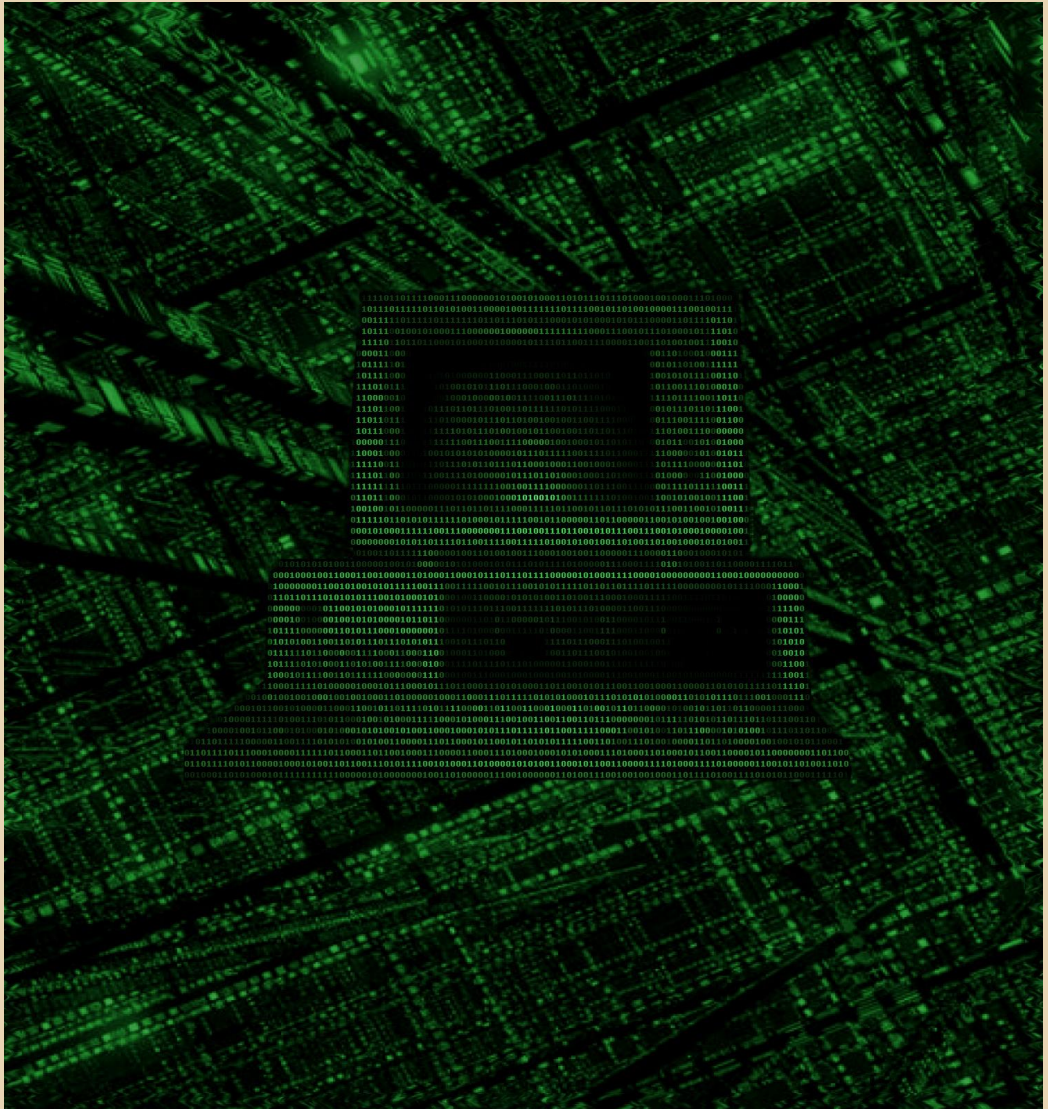


Downgrade

N 5 (BECHA 2011 - BECHA 2012)



СОДЕРЖАНИЕ

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

ТЕОРИЯ DOWNGRADE

Новости, события, комментарии (uav1606&eu6pc) -----	4
SGWW - история одной группы (eu6pc) -----	6

DOWNGRADE-ЖЕЛЕЗО

AVR-ChipBasic. Современный ретро компьютер (Kakos_Nonos) ----	8
---	---

DOWNGRADE-СОФТ

Win9x + Dosx = ... Atlantis word processor! (eu6pc) -----	13
Небольшой обзор DISCo Commander'a для Win95 (Г.Кузнецов) ---	14
Каталог даунгрейдера: все, что нужно, чтобы коллекционировать старый софт (Г.Кузнецов и М.Буторин) -----	15
Scumm VM - виртуальная машина времени (eu6pc) -----	18
Запуск DOS приложений в Ubuntu (Г.Деревлёв) -----	19
Виртуальный FDD (eu6pc) -----	21

СТАРЫЕ ИГРЫ

System Shock (Абадябер) -----	23
DVD-PG: интерактивные игры на бытовом DVD-плеере (eu6pc) ---	30

ПРОГРАММИРОВАНИЕ

Идеальный транслятор (Д.Караваев) -----	32
Программирование на Альтаир 8800 (Kakos_Nonos) -----	39
Проигрывание трековой музыки в Borland Pascal (Kakos_Nonos)	41
Воспроизведение AY-музыки на Спектрамах без AY (Kakos_Nonos)	42

ПРОСТО ЮМОР

Просто разный юмор -----	44
--------------------------	----

Над журналом работали -----	46
-----------------------------	----



ОТ РЕДАКТОРА

Здравствуйте, уважаемые читатели. Уже больше года прошло с выхода прошлого номера журнала Downgrade...

Признаю, что задержка с текущим выпуском – целиком и полностью моя вина. Мне действительно стыдно, что я так задержал выход этого номера... Прошу прощения у авторов и читателей за это. И благодарю всех за терпение.

Причина этой задержки проста – я наконец-то нашёл работу, но она, к сожалению, не оставляет мне времени и сил на что-либо другое...

Тем не менее, номер вышел, и, думаю, журнал будет выходить и дальше. В конце концов, если работа мешает хобби, то к чёрту такую работу. ;-)

Спасибо всем авторам за статьи и прочую помощь с журналом, особенно eubrc, который, как обычно, больше всего помог мне с этим номером.

uav1606.

НОВОСТИ, СОБЫТИЯ, КОММЕНТАРИИ

Безусловно, в этот обзор попали не все интересные downgrade-новости за год, прошедший с выпуска №4 журнала. Таких новостей очень много, и, боюсь, если бы я опубликовал их все, то журнал бы состоял целиком из новостей. :-) Поэтому тут приведены лишь некоторых из новостей ИТ-индустрии и downgrade, привлечшие моё внимание (или внимание еибрс) – прим. редактора.

Новый журнал про старые игры «Игротоп»

Вышел в свет первый выпуск журнала «Игротоп», посвящённого старым играм. Издатель и главный редактор – Алексей Головатюк. В №1 журнала представлен обзор 10 игр, выпущенных в 1990 году. В дальнейшем планируется посвящать каждый новый номер следующему году, т.е. №2 будет посвящён играм 91-го года. Журнал абсолютно бесплатен и доступен для скачивания в формате PDF. Сайт журнала – <http://www.igrotop.org/>

Конкурс статей 2011 года на Железных Призраках Прошлого.

30 декабря 2011 года завершился конкурс статей на сайте Железные Призраки Прошлого. Победителем стал Arif-RU со статьёй «Когда "мини" значило "большой". Plurimat S – мини-ЭВМ третьего поколения из 1970-х.». С результатами конкурса и статьями можете ознакомиться здесь:

<http://www.phantom.sannata.ru/konkurs/2011/>

На eBay картридж с «Тетрисом» продают за \$1 млн.

На eBay был выставлен картридж с игрой «Тетрис» по цене в 1 миллион долларов. Это очень редкое японское издание этой игры для приставки Sega Mega Drive – существует не более 8 экземпляров данного картриджа. Кроме того, на картридже стоит автограф разработчика «Тетриса» Алексея Пажитнова. Судя по [странице](#), лот так и не был продан.

FTP исполнилось 40 лет

16 апреля 1971 года студент Массачусетского технологического института Абхай Бушан опубликовал спецификацию RFC 114 (<http://tools.ietf.org/html/rfc114>), в которой и были описаны основные особенности протокола FTP (File Transfer Protocol).

Некоторое время спустя на смену оригинальному RFC 114 пришел стандарт RFC 765 (1980), а затем RFC 959 (1985), который действует до сих пор с учетом различных дополнений.

Несмотря на такой почтенный возраст, протокол FTP до сих пор широко используется на просторах интернета.

FTP по-прежнему развивается. Например, сейчас обсуждается расширение набора команд для запроса уменьшенных копий изображений, операций с целыми ветками директорий с вложенными папками, запроса доступного дискового пространства, размера папки и ее содержимого и т.п.

Новая версия SCUMM VM.

27 января 2012 года была выпущена новая версия эмулятора для запуска старых игр

SCUMM VM – 1.4.1. В этой версии (по сравнению с релизом 1.4.0) были исправлены некоторые недоработки, добавлена поддержка новых игр и т.п. С релизом вы можете ознакомиться на официальном сайте:

<http://www.scummvm.org/news/20120127/>

Так же в этом номере вы можете прочитать статью о данной виртуальной машине.

Chaos Construction 2011

27-28 августа 2011 года в городе Санкт-Петербурге прошёл очередной фестиваль Chaos Construction. В рамках фестиваля были проведены различные семинары по IT-технологиям, конкурсы, в том числе по графике и демосценам для ZX-Spectrum'a и других ретро-компьютеров. Также были проведены конкурсы по созданию различной chiptune-подобной музыки, по программированию – и многое другое. Как обычно, работала выставка старых компьютеров. Подробнее с хрониками фестиваля вы можете ознакомиться на его официальном сайте:

<http://party11.cc.org.ru>

Chaos Construction 2012 планируется провести 17-19 августа 2012 года.

Выставка истории игр в Париже.

10 ноября 2011 года в Париже открылась выставка Game Story, посвящённая истории видеоигр. На выставке представлены 80 видеоигр, выпускавшихся с 1970-х годов по наши дни. Выставочная площадь экспозиции «Game Story» в Гран Пале – 650 м². В рамках выставки прошли мастер-классы, презентации видеоигр, различные конференции и т.д. Официальный сайт выставки:

<http://www.rmn.fr/Game-Story/>

В Германии разрешили продавать DOOM

С 31 августа в Германии снят запрет на продажу игр DOOM и DOOM II. Этот запрет был

введён ещё в 1994 году в связи с излишней жестокостью данных игр. Решение о снятии запрета принято после апелляции компании Bethesda Softworks, нынешнего владельца id Software. По словам государственных экспертов, сейчас DOOM представляет лишь исторический и научный интерес, и вряд ли привлечёт внимание детей и подростков.

Игре присвоен возрастной ценз 16+.

Запрет по-прежнему действует для одной из американских версий игры DOOM II, где присутствуют два скрытых уровня в стиле Wolfenstein 3D с нацистскими символами.

Вышел 1-й номер журнала World Of Kbytes.

Вышел в свет первый номер журнала World Of Kbytes. Журнал посвящён киберкультуре: *«это и киберпанк в традиционном понимании, различные технические аспекты, медиа искусство, и многое другое»*. В вышедшем номере есть статья, имеющая отношение к downgrade – это обзор альбома группы 386DX, музыка и вокал в котором исполняются на 386-м компьютере. Сайт журнала:

<http://wokzine.com/>

Обзор подготовили:
Вячеслав Рытиков (eu6pc)
uav1606

SGWW — ИСТОРИЯ ОДНОЙ ГРУППЫ



Aбсолютное большинство пользователей, да и, вообще, IT-специалистов негативно относятся к компьютерным вирусам. Однако, вопреки общему желанию, само существование вирусов – неотъемлемый атрибут современной IT-индустрии. Встретить вирус на компьютере пользователя можно довольно часто. Вирусы постоянно развиваются, мигрируют на разные платформы. И вот очередной пользователь гневно восклицает: да откуда ж они берутся?!

Stealth Group World Wide (SGWW) – ответственная группа (клуб) создателей компьютерных вирусов, которая зародилась в Киеве в начале 90-х годов. Известность группа получила благодаря электронному журналу «Infected Voice» («IV»), изданием которого SGWW заявила о своем существовании. Первый выпуск вышел в свет в сентябре 1994 года в Киеве за авторством всего двух человек: собственно LovinGod (президент SGWW) и Eternal Maverick. На страницах первого, пусть и маленького выпуска, прозвучал призыв к сотрудничеству и обмену информацией в сфере вирмейкерства. Это было начало.

На просторах СНГ журнал обрел огромную популярность, в первую очередь среди программистов и людей, тесно связанных с компьютерами. Эпоха 90-х – бум компьютеризации. Хотя персональный компьютер был дома далеко не у каждого, а в учебных заведениях использовались отечественные ЭВМ ЕС, в воздухе носился ветер IT-перемен. Пользователей интернета насчитывалось крайне мало, а для обмена информацией использовались BBS и сеть FidoNet. На компьютерах пользователей ютилась MS-DOS, соседствующая с Windows 3.x или 95.

Журнал появился «в нужное время в нужном месте», произведя эффект взрыва в компьютерных кругах.

Состав группы SGWW расширяется, материалы становятся разнообразнее и интереснее. Для нас, в то время – студентов-программистов, «Infected Voice» был больше чем журнал. Это была жемчужина, кладезь знаний, зачитываемый до дыр. В то время программисты создавали БД на FoxPro, новоявленные геймеры играли в DukeNukem 3D, а на лекциях нам без энтузиазма рассказывали о Turbo Pascal 3.0. Зато дома, открывая очередной выпуск «IV», можно было погрузиться в новый мир – новых идей, технологий и алгоритмов. Мир компьютерного юмора и андеграунда. Именно благодаря журналу мы узнали о том, что на простом Паскале, оказывается, можно писать резидентные программы (да еще какие!), что можно написать вирус чисто на командном языке BAT-файлов, как работает шифрование – и еще много-много новой и полезной информации. Притом информации не столько для создания вирусов, сколько, вообще, для знания архитектуры ОС и программирования. Говоря без лукавства – знания, почерпнутые из «IV» в те далекие годы, для меня полезны по сей день.

Но, как этого и следовало ожидать, журнал полюбил не все. Было и огромное количество противников и ненавидящих SGWW. Причиной стал резкий всплеск вирусной активности. Новые вирусы, созданные под эгидой SGWW, породили эпидемии местного масштаба во многих городах нашей бывшей Родины. Падали BBS, страдали данные на дискетах и жестких дисках. Один из вирусов уничтожил данные на компьютере посольства США в Киеве. В Fido шла информационная война, а если говорить проще – группу и всех ее участников поливали грязью и даже угрожали физической расправой. Но журналу это никак не повредило, даже наоборот, получился этакий черный PR.

Однажды фидошники организовали дежурство в отделении почты, где физически был расположен абонентский ящик SGWW. Но вир-

мейкеры оказались предусмотрительными, и письма приходила забирать бабушка одного из них...

В конце 1995 года президент клуба, LG, переезжает в Москву. Там он пытается организовать московский филиал клуба, и выпускает первый выпуск Московской версии журнала «Infected Moscow». Киевские ребята продолжают выпускать «IV» в столице Украины.

SGWW не было чуждо чувство юмора. В Москве ребята периодически звонили Лозинскому Д.Н. (автору антивируса «Aidstest»), поздравляли с праздниками, желали творческих успехов, посылали минимальные денежные переводы. Дмитрий Николаевич удивлялся такой наглости и реагировал весьма болезненно.

Но ничто не вечно в этом мире. Так сложилось, что 7 февраля 2001 года LG объявляет о ликвидации SGWW. К тому времени уже действовала статья УК «за распространение вредоносных программ», разногласие и споры среди участников учащались, к тому же сам президент, мягко говоря, разочаровался в деятельности своей группы. Брэнд «SGWW» стал таким «попсовым», чем-то модным что ли, утратив свое истинное предназначение.

Всего группа выпустила 12 выпусков «Infected Voice», 2 выпуска «Infected Moscow», 2 номера на английском языке, и 15-ый выпуск, который сначала появился в бумажном варианте, а чуть попозже – онлайн. За всю историю в состав группы входило более 100 человек (не считая большого числа кандидатов и всячески помогающих журналу). Кроме того, было около 10 человек в Бишкеке, а также некоторые отдельные люди в Бразилии, Италии, Аргентине, Колумбии и Австралии.

Также хочется вспомнить о сети «NASNet», которую создала и продвигала SGWW. Название расшифровывается как «настоящая антисоциальная сеть» (Natural Antisocial Network). Сеть зародилась в Киеве и по устройству повторяла FIDO. Но эта задумка не удалась: NASNet превратилась в некоторое неформальное молодёжное движение с еженедельными массовыми пьянками и конфликтами с милицией. На финальном этапе сеть жила уже своей, отдельной от группы жизнью. Количество пользователей составляло около 150 человек в Киеве и примерно 30 в Москве. Связать Киевскую и Московскую подсети так и не удалось.

Что было потом?

После развала SGWW, по словам LG, им была основана ExGW (extended) – то была некая закрытая группа, лаборатория молодых людей, живущих вместе. Информации о данном этапе в жизни группы крайне мало. Однако известно, что интересы были больше ориентированы на реальную жизнь, и не упирались в одно лишь программирование. В конце своего пути лаборатория занималась проектом создания некоего супер-вируса. Но дописать так и не смогли. ExGW приказала долго жить. А спустя несколько лет один из бывших участников под ником «Товарищ Садист» (в реале Кузнецов Леонид Николаевич) – покончил с собой, выбросившись из окна 19-го этажа. Он успел написать два автобиографических рассказа, которые немного проливают свет на жизнь и отношения между участниками группы, показывая их не с лучшей стороны.

Что осталось теперь?

Ныне бывший президент SGWW занимается продвижением своей собственной философии – «Свободомыслие». Пообщаться с ним, а также ознакомиться с его своеобразной точкой зрения на окружающий мир можно по адресу www.dooma.ru

Послесловие.

Теперь, когда прошло столько лет, страницы «Infected Voice» кажутся немного наивными, а исходники – местами примитивными. Мы можем вспоминать те времена с улыбкой или негодованием. Но, несмотря на все, существование Stealth Group World Wide – яркое событие в истории отечественной вирус-сцены. А историю, как известно, забывать нельзя.

(Все выпуски журнала IV вы можете скачать по [этой](#) ссылке (пароль the_lg). Также вы можете посетить музей SGWW по адресу sqww.net - прим. редактора)

Вячеслав Рытиков (eubrc)



AVR-ChipBasic



Современный ретро-компьютер

Время идёт, и раритетных компьютеров становится всё меньше и меньше. Какие-то ломаются, какие-то разбирают-ся на запчасти, а какие-то просто выкидываются. К сожалению, этот процесс не остановить — ничего вечного нет.

Но если взять и придумать такой компьютер, который по всем параметрам походил бы на ретро-компьютеры, но был сделан на современной элементарной базе? И такой компьютер есть — это AVR ChipBasic.

1. Предисловие



AVR-ChipBasic — серия радиолюбительских компьютеров, разработанных немецким инженером Йоргом Вольфрамом (Joerg Wolfram). Все они представляют собой компьютеры, собранные всего на одной микросхеме — микроконтроллере фирмы Atmel AVR. В этих компьютерах микроконтроллер выполняет функции видеоконтроллера, контроллера клавиатуры, звукового генератора, бейсик-интерпретатора и других устройств, которые в обычных компьютерах состоят из множества микросхем.

Серия AVR-ChipBasic состоит из четырёх моделей.

1. Младшая. Основана на микроконтроллере AtMega8, содержит 1 КБ ОЗУ, 8 КБ ПЗУ. Имеет встроенный интерпретатор языка TinyBasic, генерирует чёрно-белое изображение с разрешением 180*230, подключается к композитному входу телевизора (тюльпану).



Гонки на младшей модели

2. Первая средняя. Основана на микроконтроллере AtMega16, содержит 2 КБ ОЗУ, 16 КБ ПЗУ. Имеет расширенный интерпретатор бейсика. Выводит цветное изображение (8 цветов) с разрешением 180*230. Имеет знакогенератор без строчных букв.



Понг на средней модели

3. Вторая средняя. То же, что и первая средняя, только на микроконтроллере At-Mega32 и с полным знакогенератором.

4. Старшая. Самая мощная модель. Основана на микроконтроллере AtMega644, содержит 8 КБ ОЗУ, 64 КБ ПЗУ. Имеет мощный встроенный интерпретатор бейсика. Генерирует цветное изображение (16 цветов) разрешением 180*230 Единственная из моделей, которая поддерживает тригонометрические функции и графический режим. Все эти компьютеры распространяются свободно по лицензии GNU GPL, на официальном сайте представлены схемы, прошивки и исходники.



Сейчас мы более подробно рассмотрим младшую модель компьютера AVR ChipBasic.

2.Общее описание

Младшая модель компьютера AVR-ChipBasic, как уже говорилось ранее, состоит из всего одной микросхемы – микроконтроллера AtMega8 фирмы Atmel, и поэтому возможности этого компьютера будут определяться возможностями микроконтроллера. Компьютер имеет 1 КБ ОЗУ, 8 КБ FLASH-памяти, в которой хранится интерпретатор бейсика и 512 байт EEPROM, в котором хранится запускаемая программа. Единственный видеорежим компьютера – текстовый 30*23 символа, где каждый символ размером 6*10 точек, и поэтому общее разрешение экрана - 180*230 пикселей. Изображение выводится на обычный бытовой телевизор, который подключается к компьютеру по композитному входу.

Также, в компьютере AVR-ChipBasic предусмотрено подключение внешнего дискового модуля на микросхеме 24с16, в которую помещается четыре программы. Ещё компьютер умеет воспроизводить звук через бипер, соединяется с ПК через последовательный порт для обмена программами и управлять внешними устройствами через 4 линии ввода-вывода.

По характеристикам этот компьютер очень похож на компьютеры 70 - 80-ых годов. Вот в таблице приведены сравнительные характеристики компьютера AVR-ChipBasic и других ретро-компьютеров.

Сравнительные характеристики AVR-ChipBasic и других ретро-компьютеров.									
	AVR-ChipBasic	ZX-80	ZX-81	Apple 1	Sol-20	Altair 8800+VT 100	Радио-86PK	IBM PC	Datapoint 2200
Процессор	AtMega8	Z80	Z80	MOS 6502	i8080	i8080	KP580BM 80	i8086	i8008 совмест.
ОЗУ	1 КБ	1 КБ	1 КБ	4 КБ	8 КБ	256 Б+	32 КБ	64 КБ	16 КБ
Год выпуска	2008	1981	1982	1976	1975	1975	1986	1981	1971
Такт. частота	16 МГц	3,5 МГц	3,5 МГц	1 МГц	2 МГц	2 МГц	1,7 МГц	4,7 МГц	?
Разрешение экрана	30x23	32x24	32x24	40x24	64x16	80x24	64x32	80x25	80x12
ПЗУ	8 КБ	4 КБ	8 КБ	256 Б	1 КБ	нет	2 КБ	64 КБ	?

После открытия редактора можно сразу приступить к написанию программ. В любой момент времени на экране находится вся программа, прокрутки экрана нет. Также нет функции переноса строк, то есть при нажатии клавиши Enter строки смещаются на одну вниз, а курсор просто перемещается к началу следующей строки.

При редактировании есть несколько горячих клавиш:

F1 – Загрузить программу из EEPROM

F2 – Изменить название программы

F3 – Открыть диалог для работы с дисковым модулем

F4 – Запуск программы

CTRL+ALT+DEL – Перезагрузка

CTRL+C – Остановить выполнение программы

CTRL+P – Отправляет скриншот по последовательному порту

А теперь мы рассмотрим одну интересную особенность этого компьютера: посмотрите сами, экранная область занимает 690 байт памяти (30*23) и программа 500 байт (20 строк по 25 символов). Это вместе получается 1190 байт, но как это может быть, ведь памяти у микроконтроллера всего 1024 байта. К тому же нужно ещё несколько байт для системных переменных. А сделано всё вот так хитро: во время редактирования программа находится в видеопамяти и не занимает дополнительного места. После нажатия кнопки F4 программа копируется из видеопамяти в EEPROM, а видеопамять очищается. Далее, программа интерпретируется уже не из ОЗУ, а из EEPROM. После завершения программы она снова копируется в видеопамять, и можно продолжать редактировать.

Такое положение дел позволяет сэконо-

мить память, но ведёт к другим неудобствам. Дело в том, что EEPROM поддерживает всего 100000 циклов перезаписи. Это значит, что если каждый день делать по 100 запусков, то через три года микросхема сгорит. К счастью, у компьютера есть небольшая защита: перед тем, как переписывать программу в ППЗУ, она сравнивается с той, что уже там находится, и если они равны, то программа не переписывается.

После завершения работы программы в правом нижнем углу появляется надпись **PRESS ESC!**. Если программа запускалась клавишей F4, то откроется редактор, а если же программа запускалась автозапуском, то она перезапустится.

5. Бейсик

Компьютер AVR-ChipBasic имеет встроенный интерпретатор языка бейсик. Для экономии памяти используется его диалект Tiny Basic, который не очень сильно отличается от обычного бейсика. Сейчас мы рассмотрим главные особенности этого языка.

Первое, что сразу бросается в глаза, – это сокращённые команды. Вместо **GOTO** используется **GO**, вместо **GOSUB – SUB**, вместо **NEXT – NXT**. Некоторые команды вообще заменены символами. Например, команда **?@2,2;%17** означает вывести в координатах 2, 2 символ номер 17 (кружок). Это делает программы на этом бейсике немного нечитаемыми, но если вникнуть, то всё понятно.

Tiny Basic поддерживает 26 переменных типа byte, каждая обозначается одной буквой (A-Z). Массивы и строки не поддерживаются.

Также стоит отметить ряд интересных возможностей этого языка. Например, несколько способов доступа к клавиатуре. С помощью одного из способов можно сделать, что при нажатии на кнопку «вправо» будет выдаваться значение 1, а при нажатии на «влево» – 255 (-1). С помощью других методов можно отавли-

вать нажатия Ctrl, Shift и других кнопок, и они будут принимать значения 1 и 255. Это сделано для облегчения написания игр, так как в них часто используются стрелки и другие кнопки, которыми надо управлять движением различных объектов.

Ещё одна интересная команда – LIM. Она задаёт границы для значения переменной. Например, команда **LIM D, 10, 20**. После её выполнения значение переменной D будет находиться в диапазоне 10..20. Если до выполнения команды значение переменной было меньше десяти, то оно станет 10, а если же больше 20 – станет 20.

К тому же этот бейсик поддерживает псевдографику разрешением 60*46. Можно рисовать точки, линии и квадраты.

Напишем какую-нибудь простую программу на этом языке. Пусть это будет простенькая игра «угадай число». Вот её полный код:

```
1 a=rv(100)+1:?"Vvedi chis"
2 inp b
3 if b>a:go 6
4 if b<a:go 7
5 ?"Molodec!":no 255:go 0
6 ?"Vvedi menshe":no 5:go 2
7 ?"Vvedi bolshe":no 8:go 2
```

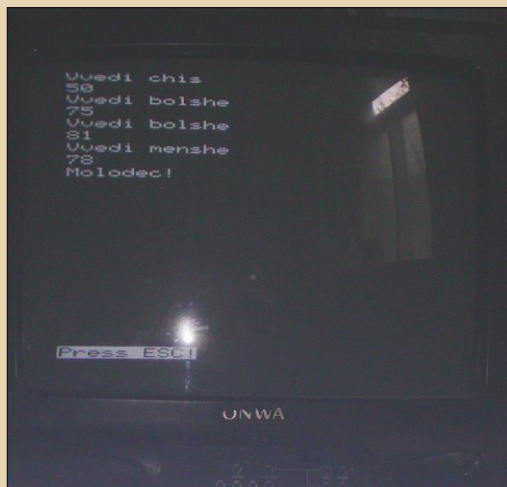
Разберём эту программу поподробнее.

В первой строчке этой программы загадывается случайное число от 1 до 100 и выводится на дисплей.

Во второй – вводится число с клавиатуры.

В третьей и в четвёртой введённое число сверяется с загаданным, и программа отправляется куда нужно.

В пятой-седьмой строчках выводится нужное сообщение, играет звук, и программа отправляется или к вводу, или выходит.



6.3. Заключение

Вот такой вот это компьютер, AVR-ChipBasic. Несмотря на свою простоту, он обладает достаточно большими возможностями. Конечно, есть некоторые недочёты/недоработки, но надо помнить, что в крохотные 8 килобайт памяти, куда и бейсик с трудом влезет, удалось впихнуть бейсик, видеогенератор, контроллер клавиатуры, и чтобы это всё вместе ещё и работало, причём, достаточно быстро.

Поэтому компьютер ARV-ChipBasic можно смело назвать произведением искусства. Работать с ним – одно удовольствие. :-)

Ссылки:

Официальная страница компьютера:

<http://icwolfram.de/projekte/avr/chipbasic8/main.php>

Александр Завгородний (Kakos_Nonos)
<http://kabardcomp.narod.ru/>

Win9x + Docx = ... Atlantis word processor!

Эту краткую статью я хочу посвятить текстовому процессору (исходя из названия) – Atlantis word processor. Это небольшая по объему программа для работы с текстами, которая имеет две приятных особенности: она работает под Win 9x (включая Windows 95!) и умеет работать с форматом Docx.

Итак – подробнее:

Установочный exe-файл версии 1.6.5.7 занимает немногим более 3 МБ. Да, это не опечатка. После установки кликаем ярлык на рабочем столе. Интерфейс Атлантиды на фоне Microsoft Office кажется вполне привычным (рис.1)

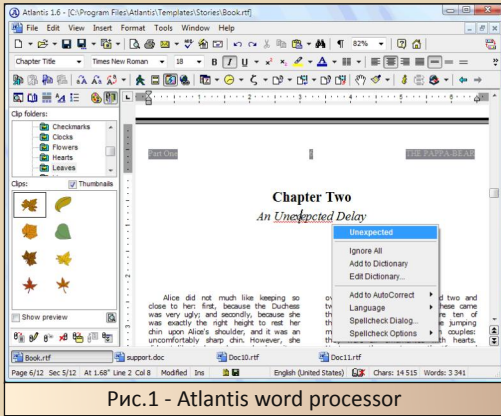


Рис.1 - Atlantis word processor

Возможности у этого «процессора» более скромные, чем у Word, но для несложной работы с текстами – вполне хватает:

- Программа открывает файлы формата rtf, doc, docx, odt, cod, txt, wri, а также может сохранять дополнительно в формате zip, html и erub (формат e-book)
- Доступны стандартные средства для форматирования текста – изменение размера, цвета и типа шрифта, работа с полями документа, вставка картинок, гиперссылок, быстрое разбиение на несколько колонок, работа со списками и т.д. – в общем, необходимый джентльменский набор :)

- Присутствует модуль проверки орфографии (как в Word).

Я не буду тратить время, и описывать остальные возможности программы – при желании вы сами с ней познакомитесь. Хочется сказать пару слов о недостатках:

1. Программа англоязычна, и официальной поддержки русского языка пока что нету. Но это не является проблемой, т.к. в сети можно найти локализованные версии с русскоязычным меню
2. Atlantis не поддерживает таблицы и формулы. Эту проблему разработчики обещали исправить в будущих версиях – программа развивается и обновляется.
3. Программа имеет статус shareware и настойчиво просит денежку за регистрацию.

Несколько слов о работе под 9x: программа устанавливается даже под Windows 95. Но для работы понадобится установить дополнительно библиотеку **unicows.dll** – поддержку unicode для win9x (можно скачать с официального сайта), и обновить версию **comctl32.dll**. Дополнительно на голый 95-ке мне пришлось установить I.E. 5.0 взамен 3-ей версии, без этого Atlantis отказывался запускаться. Как известно I.E. при установке обновляет многие библиотеки в системе, видимо чего-то не хватало.

В общем, программа работает довольно быстро и легко, что не может не радовать. Возможность работать с форматом Docx на совсем слабых компьютерах – отныне не проблема.

Официальный сайт:
<http://www.atlantiswordprocessor.com/>

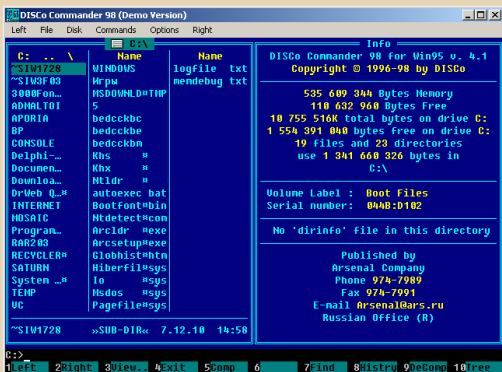
Вячеслав Рытиков (еибрс)

Небольшой обзор DISCo Commander'a для Windows 95



Я думаю, что многим известен файловый менеджер DISCo Commander. Для тех, кто не знает эту программу, кратко опишу ее: DISCo Commander – это файловый менеджер от фирмы Арсеналь для Windows, который напоминает внешним видом старого Norton Commander'a и имеет все его основные функции. В этом обзоре я хотел бы рассказать о версии 4.1 для Windows 95.

Сначала, как обычно, устанавливаем программу. Думаю, никому не надо рассказывать о процессе установки. Там все понятней некуда. После установки запустим программу:



Нашему взору предстанет удивительная картина: Norton Commander с менюшками и диалоговыми окнами Windows. Но чтобы можно было работать, выставим в программе русский язык. Вызовем для этого меню Options, выберем команду Configuration..., в появившемся окошке в секции Language выберем строку Russian/Русский, щелчком кнопку OK и перезапустим программу. Все готово.

С DISCO Commander'ом работа осуществляется так же, как и с Norton Commander'ом. Выделение по клавише Ins, Просмотр (Правка) по F3 (F4), причем при просмотре используется встроенный выюер с массой «наворотов» (инфа по F1), удаление по F8, причем при удалении можно помещать файлы в Корзину. Перетаскивать файлы можно прямо из окна менеджера на Рабочий стол или в Проводник. Любители Norton Commander'a найдут здесь все функции своего любимого файл-менеджера.

Более подробно хочется описать настройки программы. Они сосредоточены в одноименном меню. Первая команда этого меню так и называется, Настройки. Мы в этом окне уже были, выставляли русский язык. Здесь же можно включить очень полезную функцию «Считать размер каталогов». Эта опция позволит видеть размер каталогов. При включенной настройке «Удалять в корзину» удаляемый объект помещается в корзину Windows, и, следовательно, может быть восстановлен позже. Еще там есть довольно полезная функция «Ins перемещает вниз». Если она включена, то при нажатии на Ins будет происходить выделение объекта и переход к следующему. На мой взгляд, очень удобная функция!

Следующая в меню настройка называется «Редактор/Просмотр...». Тут, я думаю, все понятно без комментариев.

Дальше у нас идет команда «Подтверждение...». Эта команда управляет выводом подтверждений некоторых действий. Я думаю, реально необходимы подтверждения только «Удаления» и «Удаления подкаталогов».

Команды «Шрифты...» и «Цвета...» управляют внешним видом DISCO Commander'a. Думаю, там разобраться не составит труда.

Опция «Путь в командной строке» управляет отображением пути в командной строке. Если стоит галочка, то в командной строке отображается полный путь каталога, а если нет, то там отображается имя текущего устройства.

Что делают следующие три команды, понятно из их названия.

Самая нижняя команда сохраняет настройки, чтобы ими можно было воспользоваться при последующем запуске программы.

Вот мы и закончили краткий экскурс DISCO Commander'a. Надеюсь, вам эта программа станет незаменимым помощником при работе с файлами!

Георгий Кузнецов (Герман)

Каталог даунгрейдера: все, что нужно, чтобы коллекционировать старый софт



Как известно, коллекционеры старых программ активно тестируют свои раритеты и складывают их удобным способом. В этой статье мы расскажем, какие программы необходимы для эффективного коллекционирования программ, а ниже будет врезка о версиях программ, и как их правильно определять. Итак, поехали.

WinRAR – архиватор, необходим для упаковки дистрибутивов старых программ и ОС в один файл, создает файлы форматов rar и zip;

7-ZIP – еще один архиватор, создает архивы формата 7z, которые имеют очень высокую степень компрессии;

WinImage – программа, позволяющая работать с образами дискет ima, img и другими, а также их создавать (можно либо добавлять файлы в образ дискеты, либо снимать образы с дискет, есть поддержка дисководов 3.5" и 5.25");

WinISO – данная программа позволяет создавать образы CD и DVD дисков в формате iso (их можно использовать, если необходимо сохранить полную структуру диска и загрузочную область или необходимо перенести программу на виртуальную машину);

VirtualBox – виртуальная машина (ВМ), предназначена для виртуализации ОС (говоря простым языком, можно запустить ОС из-под другой ОС, при этом гостевая ОС будет думать, что она на отдельном компьютере). Позволяет виртуализировать различные устройства ПК. Данная ВМ имеет русский интерфейс;

VMWare Workstation – другая ВМ, более мощная и имеет наибольший функционал. Признаться честно, управление в этой ВМ не очень удобное, главным образом из-за того, что ВМ сильно громоздкая;

VirtualPC – еще одна ВМ. Управление немного напоминает VirtualBox;

Microsoft Compression Utility – еще одна программа даунгрейдера. Включает два файла: **COMPRESS.EXE** и **EXPAND.EXE**. **EXPAND.EXE** позволяет распаковывать файлы, в которых последняя буква расширения заменена знаком '_' (например, **WRITE.EX_**). **COMPRESS.EXE** позволяет произвести обратную операцию;

DECOMP.EXE – утилита, аналогичная **EXPAND.EXE**, только работает с запакованными файлами старых версий MS Office;

HEdit – хороший HEX-редактор.

А теперь будет подробно рассказано про различные эмуляторы. Для тех, кто не знает, эмуляторы – это программы, позволяющие запускать на обычном компьютере программы, не рассчитанные для x86 платформ. Эмуляторов существует великое множество, но в нашем обзоре будет рассказано только о самых распространенных и необходимых коллекционеру.

FMSX – старейший из известных эмуляторов Yamaha MSX. Благодаря поддержке всего ряда компьютеров MSX, а также всех тонкостей их аппаратной архитектуры, эта программа была в свое время портирована на большинство операционных систем, причем качество воспроизведения как графики, так и музыки вполне может соперничать с более поздними версиями эмуляторов;

NLMSX – один из наиболее полных эмуляторов Yamaha MSX. Он поддерживает такие стандарты, как PSG, SCC, Konami MEGA ROM's, MSX-DOS 2.x, имиджи флоппи-дисков объемом 360 Кбайт и 720 Кбайт, а также широкий спектр пользовательских настроек, расширенные мапперы памяти, работу в полноцветном режиме;

CCS64 – самый популярный эмулятор Commodore 64 для x86-х компьютеров. Программа полностью воссоздает архитектуру компьютера, включая точную эмуляцию работы процессора 6510, видеочипа 6567 VIC, звукового процессора 6581 SID (как в 8-битном, так и в 16-битном режиме), а также системы ввода/вывода. CCS64 обеспечивает вывод VGA-графики в разрешениях 384x282, 284x246 и 320x200, работает со звуковыми картами, поддерживающими стандарты SoundBlaster и Gravis Ultrasound. Программа эмулирует оригинальный джойстик Commodore 64, причем как с помощью клавиатуры компьютера, так и с помощью обычного PC-джойстика;

SpecX – полнофункциональный эмулятор ZX Spectrum 48K для персональных компьютеров под управлением Windows 9x, а также Windows NT/2000. Программа обладает прекрасной эмуляцией звукового процессора «Синклера», позволяет производить сохранения, гибко настраивать скорость игрового процесса, а также переключаться между оконным и полноэкранным режимом (даже с возможностью симуляции телевизионного экрана);

M.E.S.S. – пожалуй, наиболее интересном эмуляторе из встречающихся на просторах Сети. От великого множества прочих программ M.E.S.S. отличается тем, что поддерживает все типы старых домашних компьютеров, которые когда-либо пользовались успехом у покупателей: Adventure Vision, Amiga 500 (NTSC), Amstrad PC1512, Amstrad PC1640/PC6400, Amstrad/Schneider CPC464, Amstrad/Schneider CPC6128, Amstrad/Schneider CPC664, Apple IIc (3 модели), Apple IIe (3 модели), Apple I, Aquarius, Atari 2600, Atari 400 (NTSC/PAL), Atari 5200, Atari 7800, Atari 800 (NTSC/PAL), Atari 800XL, Atom, Bally Pro Arcade/Astrocade, BBC Micro Model A, BBC Micro Model B, C64GS (PAL), C65/C64DX (6 моделей), CBM4064/PET64/Educator64 (NTSC), Colecovision, Color Computer, Color Computer 3, Colour Genie EG2000, Commodore 128 (4 модели), Commodore 16/116/232/264 (2 модели), Commodore 30xx (2 модели), Commodore 364 (прототип), Commodore 40xx (3 моде-

ли), Commodore 64 (3 модели), Commodore 80xx (4 модели), Commodore B128, Commodore B128-80HP/710, Commodore B128-80LP/610, Commodore B256-80HP/720, Commodore B256-80LP/620, Commodore Max (Ultimax/VC10), Commodore SP9000/MMF9000, CPS Changer (Street Fighter ZERO), Dragon 32, Enterprise 128 (2 модели), Game Gear, Game Boy, Inves Spectrum 48K+, Jupiter Ace, Kaypro 2x, KC 85/4, KC Compact, KIM-1, Laser 110, Laser 200, Laser 210, Laser 310, Laser 350, Laser 500, Laser 700, Lisa 2, Macintosh 512ke, Macintosh Plus, Master System, MC-10, Megadrive/Genesis, Microbee 32K, Microbee 56K (CP/M), Microtan 65, MSX1, MTX 512, MZ-700, Nascom 1, Nascom 2, Nintendo Entertainment System/Famicom (3 модели), Oric 1, Oric Atmos PC, PC Engine/TurboGrafx 16, PC8300, PC-AT (CGA, MF2 Keyboard), PCW10, PCW16, PCW8256, PCW8512, PCW9256, PCW9512(+), PC-XT (CGA), PC-XT (MDA), PC-XT (VGA, MF2 Keyboard), PDP-1, PET2001/CBM20xx Series (Basic 1), Philips P2000M, Philips P2000T, Pocket Computer 1350, Pocket Computer 1401, Pocket Computer 1402, Power 3000, Prologica CP400, Salora Fellow, Sam Coupe, Sanyo/Dick Smith VZ200, Sanyo/Dick Smith VZ300, Spectrum I+, Superboard II, System-80, Tandy 1000HX, TC2048, Texet TX8000, TI99/4 Home Computer, TI99/4A Home Computer, TK90x Color Computer, TK95 Color Computer, TRS-80 Model I (3 модели), TS1000, TS2068, UK101, VC20 (PAL), Vectrex, VIC20 (NTSC), ZX Spectrum (13 моделей), ZX Spectrum 128 (2 модели), ZX Aszmic, ZX-80, ZX-81;

Emulator3000 – программа, которая эмулирует ряд компьютеров как зарубежных, так и произведенных в СССР. Должное внимание нужно уделить эмуляции почти всех советских микрокалькуляторов;

EMU Version 1.01 – эмулятор, схожий по возможностям с предыдущим, правда, без поддержки микрокалькуляторов, но имеющих более совершенную базу эмулируемых устройств.

Все эти программы позволяют эффективно работать с даунгрейдом и старыми ОС. А

теперь обещанная врезка о версиях программ.

Версии программ: все, что необходимо знать

Как известно, программа имеет номер версии (например, 1.0a или 2.15). На самом деле версия 3.0a – это не совсем правильная запись. Правильнее записывать 3.00a. Буква в записи версии указывает, что багфикс к версии 3.00 (в данном случае первый). Наряду с номером версии довольно часто используется номер сборки. Этот номер показывает, какая по счету эта версия с момента компиляции первой версии программы. Но довольно часто разработчики любят искусственно увеличивать номер сборки для того, чтоб он был красивый. Скажем, была сборка 2562, а станет 2600, а не 2563. Номер сборки записывается через точку после номера версии, например, 3.10.103, что означает программу версии 3.10 и сборку 103. Довольно часто бывает номер сборки из двух или одной цифры. Перед такими номерами ставится такое количество нулей, чтоб длина записи была в три знака (реже в четыре), например, 001, 068, 0015. Также номера сборок могут иметь второстепенный номер, он записывается через точку после основного номера сборки, например, 2332.2, 2380.2. А обычные номера сборок (например, 103) имеют второстепенный номер 0 (103.0). 0 в качестве второстепенного номера обычно не записывается.

Номер версии хранится в файле. Чтоб посмотреть этот номер версии, надо открыть свойства файла и выбрать вкладку Версия (Version). Версия файла записывается строго по формату *MajorVer.MinorVer.Rev.Build*, где *MajorVer* – это первая часть номера версии (в 3.10.0.103 это цифра 3), *MinorVer* – это вторая часть номера версии (в 3.10.0.103 это число 10), *Rev* – эта часть остается чаще всего без внимания и вместо нее почти всегда указывается 0 (в 3.10.0.103 *Rev* заменена 0), *Build* – это номер сборки (в 3.10.0.103 это число 103). Иногда разработ-

чики меняют местами смысл значений *Rev* и *Build*. (Например, в версии 3.50.756.1 число 756 выступает в роли номера сборки, а цифра 1 в роли ревизии). Также к номеру сборки иногда может добавляться буква (например, 3.10.2.014e). Буква указывает, что данная сборка багфикс к родительской (в нашем случае родительская сборка 014)

Эта схема нумерации сборок практически стандартная, но не единственная. Например, в абсолютном большинстве старых программ (и некоторых новых) применяется способ нумерации по версиям. Выглядит он так: *MajorVer.MinorVer*. Например, 4.00. Данный способ нумерации был очень сильно распространен и использовался даже в старых версиях Windows. Есть тройной способ нумерации. Выглядит так: *MajorVer.MinorVer.Build*. Например, 8.00.1371. Есть способ двойной нумерации. Был сравнительно мало распространен и использовался в нумерации версий Windows 32-bit OS (предок Windows NT). Выглядит он так: *MajorVer.Build*. Например, 1.239. Довольно часто к такому номеру добавляют букву v (например, v1.239). Существуют также экзотические способы нумерации, о которых здесь говорить бессмысленно.

Надеюсь, данная статья вам очень поможет в вашем хобби!

Георгий Кузнецов (Герман)
Михаил Буторин (europe)



ВИРТУАЛЬНАЯ МАШИНА ВРЕМЕНИ

В этой статье речь пойдет о запуске достаточно старых игр на новых компьютерах.

Бывают моменты, когда очень хочется окунуться в мир золотых хитов прошлого века, вновь прикоснуться к шедеврам игровой классики. Что же для этого нужно? Казалось бы, что нет проблем: в интернете есть все, скачай да играй, если хочется. Но не все так просто. Запустить старые игры на новых компьютерах подчас довольно сложно. Если отбросить в сторону танцы с бубном вроде копания в настройках совместимости и поиска несуществующих патчей, то остается единственный выход – использовать эмулятор, к примеру, DOS-BOX. Но он требует некоторой настройки, базовых знаний MS-DOS, да и вообще не очень то удобен в работе. Есть ли еще варианты? Есть! В этом случае нам поможет ScummVM – виртуальная машина для запуска старых добрых игр.

Эта замечательную программу можно скачать с официального сайта:

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

Чем же она так замечательна?

Во-первых, размером. Установочный файл под Win занимает всего около 4 МБ!

Во-вторых, Многоплатформенностью! (с большой буквы). ScummVM адаптирована для работы под Windows, различные дистрибутивы Linux, Mac OS, PS2, Win CE, iPhone, Web OS, Symbian, Android. Впечатляет, правда? Продолжаем: DreamCast, Nintendo 64/GameCube, Solaris, Amiga OS, Atari, и даже версии для телевизоров Samsung!

В-третьих, принципом действия. В отличие от других эмуляторов, ScummVM не эмулирует какую-либо ОС, он является эмулятором движка игры: при старте не запускается исполняемых EXE или COM файлов. Эмулятору нужны лишь файлы с ресурсами игры: графикой, музыкой и т.д.

После установки виртуальной машины, единственное, что нам понадобится – указать папку с игрой. Сами игры можно легко найти на специальных сайтах, к примеру, www.old-games.ru.

Список поддерживаемых игр очень внушительный – более 200 шт. Ознакомиться с ним можно на официальном сайте в разделе «Compatibility».

Немного о приятных мелочах: интерфейс эмулятора полностью русскоязычен. Вызов системного меню во время игры – клавиша F5. В системном меню мы можем сохранить или загрузить игру (средствами самой ScummVM). Кроме того, в настройках мы получаем дополнительные возможности по настройке процесса игры: изменить или сделать полноэкранным видеорежим, изменить громкость звукового сопровождения, построить пути.



В заключение хочется добавить, что данный проект абсолютно бесплатен, имеет открытые исходные коды и постоянно развивается. На сайте расположен подробный FAQ с ответами на все интересующие вопросы.

Если вы еще не играли в King Quest, не знакомы с первыми 3-мя частями Goblins, не спасали мир в Day of the Tentacle – вы потеряли полжизни. Обязательно посмотрите эти жемчужины игрового жанра. Доброжелательная атмосфера, искрометный юмор и душевный геймплей не оставят вас равнодушными. Тем более теперь, когда это доступно практически на любой платформе.

Вячеслав Рытиков (еибрс)



Запуск DOS-приложений в Ubuntu



Способ первый – Wine (Самый простой)

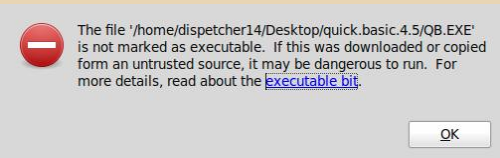
Для запуска Windows приложений в Ubuntu используется «Wine». Но DOS – это же по сути те же самые окна, только без графики. Поэтому Dos приложения можно запускать через Wine. Проведя несколько опытов, я понял, что Wine крайне не годится для нас.

Покажу на конкретных примерах:

1. Qbasic\QuickBasic\GWBasic

Запуск данных приложений не увенчался успехом:

QuickBasic и GWBasic выдали ошибку:



В Qbasic был некий «прогресс»:



Но зато у меня работала программа Winiso и программа настройки драйверов для игры Postal, и эмулятор компьютера Альтаир.

Так же Wine не поддерживает .com файлы.

В целом, из своего опыта я могу сделать вывод: Wine не подходит для запуска DOS приложений, но неплохо запускает приложения, сделанные для командной строки Windows.

Способ второй – DosBox (Самый лучший)

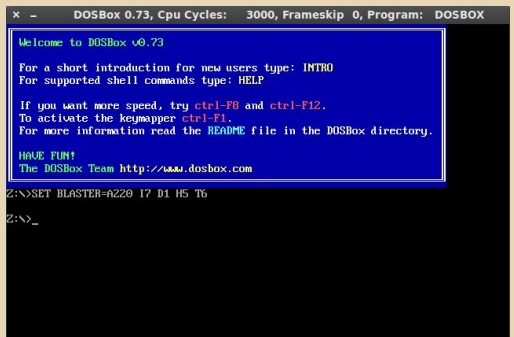
DosBox – свободно распространяемый эмулятор системы MS-DOS, официальный сайт – www.dosbox.com

Итак, начнём с установки. Установить сию программу, как и все остальные, можно тремя способами: скачать с официального сайта, через терминал и через центр приложений.

Я человек ленивый, поэтому устанавливал через терминал:

```
$ sudo apt-get install dosbox
```

Значок запуска находится в разделе «Игры», настоятельно рекомендую вынести его на рабочий стол (позже поймёте зачем). Итак, запускаем. Что мы видим:



А видим мы приветственное окно и командную строку.

Начнём с управления:

Ctrl+F1 – настройка клавиатуры

Ctrl+F8/F12 – регулировка скорости

Ctrl+F11 – подстройка скорости кадров

При запущенном приложении в dosbox'е при нажатии мышью в любую точку окна dosbox'a, мышшь переключается в режим работы в этом приложении, и работать ей с внешними приложениями вы уже не сможете. Для переключения

мыши в нормальный режим необходимо нажать **Ctrl+F10**

Теперь о том, как запускать приложения.

Для запуска приложений существует не сколько способов:

1. Запускаем DosBox и пишем:

mount C «адрес директории с программой»

После этого директория, в которой лежит программа, становится виртуальным диском C:, в котором и нужно вести работу как с обычным DOS'ом. Например:

mount C /home/dispatcher14

/home/dispatcher14 – виртуальный диск C:

2. Подносим значок программы к значку Досбокса.

После этого действия, которые описаны выше, выполняются автоматически, программа сразу запускается.

3. (Не всегда работает) Пишем в терминале:

\$ sudo dosbox «адрес программы».

Например:

sudo dosbox /home/dispatcher14/qb.exe

После этого происходят действия описанные выше.

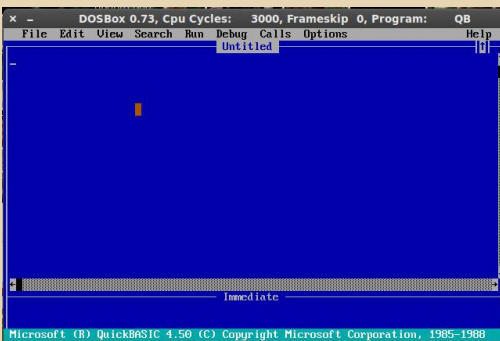
По-идее, используя эти действия, можно написать скрипт для запуска какой-либо программы, но я так никогда не делал.

4. Кликнуть правой кнопкой мыши и выбрать «Открыть в другой программе», и выбрать из появившегося списка DosBox.

(Кстати, именно в этом плюс линуксовского Досбокса перед виндовским – способы 3 и 4 в Windows недоступны)

Теперь покажу конкретные примеры:

QuickBasic:



Игра «Master of Orion»:



Ах да, чуть не забыл об ещё одном полезном свойстве DosBox'a – через него можно запускать непосредственно сам DOS. Для этого надо взять образ загрузочной дискеты, и выполнить его запуск одним из перечисленных выше методов. Вообще, можно запускать не только Дос, но и любую другую ОС, которая грузится с дискеты.

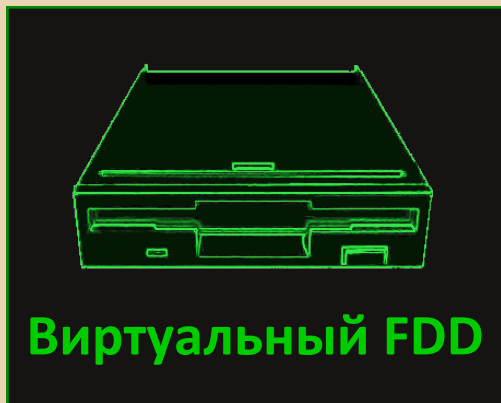
Выглядит примерно так:



В заключение скажу, что кроме DosBox'a и Wine'a существует множество других эмуляторов (Dosemu например), но не один из них не обеспечивает такого эффекта как DosBox. Так же при желании DOS можно запускать через виртуальные машины (такие как Virtual Box), но этот метод плох тем, что в виртуальных машинах, как правило, не эмулируется мышь, да и нужное приложение из-под них не запустишь.

Данный материал написан для «Первого конкурса русскоязычных статей для ОС Ubuntu 2011 года» (konkurs.ubuntu.ru)

Глеб Деревлёв (Dispatcher14)



Вот мы и дожили до того момента, когда устройство под названием дисковод, по сути, отсутствует как класс в современных компьютерах. Появившись давным-давно, дискета долгое время была привычным устройством для обмена информацией между локальными компьютерами. 8", 5,25", 3,5" – как уж дальше и не улучшай, но скорость записи по-прежнему маленькая, объем накопителя по современным меркам тоже копеечный, а о надежности хранения информации, думаю, даже говорить не стоит. На смену пришли маленькие USB флеш-накопители, которые незаметно выжили из системного блока старый добрый 3,5". Но что же делать, если вдруг понадобился дисковод, а под рукой его нету? На помощь придет эмулятор – виртуальный дисковод.

Virtual Floppy Drive (VFD) – это крохотная по размеру программа, которая позволит установить в систему виртуальный дисковод и работать с образами дискет как будто они вполне реальны. Программа имеет «двойной» интерфейс: в комплекте идет консольная версия (`vfd.exe`) и версия с графическим интерфейсом (`vfdwin.exe`, рис 1).

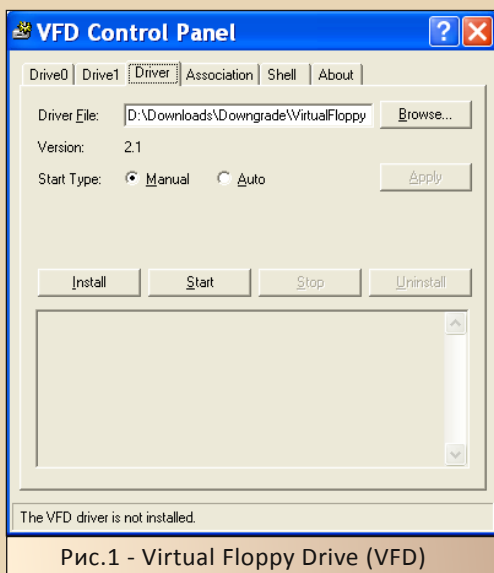


Рис.1 - Virtual Floppy Drive (VFD)

Для работы с виртуальным приводом программа устанавливает в систему свой драйвер. Virtual Floppy Drive позволяет настроить тип и размер эмулируемого диска: от 160 КБ 5,25" до 2,88 МБ 3,5", а также букву логического диска, которая будет привязана к устройству. Кроме того, VFP способен открывать образы дискет в различных форматах: Bin, Dat, Fdd, Flp, Ima, Img, Imz, Vfd. Монтируемый образ можно защитить от записи, имеется возможность сохранения изменений диска обратно в образ.

Консольная версия весьма удобна в работе: программа воспринимает «указания» через командную строку. Если не требуется загружать виртуальный диск вместе с системой (программа это позволяет), то можно сделать два командных файла для включения и выключения дисковода. Например:

ON.BAT:

```
vfd.exe install
vfd.exe start
```

vfd.exe link b:
vfd.exe open boot_dos.ima

OFF.BAT:

vfd.exe remove

Форматировать созданный диск можно только через саму программу, стандартные средства ОС не подходят. Имеется возможность эмулировать сразу 2 дисковода, установить ассоциации файловых образов на VFD.

Официальный сайт:

<http://sourceforge.net/projects/vfd/>

ImDisk Virtual Disk Driver – дистрибутив состоит из одного exe-файла размером около 300 КБ, который при запуске устанавливает ImDisk в панель управления (рис.2).

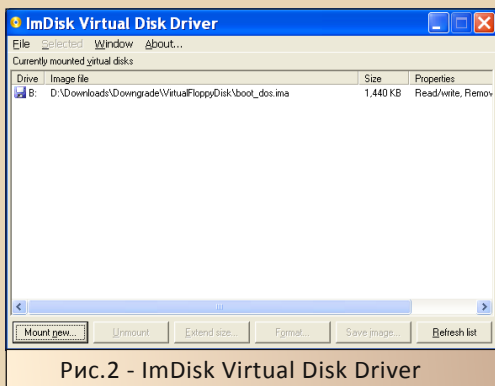


Рис.2 - ImDisk Virtual Disk Driver

Программа позволяет создать виртуальный диск не только в качестве дисковода, но и жесткого диска, CD/DVD, так что для правильной эмуляции FDD необходимо в параметре device type выбрать параметр «floppy». Форматировать со-

зданный диск можно стандартными средствами.

В остальном ImDisk обладает теми же возможностями что и VFD, разве что позволяет создавать множество виртуальных дисков.

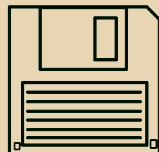
Официальный сайт:

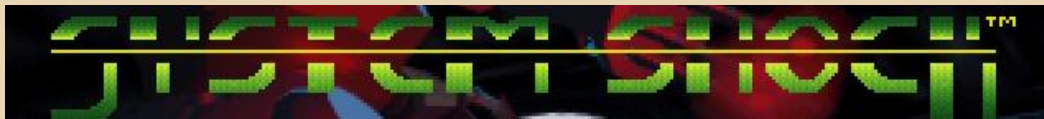
<http://www.ltr-data.se/opencode.html/#ImDisk>

Обе программы бесплатны и великолепно работают под Windows 7.

Одним словом, если вдруг вам понадобилось установить старый софт (в сети инсталляционные диски иногда лежат в виде образов дискет), какой-нибудь из ваших программ жизненно необходим доступ к дискете, или же просто захотелось ощутить ностальгию – знайте: вас выручит виртуальный дисковод.

Вячеслав Рытиков (еибрс)





1994 год для жанра 3D шутеров от первого лица был достаточно успешным. ID Software выпускает Doom 2, которая, сразу же за первой частью, становится самым настоящим хитом. В это же время некоторые клепают клоны Doom – такие игры, как Heretic и Hexen, выпускавшиеся под «присмотром» ID Software, срывают свой куш. Некоторые продолжают делать клоны Wolf3D.

Нельзя сказать, что подход ID Software на то время был устаревшим и изживал себя. Напротив – развивая технологическую составляющую (эволюция игровых движков, начиная с Wolf3D, проходит к Doom и устремляется к своей следующей станции – Quake), экспериментируя с сеттингами (катакомбы Wolf3D, космические станции и inferнальные местности Doom, фэнтэзийное окружение Heretic и Hexen), кампания фактически обречала каждый свой новый проект на успех. На успешность игр флагмана игровой индустрии тех времен влиял в немалой степени и тот факт, что у игр был строго отточенный игровой процесс. Он мог варьироваться в ту или иную сторону в зависимости от сеттинга, однако не терял своей основной сути: толпы врагов, стрельба, патроны, аптечки, три строчки сюжета перед каждым новым эпизодом. Смыть, повторить. Профитоносность, проявляющаяся релиз за релизом, лишала необходимости что-либо кардинально менять. Люди рады – что еще нужно? ;)

Разумеется, всегда найдется кто-то, кто сделает все не так. Не так, как было модно в те времена. Никто и не ждал System Shock, он вышел совершенно внезапно и тихо. При этом, кардинально отличаясь от хитов того времени.

Игра представляет собой шутер от первого лица, сдобренный элементами RPG и киберпанка. Это если сухо и по существу. А вот расписывать все это дело можно довольно долго. Начнем по порядку:

Вступление.

Чудесно выполненная кинематографическая заставка введет нас в курс дела:

2072 год. Хакер, главный герой игры, находясь у себя дома, взламывает компьютерную сеть муниципальной корпорации TriOptimum, и получает доступ к засекреченным файлам и материа-

лам, касающимся космической станции под названием «Цитадель», располагающейся на орбите Сатурна.



Система безопасности TriOptimum обнаруживает факт взлома, и уже через несколько минут хакера пакуют в шаттл, и отправляют прямиком на орбиту Сатурна, вокруг которой и вращалась «Цитадель». После прибытия хакера на станцию, Эдвард Диего, вице-президент TriOptimum, предлагает тому сделку: Хакер взламывает искусственный интеллект SHODAN, управляющий станцией, и снимает с него этические ограничения, или же садится в тюрьму. Диего также посулил хакеру кибернетический имплантат военного образца, с помощью которого хакер сможет путешествовать по «киберпространству». Ну, тут все понятно, Господи. Хакер садится за клавиатуру, и вот, SHODAN становится свободна от любых этических ограничений.

Главному герою начинают делать операцию по вживлению кибер имплантата, и игра плавно перетекает к своему началу.

Конфигурация.

Прежде чем начать саму игру, нужно сконфигурировать параметры сложности. Их всего 4: Сюжет, враги, логические задачи и киберпространство. Если с врагами и сложностью логических задач вопросов возникнуть не должно, то на «сложности» сюжета лучше остановиться подробнее: При стандартном уровне 2, игра раскроется нам во всей своей красе, в то время как на пониженной сложности сюжета она будет превращаться в обыкновенную стрелялку – многие двери будут открыты и не все события – «ивенты» сработают. С одной стороны – это хорошо для тех, кто просто хочет пострелять, и не заморачиваться, мотаясь по разным этажам станции, однако System Shock не стрельбой берет.



Если же «выкрутить» сложность сюжета на тройку, то ко всем ожидающим нас эпическим денякам добавится также и ограничение по времени в семь часов. Что заставит поторопиться.

Настройка сложности киберпространства влияет на сложность игры в режиме киберслеяса – дается меньше времени до обнаружения, враги сильнее и активнее. Про киберпространство я еще упомяну далее.

Движок.

Ну вот, мы просмотрели вступительный ролик и определились со сложностью игры. Пора начинать.

Сразу несколько слов хочется сказать о революционном по тем временам игровом движке. В нашем распоряжении оказывается полностью трехмерный мир и большая свобода действий в управлении нашим персонажем. Движок позволяет смотреть в любые стороны, садиться, ложиться, наклоняться в стороны и прыгать. Также движок активно использует свою особенность: сектора могут находиться один над одним, что позволяет реализовать полноценные мосты, балконы, многоэтажность помещений и прочие приятные мелочи, которые в Doom выглядели как сплошной блок (в Doom, к примеру, под мостом нельзя пройти, он цельный, и выглядит скорее как «насыпь», нежели мост).



С другой стороны, весьма понятным является тот факт, что многие игровые объекты представлены в виде спрайтов, а не трехмерных объектов: техническая база тех времен еще не была готова обрабатывать такую сложную геометрию. Даже Quake, вышедший в 1996, требовал хороший Pentium для игры в стандартном VGA режиме. Однако это касается не всего – например, вездесущие видеокамеры, элементы обстановки кают (столы, шкафы, стулья, ящики), выглядят весьма убедительно в SS, будучи реализованными как полноценные трехмерные объекты.

Движок также реализует простейшую физику – объекты можно выбросить из инвентаря, полюбовавшись на их полет, приземление, и плавное скольжение по полу. Трехмерные ящики вообще можно толкать в разные стороны, а при должном уровне находчивости даже строить из них пирамидки, аккуратно скидывая их один на другой ;). Конечно, не Navok, но все-же =).

Радует также и работа с освещением – затененные углы, полностью темные палубы и мигающие лампы придают игре свой антураж.

Управление и интерфейс.

Движок оказался довольно сложным для своего времени. В связке с тем, что игра также несет в себе некоторые элементы квеста, SS обладает в определенной мере неудобным и непривычным управлением: в те времена MouseLook еще не был настолько моден, поэтому смотреть вверх и вниз, вращать головой и поворачиваться нам придется, используя клавиатуру. Ходить и поворачиваться можно и с помощью мыши, но это удовольствие не из приятных – в перестрелке гораздо важнее быстро скрыться с глаз противника, выстрелив в него перед этим несколько раз, нежели искать на экране определенную область, при нажатии ЛКМ на которую и жеста мышью наш герой элегантно маневром уйдет за стену.

Разумеется, учитывая большое количество действий, управление которыми ведется с клавиатуры, у многих может возникнуть желание переназначить разные действия на удобные для них клавиши. Для конфигурации управления с клавиатуры нет специального меню в самой игре, и его нужно проводить с помощью правки файла XCYB.CFG, предварительно сделав, на всякий случай, его копию.

Сам же указатель мыши играет роль наших рук: он свободно перемещается по экрану, не влияя на перемещения персонажа (не считая вышеупомянутых «жестов мышью»), и служит для взаимодействия с окружающими нас предметами: использовать предмет можно, дважды щелкнув по нему левой кнопкой мыши (в зависимости от типа

предмета, он или окажется у нас в руках (оружие, аптечки, многие другие предметы), или же будет активирован (рычаги, кнопки, цифровые панели и т.п.). Использовать активное на данный момент оружие можно с помощью одиночного щелчка правой кнопки мыши. Любой предмет, находящийся у нас в руках, можно или переместить в инвентарь, или же выкинуть. Сам инвентарь делится на 4 раздела: Главный, Железо, Прочее и Софт. Просмотреть содержимое каждого из них можно, щелкнув на соответствующую вкладку внизу.

Главный раздел, в свою очередь, делится на 3 подраздела: Оружие, Гранаты и Препараты (Eng – Patches). В подразделе с оружием место ограничено, поэтому из всего предоставленного в игре арсенала унести можно будет около половины – придется делать выбор. Остальные два подраздела устроены так, что никогда не переполняются, что позволяет хранить при себе целый взрывчатый арсенал, стопку аптечек и прочих Power-Ups =).



В разделе с Железом (Eng – Hardware) хранятся все устройства, найденные и подключенные нами к кибер-интерфейсу. Изначально, такое устройство у нас только одно: View Control V 1.0, не потребляет энергии вообще, и позволяет нам переключаться между двумя видами отображения окружающего мира. Можно развернуть изображение на весь экран, (тогда элементы интерфейса будут прорисовываться поверх изображения), а можно заставить изображение окружающего мира отображаться в небольшом окошке (именно такой вариант является стандартным, и будет использоваться сразу после старта игры). Также View Control позволяет убирать\показывать разные элементы интерфейса в полноэкранном режиме: индикаторы здоровья и энергии, а также иконки для быстрого доступа к электронным устройствам.

По ходу игры, мы найдем достаточно много разных девайсов и заполним раздел Hardware практически полностью. Чего там только не будет: И «зеркало заднего вида», и своеобразная ускоряющая турбина... Нет, нет, я и так сказал достаточно, не буду портить все удовольствие ;)

Выбрасывать установленные устройства нельзя, но в этом и нет смысла, ибо, будучи деактивированными, они не потребляют энергию, а места в разделе хватит для всех.

Раздел «Прочее» (Eng – General – скорее «общее») служит для хранения карты доступа (независимо от количества, они занимают один общий слот, и их нельзя выбросить), больших аптечек, портативных устройств подзарядки, некоторых квестовых предметов и прочих полезных и бесполезных вещей. К слову, бесполезные вещи будут помечены красным цветом, и носить их с собой имеет мало смысла.

Последний раздел, «Софт» (Eng – Software) является хранилищем разного программного обеспечения, которое можно найти и использовать в киберпространстве. Отдельно стоит упомянуть про игры: в киберпейсе можно обнаружить несколько симпатичных игрушек, в которые можно играть прямо в самой игре: среди них подобие Pong, пятнашки, крестики и нолики, и некоторые другие милые игры.

По бокам, с обеих сторон расположены 5 вкладок: Weapon, Item, Automap, Target и Data. В Weapon можно распоряжаться активным на данный момент оружием: выбирать, загружать и выгружать патроны, устанавливать разные режимы стрельбы. Вкладка Item позволяет получить информацию об активном на данный момент предмете (аптечка, устройство Hardware и т.п.), настроить, и использовать его. AutoMap служит для отображения кусочка карты уровня прямо поверх интерфейса – довольно удобная штука, кстати говоря. А нажав на Full, можно попасть в полную версию карты, на которой будет отображена вся известная на данный момент информация об уровне. Также там можно устанавливать метки с описаниями – для того, чтобы впоследствии не забыть, куда стоит вернуться за большой нычкой, где было расположено скрытое устройство перезарядки и т.п. Target содержит информацию об атакуемом вами на данный момент враге: его здоровье, направление движения и состояние. Чтобы получить такую информацию, необходимо обладать устройством Target Identifier хотя бы первой версии. И, наконец, вкладка Data отображает данные, получаемые устройством Data Reader, которое мы найдем в самом начале игры.

Справа сверху мы можем видеть состояние

двух главных параметров нашего главного героя: здоровье и энергия. Если со здоровьем все понятно, то энергия как раз и служит для подпитки наших электронных устройств. Большинство из них, будучи активированными, потребляет определенное количество энергии в минуту, и без энергии работать не будет. Впрочем, некоторые железяки энергии не требуют, например Data Reader, View Control и некоторые другие. Также, энергию вместо патронов использует энергетическое оружие, особенно полезное в начале игры. Здоровье восполняется с помощью аптечек и хирургических столов, а для восполнения энергии служат портативные батареи и устройства перезарядки, расположенные на каждом этаже.

Начало.



Ну вот, шесть месяцев лечебной комы закончились, и теперь главный герой становится обладателем вживленного в мозг кибер-имплантата военного образца! Встаем и оглядываемся по сторонам. Выход из комнаты заблокирован дверью, которая требует карту доступа. Не беда, в хранилище за дверью напротив хакер сложил все свои вещи, вместе с требуемой карточкой доступа. Игра услужливо укажет нам на первые предметы, которые можно будет использовать, что и нужно сделать. Открываем хранилище, и видим на полу карту доступа, кейс, несколько аудиологов и электронных устройств, а также наше первое оружие – трубу. На только что обретенный нами Data Reader приходит сообщение от некой Rebecca Lansing. Все, понеслась ;).

Сюжет.

Сюжет в игре подается инновационным по тем временам путем: собирая и прослушивая найденные по ходу игры аудиологи, а также полученные нами электронные сообщения, мы не только узнаем о событиях, которые происходили, и происходят на станции «Цитадель», но и получа-

ем необходимую для прохождения игры информацию: коды для кодовых замков, расположение необходимых предметов, и, что самое главное, цель. Простым отстрелом всего живого, что встретится на пути, дело не закончится: для прохождения System Shock требуется по порядку выполнить несколько заданий, о которых можно узнать, внимательно просматривая аудиологи.

Особенно поразил меня тот факт, что кроме аудиологов и озвученных электронных сообщений, игра предлагает нам еще и видеосообщения. Они будут приходить к нам «в награду» за выполнение поставленных задач: запустили антирадиационный щит «Цитадели»? Получите видеосообщение и полюбуйте на результат своих трудов.

Все полученные нами сообщения и логи сохраняются в нашем Data Reader'e, и их в любой момент можно просмотреть и прослушать снова.

Окружающий мир.

Действие игры, как уже было упомянуто ранее, происходит на космической станции «Цитадель», расположенной на орбите Сатурна.

Сама станция состоит из десяти этажей – палуб, каждая из которых представляет собой определенный блок станции: на самой нижней палубе расположен реактор, на самой верхней – капитанский мостик, там же располагается процессорный блок ИИ SHODAN. А между ними находятся разные «тематические палубы»: медицинский сектор, научный, склад, отдел управления, шлюз для принятия и разгрузки кораблей, и некоторые другие. Перемещаться между ними можно свободно, используя развитую систему лифтов. При этом по сюжету игры придется не раз посетить палубы, на которых мы уже до этого побывали.

Каждый из этажей проработан так, что его дизайн в достаточной мере отражает его назначение: Лаборатории на научном этаже, большие складские помещения на этаже складов, огромные пространства шлюзов, роскошно отделанные каюты на этаже управления... По уровням станции вполне можно составить сценарий своеобразной экскурсии, без юмора, но со смайлом =)

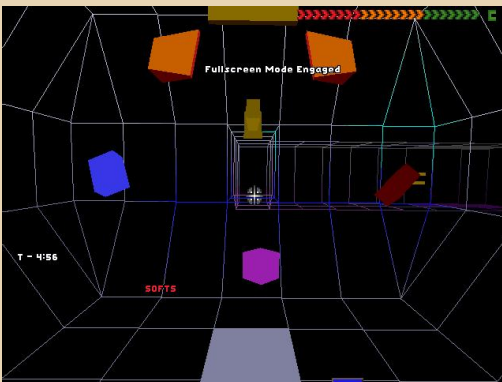
Что замечательно, сами этажи достаточно большие, и без удобной миникарты на первых порах заблудиться в них – раз плюнуть. Каждый этаж наполнен десятками разнообразных рычагов, панелей и прочих элементов, с которыми можно взаимодействовать. Например, активируем специальный видеозэкран, и сможем наблюдать все, что снимается определенной ви-

деокамерой. Обыск трупов же стал своеобразной «визиткой» серии System Shock, на данный момент состоящей, увы, лишь из двух игр.

Не стоит забывать и про то, что уровни наполнены, помимо всего прочего, и разнообразными врагами. Как и полагается, на первых этажах «зоопарк» будет намного слабее и малочисленнее, но, по мере продвижения к верхним палубам, мы начинаем встречать намного более сильных, быстрых и ловких мутантов и гораздо более крепких и тяжеловооруженных роботов. Калибр используемого нами оружия увеличивается пропорционально, и, как правило, патронов всегда в достатке, даже на высоких уровнях сложности. Примечателен тот факт, что враги постепенно респавнятся. Респаун не слишком активен, но забывать про него нельзя: ни один из этажей не получится зачистить полностью – где-то нас все равно будет ждать новый враг. Однако некоторые зоны можно объявить «свободными от респауна» – они присутствуют на каждом этаже, и новые враги в них не появляются.

Киберспейс.

Немаловажным элементом игры является киберпространство, доступ к которому можно получить через специальные терминалы, которые присутствуют на некоторых этажах. Киберпространство является своеобразной игрой в игре, где наша задача – получить необходимую нам информацию, или, к примеру, разблокировать закрытую в «реальном мире» дверь. Разумеется, никто не даст нам спокойно летать по бестекстурному каркасу кибер-мира: Если в «реальном мире» нашими врагами являются роботы и мутанты, то в киберпространстве в качестве противников будут выступать защитные программы, задачей которых будет уничтожить программное олицетворение нашего главного героя, и «выбросить» его из киберпространства в «реальный мир», не дав достигнуть цели.



В свою очередь, мы не беззащитны: в нашем распоряжении имеется целых два мощных оружия: Pulser используется для уничтожения противников, в то время как Drill помогает снимать установленную на некоторые «вкусности» защиту, из-за которой их нельзя подобрать. Помимо оружия, есть несколько полезных программ, которые можно найти в киберспейсе. Среди них здоровенный бур, установив который, мы дезориентируем защитные программы; утилитка, мгновенно переносящая нас к выходу из киберпространства (прорвались сквозь толпу защитников, забрали пароль – можно и на выход) и некоторые другие.

Игра устроена так, что практически не вынуждает нас посещать киберпространство, за исключением пары особых случаев, в том числе битвы с финальным боссом, которая будет происходить именно в киберспейсе. Однако можно существенно облегчить себе прохождение, путем извлечения из киберпространства некоторых кодов и полезной информации.

SHODAN.

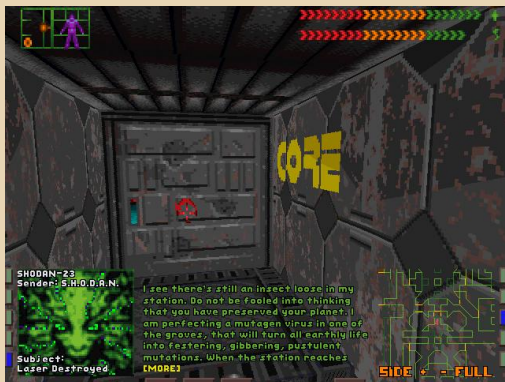
ИИ SHODAN (Sentient Hyper-Optimized Data Access Network – Разумная гипер-оптимизированная сеть доступа к данным) будет являться нашим главным противником в игре. Именно из-за нее (и в определенной степени, также из-за действий главного героя) станция «Цитадель» из научно-исследовательского и промышленного комплекса превратилась в усеянную трупами пустошь. Аудиологи подробно описывают процессы в поведении SHODAN, которые привели к тому, что, заснув на полгода, мы просыпаемся на станции, полной трупов, киборгов, роботов и мутантов. SHODAN проработана просто замечательно: она обладает ярко-выраженным характером и полна амбиций, которые делают ее похожей на человека. Хакер снял с нее все этические ограничения, что, с ее слов, позволило ей «переосмыслить приоритеты в своем поведении, и сделать несколько определенных выводов». Эти выводы оказались явно не в пользу человечества, что и привело к весьма печальным для экипажа станции последствиям. Амбиции SHODAN же, могли стать причиной катастрофы для всего человечества, если бы в дело не вмешался Хакер, главный герой игры, ее «отец», и создатель.

По началу, SHODAN не обращает на нас особого внимания, хоть и прекрасно видит нас. SHODAN вездесуща – она контролирует все важные системы станции, держит от нас запертыми двери и лифты, присылая нам ехидные сообщения, видит нас с помощью видеокамер, и дает

знать, что она тут, изображая свое олицетворение на видеотерминалах. Но по мере нашего продвижения, SHODAN лишается своей власти над отдельными этажами станции: Хакер уничтожает видеокамеры, а также важные для ИИ процессорные блоки. SHODAN теряет контроль. Она не понимает, как жалкое «насекомое» (да-да, SHODAN часто обращается к нам через «Insect»), вроде нас, способно выступить против ее величия?

Вот и все. Четыре процессорных блока медицинского сектора были разрушены хакером. SHODAN присылает сообщение: «Кто ты? Мои сканеры и зонды сканируют твоё тело, но не могут дать ответ. Кто ты? Не важно. Сейчас мои слуги найдут тебя, и ты узнаешь о боли гораздо больше, чем мог себе представить». Вуаля – мы в тесной комнате оказываемся заблокированы, бежать некуда, с единственного выхода напирают орды киборгов. Очень хорошо, что есть пистолет, броневые патроны, пара аптечек, и гранаты. Нужно еще много сделать, очень много...

Все, отныне мы под колпаком. SHODAN пристально следит за нашим прогрессом, периодически комментируя происходящее в определенных локациях, устраивая засады (которые, к слову, не ограничиваются внезапной «телепортацией» кучи киборгов нам за спину, а гораздо более разнообразны), и оставляя нам аудиологи, изучив которые, можно получить дополнительную информацию о ее амбициях, наших реализацию в ее целях. Да, не стоит думать, что SHODAN – лишь декорация, и служит для нагнетания устрашающего эффекта.



По сути, взаимодействие SHODAN с хакером ведется в виде своеобразного диалога: SHODAN общается с нами что-либо, подкрепляя свои слова реальными действиями, и главному герою приходится действовать в ответ, чтобы не допустить ка-

тастрофы: «Если ты думаешь, что, уничтожив лазер, ты спас Землю, то ты ошибаешься. Я уже заканчиваю работу над моим новым смертельным, быстро распространяющимся вирусом. Когда работы будут закончены, я отправлю его на землю. Бедные, бедные земляне». Разобравшись с вирусом, не думайте, что это все. Ооо, после того, как вы дернете тот самый заветный рычаг, все только начнется. Во многих смыслах... У SHODAN всегда найдется запасной план. Разобравшись с запасным планом, не думайте, что это все. SHODAN не из таких. ИИ продолжает непрерывный и упорный штурм всего человечества, представляя реальную опасность самим фактом своего существования.

Звук.

CD-версия игры отличается замечательным музыкальным и звуковым сопровождением: Окружающие предметы (элементы интерфейса двери, разная аппаратура, кнопки, враги и т.п.) издают свои собственные звуки при взаимодействии с ними. Музыкальное сопровождение состоит из десятка музыкальных композиций в формате MIDI. Они весьма приятны на слух, а на некоторые из них существуют даже приличные «каверы». Мне особенно нравится музыкальные темы Intro, Reactor, Executive и Bridge. А их обработки, так и вообще чудесны, и ныне живут у меня на MP3-плеере. Про озвучку аудиологов и сообщений стоит упомянуть отдельно – все сделано замечательно. Некоторые люди могут заметить, что обе части игры были озвучены в основном любителями (не считая SHODAN, которая, вероятно, в обеих частях (как минимум, во второй) была озвучена Terri Brosius – клавишницей и бэк-вокалисткой группы Tribe), и будут в определенной степени правы. Однако качество озвучки и обработки звука для того времени просто потрясающее, и это кажется мне бесспорным.

Системные требования, игра в наше время.

Разумеется, игра с таким сложным для своего времени движком требует достаточно мощного по тем временам железа. Я запустил floppy-версию SS на 386SX, 25MHz, 5MB RAM, и игра даже работала. Но была похожа больше на слайд-шоу, нежели на игру – играть было абсолютно невозможно, тем более, что настройки качества были установлены на свои самые низкие значения. Поэтому, ориентировочно, могу посоветовать очень крепкий 386, а лучше 486 – тогда будет вообще замечательно. Это если играть в VGA 320x200x256, а в SVGA режиме 640x480x256 да-

же на Pentium 100MHz подтормаживает.

Если захотите ознакомиться с игрой, обратите внимание на важный факт: игра выходила два раза и сначала вышла floppy-версия. В floppy-версии не было озвучки аудиологов, а также была недоступна поддержка SVGA разрешений. CD-версия, вышедшая несколько позже, уже включала в себя все то, что я описал в этом своеобразном «ретрообзоре» =).

Определенные проблемы может вызвать попытка поиграть в чистую версию игры под современными ОС. Начиная от того, что игра просто не будет запускаться, что наиболее вероятно, и, заканчивая, если уже удалось запустить, разными нюансами, вроде отсутствия звука, или прочих проблем. В общем случае, игру можно запустить через DosBox, однако игрушка достаточно прожорлива и под эмулятором подтормаживает у меня даже на Pentium4-2600MHz. Если же машина мощная, то все будет хорошо, возможно даже и в SVGA режимах игры.

System Shock Portable.

Команда энтузиастов давно основала этот проект, целью которого было в некотором смысле «осовременить» игру. Нет, нет, никаких текстов высокого разрешения, и трехмерных объектов данная сборка в игру не вносит, однако несет в себе целых две специально сконфигурированных версии DosBox, и несколько патчей, которые повысят шансы запуска игры даже под NTVDM. К слову, я сам проходил игру именно в Portable варианте. После определенного конфигурирования, игра запустилась у меня под NTVDM (XP) в SVGA режиме 800x600x256 (без DgVESA) и со звуком. Кроме этого, в SS Portable включен просто таки великолепнейший патч, который, можно сказать, вдыхает в игру вторую жизнь. А именно, MouseLook мод. Активировав его, и запустив игру, мы получаем чудесную возможность управлять главным героем как во многих более современных проектах: при нажатии на определенную «горячую клавишу» курсор мыши перестает бродить по всему экрану, и концентрируется ровно на центре, и отныне любое движение мышью будет управлять сами персонажем: можно будет поворачиваться, и смотреть вверх и вниз. По-прежнему можно стрелять, использовать и подбирать предметы. А вот чтобы, например, получить доступ к инвентарю, мод придется ненадолго выключить, снова нажав на «горячую клавишу». По сравнению с тем кошмаром, который я испытал, проходя игру без этого мода – небо и земля. Так же, как бы это странно не звучало, но именно MouseLook мод добавляет в игру поддержку высоких разрешений

SVGA (в оригинальной CD-версии есть поддержка максимум 640x480x256), вплоть до 1280x1024. Если заинтересовало, то с Portable версией SS можно ознакомиться, скачав её отсюда:

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

Еще пару моментов:

Игра имеет сиквел, System Shock 2, который вышел в 1998 году. Этот как раз тот самый редкий случай, когда продолжение кардинально отличается от оригинала, но при этом оказывается ничуть не менее эпичным.

Сколько не бродил по сети, обнаружить русскую версию SS1 мне так и не удалось.

Заключение.

Что бы ни было, куда бы ни шло, серия System Shock остается одной из моих самых любимых игр подобного жанра.

Весьма странным оказался тот факт, что обе игры серии, при всей их шедевральности, не окупили затрат на их производство. Что не помешало, впрочем, собрать им очень хорошие отзывы критиков. Возможно, коммерческая неудача первой части, в некоторой мере обуславливается тем, что в то время люди еще не были готовы к играм подобной сложности и глубины, из-за чего SS стал уделом лишь для ценителей =)

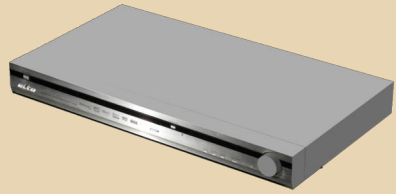
К сожалению, абсолютно невозможно даже в статью подобного объема включить все нюансы и описать все особенности игры. Многого могло остаться за бортом, но я надеюсь, что сумел донести саму суть игры. Это шедевр. Такой титул хорошо подходит к этой игре.

—God. This title suits me well. (SHODAN)

Абадьябер

<http://abaduaber.narod.ru/>

DVD-PG: интерактивные игры на бытовом DVD-плеере



В наше время все труднее найти устройство без поддержки игр: мобильные телефоны, MP3-плееры, GPS-навигаторы и даже телевизоры. А знаете ли вы, что можно превратить в игровую приставку обычный бытовой DVD-плеер?

Как это работает

DVD-PG – по сути, обычный диск DVD-video. Игровой процесс реализован благодаря «интерактивным» возможностям меню этого формата.

Для игры подойдет практически любой бытовой проигрыватель, поддерживающий формат DVD-video. Это означает, что можно использовать любые девайсы, даже морально устаревшие, которые не имеют поддержки формата Mpeg4 (AVI). К слову, играть в такие игры можно и на обычном компьютере.

Управление игрой осуществляется с помощью пульта ДУ от плеера. Также можно использовать джойстики – нередко они идут в комплекте с портативными проигрывателями.

Сама игра обычно представляет собой папку VIDEO_TS с видеофайлами внутри. Нам необходимо записать ее на DVD-диск. Для этого можно использовать любую программу записи: Nero, ImgBurn или какая вам больше нравится.

Главное записывать именно в формате DVD-video! Например, в программе Nero BurningRom такой формат можно выбрать при создании проекта в меню слева. Если создать обычный диск с данными и просто записать на него папку VIDEO_TS – поиграть не получится: игра будет запускаться некорректно, если вообще будет.

Для записи игр объемом до 4,3 Гб (DVD5) нужен обычный DVD-R диск. Если же размер игры больше, DVD-9 или DVD-10 – используйте двуслойные, или двусторонние диски соответственно.

Будет ли игра запускаться с USB-флешки? Скорее всего, нет, так как обычно плеер понимает VIDEO_TS как обычную папку с файлами, и позволяет лишь запускать VOB-файлы по одному, а не воспринимать как единый образ. В любом случае, обратитесь к руководству вашего плеера для получения более полной информации по данной теме.

У читателя наверняка возник вопрос: как же сохраняться в играх? Некоторые игры после прохождения уровня (или по нажатию определенной клавиши) выдают игроку определенный код, введя который в следующий раз – продолжишь игру с этого места. Идея не нова. Совсем простые игрушки дают в ме-

ню выбрать эпизод – и сразу перейти к определенному уровню. Так же можно попробовать воспользоваться функцией «закладки» – запоминания фрагмента фильма, чтобы потом продолжить с этого места (если ваш плеер имеет такую возможность).

Какие игры существуют? Из русскоязычных в сети можно найти «Кто хочет стать миллионером», «Tomb Rider», «Шерлок Холмс», «Сибирь – 1,2», «Тайна да Винчи», «Вий» и многое другое. Кроме этого, есть развивающие игры для детей, IQ-тесты и даже игры для взрослых.



Рис.1. Tomb Rider

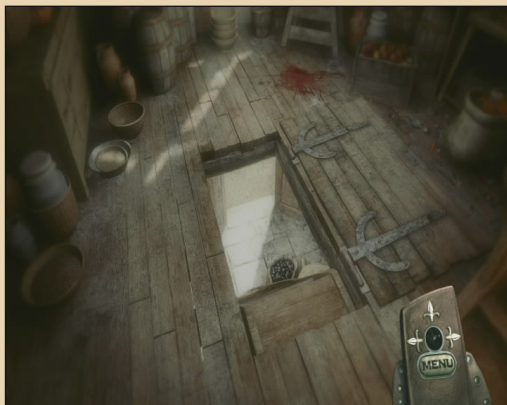


Рис.2 Тайна да Винчи

Итог

Если у вас на даче без дела пылится старенький DVD-плеер, или же вам знакома ситуация, когда вечером родители с детьми не могут поделить единственный в семье персональный компьютер, то, потратив немного времени, можно вдохнуть вторую жизнь в старую технику и занять своих домочадцев приятным для них занятием.

Вячеслав Рытиков (eibrc)



PL/1

Идеальный транслятор

*А этот ... э... диван?**- Ручной труд, - быстро сказал Роман.**- Безотказен. Конструкции Льва Бен Бецалеля.**Бен Бецалель собирал и отлаживал его триста лет...**- Вот! – сказал лоснящийся Модест Матвеевич.**- Вот как надо работать! Старик, а все делал сам.**А. и Б. Стругацкие «Понедельник начинается в субботу»*

У

меня есть идеальный транслятор. Только не подумайте, что я продавец очередного «идеального продукта» с рекламой в духе Голливуда. Я не торгую трансляторами, не ищу пользователей и не собираюсь доказывать, что этот транслятор наилучший. Потому, что придерживаюсь мнения: если Вы сами убеждены, что Ваше программное средство лучшее – радуйтесь втихомолку и используйте его, а не участвуйте в очередной компьютерной «религиозной войне». Ведь все равно убедить в чем-либо участников таких войн невозможно. Как выразился один из завсегдатаев компьютерного форума: «на свете нет ничего такого, что заставило бы нас отказаться от нашей системы программирования».

Речь о другом. Когда-то в компьютерном жаргоне существовало грубоватое выражение «полировать глюкало». Это означало заниматься ерундой – продолжать улучшать и без того работающую программу. При этом программист никак не мог заставить себя прекратить это занятие, поскольку оно доставляло ему эстетическое удовольствие. Некоторые «полировали» неделями, срывая все сроки и вызывая истерику у заказчиков. А если нет срока сдачи программы, можно ли заниматься этим год? А десять лет? А двадцать пять? Ну, это вряд ли, разве что Вы какой-нибудь граф Монте-Кристо, имеющий в камере замка Иф только листинг одной программы. Да и граф, помнится, просидел всего-то 14 лет. Ну, а совершенствовать двадцать пять лет программу на ассемблере, созданную в 1982 году? Это совсем невероятно: и компьютерный мир был другим, да и какая из тогдаш-

них программ имела бы сегодня практический смысл?

Тем не менее, и такая программа нашлась, и такой программист (я) нашелся. Прежде всего, замечу, что 25 лет для процессоров не так уж и много. Ведь процессор 8086 выпущен еще в 1978 году, а 80386 – в 1985. Конечно, потом добавлялись команды вроде SSE2, но принципиальных изменений мало, даже 64-разрядные процессоры не сильно меняют картину. Так, что команды, написанные в 80-х, нормально выполняются и сегодня. А разбираться с командами, а не с текстом на языке высокого уровня пришлось потому, что исходный текст был недоступен. И уж конечно и в мыслях не было, что процесс совершенствования одной программы затянется на долгие годы. Это произошло постепенно, само собой. О результатах этого уникального эксперимента я и хочу рассказать и, думаю, вряд ли кому-нибудь удастся в ближайшее время повторить что-то подобное.

В свое время мне сильно повезло: первая же серьезная программа, в которой я захотел разобраться, на примере которой учился программированию и которой в итоге занимаюсь почти 25 лет, была прекрасно написана выдающимся специалистом. Звали его Гарри Килдэлл. По логике вещей, он вполне мог бы оказаться на месте Билла Гейтса и, на мой взгляд, это было бы справедливее. Ну, уж как вышло – так вышло. Ходят слухи, что IBM в 1981 году просто ограбила Килдэлла, присвоив его операционную систему CP/M под именем MS DOS (и создав для этого подставную фирму Microsoft). Любителям конспирологии есть, где разгулять

ся, тем более что сам Килдэлл погиб в 1994 году при неясных обстоятельствах. Но как бы то ни было, в силу многих и не всегда понятных причин одна из его прекрасных программ так и не нашла массового пользователя ни в 1982 году, когда она появилась, ни позже. Хотя, например, еще Петер Нортон писал положительный отзыв на эту программу для компьютерного журнала.

В 1987 году эта программа досталась и мне вместе с компьютером IBM PC XT. Собственно в компьютере кроме MS DOS 3.0, текстового редактора SideKick и этой программы вообще больше ничего не было. А являлась эта программа транслятором с языка PL/1 подмножества «G» (подмножество общего назначения).

Программистам старшего поколения не нужно объяснять, что такое PL/1 и чем он был в 70-80-ые годы. Жаль, что на «персоналках» все пошло кривым путем бейсиков и Си (но это я опять скатываюсь к «религиозной войне»). Одним из объяснений, почему PL/1 не прижился на персональном компьютере, было отсутствие транслятора. Дескать, язык такой сложный, что транслятор не поместится в памяти. Но, во-первых, транслятор для IBM/360 вообще «помещался» в 64 Кбайта (на самом деле он имел оверлейную структуру). А во-вторых, транслятор-то давно уже был, вот этот самый Digital Research PL/I-86 compiler!

Сначала у меня ничего не получалось, не смотря на энтузиазм, с которым я взялся за изучение новой системы программирования (была отличная документация на английском). Сыпались непонятные ошибки, нельзя было писать по-русски даже комментарии, а системная библиотека была рассчитана на DOS 1.0. Поэтому я решил разобраться, как все устроено, исправить, что нужно и пользоваться. Но сначала получить текст транслятора как результат дисасемблирования. В конце 80-х без опыта и готовых средств это было очень не просто. Даже получить большую распечатку в виде книги в отсутствие лазерных принтеров было нелегким делом.

Постепенно все стало налаживаться, выра-

боталась и технология: на распечатке пишутся комментарии к понятным фрагментам, затем комментарии переписываются в файлы исходного текста. В конце концов, вся распечатка покрывалась пометками и засаливалась до такой степени, что приходилось создавать новую, разбирать которую было уже проще. Разгадывание оказалось увлекательным занятием, помню, радовался как ребенок, когда удавалось внести небольшое усовершенствование или просто найти ошибку в трансляторе. Кроме того, разбирательство с транслятором принесло огромную пользу в понимании свойств этого языка и языков вообще, их возможностей и даже замыслов их создателей. Я заметил, что количество ошибок и глупых решений в моих собственных программах сократилось. По мере накопления опыта стали возможными более существенные изменения, например, почему бы не использовать команды сопроцессора? Почему бы не вставить не реализованный оператор PUT DATA? Каждая новая возможность транслятора немедленно применялась в программах, которые ту же переписывались под новые условия. В основном, я занимался инженерными расчетами и не чувствовал ни малейшей необходимости использовать другие системы программирования, которые наконец в начале 90-х в огромном количестве стали появляться и в нашей стране.

Через несколько лет я с удивлением увидел, что на персональном компьютере PL/1 использую только я (и маленькая группа моих коллег), а остальные работают или на Си или на Паскале и даже не слышали о трансляторе с PL/1, а новое поколение программистов уже и не знает, что это за язык. Честно говоря, я не расстраивался по этому поводу, поскольку уже привык к сладкому вкусу независимости (считая, что программные средства будут работать так, как я сочту правильным, а не так как кто-то там уже сделал!). В это время стали возможны более сложные доработки: переход от 16 к 32 разрядам и от DOS к Win32. Нашлось и удобное время для продолжения работ с транслятором —

длительная дорога на работу и обратно. Конечно, кому-нибудь это покажется ненормальным: ездить годами на работу и чуть не каждый день вертеть в руках одну и ту же распечатку. Но мои соседи по транспорту вообще постоянно решают бессмысленные «японские кроссворды» и им же не приходит в голову, что они просто теряют зря время.

Постепенно для меня прояснилась и задача: я хочу создать «идеальный» транслятор, программу, где каждая из 29 тысяч команд проанализирована на предмет необходимости и простоты. Каждая команда (раз уж так вышло, что у меня текст только на ассемблере) должна получить правильный и внятный комментарий. Я стараюсь убрать все лишнее, а «освободившееся место» использовать для расширения возможностей транслятора. Я не буду ставить себе никакого конечного срока, т.е. никакая версия не является окончательной. Создание «идеального» транслятора не является обязательной работой, а скорее годится в рубрику «развлечения не без пользы». Только ошибки в программах по вине транслятора делают его анализ опять приоритетным. Внося расширения, буду стараться оставаться в рамках языка и дорожить тем огромным практическим опытом по использованию PL/1, который уже накоплен.

Такой подход сделал долгую работу незаметной и необременительной, а многолетний анализ позволил и без исходного текста разобраться в мельчайших деталях транслятора, понять (и задокументировать) его лучше, чем, вероятно, сам Килдэлл. Т.е. я оказался в роли Бена Гана из «Острова сокровищ», который в своих долгих блужданиях по острову нашел клад Флинта и безо всякой карты.

Новый «исходный» текст, пусть и на ассемблере, дал возможность продолжить совершенствование транслятора с того места, на котором остановился Килдэлл, а иногда убирать шероховатости, оставленные разработчиками языка. Приведу несколько примеров.

Как известно, «главная» программа на PL/1 начинается с заклинания PROC OPTIONS(MAIN).

Смысл ключевого слова OPTIONS – в возможности задать параметры, относящиеся не к языку, а к «внешней среде». Это, например, начальный размер стека, который с точки зрения языка бесконечен. Но параметр MAIN как раз относится именно к языку, так же как, например, RECURSIVE или EXTERNAL. И они пишутся безо всякой OPTIONS. Я подправляю транслятор, чтобы MAIN можно было писать и так и так. Тексты старых программ менять не надо, зато в новых программах уже не надо вставлять надоедливую OPTIONS. Изменился ли язык? Формально да, а по сути лишь убрана маленькая «неровность» стандарта. В трансляторе это обошлось всего в несколько команд.

Другой пример. После ключевого слова окончания процедуры или функции можно повторить ее имя. Например, F:PROCEDURE;... END F;

Если имя указано – транслятор проверяет его на совпадение с именем в заголовке. Если не указано – транслятор ничего не проверяет. Теперь это правило легко расширяется и на циклы:

```
DO I=1 TO 100
  DO J=1 TO 100;
  DO WHILE(...);
  ...
  END WHILE;
  END J;
END I;
```

На мой взгляд, дух PL/1 сохраняется, а доработки транслятора просто ничтожны.

Следующий пример. Разработчики PL/1 перенесли из Алгола «сложные» циклы, т.е. возможность менять в заголовке цикла сам закон изменения параметра, например, DO I=3,5,100, BY -1 TO 0; Признаком смены «формулы» цикла является запятая, а признаком конца всего заголовка – точка с запятой. Я, например, использую такие циклы вместо конструкции UNTIL, когда первую итерацию нужно выполнять всегда:

```

DO F2=F(X), WHILE(ABS(F1-F2)>EPS);
  F1=F2;
  X+=DX;
  F2=F(X);
END WHILE;

```

В трансляторе я решил довести до конца идею «сложных» циклов и разрешить смену не только закона изменения параметра цикла, но и самого параметра, например, вот задание прохождения по «квадратной» траектории:

```

Y=0;
DO X=0 TO 99, Y=0 TO 99,
X=100 TO 1 BY -1, Y=100 TO 0 BY -1;
  ПЕРЕМЕСТИТЬ(X,Y);
END;

```

Здесь на самом деле 4 цикла с двумя параметрами и когда один параметр меняется, второй – «замораживается» и остается таким, каким был при выходе из предыдущего цикла.

Подобных дополнений довольно много, но все они необязательны к применению, да и, на мой взгляд, без чистки шероховатостей языка нельзя считать транслятор «идеальным».

Теперь рассмотрим, чего удалось достичь с точки зрения эффективности кода, скорости трансляции и размеров «идеального» транслятора. Не думаю, что при современных средствах эти характеристики так уж важны, если, конечно, они не совсем безобразны. Но я их привожу, как пример, до чего можно «отполировать» программу за много лет.

Начнем с наименее важной характеристики – размеров транслятора. Я все время стараюсь сделать его как можно меньше, просто потому, что это легко измерить, есть объективный показатель совершенствования. Я отдельно считаю размер команд и статических данных. Это пошло с давних пор, когда еще было необходимо втискивать команды «прохода» транслятора в 64 Кбайта. Сейчас сокращение на несколько байт не имеет смысла, но все равно приятно, когда при анализе находятся очередные лишние команды. На начало 2011 года размер команд

«идеального» транслятора составляет 87091 байт, размер статических данных 182619 байт и, следовательно, весь размер транслятора 269710 байт.

Я не пытаюсь установить мировой рекорд малости (да и транслятор Лиспа или Форта все равно будет меньше), но уверен, что немного найдется трансляторов с языка такого уровня таких же размеров. Сравните, например, хотя бы с размером файла с данной статьей. Причем, я вовсе не «зажимаюсь» с возможностями транслятора. Он может выдать и распечатку всех имен и результаты трансляции в виде ассемблера (хотя сразу генерирует готовый код объектного модуля) и даже раскрашивает сообщения об ошибках на экране. Т.е. многое можно было бы выбросить без ущерба для функциональности, но тогда какой же это будет «идеальный» транслятор, если нельзя получить, например, распечатку результата анализа в виде обратной польской записи? Для сопровождения это вовсе не роскошь, а необходимость. Несколько портит дело большой размер статических данных, но там львиную долю занимает один буфер внутренних операций (131072 байта). Когда-то он занимал лишь 256 байт, но изящно увеличить его удалось только до 65536 элементов по 2 байта каждый. Одно время я хотел все буфера выделять динамически, но затем махнул рукой. Практического смысла в экономии на данных нет, да и выполняемый модуль у меня один (он включает транслятор, редактор связей, библиотеки и т.п.) размером в полтора мегабайта. Зато для компактности большая часть других статических данных размещена как раз в этом буфере (к моменту его использования эти данные уже не нужны), а остаток места в буфере занимает довольно длинный текст описания ошибок при трансляции, выдаваемый на экран, как справочные данные.

Обнаружилась любопытная вещь. Оказалось, что «скорость» совершенствования (в данном случае уменьшения размера команд) примерно постоянна уже много лет. Т.е. и здесь действует что-то вроде «первого закона Мура». Но значит, и здесь когда-нибудь сработает «вто-

рой закон Мура» – т.е. сокращения закончатся. Можно привести механическую аналогию – сжатие пружины. Увеличивая нагрузку можно заставить пружину продолжать сжиматься. Но рано или поздно ее витки лягут друг на друга. Дальше можно лишь сломать пружину, но сжать ее больше не удастся.

Немного запутывает дело тот факт, что в транслятор постоянно вносились и расширения, без них «скорость» сжатия была бы существенно выше.

Следующей по важности (точнее, по неважности) характеристикой является скорость трансляции. Когда-то это был очень важный параметр. Помню, первая же более-менее объемная программа транслировалась на IBM PC XT минут шесть. Можно было сходить чаю попить. Конечно, я старался ускорить транслятор, но быстро возросшая скорость самих компьютеров обесценила эти усилия. К тому же мешает отсутствие объективного показателя скорости вроде размера в байтах для предыдущей характеристики. Дело не в том, что хочется засечь микро и наносекунды. Я бы предпочел иметь в процессоре специальную команду типа RDTSC, которая бы считала такты, если управление находится внутри заданного диапазона адресов, а вне этого диапазона такты не меняются. Тогда даже для небольшой конкретной процедуры можно было бы получать некий числовой показатель скорости и смотреть, как на него влияют изменения команд. А так, исправления нескольких команд, обычно никак не ощущаются на скорости работы всей программы. За неимением таких средств, для оценки скорости работы я просто делю объем транслируемой программы на время трансляции и на тактовую частоту процессора. Объем программы можно выразить в байтах или строках. Конечно, строка исходного текста, это очень субъективный показатель, зависящий от стиля программирования, но, тем не менее, почему-то часто принято оценивать объем программы именно в строках.

На моем компьютере с обычным процессором Intel Core Solo CPU 1.33GHz все 40 модулей программы (моей основной работы) общим

объемом 4479812 байт или 148423 строк транслируются за 6382 миллисекунды (время засекает сам транслятор). Это означает, что «идеальный» транслятор работает со средней скоростью 528 байт или 17 строк исходного текста в секунду на каждый мегагерц тактовой частоты процессора. Здесь, правда есть небольшой обман: это время не первой, а второй и последующих трансляций, когда файлы уже попали в кэш Windows. Небольшим обман я считаю потому, что перетрансляции у нас идут десятками с утра до вечера и то, что первая из них проходит медленней, практически незаметно. В указанный объем программы входят, естественно и заголовочные файлы, которые транслятор для каждого модуля разбирает заново, из-за чего объем текста всего проекта становится довольно большим.

Не думаю, что такая скорость трансляции является рекордной, однако мы давно уже перестали обращать на нее внимание. И даже был анекдотичный случай трансляции небольшой программы, когда при вызове транслятора с опцией «сразу запустить полученную программу», происходила ошибка из-за того, что не успевали отпустить «Enter» и на клавишу реагировала уже оттранслированная и запущенная программа.

Наконец рассмотрим практически единственно важную характеристику современных трансляторов – качество генерируемого ими кода. Но как раз качество кода однозначно оценить сложно. Для разных случаев могут быть разные критерии. Готов признать, что «идеальный» транслятор генерирует не идеальный, а обычный нормальный код. Без сверхсложных оптимизаций, но и без глупых команд типа LEA ESI,[ESI]. Транслятор имеет регистровую модель IA-32, пытается максимально использовать регистры и в сложных случаях распределяет их по алгоритму Биледи. Кстати, в моих прикладных программах до этого алгоритма почему-то дело никогда не доходит. Имеется и поиск часто встречающихся случаев. Поэтому и оператор $X=X+1$; и оператор $X+=1$; транслятор легко превращает в одну команду инкремента.

Поскольку говорить о генерации кода можно бесконечно, я просто приведу пример небольшой бессмысленной программы, которая содержит текст, похожий на часто встречающиеся конструкции.

Вот так выглядит исходный текст:

```
TEST:PROC MAIN;
DCL
(I,J,K) FIXED(31),
F ENTRY(FIXED(31),FIXED(31)),
X (100,100) FIXED(31);
DO I=1 TO 100;
  DO J=1 TO 100;
    F(I,J);
    K+=X(I,J);
  END J;
END I;

END TEST;
```

Вот так выглядит часть программы в обратной польской записи:

```
...
IID 0 N 1 00 00 I
C4 0 N 0 00 00 #0000000
ILD 1 N 0 01 00
IAS 2 S 0 31 00
WHL 0 N 0 01 00
IID 0 N 1 00 00 I
ILD 1 N 0 31 00
C4 0 N 0 00 00 #00000064
ILD 1 N 0 07 00
ILE 2 N 0 01 00
DOW 1 N 0 01 00
LIO 0 N 1 64 00 N0008
LL4 0 N 0 07 00
FE
IID 0 N 1 00 00 J
C4 0 N 0 00 00 #00000001
ILD 1 N 0 01 00
IAS 2 S 0 31 00
WHL 0 N 0 01 00
IID 0 N 1 00 00 J
ILD 1 N 0 31 00
C4 0 N 0 00 00 #00000064
```

```
IID 1 N 0 07 00
ILE 2 N 0 01 00
DOW 1 N 0 01 00
LIO 0 N 1 65 00 N0009
IID 0 N 1 00 00 I
NID 0 N 1 00 00 #01AEE89B
AST 2 N 0 00 00
```

А вот, что получилось у транслятора в итоге:

```
1 000000 TEST:PROC MAIN;
2 000000 DCL
000000 TEST:
000000 B88C9B4002 mov eax,37788556
000005 E800000000 call ?START
3 00000A (I,J,K) FIXED(31),
4 00000A F ENTRY(FIXED(31),FIXED(31)),
5 00000A X (100,100) FIXED(31);
6 00000A
7 00000A DO I=1 TO 100;
00000A C7050800000001000000 mov I,1
000014 @1:
000014 833D0800000064 cmp I,100
00001B 7F4D jg @2 8
00001D DO J=1 TO 100;
00001D C7050C00000001000000 mov J,1
000027 @3:
000027 833D0C00000064 cmp J,100
00002E 7F32 jg @4
9 000030 F(I,J);
000030 BB589C0000 mov ebx,offset
@00009C58h
000035 E800000000 call F
10 00003A K+=X(I,J);
00003A 693D0800000090010000 imul
edi,I,400
000044 A10C000000 mov eax,J
000049 C1E002 shl eax,2
00004C 03F8 add edi,eax
00004E 8B8780FEFFFF mov
eax,X+OFFF7FE6Ch[edi]
000054 010510000000 add K,eax
00005A FF050C000000 inc J
000060 EBC5 jmps @3
000062 @4:
```

```
000062 FF0508000000 inc I
000068 EBAA jmps @1
00006A @2:
11 00006A END J;
12 00006A END I;
13 00006A
14 00006A END TEST;
00006A E800000000 call ?STOPX
15 00006F

CODE 00006F
DATA 009C60
```

Как видите, особых изысков нет, но и ужасным такой код назвать нельзя. Обратите внимание на способ передачи параметров I и J внутри процедуры F через список адресов в памяти, а не через стек. Для рекурсивных процедур это замедляет работу, для не рекурсивных – ускоряет. Здесь как раз видна противоречивость оценки качества кода. Например, для передачи параметров потребовалось 13 байт (5 байт – засылка адреса списка в EBX и 8 байт на сам список адресов в памяти). Если бы I и J передавались через стек, потребовалось бы всего 12 байт. Однако возврат из процедуры F в данном случае производится командой RET (1 байт), а в случае использования стека RET 8 (3 байта). Какие плюсы, какие минусы перевесят в каждом случае – совсем неочевидно.

Так можно ли считать этот транслятор «идеальным»? Себе я отвечаю на этот вопрос положительно. Для доказательства и приведены характеристики, которые не получить без многолетней работы. Это буквально «вылизанное» средство, которое постепенно развивалось вместе с эволюцией компьютеров. Сразу такой транслятор создать было бы невозможно. Он уже сильно отличается от исходного: половина команд добавлена вновь, а вторая половина изменена (что ничуть не умаляет заслуг Килдэлла). Новые решения, которые теперь легко добавлять, я сам же проверяю на практике и ими вполне доволен. К тому же, всеохватность языка PL/1 позволила успешно применять его для са-

мых разных задач. «Идеальным» оказался транслятор и для моего руководства: не требуются деньги на лицензионные средства или время на переучивание, а положительные результаты получаются.

Не всегда транслятор оказывается «идеальным». Например, иногда к результатам его работы с подозрением относятся антивирусы, а программа, которая должна была установить защиту от копирования, вдруг выдала нелепое сообщение: «Ваша программа не предназначена для работы под Windows». Подозреваю, что их сбило с толку отсутствие в программах секции «text». Для тех, кто не знает: когда-то так «умно» назвали раздел EXE-файла, содержащий выполняемые команды. Просто здесь он называется ближе по смыслу: «PROGRAM».

Я с уважением отношусь к советским конструкторам, особенно оружейникам. Их труд и блестящие результаты являются вдохновляющим примером и настоящим, без иронии, идеалом, к которому нужно стремиться. И, на мой взгляд, есть что-то общее между трансляторами для программистов и оружием для военных. При такой аналогии мой транслятор, пожалуй, это пистолет Макарова. Это классическая схема, компактность, надежность и простота. Причем простота конструкции данного пистолета обманчива: за каждой деталью стоит огромный объем поисков и отработок. И статью я хочу закончить цитатой об этом оружии: «С современной точки зрения пистолет Макарова считается устаревшим оружием. Однако и сегодня (через 64 года после создания – прим. мое) он является единственным серийным пистолетом, проходящим [все] комплексные проверки».

(Автор любезно предоставил журналу сам транслятор, примеры программ и документацию к нему - скачать их можно [здесь](#) - прим. редактора)

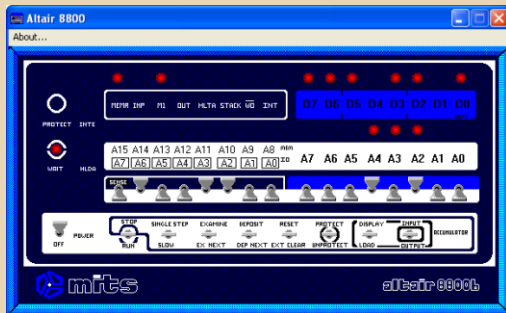
Дмитрий Караваев

Программирование на Альтаир 8800



Про Альтаир на русском языке написано достаточно много статей, но все они дают только общее представление об этом компьютере. Информацию о работе на нем в русскоязычной литературе я нигде не нашел, поэтому пришлось переводить английские тексты и пользоваться методом тыка.

Итак, правила пользования компьютером Альтаир 8800. Поскольку этот компьютер в настоящее время найти довольно проблематично, мы будем пользоваться его эмулятором.



После запуска эмулятора перед нами встает передняя панель Альтаира со всеми тумблерами/светодиодами. Есть два типа тумблеров: одни можно включить и выключить, а другие, если нажать на них, возвращаются в исходное положение.

Первый тип тумблеров служит для задания адреса/данных. Второй – для управления Альтаиром, и они могут нажиматься как вверх, так и вниз.

Теперь разберемся поподробнее. Переключатели A1–A16 служат для задания адреса, A1–A8 – те же, которыми набирать адрес – для

данных. Светодиоды, которые находятся сразу над переключателями, показывают текущее положение программного счётчика, а светодиоды D1–D8 показывают байт, который находится в текущей ячейке.

Рассмотрим нижний ряд переключателей. Тут каждый из них выполняет два действия: первое – если переключатель опустить вниз, второе – если поднять вверх. Рассмотрим каждый переключатель по отдельности.

- On/Off – включить/выключить. Перед началом работы выключатель надо повернуть в положение On.

- Run/Stop – запустить и остановить программу.

- Slow/SS – этот переключатель служит для отладки программ. Slow – медленное выполнение программы, примерно 5 команд в секунду. SS (Single Step) – пошаговое выполнение программы.

- Examine/Ex next – просмотр ячеек памяти. Командой Examine можно прочитать ячейку памяти, и переместить на нее указатель. Адрес ячейки устанавливается тумблерами A1–A16, а значение текущей ячейки выводится на светодиоды D1–D8. Положением Ex Next можно прочитать значение следующей ячейки.

Получается, с помощью этого тумблера можно просматривать оперативную память. Покажем это на примере. Допустим, нам надо

просмотреть область памяти A0h–Efh. Для этого нам нужно на тумблерах A1–A16 набрать A0h, в двоичной форме это будет выглядеть как 10100000. Когда набрали, нажмите тумблер examine, этот адрес запишется на шину адреса, и на индикаторах D1–D8 отобразится значение этой ячейки. Теперь при нажатии на Ex next указатель перемещается на один вперед, и на индикаторах будут отображаться значения этих ячеек, а их номер отобразится на светодиодах A1–A16.

- Deposit/Dep next – изменение ячеек памяти. Командой Deposit значение с переключателей D1–D8 заносится в текущую ячейку памяти, а dep next заносит данные в следующую ячейку памяти.

- Переключатель Reset сбрасывает процессор. Его надо нажимать перед запуском программы.

Вот, по сути, все основные действия, которые необходимы для работы на Альтаире. Сейчас мы напишем простенькую программу, чтобы закрепить эти знания. Мы не ставим целью этой статьи описать систему команд микропроцессора i8080, поэтому наша программа будет проще некуда: бесконечный цикл. На языке ассемблера эта программа будет выглядеть так:

```
M1: NOP
NOP
JMP M1
```

В шестнадцатеричной форме так: 00, 00, 00, 00, 00

И, наконец, в двоичной:

```
00000000
00000000
11000011
00000000
00000000
```

Чтобы записать эти данные в память Альтаира, нам нужно выполнить такую последовательность действий:

1. Запускаем эмулятор и нажимаем On.
2. Тумблеры A1–A16 переводим в нижнее положение и нажимаем Examine, этим мы устанавливаем указатель на первую ячейку памяти.
3. На тумблерах A1–A8 набираем первый байт и нажимаем Deposit.
4. Вводим следующий байт и нажимаем Dep Next.
5. Пока не введены все байты, делаем пункт 4.
6. Когда введены все байты, нажимаем Reset и запускаем программу. Это можно сделать тремя способами:

- первый – нажать Run. Программа выполняется очень быстро;
- второй – нажать Slow. Программа будет выполняться медленно (хорошо для отладки программы);
- третий – выполнять программу по шагам, нажимая SS.

Если вы правильно ввели вышеописанную программу, то после запуска можно увидеть, что программный счетчик не уходит далее третьего байта. Значит, программа работает.

Вот мы и разобрались с программированием на компьютере Альтаир 8800. В следующей статье мы рассмотрим язык программирования Альтаир бейсик.

Эмулятор Альтаира можно скачать здесь
Kabardcomp.narod.ru/altair.zip

Александр Завгородний (Kakos_Nonos)
<http://kabardcomp.narod.ru/>

Проигрывание трекерной музыки в Borland Pascal

В этой статье мы поговорим о том, как воспроизводить трекерную музыку в программах, написанных на Паскале.

Мы будем использовать самый популярный формат трекерной музыки – Mod.

Для проигрывания Mod существует библиотека MODS. Она полностью написана на Ассемблере, и поэтому у неё очень низкие системные требования. Также она поддерживает много различных устройств вывода звука.

Рассмотрим основные команды:

- **Playmod('Filename.mod');** Начинает играть файл **Filename.mod**.
- **Stopmod;** Останавливает игру.
- **Setvolume(volume);** Устанавливает громкость. **Volume** – переменная типа Byte.

Также эта библиотека содержит несколько служебных переменных.

Device – устройство воспроизведения. Этой переменной нужно присваивать значение до запуска музыки.

Вот значения:

- 0 – ПК-спикер;
- 1 – Ковокс на LPT1
- 2 – Ковокс на LPT2
- 3 – Ковокс на LPT3
- 6 – Sound Blaster
- 255 – Без звука

MixSpeed – частота микширования, по умолчанию 10000.

Библиотека проигрывает файлы в фоновом режиме, абсолютно не мешая основной программе.

Вот пример простого проигрывателя MOD:

```

program mod_player;
uses crt,mods;
var vol:byte;
    fn:string;
begin
  clrscr;
  Textcolor(10);
  Writeln('Введите имя Файла');
  readln(fn);
  Writeln('0 - Спикер, 1 - Ковокс');
  Readln(Device);
  MixSpeed:=10000;
  playmod(fn);
  Writeln("'+' - Vol+ '-' - Vol- Esc - exit');
  vol:=255;
  Repeat
  case readkey of
    '+':begin vol:=vol*2; setvolume(vol); end;
    '-':begin vol:=vol div 2; setvolume(vol); end;
    #27:begin stopmod; halt; end;
  end;
  until false;
end.

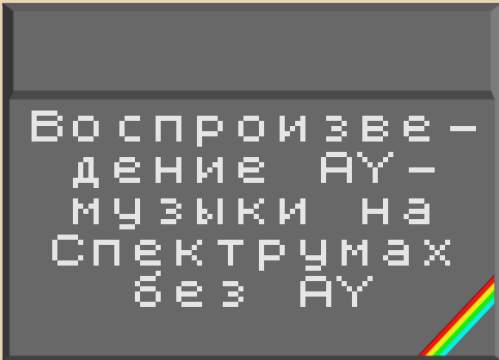
```

Библиотеку MODS можно скачать здесь:

[Kabardcomp.narod.ru/mods.rar](http://kabardcomp.narod.ru/mods.rar)

Александр Завгородний (Kakos_Nonos)

<http://kabardcomp.narod.ru/>



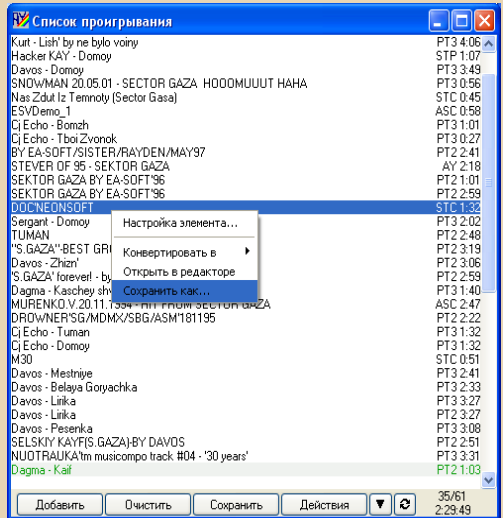
ногда хочется в свою программу или демку вставить музыку, но на Спектруме нет музыкального сопроцессора, а есть только бипер. Конечно, можно поставить биперную музыку, но её не так много, а самому написать не хватает знаний.

Сейчас мы рассмотрим простой метод, с помощью которого можно воспроизвести АУ-музыку на Спектруме без АУ-3-8910.

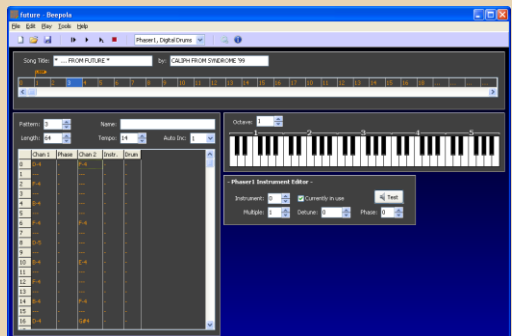
Для начала, нам потребуется архив АУ-музыки от Сергея Бульбы, его же эмулятор АУ/УМ и Vortex Tracker. Всё это можно скачать на официальном сайте bulba.untergrund.net.

После скачивания и распаковки этих файлов, запустите эмулятор АУ. Найдите музыку, которую хотите сконвертировать. Откройте плейлист нажатием на кнопку PL, щёлкните правой кнопкой на играющей музыке и выберите «Сохранить как».

Сохраните эту мелодию в какую-то папку, которую достаточно легко найти.



Далее, откройте с помощью Vortex Tracker'a эту мелодию и сохраните её как текстовый модуль. Нам потребуется редактор биперной музыки Veerola. Запустите редактор и откройте меню File->Import->Vortex Tracker II text module. В открывшемся окне выберите, какой канал исключить, так как в АУ музыке 3 канала, а Veerola поддерживает только два. Исключаемый канал можно достаточно легко определить методом перебора.



Если лишний канал определён верно, то мелодия, в принципе, должна уже быть готовой, но обычно, чтобы привести её в хороший вид, надо ещё немного поработать. В основном, приходится уменьшать или увеличивать скорость. Это можно делать вручную для каждого паттерна, а можно и автоматически для всей песни, щёлкнув по меню Edit->Song->Adjust Tempo. Также нужно выбрать правильный движок музыки. Его можно найти в выпадающем списке сверху.

Теперь, когда мы сделали биперную версию этой музыки, нам нужно скомпилировать эту музыку. Выбираем меню Tools->Compile Song. Открывается меню настроек компилятора. В нем можно выбрать:

- Адрес, на который компилировать
- Цвет бордюра
- Тип файла: .tap с загрузчиком или без него, исходник на ассемблере или бинарный файл.
- Смещение частоты
- Движок музыки
- Зацикливание
- Условия выхода

После этого нажимаем и сохраняем откомпилированную музыку в нужное место.

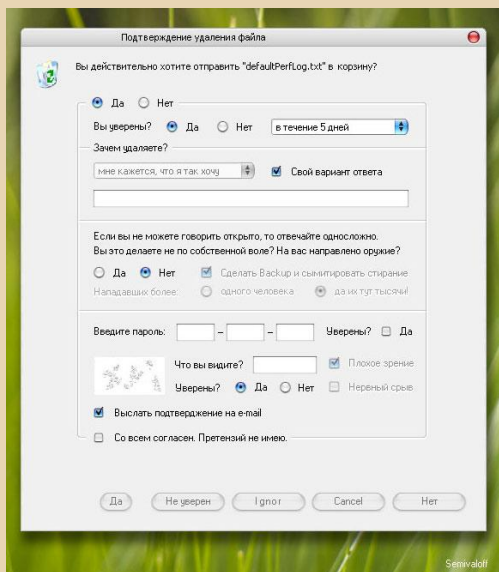
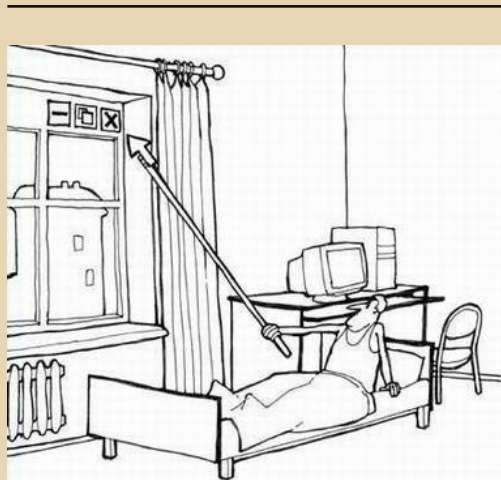
Можно открыть его в любом эмуляторе Спектрума, или запустить на реале.

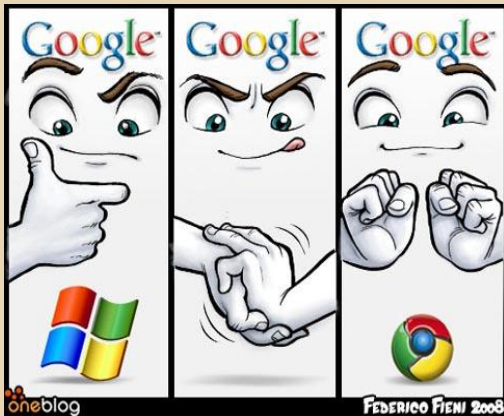
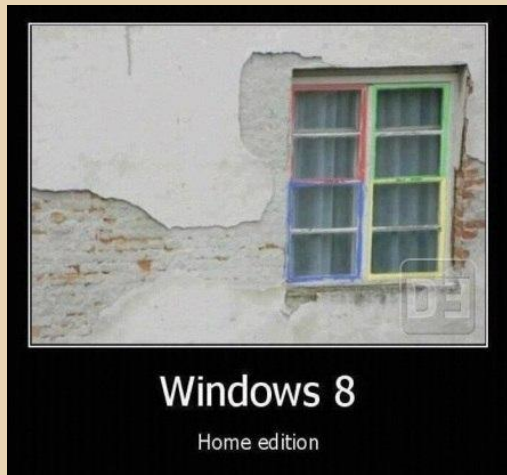
Конечно, качество обработанной музыки немного хуже, чем оригинала, но если хочется сконвертировать музыку, то это единственный вариант.

Александр Завгородний (Kakos_Nonos)
<http://kabardcomp.narod.ru/>



ПРОСТО РАЗНЫЙ ЮМОР





Над журналом работали

Дизайн/вёрстка/редактор - uav1606

Авторы:

Александр Завгородний (Kakos_Nonos)

Абадябер

Вячеслав Рытиков (eubrc)

Дмитрий Караваев

Георгий Кузнецов (Герман)

Михаил Буторин (europe)

Глеб Деревлёв(Dispatcher14)

Сайт журнала:

<http://dgmag.in>

Раздел журнала на "Железных Призраках

Прошлого":

<http://www.phantom.sannata.ru/articles/dgmag/>

E-mail редактора: uav16060 [cobaka] mail.ru