

# DOWNGRADE



N21 ' 2017



# СОДЕРЖАНИЕ

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

## ТЕОРИЯ DOWNGRADE

● Новости, события, комментарии (uav1606, eubpc)-----	4
● Интервью с bearwindows (bearwindows, eubpc, uav1606)-----	6
● Книжная полка: MicroCAP V - ваш выбор для виртуальной лаборатории (Андрей Шаронов)-----	18

## DOWNGRADE-ЖЕЛЕЗО

● Обзор сканера HP ScanJet IIp (uav1606)-----	20
● Моя идеальная DOS-машина (Роман Азарин)-----	24

## DOWNGRADE-СОФТ

● Stunnel - интернет с SSL для старых программ (А.Шаронов) -	26
● Electronics WorkBench - компьютер моделирует электронику (Андрей Шаронов aka Andrei88) -----	29
● Конкурент (Андрей Шаронов aka Andrei88)-----	33

## ПРОГРАММИРОВАНИЕ

● Новые средства разработки для DOS (Пётр Семилетов)-----	42
● push hl,de,bc,af (Sh) -----	45

## СТАРЫЕ ИГРЫ

● NFS Hot Pursuit 2 через Wi-Fi (А.Шаронов aka Andrei88)---	53
---	----

## РАЗНЫЙ ЮМОР

● Просто разный юмор -----	56
● Над журналом работали -----	57

## От редактора

Уважаемые читатели, приветствую вас на страницах номера 21 журнала Downgrade.

Темой этого номера должно было стать «Практическое использование старой компьютерной техники в наши дни». К сожалению, статей по теме прислали очень мало.

Да и вообще, набор статей этот раз затянулся больше обычного.

Даже не знаю – может быть, следует вообще сделать перерыв в выпусках журнала? Например, на год-два? Потому что в последнее время находить материалы для журнала уж слишком тяжело...

Перенос сроков приёма статей по десять раз вперёд смотрится не очень хорошо...

Ещё и многие авторы куда-то пропали – например, Юрий Литвиненко, wormsbiysk, Kakos\_Nonos, да и многие другие – без их статей как-то грустно...

Ну да ладно, долой упаднические настроения. :-)

Посмотрим, как сложатся дела с номером 22 – а там видно будет.

Приятного чтения.

uav1606

# НОВОСТИ, СОБЫТИЯ, КОММЕНТАРИИ

## MP3 стал бесплатным

23 апреля 2017 закончились сроки действия патентов на формат и прекращён сбор лицензионных отчислений.

Об этом заявили представители Fraunhofer Institute – института, который, собственно, разработал MP3 и владел соответствующими патентами.

Спустя 24 года после публикации формат наконец может считаться общественным достоянием.

Оригинал заявления Fraunhofer Institute здесь:

<https://www.iis.fraunhofer.de/en/ff/amm/prod/audiocodec/audiocodecs/mp3.html>



## Chaos Constructions 2017

26-27 августа в Санкт-Петербурге прошёл очередной фестиваль Chaos Constructions.

Как всегда, посетителей ожидала выставка ретрокомпьютеров, демосцена, конкурсы, электронная музыка и многое другое.

Ознакомиться с результатами фестиваля можно здесь:

<https://chaosconstructions.ru/>

<https://vk.com/chaosconstructions>

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

<https://youtube.com/watch?v=smSCL2vU9AA>

## Doom II за \$3000

Недавно Джон Ромеро выставил на eBay набор оригинальных дискет Doom II (одним из разработчиков которой он и является). После

83 ставок комплект из пяти 3.5" дискет был продан за \$3150.

Ссылка на лот:

<http://www.ebay.com/itm/272734734257>



## Умер Андрей Чернов

На 52-м году жизни скончался программист Андрей Чернов (**ache**). Он известен как создатель кодировки KOI8-R, автор почтовой программы UUPC/@, а также считается одним из создателей Рунета.

Сайт Андрея – <https://ache.vniz.net/>



## 25 лет Mortal Kombat

9 августа 1992 года Midway Games выпустила первую часть файтинга Mortal Kombat.



Изначально игра была предназначена для аркадных автоматов, а домашняя версия для консолей Super NES вышла в 1993 году.

С тех пор вышло множество продолжений (последняя игра серии – Mortal Kombat X – выпущена в 2015 году), было снято несколько фильмов, сериалов, написано несколько книг о вселенной Mortal Kombat и т.д.

И даже спустя четверть века игры серии остаются популярными во всём мире.

### В Харькове открылся музей компьютерной техники

29 августа в Харькове открылся, вероятно, первый на Украине музей компьютеров и программного обеспечения. В экспозиции было представлено более сотни различных экспонатов. Среди них Apple II, Osborne 1, компьютер BBC и многое другое.

Официальные страницы:

<https://www.sncmuseum.org/>  
<https://facebook.com/sncmuseum/>



### Sega дарит свои игры пользователям мобильных устройств

Компания Sega заявила о новой инициативе под названием Sega Forever. Её суть – в портировании старых приставочных игр под мобильные платформы Android и iOS – причём они будут доступны совершенно бесплатно.

В «первую волну» вошли игры Sonic the Hedgehog, Comix Zone, Altered Beast, Kid Chameleon и Phantasy Star II.

Позже компания обещает расширить этот список, добавляя в него новые игры каждые две недели.

Официальный сайт:

<http://forever.sega.com/>

### Смартфон на Windows 95?

Как бы мог выглядеть смартфон, если бы его выпустили в середине 90-х? Henrique Perticarati решил поделиться своими идеями об этом, создав нечто вроде рекламной презентации WinPhone 95.

Жаль, что воплотить его в «железе» пока никто не взялся.

Подробнее с концепт-артом вы можете ознакомиться здесь:

<https://www.behance.net/gallery/56616085/WinPhone-95>

### Grand Theft Auto на калькуляторе

Конечно, как всегда, речь идёт не о совсем обычном калькуляторе, да и запустить игру удалось не напрямую, а на эмуляторе.

Некто под ником **RogueConditional** использовал калькулятор Texas Instruments TI-Nspire CX с процессором ARM. Сделав джейлбрейк, ему удалось запустить эмулятор Game Boy Advance, а на нём уже, соответственно, игру GTA Advance (2004 года выпуска). Подробнее здесь:

<https://4pda.ru/2017/09/05/346525/>

*Прим. ред.: в обзоре использовались изображения с сайта [wikipedia.org](http://wikipedia.org)*

---

Обзор подготовили:  
Вячеслав Рытиков (eubrc)  
uav1606



# Интервью с bearwindows

*Вашему вниманию предлагается интервью с bearwindows – больше всего он известен как разработчик универсального VESA-драйвера VBEMP для различных платформ, но также на его счету множество других программ, утилит и сборок, имеющих отношение к Downgrade.*

**Расскажите, пожалуйста, немного о себе. (Любую информацию на Ваше усмотрение.)**

Скажу, что я живу в РФ и мне хорошо за 30.

**Когда у вас появился первый домашний компьютер? Что это была за модель?**

Это было в начале 90-х годов. Мне был подарен клон SPECTRUM (<http://speccy.info/Коллибри>) (на отечественной электронной базе, внутренний БП) + «Рубин Ц-208». К телевизору была куплена плата сопряжения УМ1-5, организующая НЧ-видеовход, чтобы не пользоваться антенным и получить более чёткую картинку. Для антенного входа был куплен запасной преобразователь RGB-сигнала.

Покупалась литература известных издательств «Солон», «Питер» по играм и программированию, по TR-DOS.

Потом на замену был куплен такой более симпатичный аналог с более удобной клавиатурой и внешним БП в металлическом корпусе.



Металлический шильдик только на нём другой, как тут:



При длительной эксплуатации БП грелся. В данную модель внутрь корпуса была установлена плата с TR-DOS на основе KP1818BГ93.

При эксплуатации разных моделей были схожие проблемы – горели микросхемы. В моём случае это коснулось ВГ93, Z80.



Один раз сгорела ЭСПЗУ ROM. Эта микросхема вообще была без опознавательных знаков, что было странно.

Для считывания кассет использовались в разное время – «Романтик-306», «Легенда-404», «Электроника-302-1».



Первым PC-совместимым был в 1995 г. легендарный IBM PC 5150 (2xFDD 360K, 256K памяти, МДРА-адаптер) + IBM 5151 12" TTL моно монитор.





К нему позднее появился более эргономичный моно монитор PACKARD BELL – от него меньше глаза устают – и более удобная клавиатура AT/XT. Какое-то время в корпус от 5150-го была с трудом временно «вкорячена» обычная для тех лет материнская плата Baby AT на базе впаянного 80386DX-40 на 30-pin SIMM.

Следующим компьютером было что-то с PENTIUM 1 на 430-м чипсете. Система была с двумя картами/мониторами SVGA/SVGA (или EGA/EGA?) + Hercules/Mono.

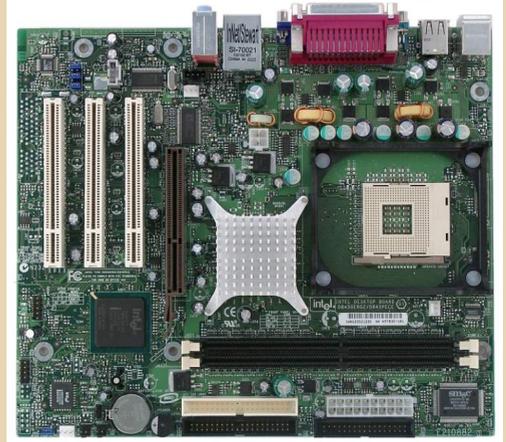
Интересный факт: в 2000 г. я застал работающую (порядка ~10 лет) DOS-сеть на Token Ring с сервером IBM PS/2 Model 80 (80386).

Затем году в 2001-2002 появился компьютер на базе ASUS PVI-486SP3 + AMD 5x86 133.

Это, на мой взгляд, одна из лучших поздних плат под 80486 – есть VLB, PCI, Flash BIOS, Onboard IDE (PIO4).



Следующим компьютером стал в 2004 г. PENTIUM 4 2.40 (S478) + INTEL D845GERG2 + ATI Radeon All-In-Wonder 9000 + 2 GB.



Данная машина у меня основная. Стоит на ней большой «зоопарк» из:

1. DOS+Windows 2.0/Windows 3.0/ Windows 3.1
2. Windows 98 SE
3. Windows NT3.1
4. Windows NT3.51
5. Windows NT4.0
6. Windows 2000 (основная система)
7. Windows XP

Дополнительно в 2008 г. была куплена машина на базе Core 2 Duo E8500 + 8 GB RAM

Видеокарта стоит RADEON HD7750 Ultimate с пассивным охлаждением.

**Где и как вы научились программировать? Это было сложно?**

Учился самостоятельно. Высшее образование у меня вообще не по IT-направленности.

В момент увлечения ZX в основном играл в игры и немного начинал программировать на BASIC-е.

Мне всегда было интересно, как на низком уровне работает периферия в компьютерах.

Когда был SPECTRUM, по крупицам изучал, как работает ULA, редкая периферия вроде



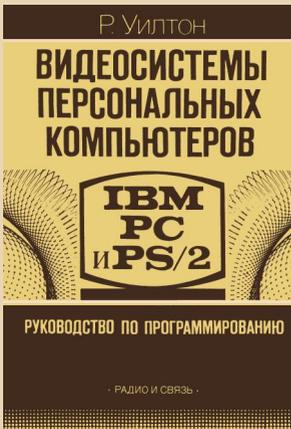
OPUS DISK и ZX MicroDrive, как там идёт перехват управления и задействуется теневого BIOS. По ходу писал небольшие программки на BASIC – можно сказать, что это были графические демки, демонстрирующие возможности платформы на Z80.

В 1995 г., с появлением нормального IBM PC и 80386-й платы, ZX был постепенно заброшен, взялся за программирование и тут. В первую очередь интересовало, как организована работа «железа» тут – видеокарты, звуковые, сетевые карты. Писал на QBASIC-е, ассемблере под DOS – разные видеодемки, вьюеры для графических файлов (BMP, WMF), программки для передачи данных по COM/LPT.

Много экспериментировал с текстовым режимом – ASCII art, скроллинг, спецэффекты разные, даже вьюер писал для текстового (!) режима (BMP-файлы, граф. ресурсы Wolfenstein 3D). Эмулировал AdLib на PC speaker (для MIDI и музыкальных тем из Wolfenstein 3D).

Неудовлетворённый возможностями штатных драйверов, написал программку для распечатывания картинок на матричных принтерах (совместимых по языку с ESC/P от Epson). Недавно переписал её для Win32.

Вот моя настольная книга в 90-х:



Видеосистемы персональных компьютеров IBM PC и PS/2. Руководство по программированию. Уилтон Р. Год изд.: 1994.

Раньше я её в библиотеке брал, а позднее всё-таки купил – и не жалею!

Затем потребовался более «взрослый» язык. Хотя долгое время всё равно тянуло к BASIC-у – на нём было проще набросать алгоритм и затем перенести на другой (более сложный) язык.

В начале 2000-х перешёл на Паскаль (DOS/WIN16 Microsoft и Borland). Писал под DOS, Windows 2.x (<http://bearwindows.zcm.com.au/win2xru.htm> – это уже тогда была экзотика), Windows 3.x. Разбирал, как WIN16 API эволюционировал от WIN 2.x до более привычных WIN 3.0/3.1. Писал программки, позволяющие просматривать и редактировать BMP- и WMF-файлы – имейте в виду, что для Windows 2.x – это «чуждый», неизвестный формат, алгоритм делался вручную. Это с выходом Windows 3.0 эти форматы стали стандартом.

Когда учился в универе – там был Паскаль – у меня вся группа списывала. :)

Немного ковырял FoxBASE – понравилась там система меню. А так – лучший GUI для DOS – это Norton Utilities 7.0/8.0.

Для Win32 перешёл на FreePascal. В 2005-2006 г. наконец руки дошли до C/C++. Впоследствии очень жалел, что не перешёл на C раньше.

Пользовался Borland C++ (DOS), Microsoft C++/VC++ (DOS, Win32/64), OpenWatcom C++ (DOS, Win32, OS/2), DJGPP (DOS).

Традиционно писались всякие демки, тесты, системные утилиты.

Несколько лет изучал 3D API (DirectX7/8/9/OpenGL/allegro/unity3d). Портировал OpenGL-демки, игры с Windows на OS/2.

**Расскажите о драйвере VBEMP – как вам пришла в голову идея его создания, кто вам помогал, были ли какие-нибудь трудности?**

Началось это ещё где-то году в 1998-1999, когда я долго сидел на EGA-мониторе, и меня бесило, что 16-цветная EGA-палитра не 100% соответствовала VGA-шной и многие игры/программы выглядели странно. Монитор был к то-



му же с поддержкой 640x480x16 цветов (т.н. SuperEGA). Тогда я трудом достал WINDOWS 3.1 DDK и решил впервые в жизни написать, т.е. переписать видеодрайвер. Была поставлена задача – максимально адаптировать EGA-драйвер под VGA-палитру. До кучи немного переделать значки кнопок интерфейса Windows 3.1 а-ля Windows 95/98. Что и было сделано. Драйвера эти сохранились, но без исходников.

В 2000 г. произошло моё не первое, но более вдумчивое знакомство с Windows NT 4.0 на Pentium 1. Мне очень понравилась эта система своей стабильностью, ориентацией на бизнес-сегмент и переносимостью на разные CPU – MIPS, DEC ALPHA, PPC, а впоследствии это «понаправилось» распространилось на всю NT-линейку. Кажется, что Windows NT медленнее, чем Windows 9x, но при наличии большого количества RAM NT 4.0 начинает работать отлично. Эх! А память тогда была дорогая! Не очень люблю Windows 9x (хотя какое-время работал на ней) – единственный её плюс – что она более оптимизирована под слабые машины, т.к. там всё ядро и драйвера написаны на ассемблере. Лучше совместимость с multimedia, играми, DOS. При этом, не разделяю мнение, что Windows Millenium провально плох – там многие вещи (USB Storage, UPNP, поддержка WDM-драйверов) доведены до ума – именно на нём впоследствии будет отлаживаться VBEMP 9x. Что же касается поддержки DOS – то я считаю, что для DOS-программ есть чистый DOS. А нормальный Windows – это Windows NT/2K/XP и далее. Как в рекламе Джеймса Бонда – агента 007 – «Windows NT – No Substitute!». Время показало, что спустя несколько лет настоящая популярность к NT-ядру придёт – это произошло, когда в 2001-м вышла Windows XP.

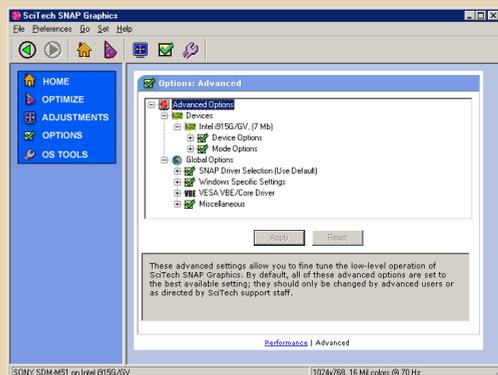
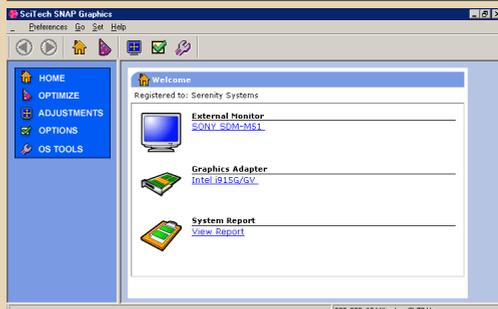
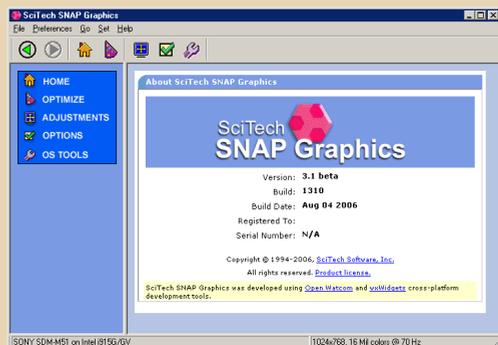
Теперь вернёмся к VBEMP.

В середине 1990-х существовал такой замечательный резидентный DOS драйвер – UNIVBE (<https://ru.wikipedia.org/wiki/UniVBE>). Его особенность была в том, что он «лечил» и «доводил до ума» некоторые BIOS видеокарт для лучшей совместимости с играми и про-

граммами, использующими видеокарту по VESA- стандарту.



Ему на смену пришел SciTech Display Doctor, а позднее и ScitechSoft SNAP Graphics (System Neutral Access Protocol) для x86 процессоров и разнообразных операционок MS-DOS, OS/2, Microsoft Windows (CE, NT, 2000, XP), QNX, SMX (the SunOS/Solaris port of MINIX), Linux, On Time RTOS-32, Ununinium OS.



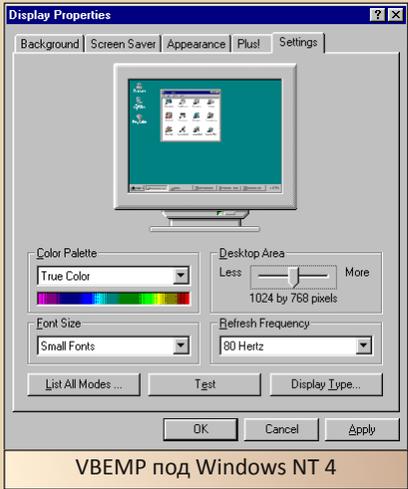
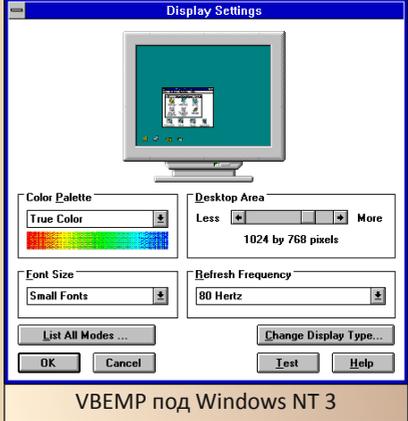


В 2005 г., после знакомства с ReactOS, появилось желание написать VESA-драйвер для Windows NT. А тут как раз – в 2004 г. – Windows NT 4.0 Microsoft сняли с поддержки, а в 2006-м выпуск SNAP Graphics был прерван, код был продан Alt Richmonds Inc., а SciTech прекратила своё существование – грандиозная затея написать универсальный (заметьте, аппаратный, а не транслятор вызовов из VESA BIOS!) драйвер «для всего и вся» провалилась – а жаль! Там были зачаточные возможности 3D-ускорения, поддержки нескольких мониторов, оверлея для воспроизведения видео, был аналогичный SNAP Audio для звуковых карт.

Имя VBEMP (VBE MiniPort) было взято из ReactOS – тогда (в 2004-2005 г.) не очень доведённого до ума проекта по созданию open source версии ОС, совместимой с Windows NT и далее. Код оттуда я взять не мог, т.к. он не был 100% совместим с Windows. Были консультации с разработчиками (теперь бывшими) из команды ReactOS. А потом помогать особо никто не помогал (кроме немногочисленных тестировщиков) – наиболее сложно шла разработка версии для Windows NT4/2000 – т.к. там нет стандартного способа доступа к видео BIOS, всё делается «хаками» и работает не так чётко и стабильно, как в Windows XP/2003, где наконец Microsoft написала свой VESA-драйвер (VGA.SYS) и, соответственно, нормальный интерфейс доступа к вызовам INT 10h из защищённого режима. Откровенно говоря,

именно в Windows XP/2003 и далее VBEMP практически бесполезен. Если только вам не нужна поддержка VESA 3.0 (для LCD она бесполезна, и кроме видеоадаптеров Intel сейчас VBE 3.0 нет нигде), сна/гибернации и видеорежимов < 640x480. Много проблем доставляли баги в BIOS-ах разных производителей, были проблемы с картами, у которых два и более выхода (VGA и DVI, например).

Доходило до смешного, что случайно оставленные мною в коде ошибки помогли драйверу проинициализироваться и нормально работать. Так вышло, например, с VBEMP NT для Windows NT 3.1/3.5x.

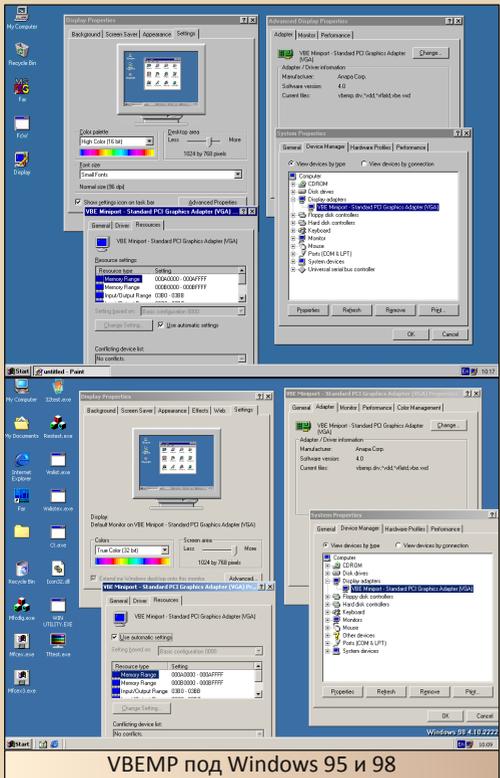


Дополнительно был выпущен аппаратный VBEMP NT-драйвер для карт Cirrus Logic CL-GD54xx, который умеет VESA 3.0, WideScreen-режимы. За основу был взят и адаптирован код из Linux cirrusfb.

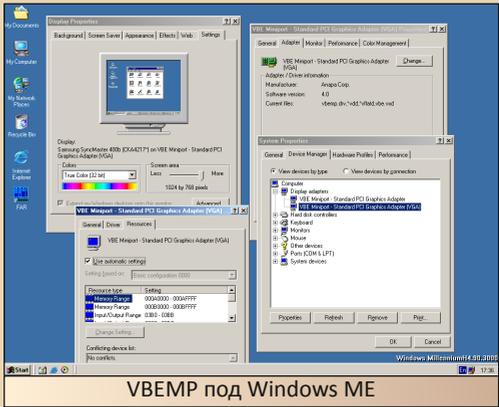
Этот драйвер хорошо работает с эмулятором QEMU, если настроить в нём эмуляцию Cirrus Logic.

В процессе разбирательств выяснилось, что многие из карт серии формально не умеют true color (16/24/32 bpp), но фактически, при правильной настройке таймингов, позволяющей такую цветность установить. Также было выяснено, что многие из ISA/VLB-карт умеют линейный фреймбуфер, что радикально ускоряет перерисовку экрана.

Позже пошёл в разработку и VBEMP для Windows 9x – он появился в 2008 г., и даже сейчас имеет некоторые проблемы – например, «мусор» на экране и нестабильная работа в DOS WDM.



VBEMP под Windows 95 и 98



VBEMP под Windows ME

Очень мало официальной документации от Microsoft. Всё пишется вслепую методом проб и ошибок. Тяжело отлаживать, т.к. на отлаживаемой машине нужен встроенный настоящий COM-порт, а не PCI-плата с COM. Какой-нибудь COM нужен и для отладчика. Это касается и NT-систем. При малейших ошибках система падает в BSOD или показывает чёрный экран, «срывает» синхронизацию.

В процессе заинтересовался Native API, прочими типами драйверов. Думаю, что это выходит за рамки данной статьи.

**На каком языке программирования написан драйвер? Какие использовались дополнительные инструменты?**

VBEMP NT (для Windows NT/2K/XP) написан на обычном C с вставками на ассемблере.

Дополнительно использовался статический анализатор C/C++ кода Microsoft PreFast for drivers.

VBEMP 9x (для Windows 9x) написан полностью на ассемблере.

Использовался IDA Pro для реверс-инжиниринга некоторых файлов, отвечающих за видеоподсистему в Windows 2000/XP.

**VBEMP не обновлялся более двух лет. Планируете ли Вы дальше работу над проектом? Появятся ли в нём какие-нибудь новые возможности?**



Основной функционал VBEMP NT давно реализован.

Была идея дооснастить VBEMP для NT3/NT4 двойной 2D-буферизацией для ускорения перерисовки. В Windows 2000 и далее она штатно есть. Но добиться стабильной работы переделанной библиотеки **framebuf.dll** не вышло, и код был отдан сообществу ReactOS (в коде ветка `drivers/displays/framebufacc`, если я не путаю). Вообще, многие из моих «хаков», фиксов впоследствии ушли в VBEMP от ReactOS.

Рассматривался вариант добавить поддержку пользовательских (т.е. любых) режимов для VBEMP NT. В итоге появился на основе «915resolution: Intel Video BIOS Hack»:

1) для Win NT/2K/XP: отдельный драйвер **IEGDNT.SYS**, который работает только с видеокартами INTEL;

2) для Win9x: патчер `inteldmp` и комплект резидентных BIOS для него с аналогичным функционалом.

В этом году по многочисленным просьбам на сайт VBEMP добавил «шпаргалку» по возможностям, как добавить к нему 3D-«ускорение».

### Могут ли наши читатели чем-то помочь в дальнейшей разработке?

Логами от видеокарт (очень интересую всякие редкости, а не стандартные вроде AMD/NVIDIA/INTEL). Подарить/выменять видеокарту, какой у меня нет. Ну и финансово конечно.

Способ получения логов описан здесь: <http://bearwindows.zcm.com.au/vbempru.htm#9>

Что такое «Корпорация АнаПа»? (Упомянется на вашем сайте.) Ну и, может быть, заодно расскажете про происхождение Вашего ника?

Это такой каламбур из разных слов, инициалов. Секрет, короче. А по поводу **bearwindows** скажу так:

1) **bear** – это есть такой mascot (талисман?) «Microsoft Bear»:

[https://en.wikipedia.org/wiki/List\\_of\\_Easter\\_eggs\\_in\\_Microsoft\\_products](https://en.wikipedia.org/wiki/List_of_Easter_eggs_in_Microsoft_products)

Есть ещё глагол «to bear» и гора Медведь (Аю-Даг) в Крыму.

А к живым медведям и к Медведю **bearwindows** отношения не имеет. И мёд я не люблю. ☺

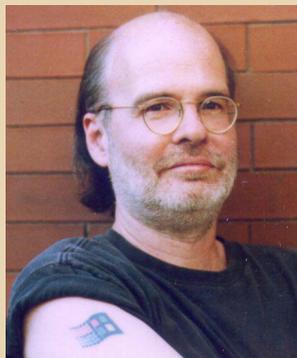
Дома есть несколько игрушечных медведей, подаренных мне чисто случайно.



2) **windows** – это понятно, откуда – «Microsoft Windows».



Есть такой известный писатель Charles Petzold – один из первых авторов книг по программированию Windows и OS/2.



Я тоже люблю Windows, но не настолько.

Расскажите немного о OS/2 Multi-Bootable Recovery CD/DVD – что это за диск, для чего предназначен, какие у него возможности?

Начнём с того, что для меня OS/2 Warp – эта такая (незаслуженно забытая) модульная операционка полу-Windows, полу-Linux. Обладает минимальными требованиями к «желе-



зу», отличной надёжностью, стабильностью (использовалась в банкоматах), возможен запуск без графической подсистемы, но уже в защищённом режиме CPU. Легче Linux-а в освоении. Хорошая программная совместимость между разными версиями OS/2. Программирование довольно похоже на WINAPI из Windows. Подсистемы для запуска DOS/WIN16-программ и WIN32-программ через Odin (аналог Wine). В 2006, с выпуском FixPack 6 для Warp 4.5x, IBM прекратила обслуживание OS/2, и сейчас в нынешнем виде эта ОС поддерживается небольшой группой людей, которые пишут драйвера для неё, пишут и портируют с Linux-а современный open source софт вроде GCC, Fire-Fox/Seamonkey и OpenOffice. В данный момент выпускается под брендом eComStation, ArcaNoae ArcaOS, BlueLion.



OS/2 Recovery CD делался как вариант eComStation Demo CD, но:

- 1) с меньшими требованиями к целевому компьютеру;
- 2) для live-загрузки, восстановления системы или её клонирования;
- 3) «Multi-Bootable», т.е. с разными версиями OS/2 (v3, v4), разными ядрами (W4, UNI, SMP), разные методики эмуляции носителей загрузки – FDD, HDD, MemDisk (т.е. RAMDisk) для разных типов BIOS (местами очень «привередливых»).

Сейчас проект заморожен, пользуйтесь аналогичным диском от Team BOOT/2 OS/2 (Ва-

лерия Седлецкого, участника проекта osFree, **valerius** на irc канале #os2russian)

**Вы, кажется, какое-то время назад занимались поддержкой OpenGL на видеокартах S3 Virge/Trio3D – расскажите, что из этого получилось?**

Это была [статья на «ПП»](#), рассматривающая аспекты OpenGL-ускорения на базе S3D Toolkit и стандарта OpenGL MCD ускорения для Windows NT 4.0/2000 (не только для S3).

**На Вашем сайте есть множество утилит для DOS и Windows – расскажите о, на Ваш взгляд, наиболее интересных.**

Всякий разный софт – <http://bearwindows.zcm.com.au/myproj.htm>

Небольшой список для примера. Указанный софт собран для Win32 (большинство из примеров рассчитано на максимально широкий спектр совместимых ОС, включая Win32s, NT 3.51, Win95, NT4.0):

- BltTest.exe** – тест скорости «блиттера» для VBEMP – обмен растровыми блоками между системной и видеопамятью (GUI).
- CF.EXE, lamp.exe, LTEST.EXE, LTEST10.EXE** – тесты OpenGL (GUI).
- constest.exe** – перечислитель устройств (консоль).
- ctbeta2.exe** – тест самодельного интерфейса проводника, требует icon32.dll (FreePascal, GUI).
- deskt.exe** – тест видеорежимов (консоль).
- dg.exe** – показывает геометрию дисков (консоль).
- EDP.exe, EDP\_MD.EXE, mfcex.exe, MFCEX\_MD.EXE, mfcex3.exe** – самописный диспетчер устройств (GUI).
- enump.exe** – листинг процессов (консоль).
- FIXTIME.EXE** – приводит дату время файла к дате времени компиляции, проверяет контрольные суммы (консоль).
- GetCRYPT.exe, ReadSPC.exe, whql\_ex.exe** – показывает подписи файлов (консоль).
- getif.exe, iphlptst.exe, iphlptst5.exe** – показывает сетевые интерфейсы (консоль).



- glvars.exe, GLXINFO.EXE** – показывает информацию о OpenGL-свойствах (консоль).
- MAKEMODE.EXE** – заполняет ветки реестра видеорежимами для VBEMP 9x (консоль).
- mfcldg.exe, win32t.exe** – тест видеорежимов (GUI).
- NATIVE.EXE** – тест Native API (native).
- NETE.EXE** – экспорт учетных записей в файл (консоль).
- NTOBJ.EXE** – внутренние объекты NT (консоль).
- peread.exe** – сканирует PE-заголовки файлов (консоль).
- rasmon2k.exe** – вызывает в Windows 2000 монитор удалённого доступа из NT4 (GUI).
- rastst.exe** – тест RAS-соединений (консоль).
- RESTEST.EXE, treetest.exe** – тест Common Controls (GUI).
- sskit.exe** – интегратор Service Pack для NT (консоль).
- vmlistex.exe** – перечисляет видеорежимы (консоль).
- w32app.exe** – тест для Win32s (GUI).
- WIZ97.EXE** – тест Wizard'a (GUI).

**Случались ли какие-нибудь смешные случаи или курьёзы, связанные с Вашими программами?**

Всё, что я пишу из софта, обычно носит довольно «серьёзный» и специализированный, местами даже исследовательский, а не практический уклон. Можно сказать, что для таких же программистов-системщиков, как и я. Из забавного? Ну не знаю. Чаще бывало наоборот, что при регулярном общении в форумах, например, набегали всякие юзеры, которым вынь-да-положь, чтобы мои программы (VBEMP в частности) умели всё на свете и сразу. А у людей сами знаете как – одному одно надо – другому – другое. А я пишу – по-третьему, и бесплатно. В итоге я предпочитаю «срачу» в форумах – личное общение, по переписке, например.

**Расскажите о своих других, не упомянутых здесь, проектах.**

Касательно «Полигона» – там есть ещё мои материалы по:

- 1) SSKIT – консольный инструмент интеграции Service Pack в Windows NT – [ссылка](#)

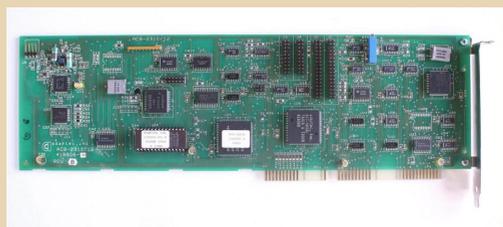
- 2) Sewell Direct FastLynx – связь сервера (Windows) с клиентом (DOS/Linux/Windows) по сети/COM/LPT – [ссылка](#)
- 3) Compushow 2000! 2.04a – выюер картинок для любой (!) видеокарты – [ссылка](#)
- 4) Бездискковая загрузка по (E)ISA-шине – [ссылка](#)
- 5) Аппаратный OpenGL на S3 Virge/Trio3D – [ссылка](#)
- 6) Связь Win2000/XP с XT по COM-кабелю – [ссылка](#)
- 7) Реализация IPX под DOS года 95-96 – [ссылка](#)

Если полистать «ПП», люблю «поумничать», когда поднимаются вопросы касательно Windows NT/2000.

Много занимаюсь темой запуска современных программ для Windows NT/2000, например, пересобирал из исходников Seamonkey 2.11 для Windows 2000.

Общался с **blackwingcat**, японцем, который самостоятельно адаптирует ядро W2K, драйвера и современные программы для работы в среде Windows 2000.

Совместно с **alter** (Александр Телятников, Киев, Украина) многие годы помогаю в разработке и тестировании UniATA ([http://alter.org.ua/ru/soft/win/uni\\_ata/](http://alter.org.ua/ru/soft/win/uni_ata/), статья на хабре [tyt](#)). Поддержка Intel AHCI и NT 3.5.1 в UniATA – это моя заслуга. По приколу переделывал (практически довнгрейдил) UniATA для MFM/RL-дисков. Есть такие контролеры, которые хоть формально и не IDE/PATA/ATAPI-совместимы, но используют аналогичные команды на низком уровне, например Adaptec ACB-2310/12:



Уже несколько лет специальная версия UniATA входит в дистрибутив ReactOS.

Материалы с моего сайта:

- 1) Многие годы собираю информацию о совместимости Windows NT линейки с железом, софтом:



Windows 3.1 + Win32s;  
 Windows NT 3.51;  
 Windows NT 4.0 (сотрудничаю с Calvin Melen «Windows NT 4.0 Reference Material» – <http://nt4ref.zcm.com.au/index.htm>);

Windows 2000;  
 Windows XP (всё планирую, да как-то лень, наверное, жду 2019 года, когда WEPOS снимут с поддержки ☺).

2) Статья 2D-акселерация в DOS – <http://bearwindows.zcm.com.au/af.htm>

3) «Хаки» VGA.SYS – <http://bearwindows.zcm.com.au/vga.htm>

4) Методичка по линуксам (уже устарела) – <http://bearwindows.zcm.com.au/lin.htm>

5) Windows. Разное – <http://bearwindows.zcm.com.au/wintips.htm>

4) Подробный разбор совместимости разных компиляторов и операционных систем от Microsoft – <http://bearwindows.zcm.com.au/msvc.htm>

В качестве эксперимента поставил Windows NT 4.0 на SuperMicro X8DT6 – практически скрестил ужа с ежом – это мы любим!



Item	Value
EVEREST V4.00.1500	
CPU Properties	
CPU Type	6x 2933 MHz
Instruction Set	x86, x86-64, MMX, SSE, SSE2, SSE3, SSE3E, SSE4.1, SSE4.2
Original Clock	2933 MHz
L2 Cache	256 KB (Asynchronous)
Multi-CPU	
CPU #1	Intel, 2933 MHz
CPU #2	Intel, 2933 MHz
CPU #3	Intel, 2933 MHz
CPU #4	Intel, 2933 MHz
CPU #5	Intel, 2933 MHz
CPU #6	Intel, 2933 MHz
CPU Utilization	
CPU #1	0 %
CPU #2	0 %
CPU #3	0 %
CPU #4	0 %
CPU #5	0 %
CPU #6	0 %

Microsoft (R) Windows NT (TM) Workstation  
 Версия 4.0 (Сборка 1381; Service Pack 6)  
 x86 Multiprocessor Free  
 01234-270-4142012-94449

Пользователь:  
 OEM  
 OEM

Всего:	1034	Физическая память (Кбайт)	
Дескрипторов	112	Всего	3136168
Потоков	17	Доступно	2980296
Процессов	17	Файловый кэш	18848
Выделение памяти (Кбайт)		Память ядра (Кбайт)	
Всего	32240	Всего	13244
Предел	5106128	Выгружаемая	11548
Пик:	38664	Невыгружаемая	1696

Процессов: 17    CPU: 0%    Память: 32240K / 5106128K

Тип адаптера  
 AnaPa Corp VBE Miniport

Изменить...    Определить

Отмена

Драйвер  
 Изготовитель: AnaPa  
 Номера версий: 10.07.0009, 4.0.0  
 Текущие файлы: vbemp.sys, framebuf.dll

Адаптер  
 Тип микросхем: VESA/SVGA  
 Тип конвертера DAC: Internal  
 Объем памяти: 16 Мбайт  
 Строка адаптера: VESA-Compatible Display Adapter  
 Информация BIOS: VESATm BIOS

Написал софт для удалённой «трансляции» рабочего стола сервера (Windows) на клиента



(Windows/DOS) – аналог MSRDP, VNC, но свой протокол на UDP-сокетах, view-only, поддержка EGA MONO/EGA/VGA/SVGA (клиент). TCP-стек для DOS – Novell или Wattcp.

Экспериментировал с NBI-загрузкой DOS по сети (80486 + карта NE2000 ISA) – работало.

Изготовил 200 MB compressed-NTFS образ Windows PE (на базе Windows 2000/XP x86/x64), который грузится по сети (PXE-протокол, сторонний драйвер WinVBlock), пользуясь на работе. Ведутся работы по драйверу бездисковой загрузки NT40.

По мелочи – делал null-модемный кабель из UTP витой пары (хорошо держит линию!), VGA-SCART БЕЗ дополнительного питания для Matrox G100 AGP, Radeon 9xxx.

**Есть ли у Вас какие-то планы на будущее – новые разработки, проекты, статьи?**

Изучаю вопросы колоризации – автоматического перевода чёрно-белых фотографий в цветные.

Это практически 3D-уравнение с 3-мя неизвестными.

Для интересующихся – несколько ссылок:

<http://demos.algorithmia.com/colorize-photos/>  
<http://richzhang.github.io/colorization/>  
<http://tinyclouds.org/colorize/>

Пишу универсальный аналог оболочки/проводника explorer.exe (интерфейс проводника Windows NT) – будет работать на всём, начиная с Win32s/Win95/98/NT/2k/XP и далее везде. Да! Win32s/NT3.51 поддерживает элементы управления/интерфейса, использовавшиеся в Windows 95/NT4.

Отлаживаю утилиту, вычищающую префиксы с LOCK на NOP из системных файлов Windows NT (kernel32, win32k, ntdll). Для

мультипроцессорных/мультиядерных систем они нужны, а для обычных – только лишние поддержки для CPU.

**Каковы Ваши пристрастия в пёстром семействе x86-го софта, железа?**

*Из железа:*

Процессоры INTEL.

Материнские десктопные платы INTEL (ну очень жаль, что INTEL, начиная с 9-й серии, ушла с рынка).

Видеокарты ATI/AMD (это скорее дань уважения длинной истории бренда).

CRT-мониторы.

HDD Seagate.

Принтеры HP (за определённую совместимость внутри большого семейства, да и просто HP-шек много есть в природе).

IBM Lenovo/ThinkPad (строгий деловой дизайн, удобная клавиатура, ориентация на OS/2 – у ранних моделей).

ASUS EEE PC – всё хорошо, кроме неудобной клавиатуры.

Категорический противник всякого разгона – всё должно работать штатно. Это залог «долголетия» железа.

*Из софта:*

Не люблю Google Chrome, Office 2007+, Windows Vista/7 и далее, но вынужденно пользуюсь, как многие. :(

Люблю Opera Presto, Seamonkey.

Когда программирую и нужна справка по WIN API, пользуюсь MSDN October 2001 (\*.chm) или справкой из MS VC 4.0 (\*.hlp). Мне не нравится нынешний help на www.microsoft.com и т.н. новый MSDN Help.

**Коллекционируете ли Вы что-нибудь?**

Собираю рабочие видеокарты.

Из того, что навскидку ищущу:

Cirrus Logic 5428/5429 ISA;



Cirrus Logic 5446 2 MB PCI;  
Cirrus Logic 5480 PCI;



Matrox-ы G200 и новее на AGP/PCI;  
Tseng ET6000/6100 PCI;  
NVidia до Geforce3 VGA+DVI AGP/PCI (рабочий VESA 3.0 для опытов!).

Редкости вроде – Alliance ProMotion, ARK Logic, Avance Logic, Chips & Technologies, InteGraphics, iXMICRO, Number Nine, NeoMagic, OAK, SiS и пр.

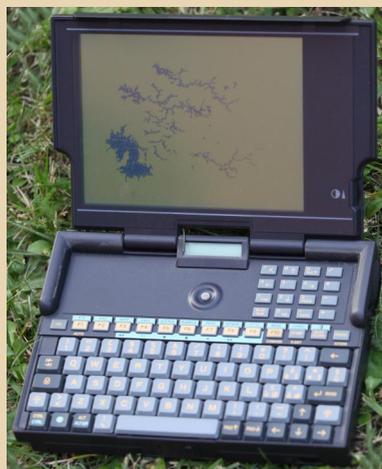
**Что бы Вы хотели пожелать нашим читателям?**

Если вы доунгрейдер – не бросайте это дело! Современное железо не так интересно ковырять, оно менее надёжно, не очень ремонтпригодно. К качественно сделанным «брендовым» вещам привыкаешь (это касается не только «железок»). Безликие современные железки (не все, конечно), доживающие до своего срока гарантии и приходящие в негодность – это как-то грустно и скучно.

Также я считал и считаю, что к каждому компьютеру можно подобрать соответствующий аутентичный софт и работать на нём более-менее комфортно – будь это 80386-я машина или Pentium 2, например. Иногда попадаются просто интересные экземпляры с удобной компоновкой, запоминающимся дизайном. Которые просто приятно держать в руках.

Например, вот:

[https://it.wikipedia.org/wiki/Olivetti\\_Quaderno](https://it.wikipedia.org/wiki/Olivetti_Quaderno)



**P.S.**

У моего сайта есть множество зеркал. По независящим от меня причинам не все из них обновляются.

Будьте внимательны.

На данный момент – лето 2017 г. – актуальны эти два:

<http://bearwindows.zcm.com.au/>  
<http://www.navozhdeniye.narod.ru/>

**Удачи и большое спасибо за интервью!**

*Прим. ред.: в статье использовались изображения из следующих источников:*  
[speccy.info](http://speccy.info), [www.interface1.net](http://www.interface1.net), [retrotexnika.ru](http://retrotexnika.ru),  
[www.rw6ase.narod.ru](http://www.rw6ase.narod.ru), [shizaudio.ru](http://shizaudio.ru), [ixbt.com](http://ixbt.com),  
[www.charlespetzold.com](http://www.charlespetzold.com), [www.servercase.co.uk](http://www.servercase.co.uk),  
[hwmuseum.pp.ua](http://hwmuseum.pp.ua), [vgamuseum.info](http://vgamuseum.info), [wikipedia.org](http://wikipedia.org)

**bearwindows**  
**Вячеслав Рытиков (eu6pc)**  
**uav1606**



## MicroCAP V – ВАШ ВЫБОР ДЛЯ ВИРТУАЛЬНОЙ ЛАБОРАТОРИИ



Книжная  
полка



восьмидесятых годов журнал «Радио» стал публиковать статьи, так или иначе относящиеся к компьютерной тематике. Сперва были статьи, посвящённые микропроцессорному комплекту K580 и компьютеру «Микро-80» на его основе, в дальнейшем описывались компьютеры, разработанные под эгидой журнала – РК86 и «Орион-128». Однако во второй половине 90-х и начале 2000-х, когда стало понятно, что любительские компьютеры безнадежно устарели, начинают активно появляться статьи, посвящённые современным компьютерам, совместимым с IBM PC, и прикладному программному обеспечению для них. Среди упоминаемых в статьях программ были САПР для разработки печатных плат – в первую очередь P-CAD и ACCEL EDA, чуть реже – OrCAD, среды программирования для микроконтроллеров – в первую очередь MPLAB и MPASM для PIC, а также программы моделирования электронных схем. Одной из таких программ – Micro-Cap V 2.0 – был посвящён цикл статей И. Григорьева «Компьютер в домашней радиолaborатории».

Статьи цикла можно почитать по следующим ссылкам:

[http://www.chipinfo.ru/literature/radio/199905/p47\\_49.html](http://www.chipinfo.ru/literature/radio/199905/p47_49.html) – первая статья цикла («Радио» №5 1999) – краткое введение, в котором рассказывается, как можно использовать компьютер в домашней радиолaborатории и описываются некоторые программы, в частности, комбинированный прибор – генератор звуковых частот и низкочастотный осциллограф AudioTester.

[http://www.chipinfo.ru/literature/radio/199906/p56\\_57.html](http://www.chipinfo.ru/literature/radio/199906/p56_57.html) – вторая статья цикла («Радио» №6 1999) – начало работы с Micro-Cap V 2.01 – описывается, где скачать программу, особенности интерфейса, панели инструментов, а также приводится простая модель генератора на транзисторе.

[http://www.chipinfo.ru/literature/radio/199907/p51\\_53.html](http://www.chipinfo.ru/literature/radio/199907/p51_53.html) – третья статья цикла («Радио»

№7 1999) – продолжение экспериментов с генератором в различных температурных условиях, а также модель на полевом транзисторе.

[http://www.chipinfo.ru/literature/radio/199908/p66\\_68.html](http://www.chipinfo.ru/literature/radio/199908/p66_68.html) – четвёртая статья цикла («Радио» №8 1999) – описание настройки генератора синусоидальных колебаний, эксперименты с LC-фильтром, исследование АЧХ фильтра.

<http://www.chipinfo.ru/literature/radio/199910/p55.html> – пятая статья цикла («Радио» №10 1999) – создание собственных библиотечных компонентов и макромоделей, включающих в себя узлы принципиальной схемы.

[http://www.chipinfo.ru/literature/radio/199911/p55\\_56.html](http://www.chipinfo.ru/literature/radio/199911/p55_56.html) – шестая – заключительная – статья цикла («Радио» №11 1999) – добавление новых моделей в Spice-формате, моделирование цифровых схем.

Собственно, выбор программы для описания её в цикле статей, как пишет автор, обусловлен наличием в интернете вполне функциональной, хоть и названной демонстрационной, версии программы, которая имеет, разве что, ограничения по количеству элементов в моделируемой схеме. Однако сейчас даунгрейдера могут подстергать определённые подводные камни, о которых трудно было догадаться или которых вообще не могло быть в 1999-м году.

Во-первых, конечно же, скачивание программы из интернета с сайта производителя. Во второй статье даётся только ссылка на сайт, но не на файл демонстрационной версии программы. Воспользовавшись веб-архивом, вы, скорее всего, найдёте форму для получения демонстрационной версии. У меня она не заработала. © И только в последней статье цикла даётся прямая ссылка на демо-версию: [www.spectrum-soft.com/down/demo.zip](http://www.spectrum-soft.com/down/demo.zip) (кстати, по ней до сих пор можно скачать урезанную демонстрашку последней версии Micro-Cap). Версия, которая описывалась в цикле статей, может быть скачана с помощью веб-архива, а сейчас доступна и на old-dos: <http://old-dos.ru/dl.php?id=16561>.



Во-вторых – в пути к каталогу установщика не должно быть русскоязычных названий, да и вообще, лучше закинуть установщик в папку в корне диска – я закинул в папку **C:\MC52** – программа уверенно поставилась под Windows XP.

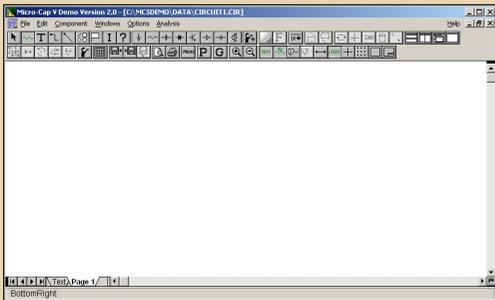


Рис. 1. Micro-Cap V 2.0, установленная и запущенная под Windows XP

В-третьих – программе нужен математический сопроцессор. Да, программа требует 486-ой процессор и 16 МБ оперативной памяти, но процессор должен быть именно DX. На моём У5S инсталлятор сразу заявил, что без сопроцессора ставиться не желает.

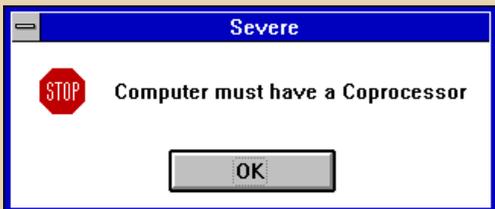


Рис. 2. Попытка установить программу на старом компьютере – нет сопроцессора – не будет и программы

Казалось бы, есть выход – Micro-Cap V 1.0 <http://old-dos.ru/dl.php?id=16552> – но и она при попытке запустить моделирование схемы вывела сообщение о переполнении при операции с плавающей точкой. А при моделировании схем на операционном усилителе и даже описанного в цикле генератора – жаловалась на неустановку различных параметров моделирования.

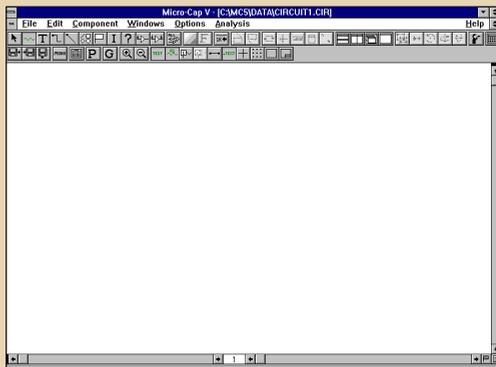


Рис. 3. Micro-Cap V 1.0 под Windows 3.11 – загружается и работает достаточно шустро, особенно в сравнении с загрузкой Electronics WorkBench

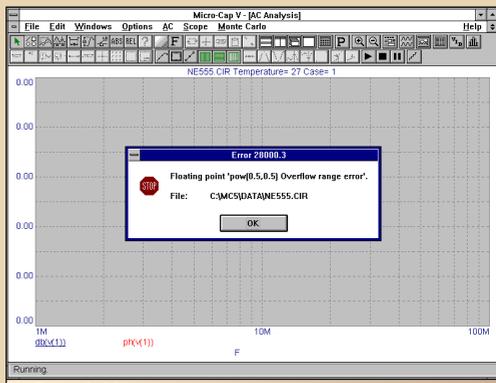


Рис. 4. Увы, при моделировании возникает ошибка переполнения

Подводя итог, можно сказать, что Micro-Cap – не та программа, которую можно освоить «с наскока» – по крайней мере, её пятые версии. В программе достаточно много хитрых настроек, например, у того же синусоидального генератора, доступ к которым ещё надо догадаться получить. Для сравнения, в Electronics Workbench генератор и осциллограф можно настроить, просто нажимая виртуальные кнопки в диалоговых окнах. Однако достаточно высокая скорость работы на старых системах, достаточно широкие возможности и симпатичный интерфейс, а также наличие такого цикла статей послужат неплохим стимулом для освоения программы.



# ОБЗОР СКАНЕРА HP SCANJET IIр (C1790A)



Недавно удалось мне купить по дешёвке – всего 300 рублей – такой вот сканер – HP ScanJet IIр (он же C1790A). Точный год выпуска установить не удалось, но, видимо, приблизительно это где-то 1994-й (или даже раньше – так как представлен он был ещё в 1992-м). Разрешение 300 dpi, 256 оттенков серого.

Общий вид сканера:



Сверху лежит кабель SCSI, что шёл в комплекте – капитальный, надо сказать – толщиной в мой мизинец. Также прилагался (справа) терминатор. Ну и сзади виден лист краткой инструкции по разблокировки сканера после транспортировки.

Шильдик:



В открытом виде:



Никаких органов управления на нём, собственно, нет – только выключатель питания сбоку да светодиод, показывающий наличие питания. А, да, ещё есть «замок», который блокирует, как я понял, механизм сканера на время транспортировки.

Сканер имеет интерфейс SCSI. Вот вид задней панели:



Видно, что верхняя крышка умеет приподниматься на своеобразных полозьях – очень



удобно, если нужно отсканировать толстую книгу. В современных сканерах такого часто не хватает.

Виден обычный разъём для питания (220 В – провод, как на всех современных компьютерах и мониторах) и два интерфейсных – оба SCSI в разном исполнении.

Для меня не очень удачно – LPT, например, был бы удобнее. Ну да ладно, попробуем его всё-таки запустить.

Проблема в том, что SCSI-адаптеры у меня есть только ISA и VLB – к современному компьютеру не подключишь. Под PCI всё никак не найду.

Итак, на свет был извлечён старый системник – Celeron 630 MHz/64 MB RAM/6 GB HDD. Установлена на нём Windows 98. И, что самое главное, есть два ISA-разъёма.

Втыкаем в ISA SCSI-адаптер Adaptec ANA-1520/22:



Windows 98 с первого раза его не увидела. Однако после запуска мастера поиска нового оборудования всё-таки обнаружила – адаптер то не PnP, так что ничего удивительного. Драйвера нашлись в самом дистрибутиве Windows. Сканер, после подключения, обнаружился сам, драйвера тоже нашлись стандартные – очень удачно.

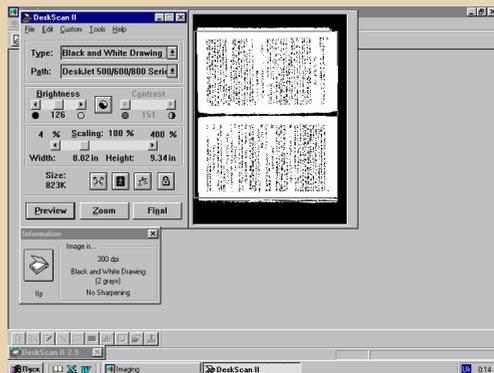
Но я всё-таки скачал родные, т.к. слишком уж бедны настройками те, что были в Windows.

Качал отсюда:

<http://www.driverguide.com/driver/detail.php?driverid=13087>

Файл **1sj166n.exe** размером в 4,7 МБ. Распаковываем, запускаем **setup.exe** из папки **DISK1**. Дальше пойдут стандартные вопросы установки, перезагружаемся – всё, теперь у нас новый интерфейс сканирования.

Для проверки сканера я воспользовался Kodak Imaging из стандартной поставки Windows 98. Жмём иконку со сканером, автоматически запустится предварительное сканирование. Не очень удобно, но это можно отключить в настройках. После чего появится примерно такой диалог:



Немного запутано, конечно, но есть справка.

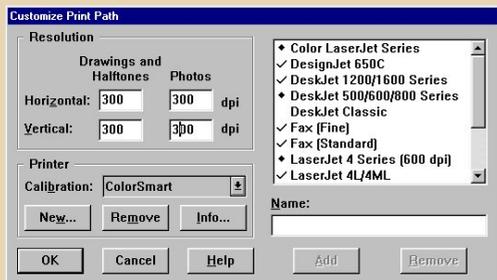
Справа окно с предварительным сканом, там можно задать область сканирования.

В меню **Type** можно выбрать предустановки – для сканированного чертежа/рисунка (**Drawing**) или фотографии (**Photo**). В первом случае получим только однобитный цвет, во втором – 256 оттенков серого. Разрешение сканирования зависит от выбора в меню **Path** – вообще, там указывается принтер, на котором мы, предположительно, собираемся печатать отсканированное изображение. Вариант, что нам нужна электронная копия, похоже, не предусмотрен. Точнее, есть вариант **Screen**, но при его выборе устанавливается разрешение всего в 75 dpi, так что это скорее предположение.

Не беда, мы можем вручную указать разрешение, для этого выберем **Custom->Print Path**. А там уже выставим для **Photos** и гори-



зонтальное, и вертикальное разрешение в 300, жмём **OK** – всё.



Правда, это сохранится только на текущий сеанс, но можно (кнопка **Add** в том же окне – см. выше) создать и собственную предустанов-

ку (собственный «принтер»), чтобы пользоваться ей постоянно.

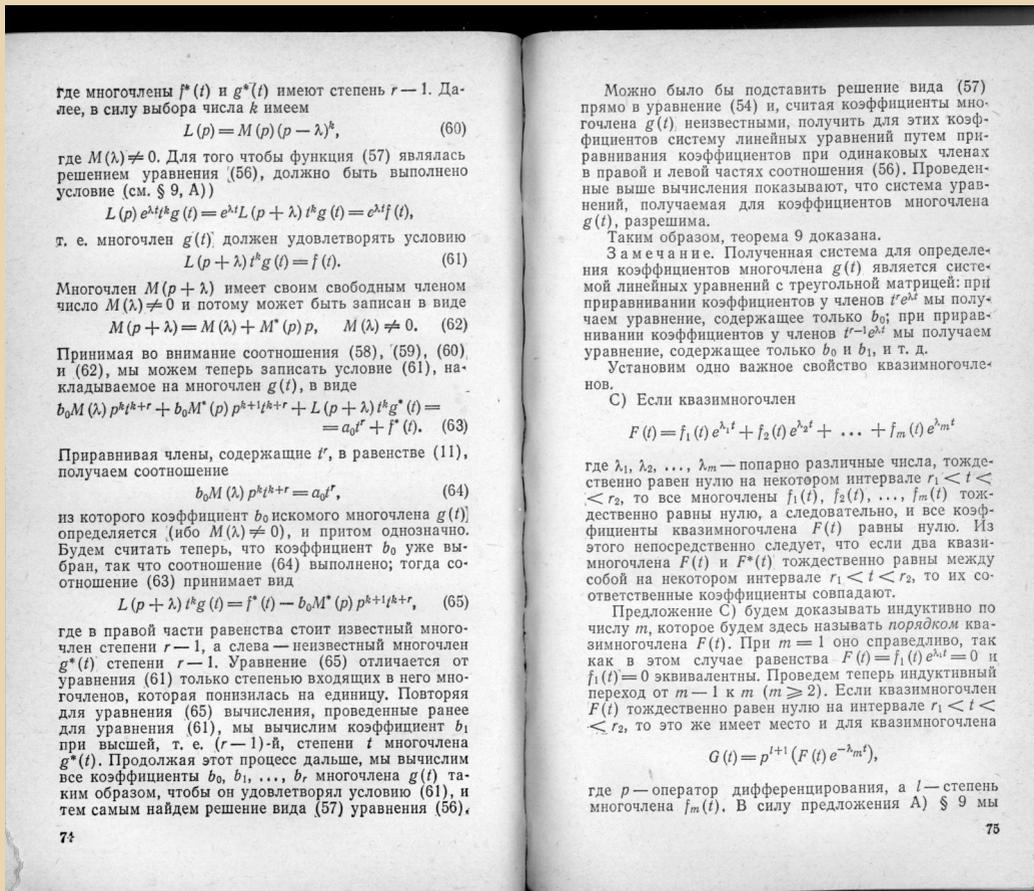
В этом же основном диалоге можно настроить яркость и контраст, увеличение, «отзеркалить» изображение, сделать из него негатив, а также изменить ширину и высоту.

Настроив всё, что надо, жмём **Final**, запускается «окончательное» сканирование. Кстати, неприятная особенность – компьютер фактически зависает на время сканирования – мышка не двигается и т.п.

Что же в итоге у нас получилось?

Ниже пример отсканированного разворота из книги, а на следующей странице – он же, но отсканированный современным USB-сканером Mustek BearPaw 1200CU Plus II.

Результат HP ScanJet IIp:



где многочлены  $f^*(t)$  и  $g^*(t)$  имеют степень  $r-1$ . Далее, в силу выбора числа  $k$  имеем

$$L(p) = M(p)(p - \lambda)^k, \quad (60)$$

где  $M(\lambda) \neq 0$ . Для того чтобы функция (57) являлась решением уравнения (56), должно быть выполнено условие (см. § 9, А))

$$L(p) e^{\lambda t} g(t) = e^{\lambda t} L(p + \lambda) t^k g(t) = e^{\lambda t} f(t),$$

т. е. многочлен  $g(t)$  должен удовлетворять условию

$$L(p + \lambda) t^k g(t) = f(t). \quad (61)$$

Многочлен  $M(p + \lambda)$  имеет своим свободным членом число  $M(\lambda) \neq 0$  и потому может быть записан в виде

$$M(p + \lambda) = M(\lambda) + M^*(p), \quad M(\lambda) \neq 0. \quad (62)$$

Принимая во внимание соотношения (58), (59), (60), и (62), мы можем теперь записать условие (61), накладываемое на многочлен  $g(t)$ , в виде

$$b_0 M(\lambda) p^{k+t+r} + b_0 M^*(p) p^{k+1+t+r} + L(p + \lambda) t^k g^*(t) = a_0 t^r + f^*(t). \quad (63)$$

Приравняв члены, содержащие  $t^r$ , в равенстве (11), получаем соотношение

$$b_0 M(\lambda) p^{k+t+r} = a_0 t^r, \quad (64)$$

из которого коэффициент  $b_0$  искомого многочлена  $g(t)$  определяется (ибо  $M(\lambda) \neq 0$ ), и притом однозначно. Будем считать теперь, что коэффициент  $b_0$  уже выбран, так что соотношение (64) выполнено; тогда соотношение (63) принимает вид

$$L(p + \lambda) t^k g^*(t) = f^*(t) - b_0 M^*(p) p^{k+1+t+r}, \quad (65)$$

где в правой части равенства стоит известный многочлен степени  $r-1$ , а слева – неизвестный многочлен  $g^*(t)$  степени  $r-1$ . Уравнение (65) отличается от уравнения (61) только степенью входящих в него многочленов, которая понизилась на единицу. Повторяя для уравнения (65) вычисления, проведенные ранее для уравнения (61), мы вычислим коэффициент  $b_1$  при высшей, т. е.  $(r-1)$ -й, степени  $t$  многочлена  $g^*(t)$ . Продолжая этот процесс дальше, мы вычислим все коэффициенты  $b_0, b_1, \dots, b_r$  многочлена  $g(t)$  таким образом, чтобы он удовлетворял условию (61), и тем самым найдем решение вида (57) уравнения (56).

Можно было бы подставить решение вида (57) прямо в уравнение (54) и, считая коэффициенты многочлена  $g(t)$  неизвестными, получить для этих коэффициентов систему линейных уравнений путем приравнивания коэффициентов при одинаковых членах в правой и левой частях соотношения (56). Проведенные выше вычисления показывают, что система уравнений, получаемая для коэффициентов многочлена  $g(t)$ , разрешима.

Таким образом, теорема 9 доказана.

З а м е ч а н и е. Полученная система для определения коэффициентов многочлена  $g(t)$  является системой линейных уравнений с треугольной матрицей: при приравнивании коэффициентов у членов  $t^r e^{\lambda t}$  мы получаем уравнение, содержащее только  $b_0$ ; при приравнивании коэффициентов у членов  $t^{r-1} e^{\lambda t}$  мы получаем уравнение, содержащее только  $b_0$  и  $b_1$ , и т. д.

Установим одно важное свойство квазимногочленов.

С) Если квазимногочлен

$$F(t) = f_1(t) e^{\lambda_1 t} + f_2(t) e^{\lambda_2 t} + \dots + f_m(t) e^{\lambda_m t}$$

где  $\lambda_1, \lambda_2, \dots, \lambda_m$  – попарно различные числа, тождественно равен нулю на некотором интервале  $r_1 < t < r_2$ , то все многочлены  $f_1(t), f_2(t), \dots, f_m(t)$  тождественно равны нулю, а следовательно, и все коэффициенты квазимногочлена  $F(t)$  равны нулю. Из этого непосредственно следует, что если два квазимногочлена  $F(t)$  и  $F^*(t)$  тождественно равны между собой на некотором интервале  $r_1 < t < r_2$ , то их соответственные коэффициенты совпадают.

Предложение С) будем доказывать индуктивно по числу  $m$ , которое будем здесь называть *порядком* квазимногочлена  $F(t)$ . При  $m = 1$  оно справедливо, так как в этом случае равенства  $F(t) = f_1(t) e^{\lambda_1 t} = 0$  и  $f_1(t) = 0$  эквивалентны. Проведем теперь индуктивный переход от  $m-1$  к  $m$  ( $m \geq 2$ ). Если квазимногочлен  $F(t)$  тождественно равен нулю на интервале  $r_1 < t < r_2$ , то это же имеет место и для квазимногочлена

$$G(t) = p^{l+1}(F(t) e^{-\lambda_m t}),$$

где  $p$  – оператор дифференцирования, а  $l$  – степень многочлена  $f_m(t)$ . В силу предложения А) § 9 мы

Результат Mustek BearPaw 1200CU Plus II:

где многочлены  $f^*(t)$  и  $g^*(t)$  имеют степень  $r-1$ . Далее, в силу выбора числа  $k$  имеем

$$L(p) = M(p)(p - \lambda)^k, \quad (60)$$

где  $M(\lambda) \neq 0$ . Для того чтобы функция (57) являлась решением уравнения (56), должно быть выполнено условие (см. § 9, А)

$$L(p) e^{\lambda t} g(t) = e^{\lambda t} L(p + \lambda) t^k g(t) = e^{\lambda t} f(t),$$

т. е. многочлен  $g(t)$  должен удовлетворять условию

$$L(p + \lambda) t^k g(t) = f(t). \quad (61)$$

Многочлен  $M(p + \lambda)$  имеет своим свободным членом число  $M(\lambda) \neq 0$  и потому может быть записан в виде

$$M(p + \lambda) = M(\lambda) + M^*(p)p, \quad M(\lambda) \neq 0. \quad (62)$$

Принимая во внимание соотношения (58), (59), (60), и (62), мы можем теперь записать условие (61), накладываемое на многочлен  $g(t)$ , в виде

$$b_0 M(\lambda) p^{k_1} e^{k_1 r} + b_0 M^*(p) p^{k_1 + i k_1 r} + L(p + \lambda) t^k g^*(t) = a_0 e^{r t} + f^*(t). \quad (63)$$

Приравнявая члены, содержащие  $t^r$ , в равенстве (11), получаем соотношение

$$b_0 M(\lambda) p^{k_1} e^{k_1 r} = a_0 e^{r t}, \quad (64)$$

из которого коэффициент  $b_0$  искомого многочлена  $g(t)$  определяется (ибо  $M(\lambda) \neq 0$ ), и притом однозначно. Будем считать теперь, что коэффициент  $b_0$  уже выбран, так что соотношение (64) выполнено; тогда соотношение (63) принимает вид

$$L(p + \lambda) t^k g^*(t) = f^*(t) - b_0 M^*(p) p^{k_1 + i k_1 r}, \quad (65)$$

где в правой части равенства стоит известный многочлен степени  $r-1$ , а слева — неизвестный многочлен  $g^*(t)$  степени  $r-1$ . Уравнение (65) отличается от уравнения (61) только степенью входящих в него многочленов, которая понизилась на единицу. Повторяя для уравнения (65) вычисления, проведенные ранее для уравнения (61), мы вычислим коэффициент  $b_1$  при высшей, т. е.  $(r-1)$ -й, степени  $t$  многочлена  $g^*(t)$ . Продолжая этот процесс дальше, мы вычислим все коэффициенты  $b_0, b_1, \dots, b_r$  многочлена  $g(t)$  таким образом, чтобы он удовлетворял условию (61), и тем самым найдем решение вида (57) уравнения (56).

Можно было бы подставить решение вида (57) прямо в уравнение (54) и, считая коэффициенты многочлена  $g(t)$  неизвестными, получить для этих коэффициентов систему линейных уравнений путем приравнивания коэффициентов при одинаковых членах в правой и левой частях соотношения (56). Проведенные выше вычисления показывают, что система уравнений, получаемая для коэффициентов многочлена  $g(t)$ , разрешима.

Таким образом, теорема 9 доказана.

З а м е ч а н и е. Полученная система для определения коэффициентов многочлена  $g(t)$  является системой линейных уравнений с треугольной матрицей: при приравнивании коэффициентов у членов  $t^r e^{\lambda t}$  мы получаем уравнение, содержащее только  $b_0$ ; при приравнивании коэффициентов у членов  $t^{r-1} e^{\lambda t}$  мы получаем уравнение, содержащее только  $b_0$  и  $b_1$ , и т. д.

Установим одно важное свойство квазимногочлена.

С) Если квазимногочлен

$$F(t) = f_1(t) e^{\lambda_1 t} + f_2(t) e^{\lambda_2 t} + \dots + f_m(t) e^{\lambda_m t}$$

где  $\lambda_1, \lambda_2, \dots, \lambda_m$  — попарно различные числа, тождественно равен нулю на некотором интервале  $r_1 < t < r_2$ , то все многочлены  $f_1(t), f_2(t), \dots, f_m(t)$  тождественно равны нулю, а следовательно, и все коэффициенты квазимногочлена  $F(t)$  равны нулю. Из этого непосредственно следует, что если два квазимногочлена  $F(t)$  и  $F^*(t)$  тождественно равны между собой на некотором интервале  $r_1 < t < r_2$ , то их собственные коэффициенты совпадают.

Предложение С) будем доказывать индуктивно по числу  $m$ , которое будем здесь называть порядком квазимногочлена  $F(t)$ . При  $m = 1$  оно справедливо, так как в этом случае равенства  $F(t) = f_1(t) e^{\lambda_1 t} = 0$  и  $f_1(t) = 0$  эквивалентны. Проведем теперь индуктивный переход от  $m-1$  к  $m$  ( $m \geq 2$ ). Если квазимногочлен  $F(t)$  тождественно равен нулю на интервале  $r_1 < t < r_2$ , то это же имеет место и для квазимногочлена

$$G(t) = p^{t+1}(F(t) e^{-\lambda_m t}),$$

где  $p$  — оператор дифференцирования, а  $t$  — степень многочлена  $f_m(t)$ . В силу предложения А) § 9 мы

Очень даже неплохо!

Сканирование в разрешении 300 dpi при 256 оттенках серого заняло где-то 35 секунд. Если сравнивать с Mustek'ом — качество кажется даже лучше. Кроме того, Mustek сканировал разворот книги при таком же разрешении (300 dpi) и тоже в оттенках серого целых 60 секунд! И из-за неудачной конструкции крышки и плохой глубины резкости изображение около корешка получилось размытым. Даже для этого результата на крышку Mustek'a пришлось сверху ещё положить груз из пары книг, иначе картинка вообще получалась мутная — из-за плохого прижима.

Что можно сказать, подводя итоги? HP ScanJet IIр очень неплохо справляется с задачей сканирования чёрно-белых документов — быстро и вполне качественно. Кроме того, мне его конструкция (особенно крышка) показалась более удобной и эргономичной, чем у того же Mustek BearPaw.

Если достану PCI SCSI-контроллер, то буду и в дальнейшем пользоваться ScanJet'ом для сканирования книг и документов.



## МОЯ ИДЕАЛЬНАЯ DOS-МАШИНА



ожалуй, чаще всего мы обращаемся к старым компьютерам, когда нам хочется поиграть в старые игры под DOS. Если это желание мимолётно, то самый очевидный и оптимальный вариант – это эмуляторы. И DOSBox тут первый в списке.

Однако по тропам ностальгии можно зайти достаточно далеко, и часто бывает, что помимо старого софта, хочется ещё и старого железа и настоящего DOSa. Радикальный подход предполагает сборку настольного ПК из оригинальных деталей соответствующей эпохи. Из плюсов тут – богатейший выбор устройств и аутентичность. А сложность состоит в поиске большого количества деталей, а также в громоздкости системного блока и монитора. И стоимость тоже может неприятно удивить. Не все готовы долго и нудно собирать компьютер и выделять под него место.

Альтернативой является покупка винтажного ноутбука. Но и тут мы сталкиваемся с ря-

дом компромиссов. Часто в ноутбуках встречаются одна (или несколько сразу) из следующих проблем: низкая производительность процессора и графической системы, отсутствие Sound Blaster-совместимой звуковой карты, отсутствие оптического привода, маленький экран, к тому же изображение может не растягиваться на весь экран, отсутствие тех или иных портов.

Например, мой ноутбук на Pentium 233 MMX – всем хорош, но изображение не растягивается на весь экран и игры отображаются в окружении траурной рамки, ну и хотелось бы производительность побольше в таких играх, как QUAKE или Duke Nukem. Есть у меня ещё Toshiba s1900 на P-4: изображение растягивается и производительности хватает, однако встроенный звук уже не работает под DOS... Все попытки как-то решить проблему со звуком ни к чему не привели.

Есть у меня ещё одна машина, которая, как выяснилось, лишена абсолютно всех вышеперечисленных недостатков. Это IBM Thinkpad A21p (модель 2001 г. – Intel Pentium III 850 МГц / 128 МБ RAM / 32 ГБ HDD / DVD-ROM / ATI Rage Mobility 128, 16 МБ VRAM / дисплей 15", 1600x1200).

Конечно, нельзя сказать, что аппаратные средства A21p соответствуют эпохе MS-DOS, но, тем не менее, машина предоставляет возможность полноценной работы в чистом MS-DOS (6.22, 7.1), со всеми её особенностями настройки конфигурационных файлов и прочего, что является неотъемлемой частью процесса.

Этот роскошный аппарат создавался как полнофункциональная альтернатива настольным компьютерам с богатыми мультимедийными возможностями и с упором на работу с графикой и видео. Продавался он по цене 3500-4000 долларов и представляет из себя небольшой шедевр технической мысли. Чего стоит один лишь теплоотвод из-под крышки матрицы, который ведёт к теплоприёмнику



центрального процессора. Для обзора всех его возможностей и особенностей потребовалась бы отдельная статья.



Но сегодня, так уж получается, что он оказался для меня идеальной DOS-машиной.

Во-первых, это прекрасный 15-дюймовый экран с разрешением 1600\*1200 (sic!), который безупречно масштабирует изображение с разрешением 320\*240.

Другим ключевым моментом является то, что ноутбук оснащен звуковым чипом CS4624, который имеет совместимость с Sound Blaster и работает в чистом MS-DOS. Более поздние чипы также заявляют SB-совместимость, однако работают уже только под Windows (как, например, CS4299).

Из других плюсов – наличие встроенного 3,5" FDD и DVD-ROM Panasonic SR-8175, для которого есть драйверы для работы в DOS. Т.е. установка любой операционной системы и переброска данных не вызовет никаких сложностей.

Бонусом идёт возможность работы с USB и PCMCIA при наличии соответствующих драйверов. Встроенные динамики, разъём PS/2 для мыши (хотя и TrackPoint отлично работает), COM и LPT довершают эту радужную картину.

Ну и, естественно, производительности хватает на любые игры. Quake начинает притормаживать только на разрешении 1024\*768, Redneck Rampage идёт в разрешении 1600\*1200!



Свой экземпляр я приобрёл за 2500 руб. Но у данной модели есть один неприятный недостаток – многие экземпляры страдают так называемым Blink of Death (BOD) – внезапное отключение под нагрузкой. Проблема тут в цепи питания. Мой экземпляр периодически отключался, когда работал под WinXP, но под DOS пока проблем не было. Работает машинка пока от сети, но никто не мешает раскошелиться на восстановление батареи и испытать все радости мобильности.

Роман Азарин





# STUNNEL – ИНТЕРНЕТ С SSL ДЛЯ СТАРЫХ ПРОГРАММ



современный интернет, а особенно Web 2.0 – бич старых машин. Казалось бы, не так давно пользователь старого компьютера мог подключиться через модем и, сидя под Windows 95, а уж тем более на Windows 98 бодро бороздить просторы всемирной паутины. Браузер Microsoft Internet Explorer не казался таким убожеством, даже если учесть, что сильно отставал в возможностях от своих конкурентов – Netscape Navigator, Mozilla и Opera, но для многих был достаточно удобным окном в мир.

Программы-оболочки NetCaptor и MyIE2 расширяли функциональные возможности «ослика», добавляя работу с вкладками и сохраняя интеграцию с другими полезными программами, которые не поддерживали конкурентов браузера от Microsoft – переводчики, менеджеры закачки, заполнители форм на сайтах и т.д.

Но времена меняются: Internet Explorer после «модернизации» стал жутко тормозным, использовать шестую версию тоже стало практически невозможно – времена, когда шестой «осёл» мог переварить и форум, и сайт с музыкой, и виртуальный клуб знакомств ушли, а ведь когда-то можно было внаглую скачать нужную композицию даже на 486-ом компьютере с Windows 3.11, и это были не далекие 90-е, а 2006-ой год.

Однако если времена требуют от даунгрейдера отказаться от старой версии браузера и хотя бы периодически посматривать в сторону обновлений, например, Firefox, которые хоть немного, но щадят оперативную память, то есть классы программ, где здоровый (да и не очень здоровый ☺) консерватизм совершенно не вредит делу. Что это за программы, спросит читатель, и тут же даст ответ, назвав,

возможно, в первую голову FTP, IRC и новостные клиенты. Всё? Нет – не всё – электронную почту забыли! Да, господа, электронная почта практически не меняется десятилетиями, и почтовый клиент, который был установлен когда-то студентом или старшеклассником, который только начал приобщаться к интернету, вполне может нормально работать и сейчас. Автор до сих пор сидит за первыми версиями The Bat! – почтовик умеет практически всё, что требуется от него сегодня – принимать письма в различных кодировках, просматривать встроенным вьюером HTML-письма (по крайней мере, версии 1.53+ это могут точно), авторизовываться при отправке писем, предъявляя SMTP-серверу логин и пароль, которые могут не совпадать с логином и паролем для сервера POP3. Про сортировку, поиск, трёхпанельный интерфейс молчу. И всё бы прекрасно, но последнее время почтовые сервера начали снова чудить – вводить шифрование трафика через SSL. Формально, если в настройках программы выбрать метод шифрования TLS, как предлагает инструкция на сайте, и указать соответствующие новые порты для сервера SMTP и POP3 (см. рис. 1) – всё должно заработать.

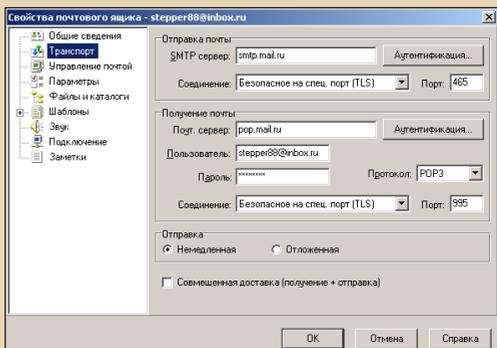


Рис. 1. Вот такие настройки предлагает сделать Mail.ru, обещая, что ваш почтовый клиент после этого продолжит работать нормально



На деле же при попытке получить почту с новыми настройками лично я в третьей версии получил сообщение об ошибке сертификата. Первая же версия вообще реагировала открытым непониманием (см. рис. 2).

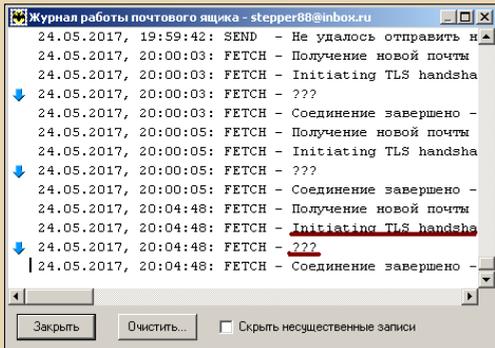


Рис. 2. Программа даже не поняла, что ей собственно ответили при соединении – подчёркнул

Чтение форумов показало, что ошибка устраняется практически сама собой – нужно лишь только поставить версию 4.x+. Но, самое интересное, что у The Bat! как раз есть средства для решения и такой проблемы. Если почтовый сервер может предоставить сертификат, его можно установить в программу, и дело решится – [http://www.donbass.net/bat\\_tls.html](http://www.donbass.net/bat_tls.html). Однако найти сертификат Mail.ru оказалось достаточно сложно. Ссылка [https://secure.mail.ru/MailRU\\_CA.crt](https://secure.mail.ru/MailRU_CA.crt) не действует. А файл, скачанный по ссылке [http://artax.karlin.mff.cuni.cz/~gurss5am/certif/MailRU\\_CA.crt](http://artax.karlin.mff.cuni.cz/~gurss5am/certif/MailRU_CA.crt), не решил проблему. Возможно, конечно, стоило скачать и установить все сертификаты, которые были в директории certif по последней ссылке, но не думаю, что это бы решило проблему.

Другим решением стала установка программы Stunnel, видимо, специально разработанной, чтобы облегчить жизнь фанатам старого софта в мире сертификатов и шифрования трафика.

Автором используется программа версии 4.56, которую можно взять по адресу: <ftp://ftp.stunnel.org/stunnel/archive/4.x/stunnel-4.56-installer.exe>

После установки программа образует свою группу в списке программ. Однако она ещё не готова работе. Необходимо сконфигурировать программу для работы с конкретными серверами и протоколами. Это делается путём правки файла **stunnel.conf**. Запустить его редактирование можно кликом по ярлыку «**Edit stunnel.conf**». В этом файле необходимо раскомментировать настройки нужных протоколов и вписать адреса серверов – глобальные и локальные, а также порты, по которым stunnel будет обращаться к внешним серверам, а также порты для локальных программ. Предполагается, что stunnel будет служить локальным сервером для программ-клиентов, располагающимся по адресу **127.0.0.1**, а получив от клиентов запрос, будет транслировать его уже внешним серверам по защищённому каналом.

Для сервера inbox.ru мне понадобилось изменить следующие настройки:

```
[pop3s]
accept = 995
connect = 110
```

```
[imaps]
accept = 993
connect = 143
```

```
[ssmtp]
accept = 465
connect = 25
```

Эти вроде бы и были такими – менять их не было необходимости. А вот следующие необходимо было раскомментировать и прописать адреса серверов:



```
[mailru-pop3]
client = yes
accept = 127.0.0.1:110
connect = pop.inbox.ru:995
```

```
[mailru-smtp]
client = yes
accept = 127.0.0.1:25
connect = smtp.inbox.ru:465
```

После сохранения настроек и, если нужно, перезапуска программы, сервер-конвертер (или его можно назвать прокси-сервером?) готов к работе. Запуск программы осуществляется ярлыком **stunnel GUI start** – в области уведомлений появится значок программы. Для того чтобы программа запускалась автоматически при старте компьютера, добавьте ярлык в автозагрузку – при установке программа этого, к сожалению, не делает.

Теперь необходимо настроить программу-клиент. Для этого переходим к настройкам почтового ящика, например, командой **Ящик->Свойства почтового ящика** или же нажатием соответствующей кнопки на панели инструментов. Открываем раздел настроек «Транспорт» и настраиваем сервера и протоколы, как показано на рис. 3. При этом стоит учитывать, что локальный адрес 127.0.0.1 должен быть свободен от других программ.

После сохранения настроек The Bat! готов получать и отправлять почту через защищённый канал SSL. При этом убедитесь, что программа stunnel запущена.

Кстати, интересное наблюдение – если при загрузке компьютера возникают проблемы с интернетом, значок stunnel не появляется в области уведомлений, что порой весьма полезно.

Конечно, в интернете, если покопаться, можно найти достаточно рецептов по настройке этой программы, возможно, что есть и описание настройки для совместной работы с The Bat!, но мне в своё время пришлось достаточно покопаться, чтобы запустить программу и заставить её работать. Здесь же я постарался изложить все проблемы, с которыми столкнулся во время настройки и начального использования программы stunnel. Надеюсь, что вы найдёте материал интересным и достаточно полезным.

Андрей Шаронов (Andrei88)

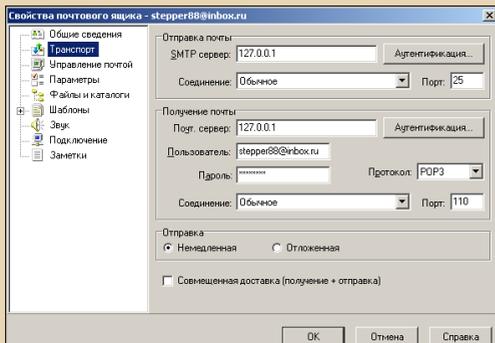
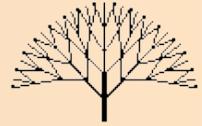


Рис. 3. Настройки The Bat! для работы с stunnel



# Electronics WorkBench – компьютер моделирует электронику



**С**истемы автоматического проектирования совершенствуются, обзаводятся новыми возможностями, включают новые функции, видоизменяют форматы файлов, дабы в более ранней версии было сложно открыть проект, созданный более поздней версией программы. Инженеры – хоть конструкторы-механики, хоть электронщики – переходят от одной версии программы к другой, а то и полностью меняя философию – от ДОСовского P-CAD к Accel EDA, а с него (вернее с P-CAD 2006) на Altium Designer. Но во всей этой гонке есть маленький островок, который может быть отрадой для даунгрейдера – программы моделирования работы электронных схем.

Да, программы моделирования совершенствуются, выходят новые версии, появляются новые продукты, которые непременно хотят Win32 и т.д., но достаточно часто студентов обучают на проверенных в бою программных продуктах, которые могут быть запущены даже под Windows 3.11. А потому даунгрейдер, если он студент, может совместить приятное с полезным, выполнив лабораторную работу на своём любимом старом компьютере.

Одной из таких программ является разработанный компанией Interactive Image Technologies пакет Electronics Workbench. В настоящий момент данная программа развивается компанией National Instruments под названием Multisim, но в сети на радилюбительских ресурсах легко можно найти Electronics Workbench 5.12, например, [здесь](http://old-dos.ru/dl.php?id=4549). Кроме того, на old-dos доступна ещё и профессиональная версия программы: <http://old-dos.ru/dl.php?id=4549>. В файле ReadMe явно указывается возможность запуска программы под Windows 3.x с установленной Win32s. Так почему бы не попробовать?

Оказалось, что программа достаточно чувствительна к версии Win32s. Например, с последней версией 1.30с автор получил от инсталлятора сообщение о том, что одна из библиотек Win32s принадлежит Windows NT и не может адекватно работать под Windows 3.11. Правда, инсталлятор (обычного EWB 5.12) нормально всё поставил. Когда же устанавливал с версией Win32s, входившей в состав драйверов софта сканера, такого сообщения не возникало.

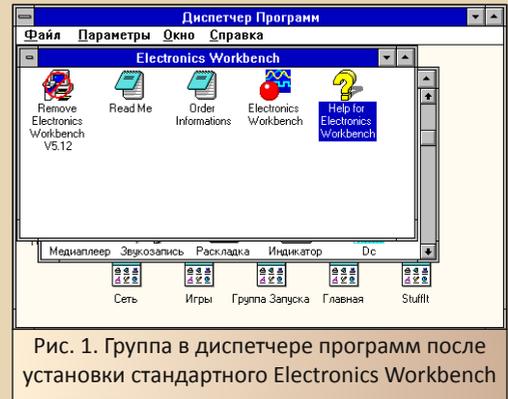


Рис. 1. Группа в диспетчере программ после установки стандартного Electronics Workbench

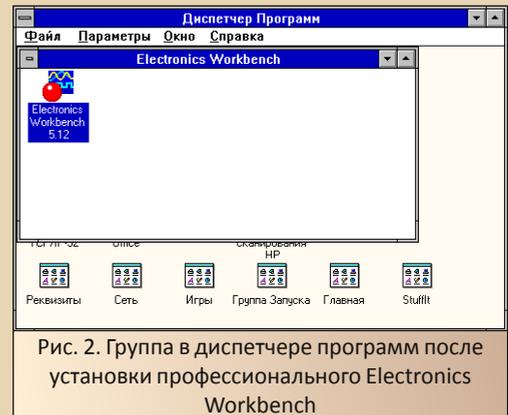


Рис. 2. Группа в диспетчере программ после установки профессионального Electronics Workbench



Кроме этого возникла другая проблема – Electronics Workbench – программа достаточно вальяжная, особенно в момент загрузки. На слабом 486-м компьютере программа загружается не долго, а очень долго. При этом может возникнуть ощущение, что программа зависла – тогда стоит нажать левую клавишу мыши. В профессиональной версии пришлось нажать клавишу несколько раз. Однако в конечном итоге открылось главное окно программы – на рис. 3 показано окно стандартной версии, на рис. 4 – профессиональной. В дальнейшем все эксперименты будем проводить в профессиональной версии.

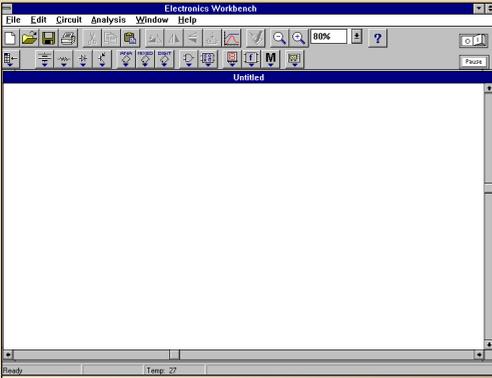


Рис. 3. Главное окно стандартной версии Electronics Workbench – загружалась долго, но работает вполне себе нормально

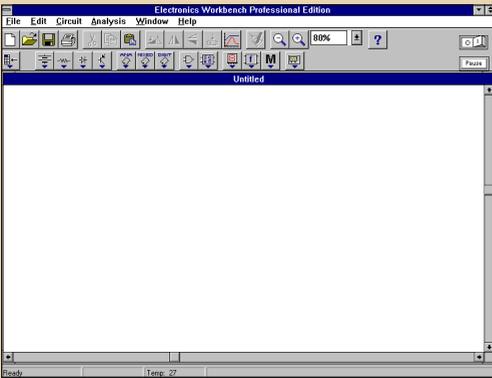


Рис. 4. Главное окно профессиональной версии Electronics Workbench – в работе разницы не видно, но показалось, что профессиональная версия ставится быстрее

В качестве эксперимента была собрана схема на операционном усилителе (см. рис. 5).

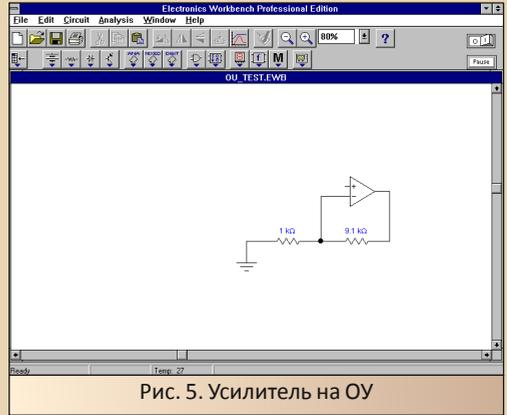


Рис. 5. Усилитель на ОУ

Из библиотеки был взят стандартный операционный усилитель и два резистора. По умолчанию резисторы имеют сопротивление 1 кОм. Чтобы поменять, понадобится дважды щёлкнуть по резистору, чтобы открылось диалоговое окно установки параметров резистора (см. рис. 6).

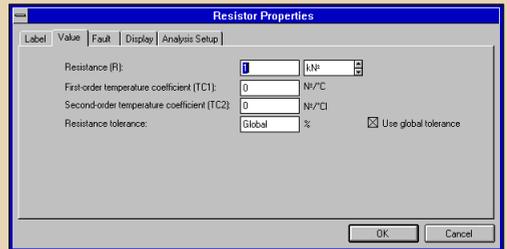


Рис. 6. Диалог установки параметров резистора

Для проверки работы усилителя на вход был подключен генератор из панели приборов (она самая правая – см. рис. 7) и осциллограф.

Перед запуском настроим генератор. Как обычно бывает – параметры по умолчанию далеко не оптимальные. Щёлкаем дважды по генератору, и открывается окно его настройки (см. рис. 8). По умолчанию частота выходного сигнала 1 Гц, а амплитуда 10 В. Автор поднял частоту и уменьшил амплитуду.



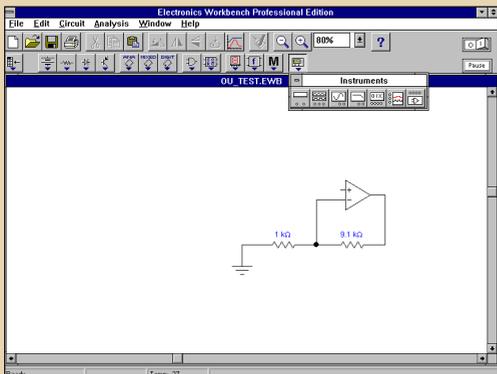


Рис. 7. Меню с измерительными приборами – каждый прибор даёт Вам в одном экземпляре – ну прямо квест какой-то

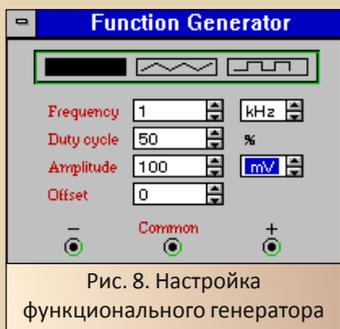


Рис. 8. Настройка функционального генератора

После подключения измерительных приборов схема приобретет вид, как показано на рис. 9.

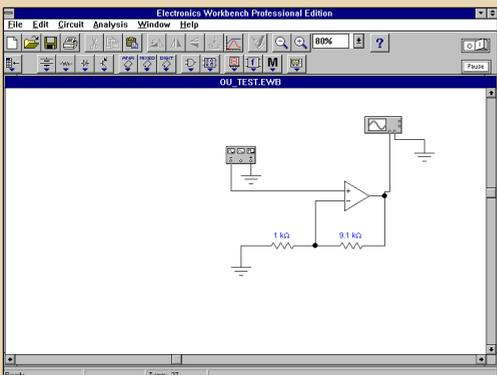


Рис. 9. К усилителю подключен генератор и осциллограф. Схема готова к моделированию

Теперь откроем двойным щелчком осциллограф. Уж не знаю – вина ли Windows 3.11, драйвера видеокарты или самой видеокарточки, а, может быть, кто-то так хорошо ломал, но изображения на экране осциллографа нет (см. рис. 10).

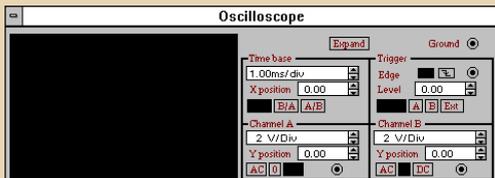


Рис. 10. Осциллограф в свёрнутом режиме – только чёрный экран

Правда, если нажать кнопку **Expand**, окно развернется и всё будет видно. Теперь можно в настройках растащить два луча вверх и вниз, настроить шаг сетки – по времени и напряжению – и запускать виртуальным выключателем питания в правом верхнем углу. Ключ на старт, и... что-то не то – оба луча показывают сигнал с выхода усилителя (см. рис. 11).

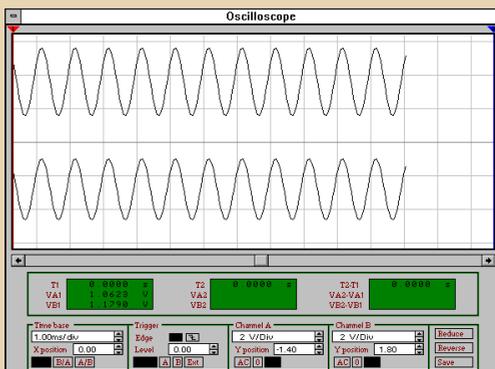


Рис. 11. Хотя у осциллографа два входа, почему-то на обоих показывается выход усилителя, хотя под Win XP, например, всё было нормально – один луч – вход, другой – выход

Когда же подаётся один сигнал, всё отображается прекрасно (см. рис. 12 – сигнал на входе, рис. 13 – сигнал на выходе).



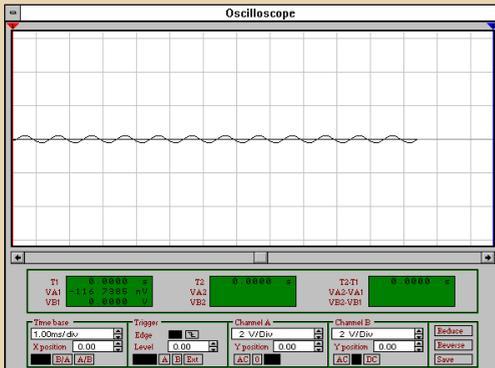


Рис. 12. Сигнал на входе усилителя – работает только один луч осциллографа – на второй ничего не подаётся

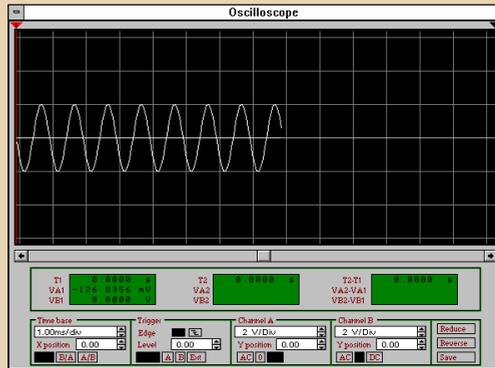


Рис. 14. Белый луч на чёрном фоне всё же привычнее

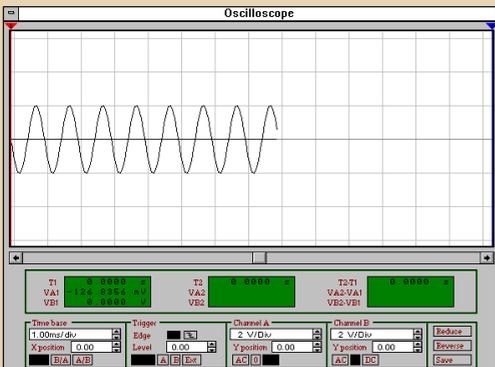


Рис. 13. Сигнал на выходе усилителя – усиление налицо ☺

Кстати, под кнопкой запуска моделирования схемы находится очень полезная кнопка – **Pause** – остановка моделирования с возможностью продолжить после повторного нажатия кнопки. Достаточно полезная кнопка, когда надо остановить исполнение и сделать красивый скриншот.

На самом осциллографе, кроме настроек параметров смещения, времени развёртки и цены деления, будут интересны кнопки **Reduce** – возвращение к маленькому объекту окна осциллографа и **Reverse** – обращение цветов – после нажатия экран приобретёт более традиционный чёрный цвет (см. рис. 14).

В принципе, если не обращать внимания на неторопливость программы, и обзавестись оптической мышкой или же хорошей шариковой с ковриком, то с программой работать можно. Стоит также учесть, что программа запускалась на 486-ом процессоре с частотой 33 МГц. Вполне возможно, что на системе с 486DX4 120 МГц или же 5x86 программа покажет куда более высокие результаты. Поэтому, если у вас есть старый компьютер, удобный стол и стул, то старичка можно применить для подготовки отчёта по очередной лабораторной работе, посвящённой моделированию той или иной электронной схемы. А заодно удивить товарищей и преподавателя занимательными скриншотами, сделанными из Windows 3.x.

Андрей Шаронов (Andrei88)



# КОНКУРЕНТ



**В** статье «И читает, и сканирует, и переводит», опубликованной в четырнадцатом номере журнала, автор рассказывал об интегрированном пакете Stylus Lingvo Office, реализующем полный цикл ввода бумажных документов в компьютер, перевода, и, если нужно – распечатки – как писали, от листа до листа. Ну и, конечно, если возникает необходимость, можно было на любом этапе передать документ в Microsoft Word или Excel. В статье вскользь упоминался конкурирующий продукт «Русский офис» компании «Арсеналь». Конечно же, очень хотелось сравнить эти два пакета, тем более что автор тогда считал, что вариант «Русского офиса», выпущенный в 1995-м году, включает в себя только переводчик «Сократ» и систему оптического распознавания BIT FineReader. Получалось, что «Арсеналь» тогда не имел ничего, кроме программы-переводчика, но хотел выпустить именно офисный пакет, потому и лицензировал FineReader. Выходило, что «Русский офис 95», назовём его так (тогда ещё они не нумеровались – всё же первая версия продукта), и Stylus Lingvo Office – продукты одной весовой категории и вполне сравнимы. Была только одна загвоздка – у автора не было «Русского офиса». Были более поздние диски, чуть позже появились образы дискет с программами FineReader 3.0 и «Сократ 2.0», но это всё же не единый продукт, хотя, вполне возможно, что программы продавались вместе – «Арсеналь» так делал, но всё было не то. И вот, недавно случилось чудо – купив очередной диск, автор с удивлением обнаружил в папке **SOCRAT\_20** (на

обложке тоже было написано, что это только «Сократ 2.0») установочный комплект полноценного офиса – с FineReader 2.0 и, собственно, переводчиком «Сократ» второй версии.

Почему статья и будет посвящена наконец найденному одному из самых ранних пакетов «Русского офиса».

## Установка и начальная настройка

Да, и здесь не обошлось без тонкостей. Возможно, и на ровном месте, но всё же.

Сам «Русский офис» можно скачать с old-dos.ru по ссылке <http://old-dos.ru/dl.php?id=16521>.

Все программы ставил на русскую версию Windows 3.11 в следующем порядке:

- MS Word 6.0 и Excel 5.0.
- «Русский офис» – инсталлятор сам определял, в каком порядке ставить FineReader и «Сократ».
- Программное обеспечение сканера – в данном случае HP ScanJet 5P – в его состав входит Win32s, поэтому она не устанавливалась отдельно.

Из тонкостей можно отметить, что инсталлятор программы устанавливает или заменяет библиотеку **CTL3DV2.DLL**, которая находится в каталоге **C:\WINDOWS\SYSTEM\**. Поэтому нужно следить, чтобы программы, которые могут её использовать, не были запущены. Автор для перестраховки просто её удалил перед началом установки. Просто так получилось, что в случае запуска инсталлятора из-под «Диско Командира» (опытным путём выяснилось, что «командир» действительно пользуется этой библиотекой) инсталлятор отказывался продолжать установку, утверждая, что не может получить к DLL-ке доступ. Даже, когда автор запускал инсталляцию с помощью функции «Выполнить» диспетчера программ, установка не желала продолжаться. Кончилось тем, что с помощью диспетчера файлов файл библиотеки был удалён и с помощью того же диспетчера был запущен инсталлятор. Только после этого программа была установлена.



При установке на русскую версию Windows 3.11 обе программы имели интерфейс на русском языке. Однако, если в вашем случае интерфейс у программ английский («Сократ 2.0» после установки на Windows XP был английским и не желал переключаться, а FineReader вполне себе переключается), язык можно переключить в настройках программ. В FineReader диалоговое окно настроек вызывается командой **Tools->Options**. В этом диалоговом окне необходимо открыть закладку **General**. В настройках можно изменить язык, а также настроить отображение панелей инструментов, подсказки при запуске программы и диалога типа текста при распознавании (см. рис. 1). После изменения языка интерфейса может потребоваться перезагрузка программы.

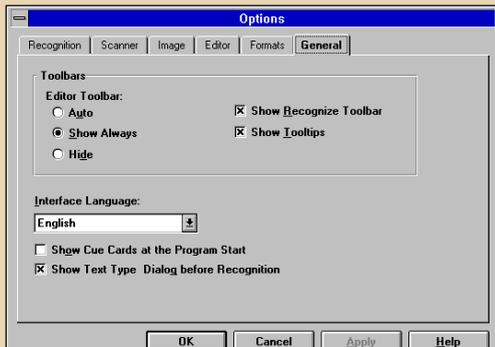


Рис. 1. Общие настройки FineReader – если программа после установки имеет англоязычный интерфейс, то это можно исправить здесь, как и настроить другие параметры

Русский язык в «Сократе» можно включить двумя способами. Первый – через диалоговое окно настройки интерфейса, вызываемое командой **View->Toolbars** (см. рис. 2). Здесь можно установить язык интерфейса, а также язык файла помощи, вызываемого по нажатию кнопки **F1** – либо английский, либо русский. Кроме того, можно изменить стиль интерфейса программы – сделать кнопки больше или меньше, закруглить или наоборот сделать

квадратными с острыми углами – стиль Windows 3.1 и стиль Windows 95 соответственно.

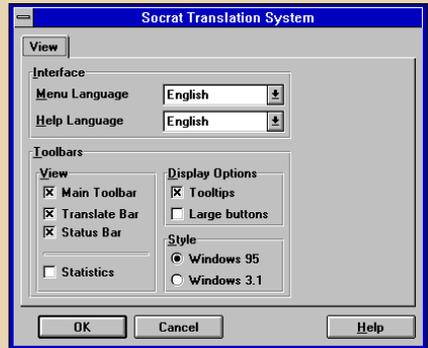


Рис. 2. Настройки интерфейса «Сократа 2.0» – настраивается и язык интерфейса, и его внешний вид

Второй способ – вызвать диалоговое окно опций переводчика командой **Tools->Options** и открыть вкладку **View** (см. рис. 3)



Рис. 3. Те же настройки, но в диалоговом окне опций переводчика

Чтобы применить настройки языка интерфейса, «Сократ» также необходимо перезапустить.

После перезапуска обеих программ можно приступать к более детальному изучению интерфейса.



## Интерфейс программ

Если проводить параллели со Stylus Lingvo Office, сразу бросается в глаза серость интерфейсов программ «Русского офиса» – ни тебе разноцветных кнопочек, разноцветной и весьма симпатичной панели управления всем комплексом, которая располагалась вверху экрана и давала возможность запустить любую программу Lingvo Office. Вроде бы и FineReader одинаковый – одна и та же версия, но тут он кажется более тусклым. Но в сторону уныние! Всё равно программы очень симпатичные! Да и сколько времени убито на поиски этого комплекта. ☺

Так как основным продуктом всё же является «Сократ», начнём с него – тем более, что для начала советуют запускать именно переводчик. Интерфейс программы показан на рис. 4. По умолчанию устанавливаются маленькие кнопки и стиль Windows 95, который легко заменить в опциях.

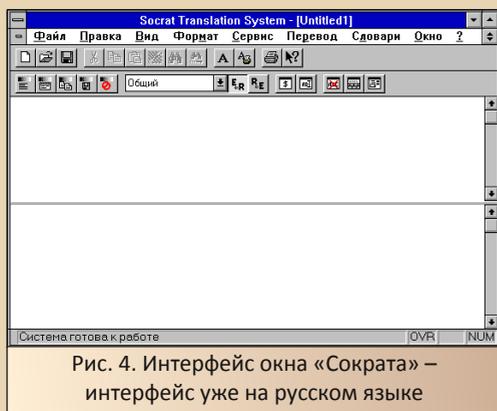


Рис. 4. Интерфейс окна «Сократа» – интерфейс уже на русском языке

Как видно, часть окна, где будет размещен текст, уже сразу разделена на две половинки – для исходного текста и для перевода. То есть программа готова уже сразу, и от пользователя не требуется создавать новый файл, как того требует Stylus.

В составе программы имеется три встроенных словаря:

- стандартный;

- компьютерный;
- бизнес-тематики.

Установить необходимый словарь можно в меню настроек программы по команде **Сервис** → **Параметры**. В открывшемся окне можно выбрать как используемый словарь, так и направление перевода (см. рис. 5). Также доступен вариант интерактивного перевода, где пользователь может влиять на процесс перевода текста – оставлять какие-то слова непереверждёнными или же исправлять неточности.

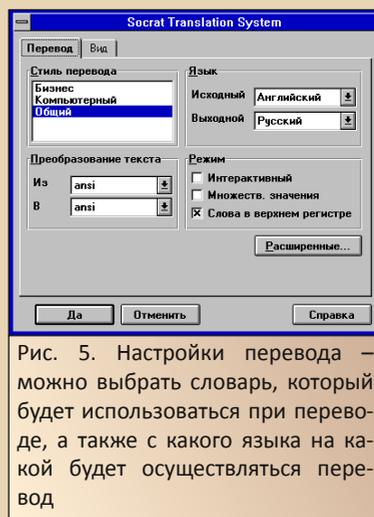


Рис. 5. Настройки перевода – можно выбрать словарь, который будет использоваться при переводе, а также с какого языка на какой будет осуществляться перевод

Кроме диалогового окна настроек, параметры перевода можно легко изменить с помощью нижней панели инструментов – правее кнопок перевода находятся меню и кнопки, дублирующие практически все настройки – выбор словаря, направление перевода, включение интерактивного режима, а также выделение слов, не подлежащих переводу.

Теперь перейдём к FineReader. Как уже говорилось ранее, «Русский офис» и Stylus Lingvo Office включают в себя одну и ту же версию программы распознавания – 2.0. С одной лишь разницей – входящий в комплект FineReader'a Lingvo Corrector в состав «Русского офиса» не включают.

Главное окно программы показано на рис. 6.



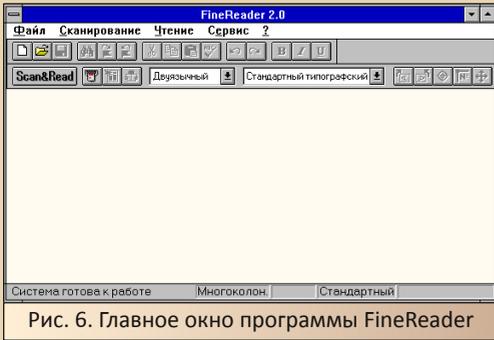


Рис. 6. Главное окно программы FineReader

На панели инструментов присутствует кнопка запуска сканирования документа с распознаванием, а также кнопки, отвечающие за отдельные операции – сканирование, распознавание, разбиение на блоки. Кроме того, можно настроить язык распознаваемого текста и выбрать его качество – FineReader имеет отдельные алгоритмы распознавания распечатки матричного принтера и текста, набранного на пишущей машинке. Так что, если вам известно, что вы будете загонять в компьютер именно распечатку с матричника, это можно указать программе явно.

Для того чтобы ввести текст, необходимо выбрать сканер в настройках программы. Это можно сделать в диалоговом окне настроек, которое в русскоязычном интерфейсе можно вызвать командой **Сервис->Опции** в закладке **«Сканер»** (см. рис. 7).

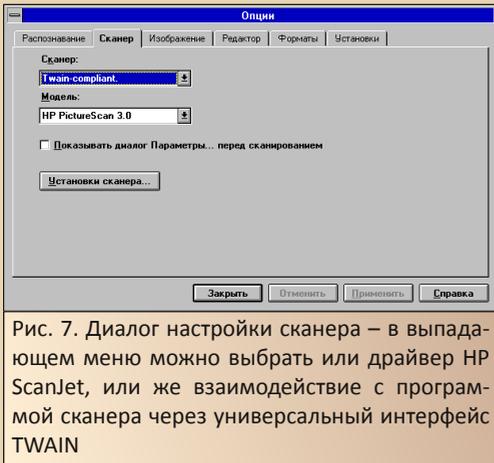


Рис. 7. Диалог настройки сканера – в выпадающем меню можно выбрать или драйвер HP ScanJet, или же взаимодействие с программой сканера через универсальный интерфейс TWAIN

В нашем случае был выбран интерфейс TWAIN и в списке **«Модель»** появилась версия программы, взаимодействующей со сканером на нижнем уровне и сопрягающейся с FineReader через TWAIN-интерфейс. После настройки можно перейти к сканированию и распознаванию документа.

Так как подходящего листочка или инструкции на английском языке не нашлось, была распечатана первая страничка первого попавшегося под руку аппнота. На ней и будем экспериментировать. Решено не запускать сканирование и распознавание в автоматическом режиме, поэтому для начала нажмём кнопку сканирования документа. Откроется диалоговое окно программы сканирования – в случае автора такое, как показано на рис. 8.

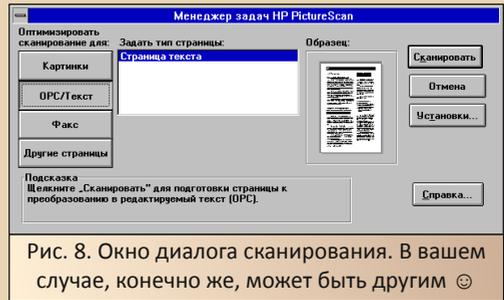


Рис. 8. Окно диалога сканирования. В вашем случае, конечно же, может быть другим ☺

Возможно, у читателя будет возможность уже на этапе сканирования выбрать необходимый участок документа, чтобы не гонять сканер (у USB-шного сканера Mustek 1248, например, это возможно), в конкретном же случае нам придётся отсканировать всю страничку.

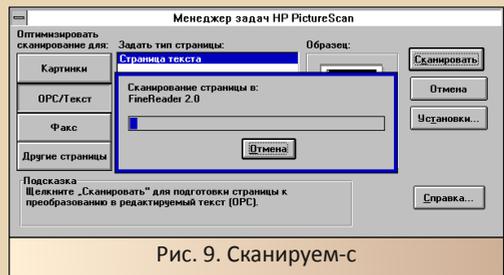


Рис. 9. Сканируем-с



После сканирования в окне FineReader откроется небольшое подокошко, где будет располагаться картинка отсканированного документа (см. рис. 10).

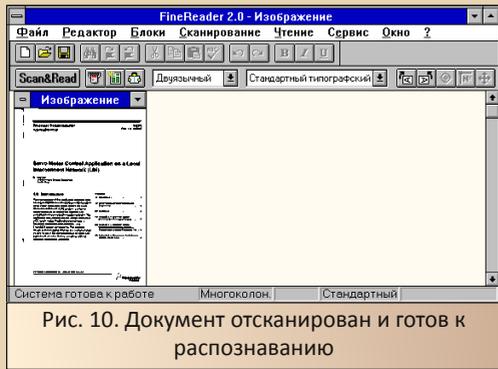


Рис. 10. Документ отсканирован и готов к распознаванию

Если вы сразу нажмёте кнопку распознавания, то система самостоятельно разобьёт на блоки и распознает, сохраняя более-менее форматирование исходного текста, всю страницу (см. рис. 11).

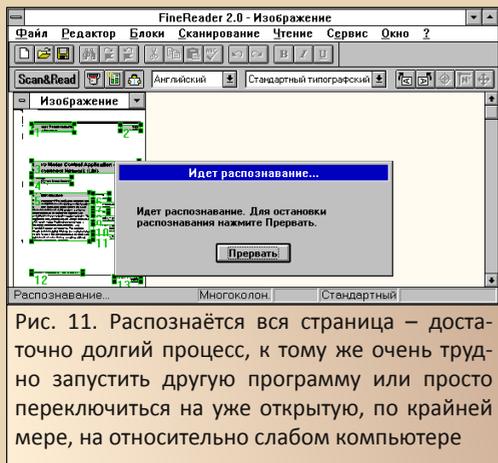


Рис. 11. Распознаётся вся страница – достаточно долгий процесс, к тому же очень трудно запустить другую программу или просто переключиться на уже открытую, по крайней мере, на относительно слабом компьютере

К сожалению, результат такого распознавания не очень пригоден для дальнейшего редактирования (см. рис. 12). Вернее, в принципе, работать с ним можно, но ради этого не стоило отдельно нажимать кнопки скани-

рования и распознавания – можно было сделать всё на автомате нажатием одной кнопки. В принципе, если в программе сканирования можно выбрать нужный участок сканируемого документа, автоматическое сканирование и распознавание вполне оправдано.

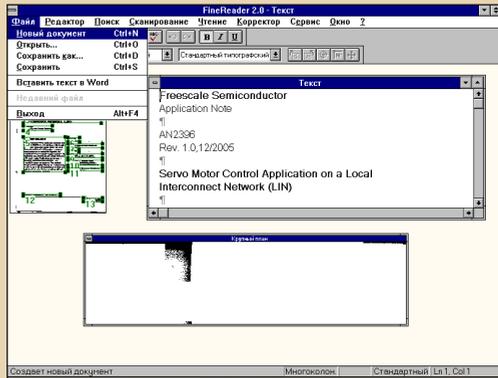


Рис. 12. Результат распознавания «на автомате» – такое можно было бы получить, нажав одну кнопку «Scan&Read»

Если же, как в случае автора, можно отсканировать только всю страницу сразу, стоит после сканирования нажать кнопку разбиения на блоки. В таком случае пользователь может самостоятельно выбрать, какие участки стоит распознавать, а какие – нет (см. рис. 13).

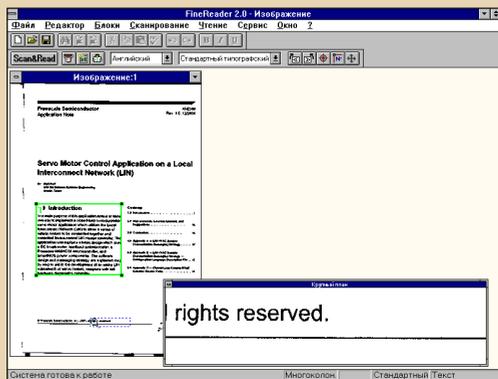


Рис. 13. Выделяем только вступление к документу – остальное распознавать не нужно – и работы программе меньше, и с результатом потом работать проще



В результате распознан только участок документа с текстом введения (рис. 14). Как раз то, что было необходимо для тестирования переводчика. В полном тексте документа этот фрагмент ещё надо было бы найти.

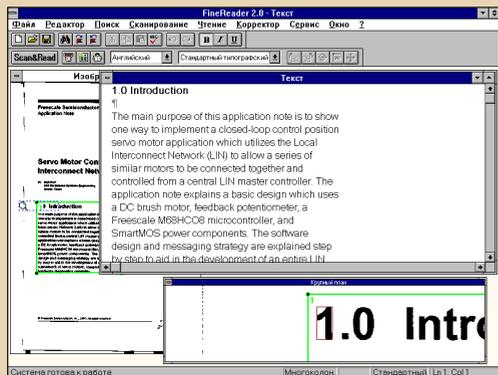


Рис. 14. Необходимый фрагмент распознан, и с ним можно продолжать работать

Перед началом распознавания программа может поинтересоваться (если не отключили в прошлый раз), какого типа распознаваемый текст – обычный или же набранный на машинке, либо распечатанный матричным принтером (см. рис. 15).

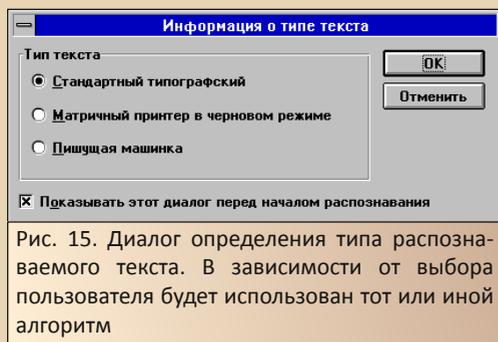


Рис. 15. Диалог определения типа распознаваемого текста. В зависимости от выбора пользователя будет использован тот или иной алгоритм

После того, как текст распознан, хотелось бы передать его в «Сократ» для перевода. К сожалению, такая функция не предусмотрена – текст можно передать только в Word (см. рис. 16).

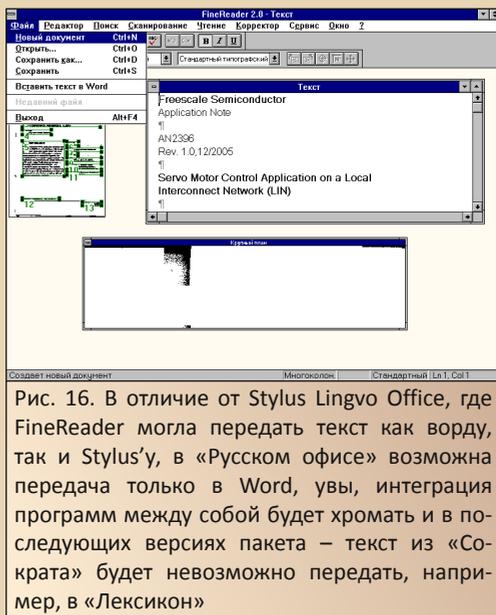


Рис. 16. В отличие от Stylus Lingvo Office, где FineReader могла передать текст как ворду, так и Stylus'у, в «Русском офисе» возможна передача только в Word, увы, интеграция программ между собой будет хромать и в последующих версиях пакета – текст из «Сократа» будет невозможно передать, например, в «Лексикон»

Поэтому сделаем просто – скопируем текст и вставим его в переводчик. Коряво, но что делать. В качестве словаря выбираем компьютерный словарь, как более-менее близкий. Запускаем перевод всего текста и... Ну, что ж, могло быть и лучше – вроде бы и компьютерный словарь воткнули, а всё равно DC (постоянный ток) был переведён как «Округ Колумбия» (см. рис. 17).

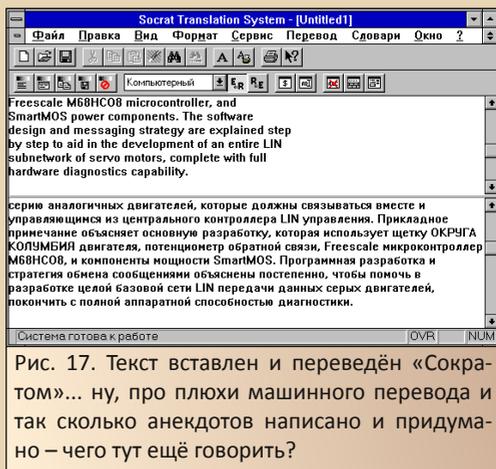


Рис. 17. Текст вставлен и переведён «Сократом»... ну, про плюхи машинного перевода и так сколько анекдотов написано и придумано – чего тут ещё говорить?



Видимо, чтоб подобные вещи повторялись как можно реже, введён интерактивный режим перевода, а заодно и возможность создания собственных словарей.

Правда, подобие интеграции программ сделано на уровне панели в Microsoft Office, но об этом речь пойдёт дальше.

### Интеграция с Microsoft Office

Что и говорить, в прошлом случались ситуации, которые с вершин сегодняшнего дня кажутся курьёзными – процессоры от AMD, работающие в материнских платах с чипсетом Intel, а порой и дающие этим платам второе дыхание.

Так и в нашем случае – ситуация достаточно занимательная – «Русский офис» – достаточно серьёзный конкурент офиса Microsoft («Арсенал» в конечном итоге был куплен российским отделением Microsoft в 2003-м году) – в те времена встраивал собственную панель инструментов в продукты Microsoft.

После установки «Русского офиса» панелька не активна, но её можно включить в диалоговом окне, вызываемом командой Вид -> **Панели инструментов**. В диалоге необходимо поставить галочку напротив пункта **Russian Office** (см. рис. 18) и панелька появится (см. рис. 19).

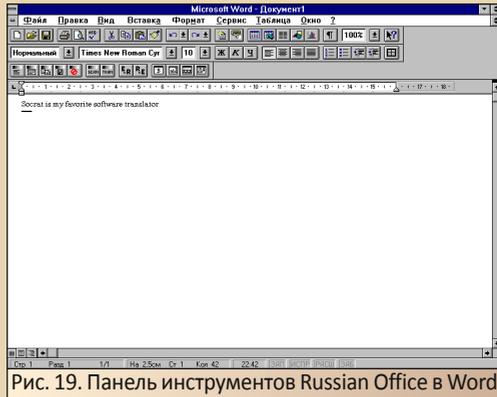
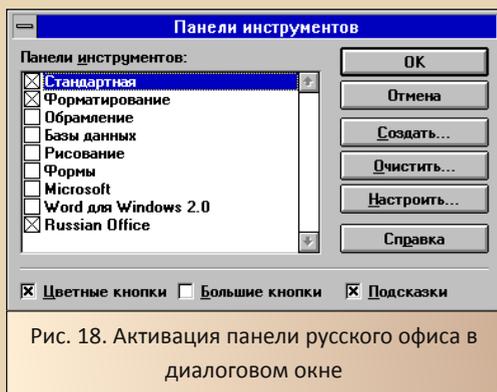


Рис. 19. Панель инструментов Russian Office в Word

Хоть в ReadMe-файле и указывается поддержка Excel 5.0, но её собирались добавить в более позднюю версию. Потому автор среди списка панелей Excel и не нашёл аналогичной.

В Stylus Lingvo Office FineReader и Stylus имели собственные панели инструментов, которые включались отдельно. Здесь же кнопки вызова FineReader и «Сократа» выполнены в едином стиле, который ближе «Сократу». Самыми левыми идут панели функций перевода – перевести выделенное, перевести окно, перевести весь файл и т.д. Кроме того, присутствуют панели выбора направления перевода, вызова диалога опций, интерактивного перевода и т.д.

Автор, ради интереса, ввёл в ворде фразу на английском языке, выделил её и нажал самую левую кнопку. В фоне запустился «Сократ» и произвёл перевод прямо в документе (см. рис. 20 – до перевода, рис. 21 – после перевода).

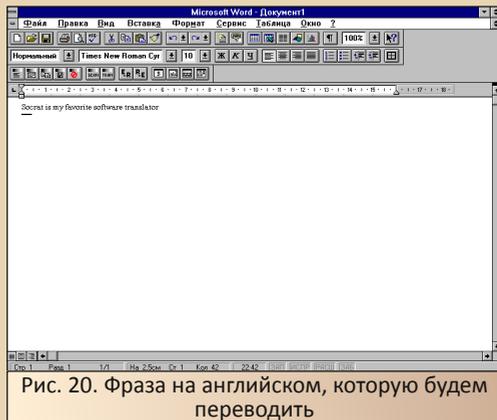


Рис. 20. Фраза на английском, которую будем переводить



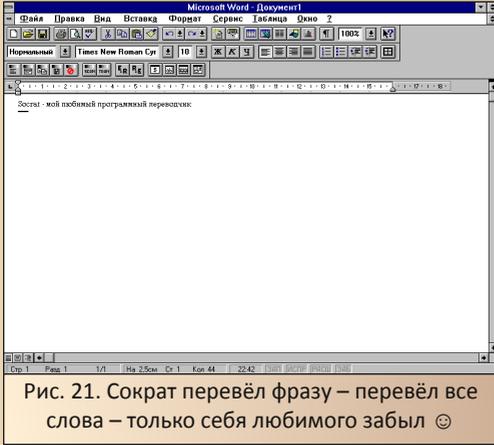


Рис. 21. Сократ перевёл фразу – перевёл все слова – только себя любимого забыл 😊

К слову, чтобы выполнить перевод в Word'e с помощью «Сократ 1.0», необходимо было сперва запустить переводчик. И только тогда появлялась панель «Русского офиса» в Word. Пока же переводчик не запущен – панели не было.

Теперь настал черёд проверить функцию сканирования и перевода. Нажимаем кнопку **TRAN** на панели инструментов и... Не знаю, как у читателя, а у автора ничего не получилось – запустился ScanManager из комплекта FineReader, и на этом дело встало – фактически зависло – Word не отвечает, а кроме менеджера сканирования ничего не желает запускаться. Что ж, не будем опускать руки. По идее, должна первой стартовать FineReader. Её и запустим. После запуска в фоне FineReader, кнопка сработала нормально. Тут же запустилось сканирование – открылось диалоговое окно работы со сканером, получилось отсканировать, распознать документ, но... Word отказался открывать буферный файл – **SOCRAT20.RTF**, который был создан в каталоге FineReader. При этом сам файл был доступен. Что делать?

Тут вспомнились настройки сканера в FineReader. Ведь программа может взаимодействовать не только с софтом производителя через TWAIN-интерфейс, но ещё и с драйверами напрямую. Не для всех моделей сканеров, но конкретно HP поддерживались. Открываем настройки и меняем интерфейс взаимодействия

(см. рис. 22). Со второго раза (пришлось перезагрузиться – иначе получал ошибку DDE) удалось выставить возможность работы со сканером напрямую.

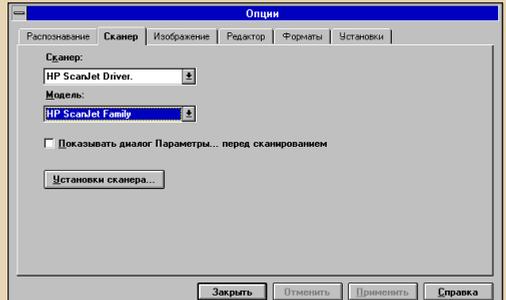


Рис. 22. Выставляем работу со сканером напрямую. Теперь FineReader будет обращаться к драйверам сканера, а не к сканирующей программе

Теперь, если нажать кнопку сканирования, сканер начнёт работать сразу, а пользователь увидит только диалог сканирования (см. рис. 23).

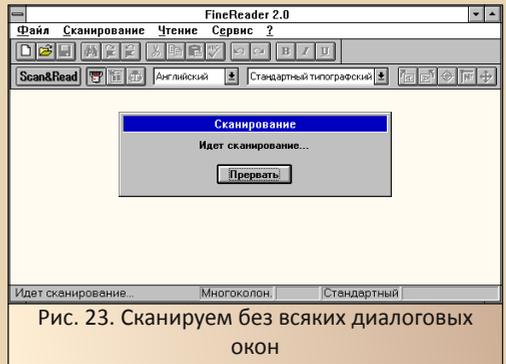


Рис. 23. Сканируем без всяких диалоговых окон

Теперь снова перейдём к панели в Word'e. В этот раз для начала попробуем вызвать только FineReader. Нажимаем кнопку **SCAN**. В этот раз FineReader запускается без всяких проблем. Сразу же запускает сканирование, потом на автомате разбивает страницу на блоки и распознаёт их. Отношение автора к этому режиму было описано выше, но тут уж ничего не поделаешь – раз полная автоматизация. 😊 В итоге Word'у передаётся буферный файл



**SOCRAT20.RTF**, а FineReader закрывается (см. рис. 24).

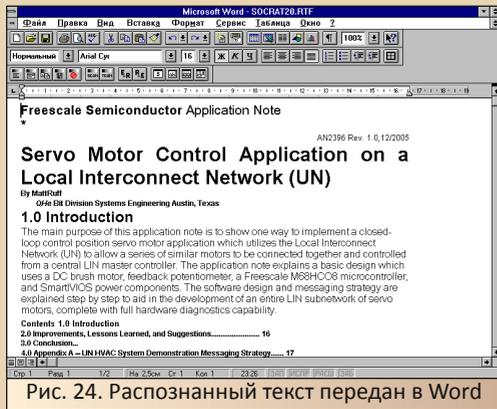


Рис. 24. Распознанный текст передан в Word

Теперь снова перейдём к кнопке сканирования и распознавания. Снова сканирование и распознавание прошло без проблем, документ был передан в Word. Только тогда приступил к работе «Сократ». Был создан новый документ, а рабочее пространство было, как будто, поделено на две половинки, как в переводчике. На верхней половинке располагался оригинал, на нижней – англоязычный текст постепенно заменялся русскоязычным. В итоге получилось то, что показано на рис. 25.

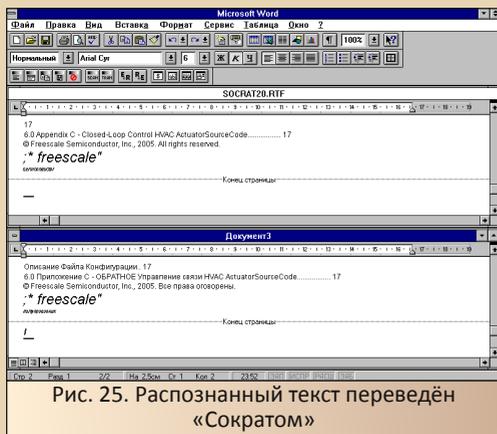


Рис. 25. Распознанный текст переведён «Сократом»

На относительно слабом компьютере процесс такого перевода был достаточно длительным. Казалось, что «Сократ» берёт текст по одному абзацу, переводит его и возвращает

назад. И складывалось ощущение, что больше трудностей ему доставляет не перевод, а взять абзац и вернуть результат. Однако всё равно переводчик справился со своей задачей, а весь комплект доказал, что является единым офисом, хоть и связанным только через панель инструментов в Word.

### Заключение

Как частенько бывает, автор хочет закончить статью одним, а жизнь требует совершенно другой концовки. Конечно, «Русский офис» выглядит куда беднее и по возможностям, и по оформлению своего конкурента Stylus Lingvo Office. Конечно, ВIT (ABBY) согласились участвовать в проекте «Арсенала» только потому, что иначе бы место их продукта заняла бы CuneiForm, и не особо заморачивались насчёт «Русского офиса». Но даже в таком виде пакет оказался функционален и, благодаря более низкой цене, мог конкурировать с творением двух гигантов – ВIT и ПромТ...

Но, оказалось, это далеко не всё. При установке «Русского офиса» был выдан список из шести программ:

- «Сократ 2.0» – машинный переводчик.
- «Цензор 1.0» – система проверки орфографии в Word, а может быть, и не только в нём.
- «Декарт 1.0» – программа ведения персональной бухгалтерии, о второй версии которой автор писал в прошлом выпуске журнала.
- «Послушные деньги» – программа для составления налоговой декларации.
- PictureMap 3.0 – растровый редактор зеленоградской компании Stoik.
- FineReader 2.0 – программа оптического распознавания текста.

Если же это перечень компонентов полного комплекта «Русского офиса» 1995-го года, то мы уже говорим не о сереньком «заморыше» из двух программ, которые не могут работать вместе без панели Word'a, а о серьёзном офисном пакете, который даже без «Лексикона» внушает уважение. Вот только одно «но» – пока что его – такого красавца – не получилось найти ни на дисках, ни на дискетах. ☹



# НОВЫЕ СРЕДСТВА РАЗРАБОТКИ ДЛЯ DOS



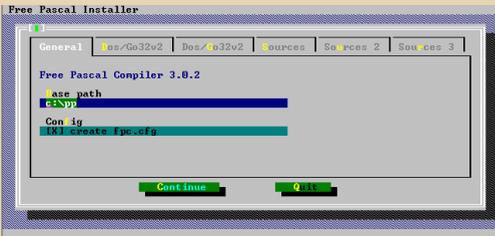
Как использовать свежайшие средства программирования под DOS, неужто свет клином сошёлся на старом Turbo/Borland Pascal и его современниках?

Нет, во-первых, существует современный и свободный аналог Turbo Pascal – Free Pascal ([www.freepascal.org](http://www.freepascal.org)), выпускаемый для множества платформ, в том числе для DOS. Free Pascal (далее FP) во многом совместим не только с Turbo Pascal, но и с Delphi. Для него существует визуальная среда разработки Lazarus и множество компонентов, однако я в этой статье намеренно ограничиваюсь только разработкой под DOS.

FP для DOS качаем отсюда:

<https://freepascal.org/download/i386/go32v2.var>

Там будут на выбор ссылки на зеркала. Скачанный архив куда-нибудь распаковываем и запускаем программу установки, где вас спросят, какие составляющие и куда устанавливать – проще оставить всё как есть.



После установки, в **autoexec.bat** добавляем строку:

```
set PATH=C:\pp\bin\go32v2;%PATH%
```

Перезагружаемся. После этого у нас будет полностью рабочий компилятор Free Pascal вместе с многочисленными библиотеками, отладчиком и рабочей средой, подобной Turbo

Pascal. Запускается она из командной строки командой **fp**.



Для проверки напишем простейшую программку на Pascal, назовём файл **pastest.pas**:

```
program pastest;
BEGIN
  WriteLn ('Hello world!');
END.
```

Чтобы откомпилировать – превратить этот исходник в **pastest.exe** – запустим среду разработки командой **fp**, откроем в ней файл **pastest.pas** (**File – Open** либо **F3**), и откомпилируем пунктом меню **Compile – Make** (либо просто **F9**). Если в исходнике вы не допустили никакой ошибки, получаем в том же каталоге файл **pastest.exe**, который при запуске выведет строку «Hello world!». Разве что, ещё – в **Options – Mode** выберите режим **Normal** вместо **Debug**, и потом компилируйте.

Для компиляции **pastest.pas** из командной строки надо в том же каталоге дать команду:

```
fpcc pastest.pas
```

Хорошо, а как насчёт Си и C++? А для DOS существует порт знаменитого компилятора GCC – DJGPP ([www.delorie.com/djgpp](http://www.delorie.com/djgpp)). Версия GCC там свежайшая, за 2017 год.

Можете выбрать и скачать сами подходящий вам набор средств для разработки [отсюда](https://yadi.sk/d/22mukpYP3J77Qo), либо заранее собранный мною единый архив (<https://yadi.sk/d/22mukpYP3J77Qo>), который надо будет просто распаковать в C, то бишь в итоге после распаковки получится **C:\DJGPP**.

Если же качаете отдельные ZIP'ы, то всё распаковываем в **C:\DJGPP**. Если чего-то в «моём» архиве не будет хватать, всегда можно потом докачать отдельно и установить. Я упаковал основные средства разработки – GCC с



поддержкой Си и С++, отладчик GDB, мощную среду разработки RHIDE, мультимедийную библиотеку Allegro, а также библиотеку Curses (интерфейс) и GRX (графика).

Затем, чтобы всё работало как надо, внесим следующие правки в ДОСовские конфиги.

**В CONFIG.SYS:**

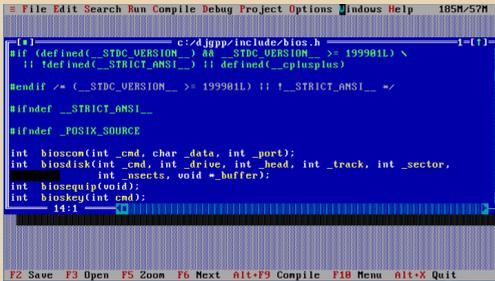
```
shell=c:\dos\command.com c:\dos /e:2048 /p
files=40
fcb=40,0
```

**В AUTOEXEC.BAT:**

```
set PATH=C:\DJGPP\BIN;%PATH%
set DJGPP=C:\DJGPP\DJGPP.ENV
```

Если вы пользуетесь не реальной DOS-машинкой, а DOSBOX'ом, то относящееся к **CONFIG.SYS** в него и пишем, а строки для **AUTOEXEC.BAT** надо добавить в файл настроек DOSBOX, в его раздел **[autoexec]**.

Попробуем теперь, как всё работает.



Вот простейшая программа на Си, которая выводит на экран сообщение «Hello world»:

```
#include <stdio.h>
int main()
{
    printf ("Hello world\n");
    return 0;
}
```

Сохраним исходник как **test.c**, и откомпилируем из командной строки:

```
gcc test.c -o test.exe
```

Если не будет никакой ругани от компилятора, то получим файл **test.exe**.

Теперь попробуем на вкус С++ и библиотеку Curses, а вернее её вариант PDCurses (Public Domain Curses), которая распространяется с DJGPP. Сразу отмечу, что в DOSBOX компиляция провалится, ибо DOSBOX не поддерживает длинные имена файлов. Для FreeDOS за такую поддержку отвечает драйвер LFNDDOS (<http://adoxa.altervista.org/doslfn>), а MS-DOS 7.10 и DR-DOS имеют встроенную поддержку. Кстати, LFNDDOS можно использовать и в более старых версиях DOS.

Напишем простенькую программку на С++, назовём её **test2.cpp**. Перед исходником дам некоторые пояснения, остальные – в комментариях самого кода.

При каждом нажатии клавиш программа перерисовывает экран, выводя на него меню из трёх пунктов, отображая текущий пункт желтым цветом на чёрном фоне, а остальные пункты – белым цветом на чёрном фоне. При нажатии на «Q» происходит выход из программы. Курсорные клавиши «Вниз» и «Вверх» уменьшают и увеличивают на единицу значение переменной **current\_element**, которая содержит номер текущего пункта меню. Элемент меню по этому номеру отрисовывается какими-то цветами, остальные – такими-то.

Строки меню хранятся в массиве **list**, который представлен вектором строк **list**.

```
#include <iostream>
#include <vector>
#include <string>
```

```
#include <curses.h>
using namespace std;
```

```
vector <string> list;
int current_element;
```

```
void update_screen()
{
    erase(); //чистим экран
```

//в цикле отрисовываем пункты меню



```

for (size_t i = 0; i < list.size(); i++)
{
    //текущий пункт
    if (i == current_element)
    {
        attron(A_BOLD);
        attron (COLOR_PAIR(1));
        printw ("%s\n", list[i].c_str());
        attroff (COLOR_PAIR(1));
        attroff(A_BOLD);
    }
    else //иначе обычным цветом
    {
        attron (COLOR_PAIR(2));
        printw ("%s\n", list[i].c_str());
        attroff (COLOR_PAIR(2));
    }
}

int main (int argc, char *argv[])
{
    //разные инициализации
    initscr();
    start_color();
    keypad (stdscr, TRUE);
    noecho();
    nonl(); //для правильного перехвата
    //клавиши ENTER

    //задаём цветовые пары,
    //1 для текущего пункта меню
    //2 для остальных
    init_pair (1, COLOR_RED, COLOR_BLACK);
    init_pair (2, COLOR_YELLOW, COLOR_BLACK);

    //заполняем вектор пунктами меню
    list.push_back ("one");
    list.push_back ("two");
    list.push_back ("three");

    current_element = 0;

    int ch = 0;
    //считываем клавиши, пока не будет нажата q
    while (ch != 'q')
    {

```

```

        //обновляем экран
        update_screen();

        //ждем нажатия на клавишу и
        //получаем её код в ch
        ch = getch();

        //проверяем, какая клавиша нажата
        switch (ch)
        {
            case KEY_UP:
                if (current_element > 0)
                    current_element--;
                break;

            case KEY_DOWN:
                if (current_element < list.size()-1)
                    current_element++;
                break;

            case 13: //клавиша ENTER
                //пользователь выбрал элемент
                //list[current_element]
                break;
        }
    }

    endwin();
    return 0;
}

```

Компилируем программу командой:

```
gxx test2.cpp -lpdcurses -o test2.exe
```

На выходе получаем **test2.exe**.

В этой статье я не ставил целью широкое освещение вопросов программирования для DOS. Задачей своей видел только описание быстрой настройки в DOS двух современных компиляторов, и проверку их работоспособности. Очевидно, что программирование для DOS живо и программировать можно, используя современные компиляторы.

Пётр Семилетов

[tea@list.ru](mailto:tea@list.ru)



# push hl,de,bc,af



Написать сей опус меня толкнула работа с демо, где пришлось использовать трюки работы со стеком, а также некоторые запросы, которые были услышаны в чатах и на форумах.

Пожалуй, я рискну рассмотреть простые примеры.

Начну с простой задачи: как очистить экран? Начинаящий спектрумист скажет: «Легко!» и напишет примерный код:

```
ld hl,$4000; начальный адрес
ld de,$4001
ld bc,6143; размер экрана
ld (hl),0
ldir
```

Здесь написано верно, но теперь понадобятся простые подсчёты: инструкция Z80 `ldir` занимает 21/16 тактов на байт, выходит  $21 * 6143 = 129003$  такта, а это очень много. При постоянной анимации такой подход не пройдёт.

Вернусь к статье в забытом журнале «ZX-Ревю»: инструкция `push r16,(r16 – hl,de,bc)` выполняется 11 тактов, причем `push` заполнит два байта. Получается  $6144 / 2 * 11 = 33792$  такта. Ого, уже лучше.

Возникает вопрос: а как применить?

Да очень просто, вот исходный текст ассемблера SjASMPPlus (`stack_cls.asm`):

```
ld hl,0
lp:
ei
halt

ld (back_sp+1),sp; запомнить значение стека
ld sp,$5800; новый указатель памяти на конец экранной памяти
dup 3072;развёрнутый цикл
```

```
push hl
edup
back_sp: ld sp,0
jr $
```

Теперь я внесу небольшие изменения в код:

```
device zxspectrum128
ORG #6000

begin

ld hl,0

lp:
ei
halt

ld (back_sp+1),sp; запомнить значение стека

ld sp,$5800; новый указатель памяти на конец экранной памяти
dup 3072; развёрнутый цикл
push hl
edup
back_sp: ld sp,0
inc hl
jp lp
jr $

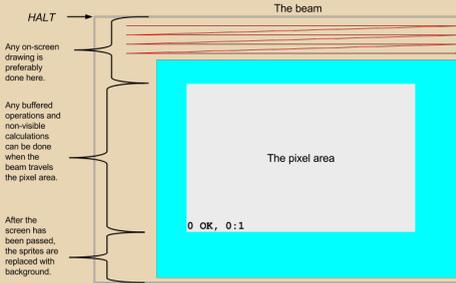
end

display /d,end-begin
savesna "!stack_cls.sna",begin
```



Все выглядит чётко, но на экране картинка немного искажается. В чём причина?

Следует описать процесс рисования:



Изображение с сайта <http://oldmachinery.blogspot.ru/2014/04/zx-sprites.html>

При выполнении команды **halt** потребует некоторая «задержка», затем начинается отрисовка экрана, в программе, которую я написал, происходит этакий «конфликт» – заполняется экран и луч попадает на область памяти – так называемое «сечение лучом»/«лучи секутся».

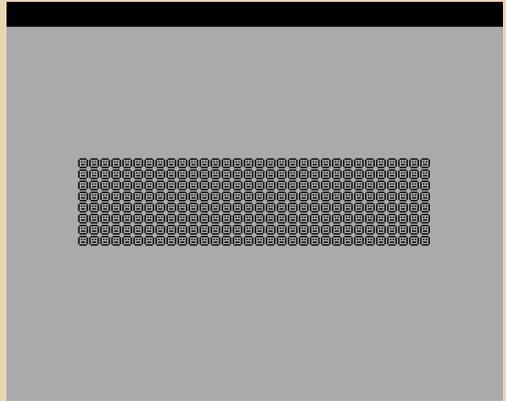
Как тут быть? Как один вариант – использовать для ZX Spectrum 128 двойную буферизацию с двумя экранами.

Или, зная структуру экрана, перехитрить железо.

Сначала нужно нарисовать узор верх ногами:

```
pat:↓
db %01111100↓
db %10000010↓
db %10111010↓
db %10000010↓
db %10101010↓
db %10000010↓
db %01111100↓
db %00000000↓
```

Теперь рисуется картинка:



Исходный текст (**stack\_pt.asm**):

```
device zxpectrum128
ORG #6000

begin

lp:
    ei
    halt

    ld (back_sp+1),sp; запомнить значение стека

    ld sp,pat
    ld a,0:out ($FE),a; рамка чёрная
    pop hl,de,bc,af
    exx
    exa
    pop hl,de,bc,af
    exx
    exa
    ld sp,$5000; новый указатель памяти на конец экранной памяти

; dup 8

    dup 128
    push hl
    edup
```



```
dup 128
push de
edup
```

```
dup 128
push bc
edup
```

```
dup 128
push af
edup
```

```
exx
exa
```

```
dup 128
push hl
edup
```

```
dup 128
push de
edup
```

```
dup 128
push bc
edup
```

```
dup 128
push af
edup
```

```
exx
exa
```

```
;edup
```

```
back_sp: ld sp,0
         ld a,7:out ($FE),a
         ; рамка белая – так по старинке вы-
         ; полняется измерение времени ис-
         ; полнения программы
```

```
jp lp
```

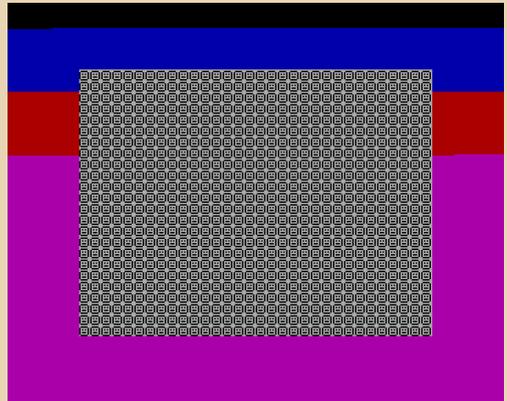
```
pat:
db %01111100,%01111100
```

```
db %10000010,%10000010
db %10111010,%10111010
db %10000010,%10000010
db %10101010,%10101010
db %10000010,%10000010
db %01111100,%01111100
db %00000000,%00000000
```

end

```
display /d,end-begin
savсна "!stack_pat.sna",begin
```

Теперь немного модифицирую программу и перейду к пояснениям.



Исходный текст (`stack_pt_fs`):

```
device zxspectrum128
ORG #6000
begin
;full screen
lp:
ei
halt

ld a,0:out ($FE),a
ld hl,$4800:call draw
ld a,1:out ($FE),a

ld hl,$5000:call draw
ld a,2:out ($FE),a
```



```

ld hl,$5800:call draw
ld a,3:out ($FE),a

jp lp

draw:
ld (back_sp+1),sp; запомнить значение
                    стека
ld (sp+1),hl

ld sp,pat
pop hl,de,bc,af
exx
exa
pop hl,de,bc,af
exx
exa

sp1: ld sp,$5000
; dup 8

dup 128
push hl
edup

dup 128
push de
edup

dup 128
push bc
edup

dup 128
push af
edup

exx
exa

dup 128
push hl
edup

dup 128
push de
edup

```

```

dup 128
push bc
edup

dup 128
push af
edup

exx
exa

;edup

back_sp: ld sp,0
ret

pat:
db %01111100,%01111100
db %10000010,%10000010
db %10111010,%10111010
db %10000010,%10000010
db %10101010,%10101010
db %10000010,%10000010
db %01111100,%01111100
db %00000000,%00000000

end

display /d,end-begin
savesna "!stack_pat_fs.sna",begin

И последняя модификация (stack_pt_fs_a):

ld hl,$5800:call draw
ld a,3:out ($FE),a
call anim ;сдвиг узора
ld a,6:out ($FE),a

jp lp

anim:
ld ix,pat
ld b,8
a_lp:
xor a
rr (ix+0) ; циклический сдвиг одной
линии узора

```



```
rr (ix+1)
jr nc,nob7
set 7,(ix+0)
nob7:
inc ix,ix
djnz a_lip ; повторить 8 раз
ret
```

Круто, всё движется в полный экран! Ну чем не демомейкеры? Осталось только придать узору вертикальное движение, оставлю такую задачу читателям.

Как работает этот замысловатый код?

Экран «Спектрума» организован следующим образом: адреса **16384-22527** (**\$4000-\$57FF**) хранят информацию о пикселях. При разрешении экрана 256x192 выходит 256/8=32 байта на строку. Следующие 32 байта отведены для линии на 8 линий ниже.

Получается:

- \$4000-\$401F** – верхняя линия;
- \$4020-\$403F** – 8-я линия ниже;
- \$4040-\$405F** – 16-я линия;
- \$4060-\$407F** – 24-я линия;
- ...
- \$40E0-\$40FF** – 56 линия.

Адреса **\$4100-\$41FF** организованы таким же образом, только описывают данные линий ниже, чем у начального адреса.

8 линий начинаются с адресов **\$4000, \$4100, \$4200, \$4300, \$4400, \$4500, \$4600, \$4700**.

С такой организацией экран можно разбить на три части:

- 1-я: адреса **\$4000-\$47FF** описывают линии 0-63,
- 2-я: адреса **\$4800-\$4FFF** линии 64-127,
- 3-я: адреса **\$5000-\$57FF** линии 128-191.

Выглядит непонятно, но такая адресация удобна для печати символов 8x8.

Наберите на Бейсике программу:

```
10 FOR n=0 TO 6911: POKE 16384+n,PEEK n: NEXT n
```

И увидите, как экран заполняется в соответствии с описанием.

Как работает инструкция **PUSH HL**?

Регистр **SP** (указатель стека) уменьшается на 1, по адресу, хранящемуся в **SP**, помещается значение регистра **H**, затем **SP** уменьшается на 1, и по адресу, хранящемуся в **SP**, помещается значение регистра **L**. Пример (**void.asm**):

```
device zxspectrum128
ORG #6000
begin
di
jr $
ld sp,$4002
ld hl,$0100
push hl
jr $
end
display /d,end-begin
savesna "!void.sna",begin
```

Посмотреть на работу удобнее в отладчике.

Довольно теории, пора разобрать пример **stack\_pt.asm**. Как было сказано, используется узор 8x8 (смайлик). Для удобства узор продублирован:

```
pat:
db %01111100,%01111100
db %10000010,%10000010
db %10111010,%10111010
db %10000010,%10000010
db %10101010,%10101010
db %10000010,%10000010
db %01111100,%01111100
db %00000000,%00000000
```

Процессор Z80 предоставляет 4 регистра-пары – **AF,BC,DE,HL** и 4 альтернативных – **AF',BC',DE',HL'**. Поэтому будет удобнее использовать все пары, их значения получаются с помощью стека:



```
ld sp,pat
pop hl,de,bc,af
exx
exa
pop hl,de,bc,af
exx
exa
```

Данные узора уже есть в регистровых парах, заполнение экрана используется так:

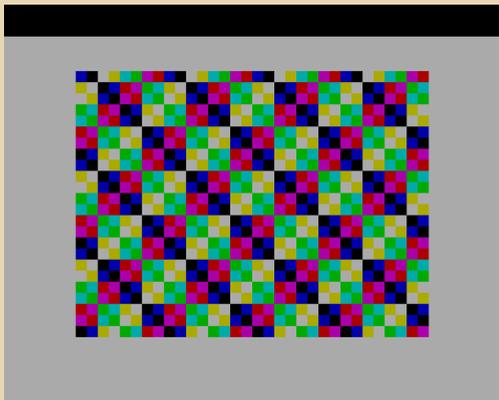
```
ld sp,$5000; новый указатель памяти
                на конец экранной па-
                мяти
```

```
dup 128
push hl
edup
```

128 раз выполненная инструкция **push hl** заполнит 256 байтов участка памяти **\$4F00-\$4FFF**, инструкция **push de** заполнит еще участок выше. Так, с использованием 8 регистровых пар, заполняется вторая треть экрана.

С другими примерами несложно разобраться, одинаковая процедура вызывается с разными параметрами (начало стека для заливки).

Однако использование стековых инструкций можно применить и по другому назначению:



Пример (**gr8z.asm**) не содержит комментариев, оставлю для самостоятельного изучения:

```
device zxspectrum128
ORG #6000
begin
```

```
zz:
ei
halt
```

```
ld a,0
out ($FE),a
ld (back_sp+1),sp
n=0
```

```
dup 8+8+8
ld sp,$5800+n*32+15
pop af,bc,de,hl
exx:exa
pop af,bc,de,hl
inc sp
push hl,de,bc,af
exx:exa
push hl,de,bc,af
```

```
ld sp,$5800+n*32
pop af,bc,de,hl
exx:exa
pop af,bc,de,hl
inc sp
push hl,de,bc,af
exx:exa
push hl,de,bc,af
```

```
n=n+1
edup
```

```
back_sp:ld sp,0
```

; нарисовать линию узора XOR  
; один недостаток – при сдвиге эта линия заливается трешем

```
ld hl,$5800
ld b,24
ld de,32
```



mm:

```
ld a,0
xor b
```

```
and 7
ld c,a
add a,a
add a,a
add a,a
or c
ld (hl),a
add hl,de
djnz mm
```

```
ld hl,mm+1
inc (hl)
```

```
ld a,7
out ($FE),a
jp zz
```

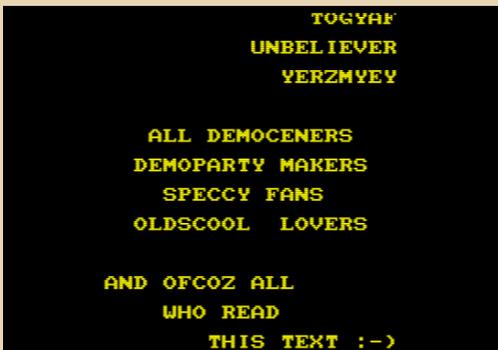
end

```
display /d,end-begin
savesna "gr8z.sna",begin
```

Один недостаток примера – это размер, код весит 1198 байт. В следующей статье я опишу, как сформировать подобные процедуры.

Надо заметить, что стек использовался в различных демо:

**Multimatograf 9** ([ссылка](#)) – вертикальный скролл текста.



**Zombie TV** ([ссылка](#)) – тот же.



Другая задача – вывод на экрана спрайта или анимация на фиксированной позиции.

**HNH 2014** ([ссылка](#)).



Для решения задачи понадобится справочник опкодов Z80 и калькулятор.

Переброска спрайта не решается **LDIR** (21/16 тактов) или цепочкой **LDI:LDI** (16 тактов).

Решается следующей процедурой:

```
POP HL
LD ($4800),HL
```

```
POP HL
LD ($4802),HL
```

```
...
```

```
POP HL
LD ($4810),HL
```

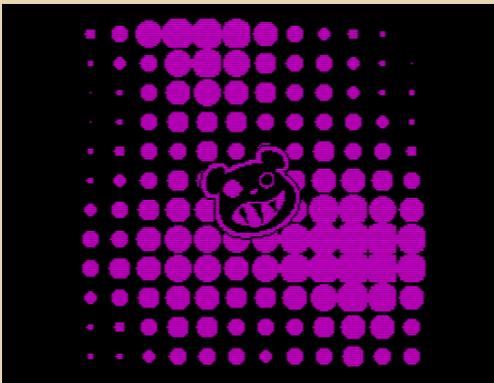
Выходит  $(16+10)/2=13$  тактов на байт.

При использовании стека нужно учитывать один нюанс: выполнение процедуры должно закончиться за 1 фрейм, иначе произойдёт прерывание, которое испортит данные, хранящиеся на стеке.

Исключение:

**Disco Bears:**

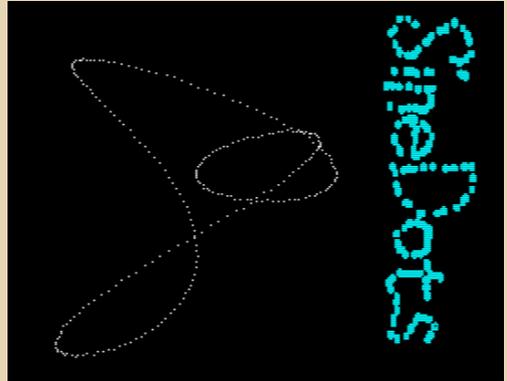
[https://zxaaa.net/view\\_demo.php?id=8940](https://zxaaa.net/view_demo.php?id=8940)



Здесь, при выводе узора, данные поднимаются со стека, и те же данные формируются заново.

**SineDots:**

[https://zxaaa.net/view\\_demo.php?id=10055](https://zxaaa.net/view_demo.php?id=10055)



Для построения точек используются адреса, которые поднимаются со стека. (И эти же адреса используются при очистке точек.)

К сожалению, время не позволяет остановиться на рассмотрении всех вопросов, связанных с деталями. До встречи в следующем номере.

*Прим. ред.: описываемые в статье исходники и готовые SNA-образы можно скачать здесь:*

[http://damaq.in/N21/push/push\\_files.rar](http://damaq.in/N21/push/push_files.rar)



# NFS Hot Pursuit 2 через Wi-Fi



**Б**ольшинство игр, как старых, так и новых, имеет режим для совместной игры по сети. В принципе, это ни для кого не секрет, но до последнего времени для домашних пользователей сетевой режим практически не имел никакого интереса. В семье обычно находился только один персональный компьютер, на который и устанавливалась игра, а к компьютеру выстраивалась очередь из желающих показать класс в стрелялке, гонках и даже в тетрисе или цветных линиях. Но времена меняются – в одной семье может быть не только компьютер, но и планшетка, телефон, ноутбук, и всё это обычно подключается к одному роутеру, который раздаёт интернет на все устройства.

Но кроме интернета, как догадался читатель, этот роутер и Wi-Fi возможности портативных устройств можно применить для сетевых баталий в... да, вы правильно поняли, например, в *Need for Speed Hot Pursuit 2*. Собственно, автор особо ничего изобретать не собирается и протестировал игру в сетевом режиме на стационарном компьютере, работающем под Windows XP, и нетбуком под управлением Windows 7. Как выяснилось, настройка сетевой игры имеет определённые тонкости, которые будут изложены в настоящей статье.

## Готовим сеть

В принципе, особо ничего готовить не нужно. Просто кое-что нужно будет проверить и зафиксировать. Во-первых, перед стартом игры следует определиться, какой из компьютеров возьмёт на себя обязанности сервера. Автор решил, что на эту роль подойдёт стационарный компьютер, подключенный к роутеру через

Ethernet. Ноут же будет подключаться через Wi-Fi.

Как уже было сказано выше, ничего настраивать нет необходимости, но необходимо узнать и записать IP-адрес компьютера. Если Вы выделили статический адрес и помните его, ничего записывать не нужно. Если же адрес назначается роутером, то необходимо узнать его и записать или запомнить.

В XP это сделать достаточно просто – открываем папку **«Сетевые подключения»**, щёлкаем по значку подключения к роутеру, после чего открывается окно состояния подключения (см. рис. 1).

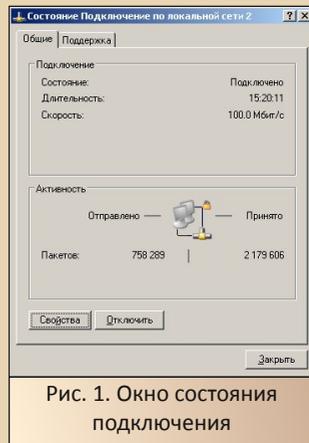


Рис. 1. Окно состояния подключения

Щёлкнув по вкладке **«Поддержка»**, Вы увидите диалог, показанный на рис. 2, где можно узнать, какой IP-адрес назначен компьютеру, если назначение адресов происходит посредством DHCP. Этот адрес нам понадобится при настройке сетевой игры, поэтому его следует либо запомнить, если он такой же простой, как у автора, либо записать.



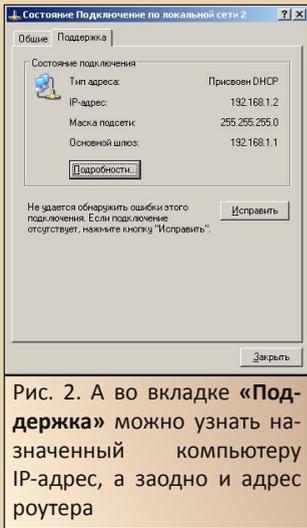


Рис. 2. А во вкладке «Поддержка» можно узнать назначенный компьютеру IP-адрес, а заодно и адрес роутера

После того, как мы узнали IP-адрес стационарного компьютера, который будет у нас игровым сервером, запустили ноутбук, подключили его к Wi-Fi (можно даже для проверки открыть какой-нибудь сайт), можно приступать к настройке самой сетевой игры.

### Настройка сервера игры

Собственно, настройка сетевой игры тоже достаточно проста, но имеет ряд особенностей. Так как нам предстоит производить настройку и на сервере, роль которого играет стационарный компьютер, и на клиенте – ноутбуке, то для успешного запуска сетевой игры важную роль играет последовательность действий – какие действия за какими производить. Что нужно сделать вначале на компьютере-сервере, чтоб клиент мог подключиться, а потом – какие кнопки нажимать на клиенте. Вроде бы мелочи, но несоблюдение последовательности, раннее нажатие той или иной кнопки могут привести к тому, что клиент не обнаружит сервера с игрой, сервер не дожждётся подключения клиента и начнёт игру раньше и всё в таком духе.

Для начала в меню игры необходимо открыть настройки сетевой игры. В зависимости от локализации, данный пункт будет назы-

ваться «Сетевая игра» или «Мультиплеер». Игра предлагает три режима совместной игры, которые можно выбирать с помощью стрелок вправо и влево на экране:

- Игра по локальной сети.
- Игра через интернет.
- Частная игра через интернет.

Казалось бы, в нашем случае организована локальная сеть, и нужно выбирать игру по локальной сети, но нет. Наш выбор – частная игра через интернет (см. рис. 3).

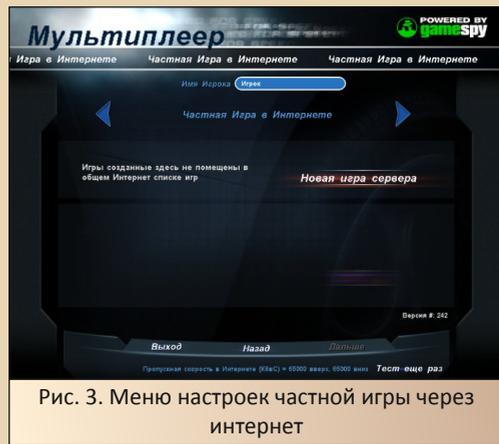


Рис. 3. Меню настроек частной игры через интернет

На компьютере, который будет исполнять роль сервера, необходимо нажать «Сервер новой игры». Откроется меню настроек игры (см. рис. 4), где можно будет выбрать трассу, настроить количество кругов и другие параметры заезда.

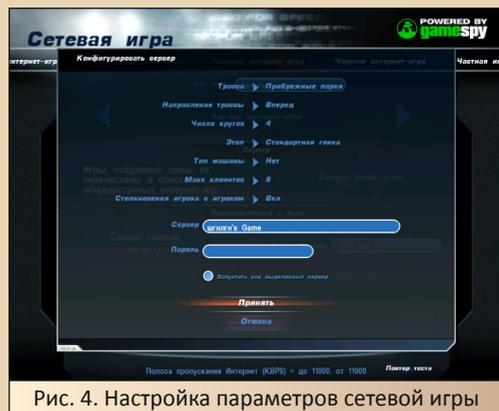


Рис. 4. Настройка параметров сетевой игры



Кстати, тип гонки всего лишь определяет либо обычную гонку, либо игру на выбывание. Режим погони с управляемыми компьютером полицейскими найти не удалось, хотя было бы интересно, чтоб кроме машин, управляемых реальными игроками, в сетевой игре были бы и управляемые компьютером полицейские. После нажатия кнопки «Принять» на сервере начнётся загрузка игры. После этого откроется меню выбора марки и цвета машины (см. рис. 5).

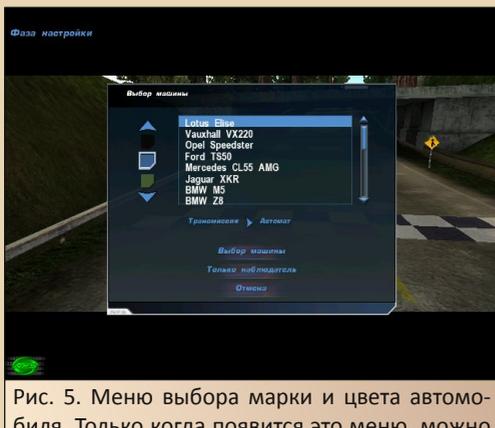


Рис. 5. Меню выбора марки и цвета автомобиля. Только когда появится это меню, можно осуществлять подключение клиента к игре. Лучше всего будет оставаться в этом меню до тех пор, пока игра на клиентском компьютере не запустится и клиент не получит аналогичную менюшку

Только в этот момент клиент может подключиться к игре. Раньше – когда будет открыто меню настройки параметров сетевой игры или же будет производиться загрузка – сервер не будет отвечать на запросы клиента, поэтому соединение не произойдёт. Но даже когда будет открыто меню выбора цвета и марки вашего железного коня, не торопитесь нажимать кнопку начала гонки. Даже если клиент успешно подсоединился к серверу, вполне возможно, что ему не хватит времени, чтобы загрузить игру. Поэтому лучше перестраховаться и держать меню выбора машины открытым, пока у клиента полностью загрузится игра и появится такое меню – тогда можно начинать гонку.

## Настройки клиента игры

Настройки клиента игры куда проще. Как и на сервере, в начальном меню игры необходимо выбрать пункт «Сетевая игра» или «Мультиплеер», а потом частную игру через интернет.

Однако на клиенте нас будет интересовать не область «Сервер», а область «Присоединиться к игре», где необходимо будет ввести IP-адрес игрового сервера, который мы глядели в самом начале (см. рис. 6), а после, когда на экране компьютера, выполняющего роль сервера, появится меню выбора машины, нажать кнопку «Соединение».



Рис. 6. Настройка сетевой игры со стороны клиента. Как видно, в поле адреса сервера введён IP стационарного компьютера

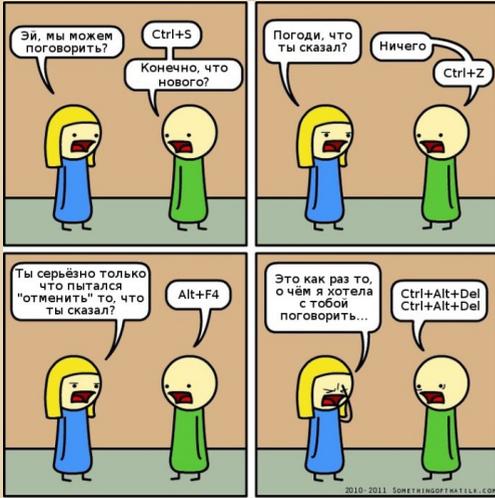
После этого начнётся подключение к игровому серверу, загрузка игры и появится, как на сервере, меню выбора марки и цвета машины. Теперь можно смело выбирать авто, красить его в нужный цвет – как клиенту, так и серверу – и начинать гонку.

Как легко понять, сетевой запуск игры лучше проводить, когда клиент и сервер находятся в одной комнате – так будет проще узнать, когда готов сервер и подключился ли уже клиент. Согласен, процедура достаточно хитрая, но ощущения от сетевого заезда всё равно с лихвой компенсируют слегка мудрёную настройку.

Андрей Шаронов (Andrei88)

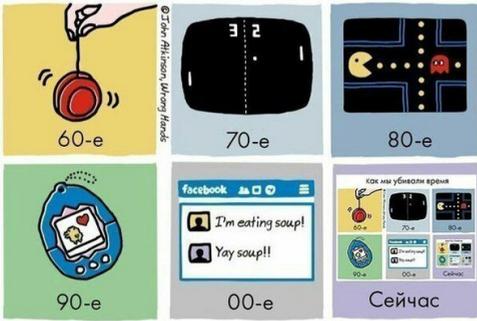


# Просто разный юмор

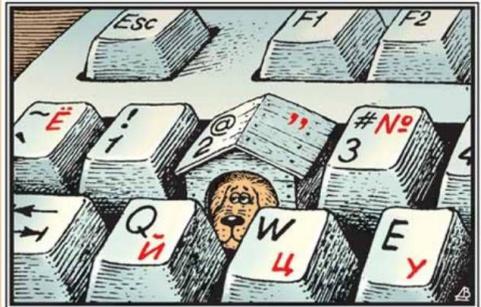


Надпись на футболке: «Жизнь слишком коротка, чтобы безопасно извлекать USB»

## Как мы убивали время



## ТИТАНИК XXI ВЕКА



Над журналом работали

Дизайн/вёрстка/главный редактор –  
uav1606

Редактор – Вячеслав Рытиков  
(eu7pc)

Авторы:

Вячеслав Рытиков (eu7pc)  
Андрей Шаронов (Andreid88)  
uav1606

Роман Азарин  
Пётр Семилетов  
Sh

Интервью:  
bearwindows

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:  
uav16060 [собака] mail.ru