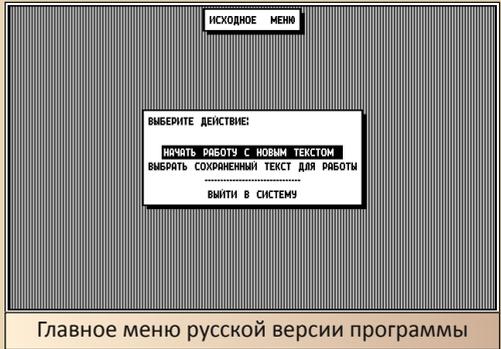
Титульный экран NewsMaster версии 1.0

Редактор NewsMaster, разработанный фирмой UNISON WORLD и выпущенный в 1986 году (первая версия 1.0), является текстовым процессором по типу Microsoft Word. Программа не только позволяет использовать разнообразные шрифты, но и делать разбивку страниц на колонки и столбцы, а также вставлять рисунки со всевозможными типами обтекания текстом из поставляющегося набора библиотек разнообразного клипарта. И, конечно же, производить вывод готового документа на печать – для этого через комплект внешних драйверов была предусмотрена поддержка весьма солидного количества принтеров того времени. Сам же редактор работает в чёрно-белом графическом режиме 06h с разрешением 640x200 пикселей.



Титульный экран русской версии – краткая информация о переводчиках

В конце 80-х – начале 90-х редактор версии 1.0 попал в СССР и был переведён на русский язык отделом 260 (адаптация И К) – к сожалению, кроме этих скудных данных с титульного экрана, про перевод или тех, кто над ним работал, больше ничего неизвестно. При переводе NewsMaster стал «Журналистом» – именно под этим именем он и стал известен широкому отечественному пользователю. Среди всевозможных применений редактора на время его выхода стоит отметить, например, печать мелкого, но читаемого текста на документах (расшифровки и прочее) – именно русская версия «Журналиста», с мелкими, но отчётливо читающимися русскими шрифтами, не раз приходила на помощь и выручала в подобных ситуациях.



Главное меню русской версии программы

Посмотреть и загрузить программу можно с сайта Old-Dos:

http://old-dos.ru/files/file_4512.html

Несмотря на свой солидный возраст, «Журналист» чувствует себя отлично не только под последними версиями DOS или DOSBox, но даже под NTVDM в Windows XP.

Однако одним из недостатков русского перевода было то, что русские шрифты были



рассчитаны на раскладку JCUKEN, а не QWERTY, в результате чего русские буквы не совпадали с теми, что нанесены на клавиши клавиатуры. Вероятно, и сами авторы делали перевод под JCUKEN-клавиатуру.

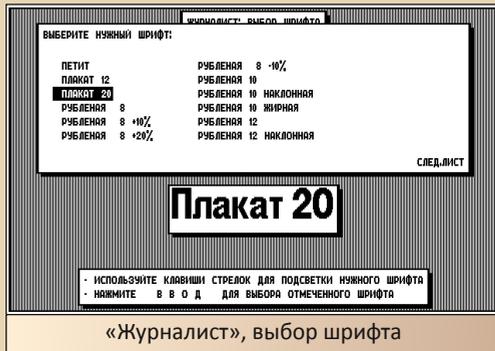
Давайте попробуем этот недостаток устранить.



Замечание. Текст ниже предполагает, что читатель на минимальном уровне знаком с языком программирования Assembler, а также работой прерываний и сегментной организацией памяти в DOS, потому что эти вещи выходят за рамки данной статьи. Далее будут разобраны только отдельные технические моменты, а также показано, как и почему для использования были выбраны именно они.

Самым простым и быстрым для реализации решением будет создание своего резидентного клавиатурного драйвера под DOS, который будет включаться и отключаться (для использования английских шрифтов) по нажатию определённой клавиши на клавиатуре. Проблема с невозможностью «нормального» перевода шрифтов «Журналиста», со вставкой туда русских букв в дополнение к английским, заключается ещё и в том, что помимо разбора файла со шрифтами (неизвестный формат), придётся ещё и как-то модифицировать программу, потому что любые символы, код которых более **7Fh (127)**, воспринимаются ей как управляющие

клавиши (**F1, F2, ...**) – вероятно, именно в силу этого авторы перевода и были вынуждены пожертвовать английскими буквами и частью знаков пунктуации, чтобы разместить на их месте русские буквы.



Но прежде чем начинать что-то делать, давайте оценим входные данные, граничные условия и прикинем примерный план работ.

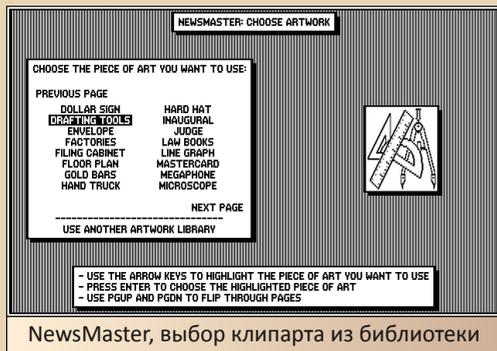
Во-первых, резидентный драйвер – это хорошо, но писать именно полноценную резидентную программу не стоит, потому что это в разы увеличит сложность разработки: нужно будет как-то решать вопрос с предотвращением повторного запуска драйвера, а также, по-хорошему, и выгрузкой из памяти без перезагрузки, что при запуске каких-либо программ после драйвера (а уж тем более резидентных и тоже перехватывающих клавиатурное прерывание) будет приводить к дополнительным лишним заморочкам. Поэтому ограничимся простым загрузчиком такого вида:

- а) Запуск загрузчика.
- б) Загрузчик перехватывает клавиатурное прерывание **09h** и через него работу с клавиатурным буфером.
- в) Загрузчик запускает дочерний процесс **NEWS.EXE**, т.е. «Журналиста».
- г) После завершения **NEWS.EXE** управление возвращается загрузчику, который восстанавливает клавиатурное прерывание назад и завершает свою работу.



По подобной схеме работают многие «трейнеры» и «кряки» для DOS-игр.

Отдельно стоит упомянуть ситуацию, когда дочерний процесс сам перехватывает клавиатурное прерывание для своих нужд, но «Журналист» этого не делает, ему вполне хватает и стандартных сервисов DOS.



NewsMaster, выбор клипарта из библиотеки

Во-вторых, неплохо будет сразу определиться с клавишей для переключения. Для удобства возьмём **Scroll Lock** – эта клавиша практически никем не используется, а световой индикатор на клавиатуре будет наглядно и просто подсказывать текущее состояние режима работы – включено или выключено¹.

В-третьих, нужно разобраться, какой русской букве какой код соответствует. Для начала будет неплохо войти в редактор и напечатать там все четыре ряда клавиш на клавиатуре – цифры и буквы (см. скриншот ниже), а также снова их, но уже с нажатой клавишей Shift².

И здесь сразу можно заметить подвох – среди напечатанных букв отсутствует «ъ» (маленький твёрдый знак). Выясним, есть ли он вообще среди набора символов шрифта. Для этого будем печатать символы вручную при помощи комбинации клавиш **Alt+число**. Начнём перебирать все печатные символы: с 32 (пробел, его можно пропустить и начать с 33) и до 127 включительно. Цифры для числа вводятся на цифровой клавиатуре справа (кеурпад) следующим образом: нажимаем и удерживаем **Alt**, затем вводим номер (например, 33), после чего отпускаем **Alt**. Итак: **Alt+33** даст символ «!»,

Alt+34 – символ «”» и так далее. К счастью, маленький твёрдый знак действительно оказался в наборе шрифта, но под номером **7Fh (127)** – это, увы, непечатаемый символ, который иначе как через **Alt+число** набрать невозможно. Впрочем, для того и пишем программу, чтобы это исправить.

Теперь разберёмся со скан-кодами клавиш клавиатуры, у которых будем менять код символа. Но для начала стоит немного пояснить, как работает клавиатурный буфер в DOS. Важными для нашей программы являются три адреса в оперативной памяти:

0040:001A – адрес указателя на голову (head) клавиатурного буфера (2 байта, WORD);

0040:001C – адрес указателя на хвост (tail) клавиатурного буфера (2 байта, WORD);

0040:001E – адрес начала, собственно, самого клавиатурного буфера (массив, 32 байта).

Клавиатурный буфер организован в виде очереди FIFO (First In – First Out, «первый пришёл, первый ушёл»). При нажатии или отпускании какой-либо клавиши вызывается клавиатурное прерывание **09h**, системный обработчик которого проверяет, что за клавиша была нажата/отпущена и, если это необходимо, помещает в буфер по адресу хвоста пару из кода символа и скан-кода клавиши, затем хвост увеличивается на 2 (1 код символа + 1 скан-код = 2 байта). Буфер кольцевой, поэтому при достижении хвостом или головой конца буфера происходит переход на его начало. При чтении из буфера через прерывание **16h** по адресу головы извлекается слово (два байта) и передаётся запросившей его программе, после чего адрес головы также увеличивается на 2, сдвигаясь к хвосту. Когда адрес головы и хвоста сравняются – буфер пуст. Если не считать (не забирать) нажатые клавиши из буфера, то туда поместится 15 клавиш, после чего компьютер будет пищать, оповещая о том, что буфер полон. Кого-то может удивить, почему 15, а не 16 (32/2 = 16), но, как уже было написано выше, когда адрес хвоста равен голове, то буфер считается пустым, поэтому положить туда 16-е

¹ После выхода или перед запуском загрузчика не забудьте выключить Scroll Lock. Да, его можно выключить программно, но DOSBox, например, физически этого не делает – индикатор на клавиатуре так и будет гореть.

² Стоит напомнить, что стандартный режим Caps Lock действует только на буквы и не действует на цифры и знаки.



значение не представляется возможным, ибо тогда адреса хвоста и головы совпадут.

Рассмотрим на простом примере – предположим, что на клавиатуре были последовательно нажаты три клавиши: «g», «h» и «j». Тогда содержимое буфера будет выглядеть следующим образом (здесь и далее все значения в шестнадцатеричном представлении):

(67 22) (68 23) (6A 24)

Теперь включим **Caps Lock** или нажмём **Shift** и снова наберём эти буквы:

(47 22) (48 23) (4A 24)

Наконец, переключимся на русский язык (при условии, что установлен русский клавиатурный драйвер для 866 кодировки) и опять наберём:

(AF 22) (E0 23) (AE 24)

На что можно обратить внимание:

а) Коды символов могут меняться, скан-коды – нет³.

б) В зависимости от кодовой страницы или установленного национального клавиатурного драйвера, коды символов могут быть какими угодно.

в) Заглавные и маленькие английские буквы отличаются только 5-ым битом в байте (+20h к значению) – мы к этому чуть позже ещё вернёмся.

Таким образом, всё что осталось – это найти скан-коды клавиш, которые нужно заметить (меняться от системы к системе они не будут), и при нажатии на них помещать в буфер уже нужные нам коды символов. Для этого напишем небольшую программу «**showcode**», которая будет показывать код нажатой клавиши⁴.

```
; FASM source code
; http://flatassembler.net/
;
; showcode.asm
; (c) SysTools 2019
; http://systools.losthost.org/
org 100h
```

```
; INT 16H: Keyboard Services
; AH = 00H - Read (Wait for) Next
; Keystroke
mov ah, 000h
int 16h
; output: AH - scan code; AL -
; character code
; character
mov [_txt+6], al
; convert scan code value
call @tohex
mov [_txt+1], dl ; str index 1,
; see below

call @tohex
mov [_txt+0], dl
; convert char code value
call @tohex
mov [_txt+4], dl
call @tohex
mov [_txt+3], dl
; DOS Fn 09H: Display String
mov ah, 009h
mov dx, _txt
int 021h
; DOS Fn 4cH: Terminate Program
mov ax, 04C00h
int 021h
@tohex:
mov dl, al
shr ax, 004h
and dl, 00Fh
cmp dl, 9d
jbe @f
add dl, 'A' - '0' - 10d
@@:
add dl, '0'
ret
; index: 01 34 6
_txt db '## # ?', 00Dh, 00Ah, '$'
```

Запустим получившуюся после компиляции программу **showcode.com**, и после нажатия упомянутой клавиши «g» увидим на экране уже известное нам:

67 22 g

³ Строго говоря, ничто не мешает клавиатурному драйверу менять и скан-коды тоже, но практического применения этому мало.

⁴ В листингах будет только минимальное количество необходимых комментариев. Детальное и подробное описание API прерываний DOS, использованных в коде программ, можно найти в справочнике TECH Help!, доступном для загрузки на сайте Old-Dos в виде интерактивной гипертекстовой справочной системы под DOS (по ссылке также доступна русская версия, правда, менее полная). TECH Help! версии 6 на английском в формате CHM можно загрузить [здесь](#).



Понажимаем на клавиши и выясним, что в общей сложности необходимые нам символы лежат в диапазоне от **02h** до **35h** включительно. Не будет только кодов для **1Dh** и **2Ah** – это левый **Ctrl** и левый **Shift** – клавиатурный драйвер не помещает их значения в буфер (впрочем, их статус сохраняется в другом месте, об этом ниже). Для удобства адресации по таблице расширим диапазон до нуля, но выкинем оттуда, помимо упомянутых левых **Ctrl** и **Shift**, также те клавиши, которые нам не интересны: **Esc (01h)**, **Backspace (0Eh)**, **Tab (0Fh)**, **Enter (1Ch)** – вместе с нулём должно получиться 7 клавиш – именно в этих местах в нашей таблице перекодировки будут стоять нули – все эти клавиши, а также клавиши со скан-кодом более **35h** мы будем передавать на обработку оригинальному клавиатурному драйверу. В итоге должна будет получиться примерно такая таблица:

```

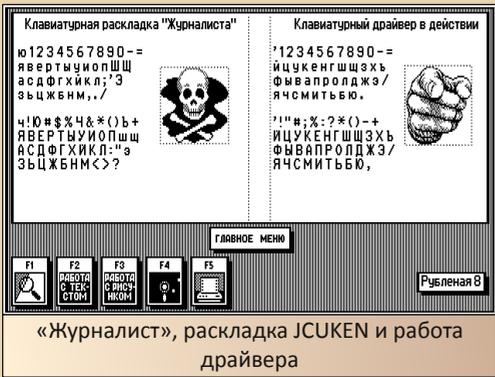
; нижний регистр
db 0,0, '1234567890-=',0
db 0, 'jцukeng{ }zh',07Fh,0
db 0, 'fywaproldv\ ',027h,0, '/'
db 'q~smitxb' .

; верхний регистр
db 0,0, '! "#; :?*( )-+',0
db 0, 'JCUKENG[ ]ZH_',0
db 0, 'FYWAPROLDV\ ',027h,0, '/'
db 'Q^SMITXB@, '
    
```

лишь на значение **+20h**. Если бы не несколько символов, которые нам нужно заменить (чтобы русские буквы не торчали в тех местах, где им быть не положено), то можно было бы обойтись одной таблицей с заглавными буквами и сэкономить 54 байта – достаточно было бы при необходимости добавлять к коду символа **+20h**, чтобы получать маленькую букву. Заодно и прояснилась ситуация, почему у символа маленького «ъ» код **7Fh** – потому что заглавный «Ъ» здесь – это символ «_» (подчерк) с кодом **5Fh**, отсюда **7Fh = 5Fh+20h**.

Возможно, у кого-то в этот момент возникнет вопрос, а почему бы, собственно, не давать всегда сначала отработать оригинальному клавиатурному драйверу, а уже затем подменять только код нового символа, по адресу хвоста? Проблема здесь в том, что при вызове оригинального обработчика тот, в свою очередь, завершая свою работу, посылает контроллеру прерываний сигнал о том, что прерывание отработало, и дальнейшее выполнение кода идёт уже не в монопольном режиме. Из-за этого «Журналист» успевае вытащить клавишу из клавиатурного буфера через функцию прерывания **16h**, и когда управление наконец-то возвращается назад к коду нашего перехватчика, то буфер уже пуст, а буква напечатана.

И осталась ещё пара важных вещей, достойных упоминания. Одна из них – это байт по адресу памяти **0040:0017** – в нём как раз хранится статус клавиш **Ctrl**, **Shift** и **Alt**, но и, важное для нас, статус **Scroll Lock** – за это отвечает 4-й бит (значение **10h**). И этот же байт нам понадобится, чтобы узнать, какую букву писать – маленькую или заглавную, за это отвечает статус **Shift** (биты 0-й и 1-й – **01h** и **02h**) и **Caps Lock** (6-й бит – **40h**). Важно помнить, что **Caps Lock**, в отличие от **Shift**, действует только на буквы, а, значит, все клавиши, которые печатают на русской раскладке цифры и символы, не должны быть затронуты в этом режиме. Для этого достаточно проверить, что скан-код в диапазоне от **10h** до **34h** включительно ⁵.



И здесь стоит вернуться к упоминанию отличия маленьких букв от заглавных – всего

⁵ В этот диапазон также попадают клавиша тильда (слева от цифры «1») и клавиша «\» (скан-коды 29h и 2Bh соответственно), но в нашей таблице их символы в верхнем и нижнем регистре совпадают, поэтому в данном случае этим можно пренебречь, однако написание полноценного клавиатурного драйвера, безусловно, потребует более строгого подхода.



И, наконец, последняя важная вещь – мы должны отдавать оригинальному клавиатурному драйверу не только те клавиши, которые нас не интересуют, но и расширенные клавиши. Расширенные клавиши состоят из последовательности двух нажатых клавиш, где первая всегда **0E0h**. Таким образом, если код первой считанной клавиши **0E0h** – то необходимо передать эту клавишу, а также следующую за ней оригинальному драйверу.

А вот теперь можно, собрав всю полученную информацию воедино, написать свой клавиатурный драйвер.

```
; FASM source code
; http://flatassembler.net/
;
; newskeyb.asm
; (c) SysTools 2019
; http://systools.losthost.org/
@head:
org 100h
; get interrupt vector 009h (keyboard)
mov ax, 03509h
int 021h
; save original vector
mov [@vect9addr+1], bx
mov [@vect9addr+3], es
; set new interrupt vector
mov dx, @vect9hook
mov ax, 02509h ; keyboard
int 021h
; move stack down before shrinking
mov bx, @tail
mov sp, bx
; save stack registers
mov [_sp], sp
mov [_ss], ss
; fill segments for @execparm
mov [_tcmd + 2], ds
mov [_fcb1 + 2], ds
mov [_fcb2 + 2], ds
; adjust memory block
mov ax, cs
mov es, ax
; bx - size in 16 bytes paragraphs
mov bx, ((SIZE + 15) / 16)
```

```
mov ah, 04Ah
int 021h
; execute program
mov dx, _filename
mov bx, @execparm
mov ax, 04B00h
int 021h
; restore stack registers
cli
mov sp, [cs:_sp]
mov ss, [cs:_ss]
sti
; restore interrupt
mov dx, [cs:@vect9addr+1]
mov ds, [cs:@vect9addr+3]
mov ax, 02509h ; keyboard
int 021h
; exit program
mov ax, 04C00h
int 021h

; hook procedure
align 2
@vect9hook:
; save registers in use
push ax
push bx
push es
push ds
push di
; data segment
mov ax, cs
mov ds, ax
; BIOS segment
mov ax, 040h
mov es, ax
; first of all check that Scroll Lock is on
test byte [es:00017h], 010h
jz @stop
; read the key
in al, 060h
; last key was extended - skip current
; one (E0 xx)
cmp [_flag], 0
mov [_flag], 0
jnz @stop
; extended key - skip it
cmp al, 0E0h
jnz @f
```



```

mov  [_flag], 1
jmp  @stop
@@:
; check that key in range
cmp  al, 035h
ja   @stop
xor  ah, ah
mov  di, ax
; check if that key should be mapped
cmp  [@key_lower+di], ah
jz   @stop
; check keyboard buffer head/tail
mov  bx, [es:0001Ch] ; tail
; 1A - head
; 1C - tail
; 1E - buff
; calculate next position
sub  bl, 01Eh
add  bl, 002h
and  bl, 01Fh
add  bl, 01Eh
cmp  bx, [es:0001Ah] ; head
; buffer not full (next(tail) == head)
jnz  @work
; keep all jumps short - move @stop code here
@stop:
; restore registers
pop  di
pop  ds
pop  es
pop  bx
pop  ax
; jump to original interrupt
@vect9addr:
; if for some reason vector was not
; saved here - perform a cold reboot
jmp  far 0F000h:0FFF0h
@work:
; at lest one Shift pressed
xor  ah, ah
test byte [es:0017h], 003h
jz   @f
xor  ah, 1
@@:
; Caps Lock only for letters
cmp  al, 010h
jb   @f
cmp  al, 034h
ja   @f

```

```

; Caps Lock enabled
test byte [es:0017h], 040h
jz   @f
xor  ah, 1
@@:
; lower or upper case
test ah, ah
mov  ah, [@key_lower+di]
jz   @f
mov  ah, [@key_upper+di]
@@:
; al - char; ah - code
xchg al, ah
; move buffer tail
xchg [es:0001Ch], bx
; put key to the buffer
mov  [es:bx], ax
; interrupt cleanup
in   al, 061h ; get value of keyboard
; control lines
mov  ah, al ; save it
or   al, 080h ; set the "enable kbd" bit
out  061h, al ; and write it out
; the control port
xchg ah, al ; fetch the original
; control port value
out  061h, al ; and write it back
; send End-Of-Interrupt signal to the 8259
; Interrupt Controller
mov  al, 020h
out  020h, al
; restore registers
pop  di
pop  ds
pop  es
pop  bx
pop  ax
; interrupt return
iret

; static data here
align 2
; scan codes from [00h..035h]
@key_lower:
db  000h, 000h, 031h, 032h, 033h, 034h, 035h, 036h
db  037h, 038h, 039h, 030h, 02Dh, 03Dh, 000h, 000h
db  06Ah, 063h, 075h, 06Bh, 065h, 06Eh, 067h, 07Bh
db  07Dh, 07Ah, 068h, 07Fh, 000h, 000h, 066h, 079h

```



```
db 077h, 061h, 070h, 072h, 06Fh, 06Ch, 064h, 076h
db 07Ch, 027h, 000h, 02Fh, 071h, 07Eh, 073h, 06Dh
db 069h, 074h, 078h, 062h, 060h, 02Eh
```

```
@key_upper:
```

```
db 000h, 000h, 021h, 022h, 023h, 03Bh, 025h, 03Ah
db 03Fh, 02Ah, 028h, 029h, 02Dh, 02Bh, 000h, 000h
db 04Ah, 043h, 055h, 04Bh, 045h, 04Eh, 047h, 05Bh
db 05Dh, 05Ah, 048h, 05Fh, 000h, 000h, 046h, 059h
db 057h, 041h, 050h, 052h, 04Fh, 04Ch, 044h, 056h
db 05Ch, 027h, 000h, 02Fh, 051h, 05Eh, 053h, 04Dh
db 049h, 054h, 058h, 042h, 040h, 02Ch
```

```
; padding_filename to 12 characters
; (8.3) format
; so anyone can replace 'NEWS.EXE' to the
; program name they might want without
; recompiling source
_filename db 'NEWS.EXE',0,0,0,0,0
_flag db 0
@execparm:
_eseg dw 0
_tcmd dw 080h, 0
_fcb1 dw 05Ch, 0
_fcb2 dw 06Ch, 0
; dynamic data here
_sp rw 1
_ss rw 1
; stack size: 32 bytes (should be enough)
_stack rb 32
align 16
; size of whole program (PSP + code +
stack)
@tail:
SIZE = ($ - @head)
```

Комментарии к листингу программы

Основная программа состоит из следующих частей:

- чтение и сохранение адреса оригинального клавиатурного прерывания;
- уменьшение размера стека (при запуске com-программы под стек отводится много места, но так как нам столько не нужно, к тому же необходимо освободить память для дочернего процесса, то уменьшаем его размер);

- сохранение регистров стека (значения всех регистров будут произвольные после возврата управления к нашей программе назад);
- заполнение сегментов в структуре для запуска дочернего процесса;
- уменьшение размера памяти, выделенного нашей программе (уменьшать размер стека нужно обязательно перед этой операцией);
- вызов дочернего процесса – **NEWS.EXE**;
- восстановление регистров стека после возврата в программу;
- восстановление оригинального клавиатурного прерывания;
- выход из программы.

Подпрограмма клавиатурного драйвера состоит из следующих частей:

- сохранение на стек используемых регистров;
- инициализация регистров для работы (текущий сегмент данных и сегмент **0040h** для чтения значений из памяти);
- проверка, включён ли **Scroll Lock** (здесь и далее, если требуемое условие не выполнено, то восстанавливаем регистры и передаём управление оригинальному драйверу);
- чтение скан-кода клавиши из порта ввода-вывода и пропуск, если предыдущая клавиша была расширенной (**0E0h**);
- если текущая клавиша расширенная – пропускаем её;
- проверка, что скан-код текущей клавиши не более **035h** (когда клавишу отпускают, то выставляется старший бит **80h** – например, при нажатии на «**g**» код нажатия будет **22h**, а отпускания – **22h+80h = A2h** – заведомо больше **035h**, поэтому данная проверка отсекает и ситуацию, когда клавиатурное прерывание **09h** вызывается при отпускании клавиш);
- проверка по массиву, что новый код не ноль – т.е. эту клавишу нужно заменять;
- последняя проверка: берём адрес хвоста, затем вычисляем следующую позицию, и если она равна адресу головы, значит, буфер полон –



возвращаем управление оригинальному клавиатурному драйверу (пусть поест за нас);

- проверка верхнего регистра: хотя бы один **Shift** нажат, а также **Caps Lock**, но только для выбранного диапазона от **10h** до **34h** (**Caps Lock** при нажатом **Shift** отменяют друг друга – буквы снова будут маленькими);

- выбор символа из таблицы для нижнего или верхнего регистра и помещение в клавиатурный буфер по адресу хвоста, а также замена адреса хвоста на новый;

- завершение работы прерывания как у оригинального клавиатурного драйвера: посылаем сигнал клавиатуре, что клавиша обработана, а также сообщаем контроллеру прерываний, что прерывание завершено;

- восстановление регистров и завершение работы прерывания.

P.S. Так как в основной статье нет подходящего места, то интересные факты, связанные с «Журналистом», будут приводиться в постскрипуме.

В Интернете с некоторым трудом в том или ином виде удалось найти 4 версии NewsMaster (русская здесь не учитывается), но, возможно, их больше:

NewsMaster 1.0 (1986)

- изначальная версия программы.

NewsMaster 1.5 (1986)

- изменён формат библиотеки шрифтов **NEWS.FLB**;

- добавлены новые шрифты;

- добавлены новые библиотеки клипарт.

NewsMaster II (1988)

- драйвера для принтеров обновлены и упакованы в файл **PRINTERS.PRG**;

- поставляется с утилитой **CAPTURE.EXE** для захвата изображений в других программах;

- поддержка изображений в формате **CAPTURE.SHP** (PrintMaster capture format);

- в текстовый файл с настройками **NEWS.CFG** вынесено большее количество настроек, которые теперь можно изменять;

- добавлены новые библиотеки клипарт;

- небольшие изменения шрифтов.

NewsMaster II (1988)

- добавлены новые драйвера для принтеров.

Исходные коды из этой статьи вместе с готовыми скомпилированными программами доступны по ссылке:

http://systools.losthost.org/files/news_prg.zip

Специально для Downgrade N27'2019

© SysTools 2019

<http://systools.losthost.org/>





ЗНАКОМСТВО СО SWEET 16

8-битный процессор MOS 6502 был разработан в 1975 году (https://ru.wikipedia.org/wiki/MOS_Technology_6502). Его появление и дешёвизна способствовали выпуску новых компьютеров. В 1977 году появился оригинальный Apple II, который намного отличался от предыдущего Apple I. У Apple II были различные видеорежимы, встроенный монитор и интерпретатор Integer Basic с целочисленными операциями, без чисел с плавающей точкой. Уже позже Стивен Возняк разработал другую версию Бейсика – Applesoft Basic. Integer Basic был безжалостно выпилен, но стал доступен на версии Apple DOS 3.3, после загрузки диска была возможность переключаться между разными версиями Бейсика, набрав **INT** или **FP**.

Кроме тонкостей и нудных нюансов, я узнал о SWEET 16 – виртуальной 16-битной машине (для удобства назову SW16), которая была включена Возняком в состав пакета Integer Basic.

(Прим. ред.: фактически, SWEET 16 – это встроенный в Integer Basic виртуальный процессор, умеющий выполнять операции с 16-разрядными числами, в отличие от 8-битного «реального» MOS 6502.)

Попытку знакомства со SWEET 16 можно смело назвать провалом – масса неточных описаний и исходные тексты, опубликованные с ошибками, не принесли никакой внятной информации. Ещё один эпизод толкнул меня на изучение информации – это публикация виртуального ассемблера и машины. Надо сказать, что ассемблер содержал ошибки, на что я указал в публице какой-то соцсети. Забавно: автор объяснился тем, что мало знает о SW16, а баг нашёл тот, кто знает мизер информации.

Поиск информации мало что дал, кроме упоминания о текстовом редакторе TED, который был написан с использованием SW16. Нашлись несколько ассемблеров:

BIG/MAC;
Merlin;
Lisa;
SC Assembler.

Зачем нужна такая загадочная машина? Иногда программисты, пытаясь познакомиться с ассемблером 6502, терпят крах. Причина проста – нужно 8-битное мышление, перейти к которому непросто.

Процессор 6502 содержит пять 8-битных регистров: **A** (аккумулятор), **X,Y** (индексные регистры), **P** (флаги результата операции), **S** (указатель стека).

16-битное слово можно разбить на две 8-битных части: **LSB** (Least Significant Byte) и **MSB** (Most Significant Byte). Вычисляются они так:

MSB=int(Word/256)
LSB=Word-256*MSB

Для значения **\$1234** получатся **MSB=\$12**, **LSB=\$34**, а в первой ячейке памяти хранится LSB, в следующей MSB.

Этот принцип используется для адресации – способа доступа к ячейкам памяти. Для 6502 существует два особенных участка памяти: нулевая страница (**\$00-\$FF**) и стек (**\$100-\$1FF**). Стек ограничен размерами, так как регистр **S** тоже 8-битный.

Нулевая страница предназначена для хранения любой информации, в том числе для обращения к адресам. Зачем это нужно? Представьте, что основной код расположен в ROM, и никакие трюки с модификацией кода не заработают.

Теперь о способах адресации. Я перечислю часть по [справочнику](#), запоминать эти режимы не нужно, важнее знать принцип обращения к памяти. Для удобства ячейки обозначены как **zr** – адрес **\$00-\$FF**, а пример написан на Бейсике, чтобы не смешивать с Z80.

Здесь перечислены только загрузки значения в аккумулятор **A**, но в некоторых случаях способы адресации можно применять и для записи значения в память.



Zeropage	Абсолютная на нулевой странице (для адресов 00H-OFFH)	LDA zp ; A=PEEK(zp)
Zeropage, x Zeropage, y	Абсолютная индексная на нулевой странице	LDA zp, X A=PEEK(zp+X)
Indirect, x	Индексно-косвенная на нулевой странице. Адрес операнда складывается из байта, следующего за кодом команды и содержимого регистра X. Важно! Старший байт суммы отбрасывается!	LDA (zp, X) A=PEEK(zp+X)+256*PEEK(zp+X+1)
Indirect, y	Косвенно-индексная. Адрес операнда складывается из двух байт в нулевой странице по адресу, указанному вторым байтом команды, и регистра Y	LDA (zp), Y A=PEEK(PEEK(zp))+256*PEEK(zp+1)+Y)

Кроме адресации, следует упомянуть некоторые операции, которые можно применить к 16 бит (**NUML** и **NUMH** – любые ячейки из нулевой страницы, **NUMH=NUML+1**):

Увеличение на 1:

```
INC NUML
BNE LABEL
INC NUMH
```

LABEL

Уменьшение на 1:

```
LDA NUML
BNE LABEL
DEC NUMH
LABEL DEC NUML
```

Сложение двух 16-битных значений:

```
num1lo = $62
num1hi = $63
num2lo = $64
num2hi = $65
resultlo = $66
resulthi = $67
```

```
add clc ;очистка флага Carry
lda num1lo
adc num2lo
sta reslo ; LSB суммы записан в память
```

```
lda num1hi
adc num2hi ;сложение MSB с учетом Carry
sta reshi ;запись в память
```

Вычитание двух 16-битных значений:

```
sub sec
lda num1lo
sbc num2lo
sta reslo
lda num1hi
sbc num2hi
sta reshi
```

Теперь о SW16. Виртуальная машина содержит 16 16-битных регистров:

- RO** – аккумулятор;
- R12** – указатель стека для вызова процедур;
- R13** – хранит результат операций;
- R14** – регистр статуса;
- R15** – Program Counter.

Для виртуальной машины предназначены байткоды, назначение которых описано так (мнемоники в разной литературе и в ассемблерах разные, поэтому удобнее ориентироваться на байткоды):

- 00 RTN Вернуться к режиму 6502
- 01 ea BR addr Смещение, переход по адресу
- 02 ea BNC addr Переход, если флаг переноса равен 0
- 03 ea BC addr Переход, если флаг переноса равен 1
- 04 ea BP addr Переход, если положительный результат
- 05 ea BM addr Переход, если отрицательный результат
- 06 ea BZ addr Переход, если результат равен 0
- 07 ea BNZ addr Переход, если результат не равен 0
- 08 ea BM1 addr Переход, если отрицательный результат
- 09 ea BNM1 addr Переход, если положительный результат
- 0A BK Выполнение опкода 6502 BRK - переход в монитора
- 0B RS возврат из процедуры SWEET 16
- 0C ea BS addr Вызов процедуры SWEET 16
- 0D,0E,0F Нет операции



О следующих байткодах: здесь применено обозначение вида **1R**, где **R** определяет номер регистра (**0-15**), так получается, что **1R** соответствует **\$10-\$1F**.

- 1R const SET Rn,value Rn=value
- 2R LD n R0=Rn
- 3R ST n Rn=R0
- 4R LBI r Загрузка байта в R0L (мл. байт) из адреса Rr, инкремент регистра Rr. R0H (ст. байт)=0. Описывается как R0=@(Rr),Rr=Rr+1
- 5R SBI r Запись R0L по адресу Rr, инкремент Rr или MA = (Rn), MA <-- (R0L),Rn <-- MA+1
- 6R LDD @Rn Загрузка регистра R0 из адреса, хранящегося в Rr, инкремент Rr на 2 или MA = (Rn), R0 <-- (MA, MA+1),Rn <-- MA+2.
- 7R STD @Rn Запись регистра R0 по адресу, хранящемуся в Rr, инкремент Rr на 2 или MA = (Rn), MA,MA+1 <-- (R0),Rn <-- MA+2.
- 8R POP @Rn Загрузка R0L по адресу, хранящемуся в Rr, декремент Rr, при этом R0H=0 или MA = (Rn)-1, R0L <-- (MA),R0H <-- 0,Rn <-- MA.
- 9R STP @Rn Запись R0L по адресу, хранящемуся в Rr, декремент Rr или MA <-- (Rn)-1, (MA) <-- R0L, Rn <-- MA
- AR ADD n R0=R0 + Rn сложение R0 с регистром Rn
- BR SUB n Сравнение R0 с Rn (вычитание) R0=R0 - Rn)
- CR POPD @n Загрузка R0 по адресу, хранящемуся в Rn, декремент Rn на 2 или MA = (Rn)-2, MA,MA+1 <-- R0, Rn = MA
- DR CPR n Сравнение Rn с R0 или R13=(R0 -(Rn, R14=флаги
- ER INC r Rr=Rr+1 или INR Rn
- FR DEC r Rr=Rr-1 или DCR Rn

Здесь приведён список, который собран из разных описаний, содержащих неточности. Например, написано, что байткоды **\$0B**, **\$0C** приравнены к «нет операции». Понять назначение было сложно, даже чтение исходного текста ещё более запутало.

Помог разобраться этот [документ](#).

С примерами использования было тоже неясно, например:

- \$300 20 89 F6 JSR \$F689 ;начало работы: вызов процедуры из Integer Basic, вслед идут байт-коды
- \$303 11 00 08 SET R1 source address ;R1=начальный адрес
- \$306 12 00 00 SET R2 destination address ;R2=адрес назначения
- \$309 13 00 00 SET R3 length ;R3=длина
- \$30C 41 LD @R1 ;загрузка R0L по адресу, хранящегося в R1, R1=R1+1
- \$30D 52 ST @R2 ;запись R0L по адресу в R2, R2=R2+1
- \$30E F3 DCR R3 ;уменьшение R3 на 1
- \$30F 07 BNZ \$30C ;если результат не нулевой, то переход на адрес \$30C
- \$311 00 RTN ;завершение выполнения
- \$312 60 RTS ;возвращение из подпрограммы для 6502

Какой ассемблер выбрать? У выбранного компьютера Apple IIe код просто отсутствует. Как вариант, можно взять один из перечисленных ассемблеров и использовать готовый код для SW16. Но для того, чтобы использовать ассемблер, понадобится время на чтение документации, что, по опыту, остудит пыл новичка. Есть другой вариант – использовать кросс-средства, с которыми удобно работать.

Я использовал [naken asm](#) – для отладки, и [ACME](#) – настроенный для C64 с макросами. Зачем нужна такая связка? Об этом позже.

Первая задача: написать самому алгоритм с использованием 16 бит. Думать долго было не нужно, готовый уже был – вычисление и построение таблицы N^2.

Сам алгоритм давно известен:

```

;http://www.cpcwiki.eu/index.php/Programming:Precalculated_square
INITSQ LD DE, 1 ;1st odd number
LD HL, 0 ;HL = 1st square number
LD B, H ;counter = 256
    
```



```

LD IX, SQTAB ;startaddress of the
                ;square table
SQLLOOP LD (IX), L ;Lowbyte to table
        INC IX
        LD (IX), H ;Highbyte to table
        INC IX
        ADD HL, DE ;add odd number
        INC DE ;next odd number
        INC DE
        DJNZ SQLLOOP ;256 times
        RET
    
```

Пробный код для ассемблеров выглядел примерно так:

```

ldy #0

jsr sw16
sub r0;set 0,0 ;r0=0
st 6 ;r6=r0
inr 6
rtn

mm
lda 0
sta sqt,y

lda 1
sta sqt+256,y

jsr sw16
add 6 ;r0=r0+r6
inr 6 ;r6+1
inr 6 ;r6+1
rtn

iny
bne mm
    
```

Генерация квадратов разбита на две части: сначала значение **RO** помещается в две таблицы старших и младших байтов. Эти значения берутся из ячеек нулевой страницы с адресами 0 и 1. По идее, можно было бы реализовать запись готовых данных с помощью других регистров, но я не догадался, как так сделать. Запись **RO** в память осуществлена как запись **ROL, ROH**, но нет удобной операции «Обмен байтами» (как у БК-0010).

Хорошо, на первый взгляд таблица получена, вернусь к АСМЕ и напишу программу снова:

```

sqt=$400
        ldy #0
        jsr SW16
!byte  $B0; sub r0 ;set 0,0 ;r0=0
    
```

```

!byte  $36; st 6 ;r6=r0
!byte  $E6; inr 6
!byte  $0; rtn

mm lda 0
    sta sqt,y

lda 1
    sta sqt+256,y

jsr SW16
!byte  $a6; add 6 ;r0=r0+r6
!byte  $E6; inr 6 ;r6+1
!byte  $E6; inr 6 ;r6+1
!byte  $00; rtn

iny
bne mm
rts
    
```

Для удобства я взял исходный текст из пакета SC Assembler 4.0, адаптировал к синтаксису АСМЕ, так стало удобно использовать кросс-ассемблер. Единственное требование – привязка кода к адресу XX89 из-за специфики кода SW16.

Осталось одно неудобство – долго и нудно писать опкоды в виде байтов. Поэтому были использованы макросы ассемблера, например:

```

;Macros for SW16
R0 = 0 ;reg definitions
R1 = 1
R2 = 2
R3 = 3
R4 = 4
R5 = 5
R6 = 6
R7 = 7
R8 = 8
R9 = 9
R10 = 10
R11 = 11
R12 = 12
R13 = 13
R14 = 14
R15 = 15
;0A 1 Break to Monitor
;0A BK Execute 6502 BRK instruction.
!macro BK {
!byte $0a
}

;0B-0F 1
    
```



```

;0B RS Return from SWEET 16 subroutine.
!macro RS {
!byte $0B
}

;0C ea BS addr Call SWEET 16 subroutine.
!macro BS brad {
!byte $0a,brad-*
}

```

;0D,0E,0F - No Operation

```

;1R 3 R<-2 byte constant
;(load register immediate)
;1n lo hi SET Rn,value Rn=value.
!macro SET .rn { ; also TRA
!byte $10+.rn
}

```

```

;2R 1 ACC<-R
;2n LD n R0 <-- (Rn).
;2r TRA r, R0=Rr
!macro LD .rn { ; also TRA
!byte $20+.rn
}

```

Полный текст в приложении. Теперь текст программы стал выглядеть иначе:

```

MYPROG

sqт=$400
ldy #0
jsr SW16
;!byte $B0; sub r0;set 0,0 ;r0=0
+SUB R0
;!byte $36; st 6 ;r6=r0
+ST R6
;!byte $E6; inr 6
+INR R6
;!byte $0; rtn
+RTN

mm lda 0
sta sqт,y

lda 1
sta sqт+256,y

jsr SW16
;!byte $a6; add 6 ; r0=r0+r6
+ADD R6
;!byte $E6; inr 6 ;r6+1
+INR R6

```

```

;!byte $E6; inr 6 ; r6+1
+INR R6
;!byte $00; rtn
+RTN

```

```

iny
bne mm
rts

```

Вызов макросов в ACME осуществлен через +ИМЯ.

Следующим испытанием стал пример, найденный в документации – очистка экрана.

Loca- tion	Hex value	Opcode	Comments
300	20 89 F6	JSR \$F689	Turns on Sweet 16
303	11 00 04	SET 1, 0400	R1 = address of screen buffer
306	12 00 04	SET 2, 0400	R2 = count of characters
309	13 E0 00	SET 3, 00E0	R3 = ASCII blank
30C	23	TRA 3	R0 = ASCII blank
30D	51	SBI 1	Store blank on screen
30E	F2	DEC 2	R2 = R2-1
30F	07 FC	BNE 030C	Make all of screen blank
311	00	RTN	Return to 6502 mode
312	60	RTS	6502 return from subroutine

Особо сложного ничего нет. Адрес экрана Apple II привязан к \$400, запись R0 осуществлена по адресу, хранящемуся в R1, R2 – счетчик цикла. Хотя, работает процедура очень медленно.

Осталось убедиться в том, что байткоды условных смещений написаны верно, так как описаны они туманно. Для этого пригодился naken.

Переходы стали описываться так:

```

;01 2 Branch Always
;01 ea BR addr Unconditional Branch.
!macro BR brad {
!byte 1,brad-*
}

```



```
;02 2 Branch no Carry
;02 ea BNC addr Branch if Carry=0.
!macro BNC brad {
!byte 2,brad-*
}
```

Остался ещё один непроверенный набор байткодов для процедур, на отладку которого был потрачен целый вечер.

С выбором задачи тоже не возникло проблем – возьмём печать десятичного числа.

```
;16bit to ascii
jsr SW16

+SET R0,0xFFFF
+SET R2,10000
+SET R4,0x400
+SET R7,$B0
+BS sub10
+SET R2,1000
+BS sub10

+SET R2,100
+BS sub10

+SET R2,10
+BS sub10

+ST R3
+BS digi
+RTN

; JSR $FDF0;COUT1
rts

sub10:
+SET R3,-1
s10:
+INR R3
+SUB R2
+BC s10
+ADD R2
digi:
+ST R6
+LD R3
+ADD R7
+SBI R4
+LD R6
+RS
```

Выглядело всё просто и красиво, но программа зависала. Копание в отладчике показало причину: я использовал стек для

хранения, а стек не инициализируется, и в таком случае равен \$FFFF. Запись по такому адресу запрещена, поэтому возврат из процедуры SW16 приведёт в другой адрес. Поэтому был добавлен +SET R12,\$20;init stack (на форуме посоветовали использовать адрес \$110, что, на мой взгляд, небезопасно).

Компиляция программы :

```
set nam=sw
acme %nam%.asm
java -jar AppleCommander-1.3.5.jar -d
master.dsk %nam%
java -jar AppleCommander-1.3.5.jar -p
master.dsk %nam% B 2182 < %nam%.bin
rem del pl.bin
pause
```

Строка 2: компиляция **sw.asm**

Строка 3: удаление с образа диска **master.dsk** файла **sw.bin**.

Строка 4: копирование файла **sw.bin** на диск **master.dsk** с указанным адресом **2182** (тот, что указан **\$0886**).

После компиляции нужно запустить образ диска **master.dsk** в эмуляторе AppleWin и набрать **BRUN SW**

Применимо ли SW16? Для меня нет, так как за время кодирования я уже знал некоторые приёмы работы с 16-битными значениями

Но, как уже сказано, неточности в описании толкнули меня на полное изучение нюансов в использовании готовых процедур.

Полагаю, что начинающий в программировании «вырастет» из использования SW16 и научится использовать алгоритмы в любой задаче.

С одной непростой задачей столкнулся и я, о чём расскажу в следующем номере.

Прим. ред.: упомянутые в статье исходники, бинарники и вспомогательные файлы можно скачать здесь:

<http://dqmag.in/N27/sw16/sw16.zip>





СУБЪЕКТИВНОЕ МНЕНИЕ О ФИДО

В девяностых годах по выходным я ходил в гости к бывшему однокласснику, «на-таскивал» его решать интегралы, а потом мы вместе играли на его IBM-е, рассматривали фото из разных «фэх», читали свежие анекдоты из «эхи»... Воображение поражала фотография стадиона, которую можно было увеличить настолько, что становились хорошо различимы лица людей на противоположных трибунах. Однако интенсивность компьютерной жизни дала себя знать очень быстро, после первой же сессии товарища из универа отчислили, и семейный совет объявил меня пособником и персоной нон грата. Так вот я и познакомился с Фидо.

Продолжение знакомства состоялось только через много лет, когда у меня уже давно был свой (и не первый) компьютер, а необходимость в стационарном телефоне перекрывал кабель с Интернетом. Компьютерщиком я как не был, так и не стал, однако интерес к Фидо сохранился. Полез в Интернет, стал искать информацию, читать всё, что рекомендуется в таких случаях... Продравшись сквозь технические заклинания типа «мейлер» и «тоссер», я выяснил, что есть как минимум один готовый комплект, в который достаточно ввести минимум необходимых данных: **fidoip**. Его я себе на ноутбук и поставил. Было это три года тому назад...

В первую очередь я стал искать признаки жизни Фидо в родном Калининграде. В конце концов, я нашёл координаты Дмитрия¹ (ныне

босс узла 2:6078/1). История Фидо в Калининграде это отдельная тема, в моей же истории получилось так, что к тому моменту, когда он ответил на запрос, я уже чуть не месяц был пойнтот на узле 2:5020/570, читал и даже иногда писал сообщения в эхи. Так началась моя двойная фидо-жизнь, которой и продолжаю жить.

Общее впечатление о Фидо: здесь интересно. Нет рекламы. Нет коммерции. Зачастую весело, потому что не всё обсуждается с самыми серьёзными выражениями лиц. Люди общаются, а не меряются авторитетами и посылают, и самое близкое – смысловое – определение к понятию «эха» – это дружеская беседа.

Малограмотность не распространена, что действительно удивляет. И что касается общения, то это просто приятно и кайфово, когда собеседник не только интересный человек и (или) хороший специалист, но он способен чётко сформулировать свои мысли. За уровнем общения и тематикой следят модераторы, но это не цензура в привычном понимании этого термина. Фидо строится на ряде принципов («полиси», свод правил сети Фидо), и правила каждой эхи так или иначе отталкиваются от «полиси». Конечно же, в Фидо, как и везде, есть свои правдолюбцы и люди близкие к природе, из уважения к ним и специально для их удобства созданы особые эхи, где их никто и ни в чём не ограничивает.

Может потому, что я подписан на не такое уж большое число эх, но именно с самодурством модераторов я не сталкивался ни разу, и за всё это время – три года – наблюдал только один случай, когда человек трудами праведными заслужил свою «награду».

Собственно, боссы и модераторы – это тот костяк, на котором всё держится. Для того чтобы сеть существовала, нужны вполне реальные компьютеры, на которых установлено соответствующее программное обеспечение, и поскольку хорошим тоном теперь является круглосуточный доступ, то и работать эти компьютеры должны круглосуточно. Хорошо,

¹ Который на момент написания этой статьи в Фидо уже около двадцати лет, и примерно десять из них – босс узла.



если есть возможность установить ноду на служебном компьютере, но это возможно не всегда, а в домашних условиях так и вовсе бывает проблематично. Появление новых узлов и существование прежних обусловлено решением (или не решением) вот такой техниче-ско-бытовой проблемы. И уже существует решение: установка ноды на смартфоне. В этом тоже есть своя специфика, но уже сугубо техниче-ского плана (размеры экрана, экранная клавиатура и т.д.).

А для того, чтобы эхи не превращались в помойки, нужно следить за порядком. У каждой эхи есть свои правила (за исключением особо выделенных немодерируемых эх), пра-вила довольно лояльные, на самом деле. И каждая из них требует внимания, а не поверх-ностного отношения.

Нужно очень ценить людей, у которых есть продолжительная мотивация заниматься тех-нической стороной дела, которые вкладывают своё время, внимание, а иногда и деньги, в существование Фидо, которым мы пользуемся.



Фидо-почту и свежие сообщения в эхах я прочитываю вечером, поскольку пакет установ-лен на ноутбуке, а не на телефоне. Иногда за вечер получается перебраться несколькими письмами или сообщениями. Принцип обще-ния в Фидо «сегодня спросил, завтра прочёл ответ» по большей части продолжает действо-вать, и лично меня это вполне устраивает. Существует хорошее и очень верное наблюде-ние: «если вы разговариваете с человеком, и он достаёт сигареты и закуривает, можете раз-ворачиваться и уходить – в беседе с вами этот человек не заинтересован». По моему мнению,

это применимо и к Фидо: если человека не устраивает существующая тут скорость обще-ния, то не собеседник ему нужен, а нечто иное: справочник-на-все-случаи-жизни, развлечение, свежие ушы, ещё что-то... он не заинтересован в людях, к которым обращается. В Интернете, кстати, ответ от нужного специалиста тоже далеко не мгновенно приходит, и зачастую поисковик выдаёт просто хлам и рекламу, вместо нужной информации.

Утверждения что «мы живём в век скоро-стей», «всё непрерывно меняется и нужно этому соответствовать», а также производные от них – не более чем мифы нынешнего перио-да истории. Мы скорее узнаем новости о происходящем в мире, но всё это поверхно-стные явления, мир – по сути – не меняется. Достаточно понять основные тенденции проис-ходящего и быть в курсе тех изменений, которые затрагивают личные интересы – всё остальное – это просто нагнетаемая истерия и информационный шум. Точка и шкала отсчёта для любых изменений, любого развития – сам человек, которого, увы, не облагораживают ни деньги, ни должности, ни суперкрутые техно-логии. Мир «непрерывно меняется» только для тех, кого эти упомянутые истерия и информа-ционный шум полностью поглотили. Им можно посочувствовать, но это не повод следовать их примеру.

Иногда сообщений в эхах за сутки набира-ется довольно много, и уже приходится или откладывать прочтение на потом, или жертво-вать какой-нибудь тематикой – пролистывать не глядя, хотя чаще всего даже в неинтересных лично мне темах высказываются очень даже интересные люди, у которых есть опыт и знания, а не один только «мобильный Интер-нет на все случаи жизни».

На данный момент «население» Фидо от-носительно невелико, не редкость видеть в разных эхах сообщения от одних и тех же людей. Хорошо было бы, чтобы «население» увеличилось, чтобы сеть Фидо не просто про-должала быть, а росла... Но для того, чтобы приходили новые люди, у них должна быть



мотивация, и тут есть проблема, которую поясню на примере: если я хочу посмотреть хороший детективный сериал в авторском переводе, но не хочу искать по торрент-трекерам и скачивать на жёсткий диск, то самое простое – зарегистрироваться в одной из социальных сетей и спокойно смотреть онлайн. То есть, порог вхождения: хочешь приобщиться, выполни ряд несложных действий. Ранее в Фидо таким порогом было умение настроить софт для работы в сети: хочешь читать-писать в Фидо – пожалуйста, ставь программу, получай пойнта и вперёд! Сейчас практически все эти доступны для чтения из сети Интернет, и если человеку интересно узнать, что именно пишу в определённой эхе, то ему нет необходимости становиться пойнтом Фидо. Сверхдоступность убивает мотивацию и интерес. Всё равно что рубим сук, на котором сидим. Не применяя кардинальных мер, эту ситуацию можно было выправить следующим образом: пускай от каждого сообщения, транслируемого таким образом в Интернет, будут доступны только первые 3-4 строчки. Для общего впечатления от сети этого достаточно, а если человеку действительно интересно – добро пожаловать!

Фидо – это сеть друзей, ей изначально присуща уникальная атмосфера, которая существует по-прежнему. Мессенджеры и социальные сети безлики, и попытки сращивания с ними Фидо, «дотягивание» её до «стандартов» ни к чему хорошему не приведут. В лучшем случае появится ещё один – среди множества – ничем не примечательный мессенджер, который придётся обогащать «фишками», чтобы он пользовался спросом хоть у кого-нибудь. Либо ещё одна, но уже «недосоциальная» сеть, которая растворится в уже существующей безликости. В уже существующих программах для работы в Фидо есть всё, что действительно необходимо для общения и работы в сети друзей, в некоторых случаях их стоит дорабатывать под современное «железо», незначительно менять интерфейс, но не нужно менять принципы, лежащие в основе сети,

потому что именно они и делают Фидо тем, чем она является.

Фидо живёт только за счёт энтузиазма, тут нет тупых репостов одного сообщения сотнями «пользователей», которые тем самым «наполняют свою страницу», нет «котиков», «лайков» и прочего. Хочется, чтобы участники от вялого созерцания – что не редкость, к сожалению – перешли к активной позиции. Как минимум часть тематических – и очень хороших – эх фактически мертвы не потому, что «люди ушли в Интернет», а потому что оставшиеся занимают пассивную позицию наблюдателей – и ничего не происходит, правда, удивительно?

Фидо живо? Да, вполне. Есть ли у него будущее? Не гарантированно, но выше я попытался отметить основные моменты, от которых это зависит. Что же касается массовости, то Фидо функционирует не за счёт коммерции, где каждый пользователь – это потенциальный покупатель или товар, Фидо – это реализация принципов, разработанных её создателями. На данный момент эти принципы в нашей цивилизации не являются определяющими. Соответственно и притягиваться к сети Фидо будут в основном только те люди, у которых принципы Фидо будут находить отклик. Толпы народа не ринутся сюда никогда, но нужно продолжать заниматься популяризацией сети, чтобы сюда пришли те, кому это будет по душе. И нужно занимать активную позицию «в этом нашем Фидо», а не просто наблюдать, потому что принципы принципами, но люди идут на людей, увлеченных идеей, а не на голую теорию и пустые эхи. Только так, через работу вот такого своеобразного магнита, Фидо может обрести новую жизнь, и в сеть будут приходиться новые и интересные люди.

На этом у меня всё. Пожелания и замечания можно присылать на указанный адрес.

Андрей Тумилевич
2:5020/570.10@Fidonet



10 апреля

1985. Встреча сисопов в городе Сент-Луис. На ней было решено разделить Fidonet на сети и регионы.

1989. Raymond Lowe (3:700/725, Гонконг) анонсировал программу Daisy – фидонет-совместимую Apple CP/M BBS.

1992. В нодлисте появилась сеть 2:5090 (Красноярск).

1997. Леонид Каганов публикует «сенсационную новость»: «Некоторые особенности образования виртуалов в Фидо на примере Алекса Экслера». После этого фраза «пять Экслеров» стала фидошным мемом.

1998. После трёх лет отсутствия международного координатора новым IC выбран Bob Satti из Канады.

12 апреля

1996. В нодлисте появилась сеть 2:4652 (Макеевка, Украина).

2002. В нодлисте появилась сеть 2:6055 (Волжский, Волгоградская область).

15 апреля

1994. В нодлисте появилась сеть 2:5079 (Сургут).

2002. Немецкая «WIF team» (WIF расширяется как «What is Fido?») анонсировала второй выпуск «Fido CD-ROM Second Edition». Первый выпуск сборника состоялся годом раньше. Компакт-диск был выпущен тиражом до 600 штук и распространялся бесплатно на автозаправках, в компьютерных магазинах и т.п. На диске содержалась общая информация о Фидо (на немецком языке) и большой сборник программного обеспечения для Fidonet.

16 апреля

1993. В нодлисте появилась сеть 2:451 (Гродно, Беларусь).

2016. На состоявшемся в Москве фестивале ASWDF с рассказом о HotdogEd (фидософт под Android) выступил Сергей Позитурин.

17 апреля

1992. В нодлисте появилась сеть 2:467 (Одесса, Украина).

18 апреля

1988. Daniel Tobias (Шривпорт, Луизиана) анонсировал программу GATEWAY для гейтования электронной почты между системами Fido/Opus/compatible и другими системами, в том числе E-Mail.

2006. Международный координатор дал официальный ответ по поводу выделения регионов exUSSR в седьмую зону. В нём были поставлены такие трудновыполнимые условия, что эту дату можно смело считать похоронами идеи Z7.

19 апреля

1996. Алекс Сотник получил узел в Германии, который стал там первой русскоязычной нодой. В этом же месяце создана эхоконференция GERMAN.RUS, первоначально предназначенная для общения русскоговорящих фидошников Германии.

2005. Стас Дегтев выложил снимок редактора GoldEd+ версии 1.1.5.

20 апреля

1985. Состоялась сисопка в Сент-Луисе, США. На ней были принято решение о серьёзной реорганизации и децентрализации Фидо, необходимость в которой была вызвана слишком быстрым ростом Fidonet. В результате в фидошной адресации появились сети, регионы и координаторы. Том Дженнингс объявляет о скором выпуске программы Fido v.10N, где уже будет учтено новое устройство сети, а также снято ограничение в 250 узлов.

1987. В редакционной статье бюллетеня Фидоньюс поднят вопрос о создании межзональных гейтов (ZoneGate). Сообщено, что мейлер SEAdog v.4.0 и транслятор нодлиста XlatList v.2.70 уже умеют обрабатывать сообщения с межзональными адресами.

1987. Опубликован список кандидатов на места в Совете Директоров IFNA (International FidoNet Association). Всего было 23 кандидата. Голоса принимались по почте (бумажной) или лично на четвёртой Международной Конференции Fidonet в августе 1987 г.



2013. Толик Вдовичев анонсировал приложение-фидопакект AfterShock для Android.

21 апреля

1999. Первый релиз файломенеджера Husky Project Htick v.0.01.

2013. Макс Лушников обновил движок wFido, добавив, в частности, возможность читать через интернет эхи без регистрации.

22 апреля

1989. Опубликована финальная версия проекта эхополиси первой зоны (General Echomail Policy 1.0). Он был принят в октябре 1990.

2002. Bjorn Felten (Гётеборг, Швеция) в редакционной статье FidoNews рассказывает о том, как настроить показ русских символов в эхоконференциях Фидо.

23 апреля

1993. В нодлисте появился регион 2:479 (Латвия) и сеть 2:4790 (Рига). Позже, с 15 октября 2003, латвийский регион получил нумерацию 2:51, а рижская сеть — 2:5100.

2010. Леонид Каганов получает узел 2:5020/313 и делится идеей по созданию веб-ноды.

24 апреля

2000. Образован проект мейлера Radius, основанный на исходных текстах Argus (by RIT labs).

25 апреля

1994. Бюллетень FidoNews получил Международный стандартный серийный номер ISSN 1198-4589. Редактором на тот момент была Sylvia Maxwell. В 2001 году этот ISSN был признан недействительным.

26 апреля

1993. Benju Arnold (2:254/1.3) публикует в FidoNews инструкцию «Как писать из Фидо в интернет и обратно».

1996. В нодлисте появилась сеть 2:5008 (Улан-Удэ).

2013. Иван Агарков выложил свои исходники php-гейта, со стороны интернета выглядящего как форум на vBulletin.

28 апреля

1989. Начало конференции GATEWAY'89 в Куинсе, Нью-Йорк.

29 апреля

1988. В связи с тем, что программа-сборщик нодлиста MAKENL достигла предела в размере diff'a, из нодлиста в срочном порядке удалены «технические» и приватные ноды; выдача новых узлов приостановлена. Координаторам зон рекомендовано в случае необходимости выпускать зональные нодлисты вместо мировых. Проблема была решена через месяц, с выходом MakeNL v.2.11.

30 апреля

1984. Tom Jennings публикует «Некоторые предварительные идеи для FidoNet» («Some preliminary ideas for the FidoNet»).

2004. Закрылся крупный хаб и гейт «InfoScience», 2:5020/238.

МАЙ

1 мая

1987. Первый поинткомплект за пределами США вышел в Нидерландах. Henk Wevers выпустил программу DUTCHIE, которая позволяла работать как с BBS, так и с нетмейлом и эхами. DUTCHIE распространялась бесплатно для некоммерческого использования, Хэнк лишь просил прислать ему открытку.

1992. В составе региона 2:47 (Литва) появилась сеть 2:478 (Калининград). С августа 2009 года сеть перешла в R50 (Россия) под номером 2:6078.

4 мая

2016. Дмитрий Каменский (Калуга) открыл параллельный доступ к wfido.ru через WEB и FTN.



5 мая

1989. Принят проект FidoNet Policy v.4.06.

2005. Андрей Елкин (Andy Elkin, 2:5030/15) анонсирует свой новый проект мейлера T-Next под Win32 (бета-релиз v.0.44).

7 мая

1997. Второй рейд итальянской полиции против сисопов BBS и Fido (первый Crackdown был тремя годами раньше). В этот раз поводом послужило подозрение в распространении детской порнографии. Конфисковано оборудование у трёх узлов Фидо региона 33. Координатор Z2 (Европа) Ward Dossche пишет открытое письмо в защиту фидошников послу Италии и итальянскому делегату Европейской Комиссии в Брюсселе.

9 мая

1992. В R2:20 (Швеция) при участии Z2C Ron Dwight состоялась встреча хабов и хостов региона.

11 мая

1994. В Италии прошла спецоперация силовых структур против сисопов BBS и FidoNet, впоследствии получившая название «Crackdown». Это был рейд против хакеров и распространителей вареца, в основном взломанных игр. Под раздачу попали 122 сисопа и держателя BBS. Были проведены обыски, конфискации, возбуждены уголовные дела. Однако доказать причастность фидошников к незаконной деятельности итальянская прокуратура не смогла. Те же BBS, где реально распространялся варез, вовремя уехали в другие страны. Тем не менее, после этого случая Фидо в Италии значительно поредело – многие были напуганы размахом репрессий.

12 мая

1987. Появление первых узлов Фидо в Латинской Америке.

1989. Для Латинской Америки в нодлисте появилась отдельная зона 4.

1995. В нодлисте появилась сеть 2:5031 (Мурманск). Ранее этот же номер принадлежал Архангельской сети.

13 мая

1993. Steve Mulligan (Непин, Онтарио, Канада) выпустил VoteFix v0.04b – программу для голосования через нетмейл и эхомейл.

14 мая

2003. На бэкбон R50 принята эхоконференция SU.TORMOZ.

2012. Юрий Лукьянец опубликовал исходники vFido – вКонтактовского приложения для доступа к фидо-конференциям и нетмейлу.

2014. Принята на бэкбон R50 эхоконференция STARPER.LIMITED.

15 мая

1988. Chris Irwin (Майами, Флорида) выпустил быстро ставший популярным мейлер D'Bridge. Первоначально программа продавалась за \$79.

2013. Сергей Блохин выпустил альфа-версию Panda – установщик фидософта под MacOS X.

2017. Michiel van der Vlist (Дриберген, Нидерланды) сообщает об обнадёживающих результатах теста протокола BinkP с использованием технологии Dual-Stack Lite (IPv6).

16 мая

1986. Состоялась сисопка в Цинциннати (Огайо, США) – «Cincinnati Area Fido Sysop Meeting». Присутствовало 16 сисопов из трёх городов штата.

2011. Michiel van der Vlist (Дриберген, Нидерланды) анонсировал созданную им англоязычную эху IPV6.

17 мая

1996. В нодлисте появилась сеть 2:5029 (Рязань).

1996. В нодлисте появилась сеть 2:5038 (Петрозаводск).

18 мая

2001. После двух лет безвластия в зоне 6 (Азия) появился зональный координатор. Им стал Bin Li из Шанхая.



19 мая

1989. Начало EuroCon III (третьей европейской конференции Fidonet) в Эйндховене, Нидерланды. Основная тема – «Демократия и Европейский Fidonet». Henk Wevers подал в отставку с поста Z2C, новым координатором выбран Ron Dwight. Принято решение о выборности всей вертикали координаторов от сетевых и до зонального. Сделаны шаги в попытке организовать EFNO (European FidoNet Organisation). Было предложено вести «членские взносы» с каждой сети в размере 4 ЭКЮ, умноженных на число узлов (как это уже было сделано в Германии). Сисопы высказались за отмену действия Policy 3 и против проекта Policy 4 для узлов Z2, предложив продолжать использовать проект Полиси Европейской версии (POLICY4E). В первой зоне (Северная Америка) некоторыми это было расценено как «Декларация независимости». Впрочем, Том Дженнингс призвал к спокойствию, заявив, что «Европа – это не наше собачье дело, а Z1 – это не полицейские силы мира».

1995. Сбор подписчиков локальной эхи YOUNG.PIONEERS самарского узла 2:5057/19 «Pioneer's Palace» (сисоп-основатель – Дмитрий Ермолаев). Начало самарской фидошной традиции «Пионеровка».

2000. В нодлисте появилась сеть 2:6023 (Обнинск).

20 мая

1985. Сообщено о выпуске Томом Дженнингсом программы Fido v.10i (предыдущая версия 10h была протестирована, но не получила широкого распространения). В ней появилась поддержка региональной и сетевой адресации в Фидо.

1991. Объявлены результаты голосования по проекту WorlPol. Большинство голосов (48,7% «против», 39,5% «за», 11,8% «воздержались») принятие WorlPol вместо Policy 4.07 было отклонено. Всего проголосовало 195 сисопов из 424 имеющих право голоса.

21 мая

1993. В нодлисте появилась сеть 2:4615 (Луганск, Украина).

22 мая

1992. В нодлисте появилась сеть 2:5045 (Владивосток).

1998. В нодлисте появилась сеть 2:5093 (Набережные Челны).

24 мая

1987. Jean Corrola представил утилиту FILELIST для обслуживания файловых запросов.

1991. Регион 2:50, именованный ранее как «Siberia» и «The_Court_Of_The_Crimson_King», теперь в нодлисте числится как «Russia».

1991. В нодлисте появилась сеть 2:5030 (Ленинград).

1991. В нодлисте появился регион 2:47 (Литва) и сеть 2:470 (Каунас).

1991. В нодлисте в составе региона 2:49 (Эстония) создана сеть 2:495 (Рига, Латвия).

26 мая

1984. John Madill публикует «Рекомендации по FidoNet» («Considerations for FidoNet»).

27 мая

1985. В Новом Орлеане начинается симпозиум DECUS, на котором с полуторачасовой лекцией о Фидо выступает Kurt Reisler (1:109/74).

1991. Vince Perriello объявляет об уходе с должности редактора FidoNews. Следующий номер выйдет уже под редакцией Tom Jennings и Tim Pozar.

1991. Jamie Penner анонсировал GateWorks – систему для гейтования эхомэйла в различные сети.

2002. Bob Short (1:105/38) сообщает о создании новой эхоконференции FIDO_SYSOP. Эха жива до сих пор.

2008. Павел Гульчук поднял у себя сервис NodeHist (история узлов в нодлистах): <http://nodehist.fidonet.org.ua>.

28 мая

1990. Сообщено, что George Pease избран новым Z1C взамен подавшего в отставку Steve Bonine.



30 мая

1988. Edward Hawes (1:106/132 – Кейти, Техас) анонсировал шахматную программу ChessLine для BBS. Игрок подключался к BBS, выбирал опцию ChessLine, делал ход и отключался. Потом то же делал другой игрок, и так далее. Программа сохраняла все ходы, а в конце партии показывала на экране шахматную доску с финальной расстановкой фигур. Впоследствии при участии James Brown программа была адаптирована для игры через NetMail, а также для онлайн-режима (при этом можно было одновременно чатиться с сисопом).

1997. В нодлисте появилась сеть 2:5042 (Благовещенск).

1997. В нодлисте появилась сеть 2:5073 (Качканар, Свердловская область).

1997. В нодлисте появилась сеть 2:5076 (Надым).

1997. В нодлисте появилась сеть 2:4655 (Енакиево, Украина).

2010. Дмитрий Игнатов предложил новый логотип Fidonet, в котором, помимо прочего, пятидюймовая дискета заменена на компакт-диск.

31 мая

2011. Владислав Коротнев собрал фидо-пакет, состоящий из binkd, crashmail и GoldED для мини-ноутбука Ben NanoNote.

ИЮНЬ**1 июня**

1984. В июне 1984 Том Дженнингс выпустил релиз первой версии FidoNet software.

1992. В FidoNews объявлено о том, что сети 2:5000 и 2:343 (Новосибирск и Барселона) стали побратимами. Об этом совместный манифест опубликовали Tirs Abril и Влади-мир Лебедев (2:343/106 и 2:5000/20 соответственно).

1997. Начало первой Международной недели BBS (International BBS Week).

1998. Приняты правила эхобона зоны 1 (Северная Америка).

2 июня

1997. В первой зоне (Северная Америка) анонсирован доступ к эхоконференциям через группы новостей (NNTP).

2000. В нодлисте появилась сеть 2:6083 (Петропавловск, Казахстан).

3 июня

1985. Ben Baker (Fido #76) и Ken Kaplan (Fido #22) публикуют документ «Обязанности координатора сети или региона».

1991. Tom Jennings (вместе с Tim Pozar) снова стал редактором Fidonews, но сразу предупредил, что это ненадолго.

5 июня

1989. Jacek Szelozynski рассказал в FidoNews 6-23 о появлении в Польше первых BBS и поинтов Фидо (на голландском узле 2:286/201).

1998. Координатором региона 2:50 (Россия) становится Юрий Сафронов.

2014. В регионе 46 (Украина и Молдова) после длительного перерыва возобновлён выпуск регионального поинтлиста.

6 июня

1997. В нодлисте появилась сеть 2:5064 (Ставропольский край).

1997. В нодлисте появилась сеть 2:5067 (Майкоп).

7 июня

1991. В нодлисте появился регион 2:46 (Украина, Белоруссия и Молдова) и сеть 2:461 (Харьков).

8 июня

1990. Начало «FidoCon OZ! 1990» – первой конференции Fidonet в Мельбурне, Австралия.

1991. Вторая Азиатская конференция сисопов BBS в Тайбэе, Тайвань.

9 июня

1989. Принят ныне действующий FidoNet Policy Document (version 4.07) – основной документ всея Фидо. За его принятие проголосовало 152 сисопа, против – 75.



10 июня

1985. Сообщено, что у Ken Kaplan (Fido #51) и его жены Sally родился ребёнок, названный Eric Samuel Kaplan. Это первый ребёнок, одним из родителей которого был фидошник.

1985. Ben Baker (1:100/76 – Роксана, Иллинойс) написал программу LISTGEN, автоматически строящую роутинг.

1991. Tom Jennings предлагает ввести PGP-шифрование для нетмейла.

11 июня

1993. В нодлисте появилась сеть 2:5027 (Орёл).

1993. В нодлисте появилась сеть 2:5028 (Рыбинск).

12 июня

1985. Начало эпохи «многосетевого» Фидо. Все узлы к этой дате перешли на адресацию, учитывающую сети и регионы. Изменился также формат нодлиста.

1995. Christian Brandt (2:2480/604) предлагает разрешить вносить в нодлист интернет-адреса (IP, также упоминаются UUCP и telnet), по которым доступен фидо-узел.

14 июня

1991. В нодлисте появилась сеть 2:466 (первоначально Беларусь, а с 22 июля 1994 – Николаев, Украина).

2002. Суд отклонил иск экскоммуницированного экс-сисопа Helmut Hullen к Ward Dossche. Helmut требовал восстановить его в нодлисте. Разбирательство продолжалось два года.

16 июня

1990. В Тайбэе (Тайвань) прошла первая Азиатская конференция сисопов BBS. В ней приняли участие сисопы и гости из Японии, Южной Кореи и Тайваня, в том числе семь женщин. Японские компьютерные журналы писали об этой конференции.

1991. В FidoNews впервые сообщено о выходе фидософта для Windows. Will Schlichtman (1:350/59 — Мэрион, штат Орегон) анонсиро-

вал редактор сообщений MsgMstr v1.21 для Windows v3.0.

2000. В нодлисте появилась сеть 2:6035 (Железногорск Курской области).

17 июня

2016. Координатором региона 2:46 (Украина и Молдова) становится Юрий Рощупкин.

19 июня

1998. В нодлисте появилась сеть 2:5017 (Сергиев Посад).

1998. В нодлисте появилась сеть 2:4622 (Тернополь, Украина).

2009. Юрий Рощупкин выпустил «Фидо-нюс по-русски». Всего вышло два номера.

21 июня

1991. В нодлисте появилась сеть 2:463 (Киев, Украина).

1991. В нодлисте появилась сеть 2:469 (Молдова).

2004. В FidoNews анонсирован домен fidonet.net для binkp-узлов, а также краткая «инструкция по эксплуатации». Впоследствии заменён доменом binkp.net.

2012. Павел Гульчук выпустил релиз 1.0.0 IP-мейлера BinkD, вскоре ставшего самым популярным мейлером.

23 июня

1986. J. Brad Hicks, 100/523, подробно рассказывает о программе ECHOMAIL (автор – Jeff Rush), состоящей из двух программ: Scanmail and Tossmail. С её помощью эхопочта приобрела практически сегодняшний вид, а в письма стали добавляться клуджи AREA и SEEN-BY. Кроме того, произошло окончательное разделение на нетмейл и эхомейл, а личная переписка в эхоконференциях стала считаться дурным тоном. В версии ECHOMAIL 1.1 максимальная длина сообщений в эхопочте ограничивалась 100 строками. В версии 1.3 сообщение могло содержать до 200 строк.

1995. В нодлисте появилась сеть 2:5084 (Караганда, Казахстан).



24 июня

1991. Fredric L. Rice (1:102/901) анонсировал спеллчекер для проверки сообщений в FTN-сетях – MsgSpell.

1994. В нодлисте появилась сеть 2:473 (Клайпеда, Литва).

1994. В нодлисте появилась сеть 2:474 (Паневежис, Литва).

2018. Под заголовком «Этот день в истории Фидо» начат постинг календаря фидошных дат (сокращённую версию которого вы сейчас читаете) в RU.FIDONET.TODAY.

25 июня

1988. Начало EuroCon II – второй европейской конференции Fidonet в городе Тил, Нидерланды. На ней присутствовало около 150 сисопов из десятка стран Европы, гости из Z1 и некоторых левонетов.

1999. В нодлисте появилась сеть 2:4531 (Витебск, Беларусь).

26 июня

1986. Принят всефидошный документ «Policy and Procedures Guide» (Version 2).

1998. В нодлисте появилась сеть 2:5068 (Череповец).

2000. В эхе REGCON.EUR опубликованы результаты голосования RC на выборах Z2C. Победил Ward Dossche из Бельгии (13 голосов); его соперник Jan H. Klug получил только 2 голоса. В семи регионах предварительно были проведены референдумы, чтобы узнать мнение простых сисопов.

27 июня

1997. В нодлисте появилась сеть 2:4616 (Северодонецк, Лисичанск, Рубежное – Украина).

28 июня

1991. В нодлисте появилась сеть 2:471 (Вильнюс, Литва).

2003. На эхобон R50 приняты эхи Su.Kaschenko.Local и Su.Kaschenko.Local.Must.Die.

30 июня

1986. IFNA, Международная ассоциация Фидонет (International FidoNet Association) зарегистрирована в США как некоммерческая корпорация. Просуществовала до 1990 года.

1995. В городе Арнем (Нидерланды) началась конференция DutchCon/EuroCon'95.

1997. В Z1 (Северная Америка) анонсирована новая система эхораздачи, альтернативная EchoBone. Она называлась «FidoSpine Distribution System for Echomail». Первоначально в неё входило 5 хабов, которые предлагали как традиционный способ получения эх, так и варианты через FTP и/или TransX-каналы.

2001. В FidoNews опубликовано интервью с Максимом Масютиным, соучредителем (и на тот момент вице-президентом) компании RITLabs. Наиболее известные продукты этой фирмы в то время – Dos Navigator, Argus и The Bat!.

2003. Состоялось торжественное открытие эхобона R50, который на тот момент состоял из двух 0-level хабов: 2:5080/1003 (Екатеринбург) и 2:5030/115 (Санкт-Петербург).

Владимир Фёдоров, 2:50/15



Игры и обучение подрастающего поколения

*Во мне идей, почти что миллион!
Как всё успеть, пока ещё не знаю сам.
С солдатиков я строю целый батальон.
Готовый дать отпор достойный, всем врагам.*

*Немного погода, из кубиков, построю дом.
Высокий – этажей на двадцать пять.
Чтоб жить уютно куклам было б в нём,
Поставлю шкаф, комод, два стула и кровать.*

*Из «Лего» сделаю крутой автомобиль.
Который мчится по ковру, до самых до дверей.
И катер, чтобы плыть в полнейший штиль
А капитан на нём пират – гроза пяти морей.*

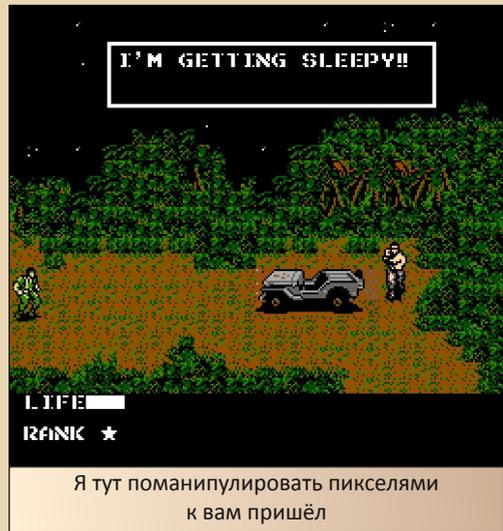
*Ещё хочу построить, крепость, звездолёт.
И на планшете в «Нинзя», хоть немного поиграть.
Готов играть, скакать и строить ночи напролёт.
Но папа строго говорит, что ночью нужно спать.*

Константин Политиц

Игры стали неотъемлемой частью нашей жизни и неотъемлемой частью образовательного процесса. Процесс информатизации не стоит на месте и внедряется в нашу жизнь. В школах уже давно проходят онлайн-тестирования, игровые тесты, поиск информации в интернете, сейчас активно осваивается технология виртуальной реальности. Внедрение видеоигр в образование было вопросом времени, и вот на сайте regulation.gov.ru появился проект, направленный на внедрение электронно-игровых комплексов в учебный процесс. Он направлен на реализацию мероприятий по разработке цифровых учебно-методических комплексов, учебных симуляторов, тренажёров, виртуальных лабораторий по предметам «Математика», «Информатика» и «Технология».

Такую идею уже апробируют частные проекты, например, для изучения математики в виде простых флеш-игр. Зная подобный опыт внедрения таких проектов и качество конечного продукта, становится немного страшно.

Отношение к видеоиграм в России всегда было консервативным. Примеров много: то им серия GTA пропагандирует насилие (с этим я согласен, и детей младшего возраста до таких игр бы не допускал); то конфликт с игрой DOTA 2 с наматыванием сами знаете чего; то «американские спецслужбы манипулируют молодежью с помощью таких проектов, как игра Metal Gear».



Ну, такой диссонанс объясняется очень просто. Подобные высказывания летят из уст закоряченных людей, которые и в игры то не играли, да и к образованию отношения не имеют. А вот инициатива подобных проектов поступает от инициативных педагогов, которые с разных уровней образования пытаются преобразить урок и вывести его на новый уровень. Каким образом будет реализован процесс – пока неизвестно.

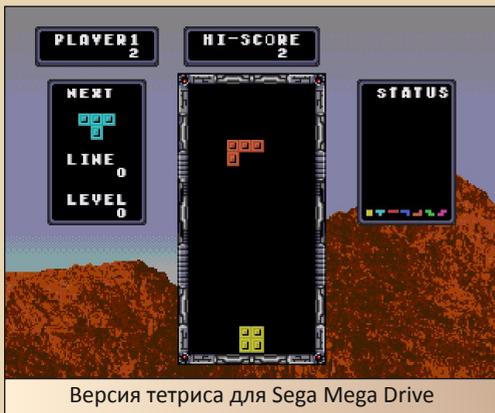
Эпоха видеоигр подарила нам множество проектов, и часть из них хорошо бы вписалась в образовательный процесс. Эта статья написана



исключительно для развлекательных целей. Давайте пофантазируем!

Развивать логическое мышление и реакцию нам помогла бы **Tetris (Тетрис)** и его клоны.

Великая игра от Алексея Пажитнова, которая породила целый культ и просто огромное количество подражателей, к примеру **Dr. Mario** или **Magic Jewelry** на NES. Проверенная временем головоломка. Рассказывать про неё смысла нет, так как найти человека, не игравшего в неё, очень сложно.



Версия тетриса для Sega Mega Drive

На уроках искусства по соседству с «Бельведером» Маурица Корнелиса Эшера идеально бы смотрелась **Monument Valley**. Головоломка от Ustwo Games, основанная на оптических иллюзиях. А приятная медитативная музыка только дополняла бы приятное восприятие.



Слева «Бельведер», справа локация из Monument Valley

Disney's Aladdin, Sonic The Hedgehog, Lion King, Flintstones, Jungle Book, World of Illusion Starring Mickey Mouse & Donald Duck, Toy Story, Mickey Mania – Timeless Adventures of Mickey Mouse – это далеко не полный список платформеров на Sega Mega Drive / Genesis. Добротно сделанные красочные «антипланшетные» игры. Многие из них легко проходятся и безобидны для детей (как, собственно, и мультфильмы от Disney). Множество моих друзей дают с часок поиграть своим детям или играют вместе с ними. Дети в восторге, особенно от Аладдина. Развивает моторику пальцев и общее мышление.



World of Illusion так и просит: пройди меня вдвоём

Для юных экономистов посоветовал бы **Theme Park**, которая выходила на множестве платформ.

Симулятор строительства собственного парка развлечений. Помогает правильно расходовать ресурсы, следить за разными аспектами. Как введение в экономику было бы отличным шагом.

Если тема развлечений будет далека, то **SimCity** могла бы подарить другой игровой опыт. Воспитать будущего мэра города? С этой серией игр это будет легко.

С крупных должностей можно спуститься к более приземленным и житейским проблемам. С этой задачей помогла бы **The Sims**. Как пособие для будущих матерей, которое пока-



зывает, что ребёнка надо кормить, одевать, мыть и ещё тратить на него деньги.



Игра может дать обратный эффект

Игра развивает и дизайнерские способности. Какими будут обои или как правильно сделать пристройку к дому – это ли не даёт фантазии для дизайнера и строительства? Идеальное пособие для уроков технологии по моделированию.

Помню, в школьные годы, когда персональный компьютер был только мечтой, мы собирались в школе поиграть в различные игры в кабинете информатики. Одной из таких игр была «Ну, погоди! Выпуск 3. Песня для зайца», игра в жанре квест. Тогда мы и не знали, что существуют интересные игры в данном жанре. Пока мы проходили данную игру или учили клавиши на уроках информатики, наш учитель ОБЖ, который был бывшим военным, играл в Aliens versus Predator. Мы исходили слюной, а учитель вояка махал кулаками перед экраном, как в симуляторе бокса через Kinect.

Квесты всегда были игрой на внимательность и сообразительность. Взять, к примеру, **Machinarium**, **Samorost**, **Botanicula**, а кто постарше вспомнит **LOOM** от Lucasfilm Games. Что не игра, то произведение искусства. Слушая саундтрек из Botanicula можно постичь дзен.

<https://youtu.be/mxfRPPEpS3g>



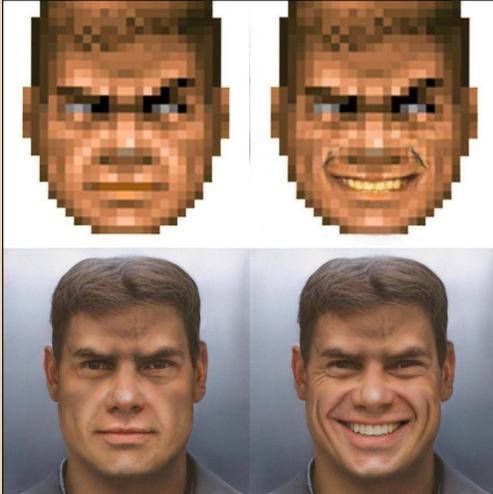
Любой скриншот можно смело ставить на рабочий стол

Вспомнить хороших игр можно ещё очень много. Одни учат нас соблюдать нормы морали, другие нарушать закон. Важность игровой формы играет большую роль. Здесь речь не о важности видеоигр, а именно об игровом методе воздействия и воспитания. Вспомните себя в молодые годы. Когда на экране была картинка, состоящая из пикселей и пары цветов, мы задумывались, мечтали, фантазировали. Иероглифы для нас были непонятны, но мы сами придумывали сюжет, воспринимали его по картинкам. Сейчас такого не происходит. Нужно разжевать весь сюжет, показать всё во всех подробностях и положить в рот. Не зря же появилось выражение «Press X to Win». Будем надеяться, что данный закон пойдёт по правильному направлению. Может, когда-нибудь мы увидим предметы «Игроматика», «Игрология», «Техноигростроительство», «Искусство пиксель-арта» или «История геймдева».

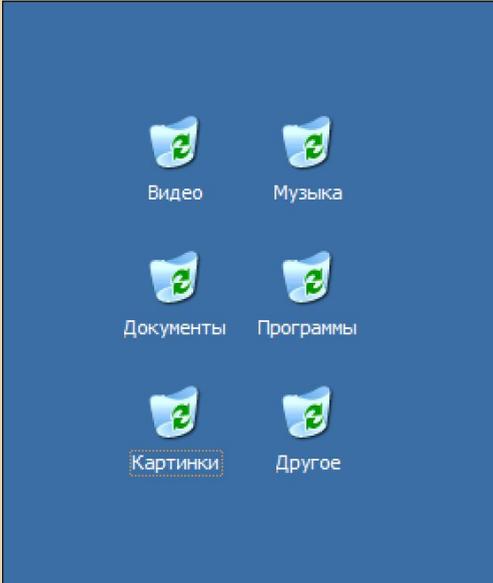
Павел Ярославцев (paHa_13)



ПРОСТО РАЗНЫЙ ЮМОР



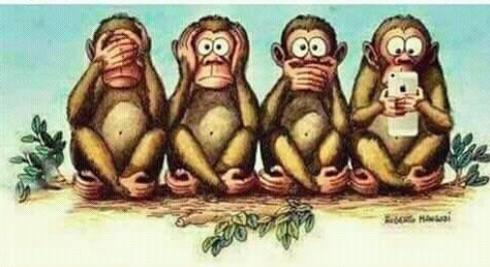
DoomGuy, обработанный нейросетью.
 Оригинал новости на reddit:
https://www.reddit.com/r/Doom/comments/bspsup/doomguy_irl_see_comment/



Раздельная утилизация мусора



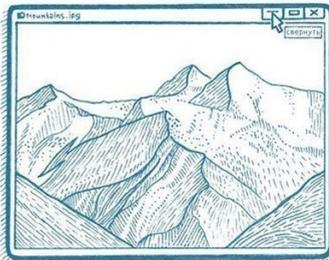
Обнаружена четвертая обезьяна - она объединяет всех предыдущих: ничего не видит, ничего не слышит и ничего не говорит.



Нетрадиционное использование компьютерных комплектующих



Я для тебя горы сверну



gdbesaulo

Если вы видите точки,
то вам пора чистить монитор



Дедушка! Расскажи нам ещё раз, как ты падал с третьего этажа



Над номером работали

Дизайн/вёрстка/главный редактор - uav1606
Редактор - Вячеслав Рытиков (eubrc)
Помощник редактора - Андрей Шаронов

Авторы:

Андрей Шаронов (Andrei88)

uav1606

SysTools

Олег Павлов

Павел Ярославцев (paha_13)

Андрей Тумилович

Владимир Фёдоров

Sh

Интервью:

В.Ф. Фёдоров

Сайт журнала: <http://dgmag.in>

Раздел журнала на "Железных
Призраках Прошлого":

[http://www.phantom.sannata.ru/
articles/dgmag/](http://www.phantom.sannata.ru/articles/dgmag/)

Группа ВКонтакте: <http://vk.com/dgmag>

E-mail главного редактора:
uav1606@cobaka@mail.ru