

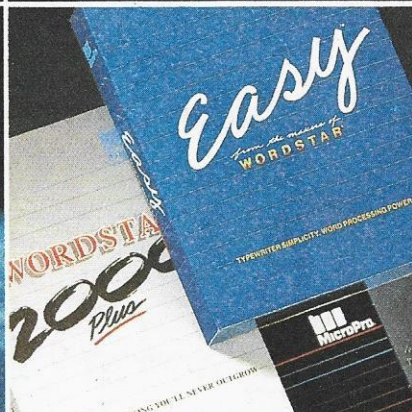
ИНТЕРКОМПЬЮТЕР

INTERCOMPUTER

2/90



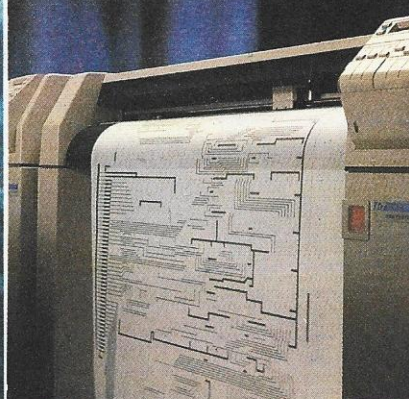
Первые
сообщения
«Пробирной
ПАЛАТЫ»



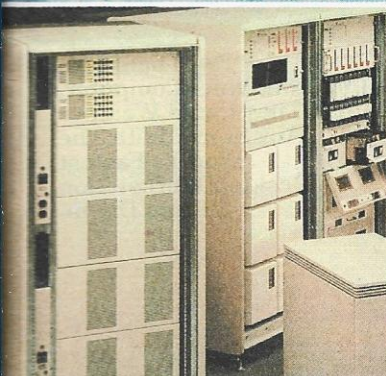
СОВРЕМЕННЫЕ
РЕДАКТОРЫ
ТЕКСТА



P-CAD
ДЛЯ
ЛЮБЫХ
ПЛАТ



ОТ PDP-11
К VAX-11



ВЫСОКОИНТЕЛЛЕКТУАЛЬНАЯ СИСТЕМА
для компьютеров типа IBM PC/XT/AT



SPELLCHECKER FOR RUSSIAN TEXTS

- избавит Ваши тексты от ошибок и опечаток;
- проверит любой русскоязычный текст по правилам орфографии и грамматики;
- позволит включать новые слова в свой словарь, содержащий 120000 слов;
- совместима с любым текстовым редактором;
- проста в обращении;
- поможет при изучении русского языка

103104, Москва,
ул. Остужева, 7, корп. 2
Телефон: 290 35 24,
299 99 04

ИНФОРМИТК
INFORMATICS



В Н О М Е Р Е

*Возможна ли
демократизация общества
без освобождения
творческого труда?*

6

*Is it possible
to democratize society
without creative activities
liberation?*

*Опыт
практической работы
с текстовыми процессорами*

15

*Experience
of practical activity
with text processors*

*Компьютеры
серии VAX-11:
особенности архитектуры*

21

*Computers
of VAX-11 series:
special features of architecture*

*“Пробирная палата”
ИНТЕРКОМПЬЮТЕРА:
Результаты тестирования
UNIX-совместимых ОС*

30

*First test report
of INTERCOMPUTER
Testing Laboratory:
UNIX-compatible OS*

*Почему проектировщики
многослойных печатных плат
выбирают
систему P-CAD?*

33

*Why do the designers
of multilayer printed circuit boards
choose
the system P-CAD?*

*Классификация
вирусов
и антивирусных программ*

37

*Classification
of viruses
and antiviral programs*

СОДЕРЖАНИЕ

СОБЫТИЯ

3

Выставка программных продуктов
фирмы Ashton-Tate
Выставка-семинар "Прикладные системы
искусственного интеллекта"
Визит представителя фирмы Informix Software
в Москву
"Социальная информатика - 90"

МНЕНИЯ

6

Информационное общество и демократия.
А.Н. Козырев
Идолы массовой компьютеризации. *С.П. Львов*

ЛИКБЕЗ

15

Текстовые процессоры для IBM-совместимых ПК.
А.В. Смородинский, О.Ю. Посудина

СЕМЕЙНЫЕ ХРОНИКИ

21

Серия VAX-11. *А.С. Богданов*
UNIX и микропроцессор Intel 80386.
В.В. Леонас
Grep и awk - опыт сравнения. *Г.Г. Кольнер*

ПРОБИРНАЯ ПАЛАТА

30

UNIX-совместимые ОС

ВОКРУГ САПР

33

R-CAD для любых плат. *А.И. Масалович*

АНТИВИРУСЫ

37

Классификация вирусов. Попытка стандартизации.
Н.Н. Безруков
Классификация антивирусных программ.
И.Ш. Карасик
Вирусы и антивирусы на IBM-совместимых ПК.



**КОЛОНКА
ИЗДАТЕЛЯ**

Что нового в "Интеркомпьютере"?

Вопрос этот не праздный, но дежурный, поскольку один из главных принципов "Интеркомпьютера" - постоянно радовать читателей чем-нибудь новеньким, сохраняя высоту стиля и качества изложения. "Интеркомпьютер" хочет быть узнаваемым и популярным.

Сегодня у "Интеркомпьютера" важное событие - при бюллетене открыта лаборатория тестирования программного обеспечения. Может быть, для советского читателя смысл этого мероприятия не совсем понятен, однако за рубежом никто не покупает пакеты программ, не проконсультировавшись с любимым компьютерным журналом. "PC Magazine" и "PC World", "Byte" и "PC Resource", "Personal Computer World" и "InfoWorld" - все эти издания регулярно публикуют результаты тестирования компьютеров и программ. При каждом журнале действует группа независимых экспертов, которые, оставаясь анонимными, выносят приговоры, окончательные и обжалованию не подлежащие. Каждая лаборатория выдает фирмам-разработчикам лучших изделий сертификат качества, который те успешно используют в рекламных целях.

Большинство зарубежных читателей именно потому и покупают компьютерные журналы, что хотят правильно сориентироваться в вопросах вложения денег. Очевидно, что тираж журнала существенно зависит от качества работы лаборатории тестирования.

У наших читателей может возникнуть вопрос еще и о том, почему подобные лаборатории работают именно при журналах. Ответ прост: любая информация, рассчитанная на широкий круг потребителей, имеет смысл только при массовом ее тиражировании.

Но вернемся к нашим делам. "Пробирная палата" (так "Интеркомпьютер" назвал свою лабораторию тестирования) будет следовать лучшим зарубежным образцам, но не копировать их, а учитывать специфику советского рынка программного обеспечения. Надеемся, что информация "Пробирной палаты" будет полезной для вас...

И еще об одном нововведении. Отныне большинство статей мы будем снабжать словарями терминов, чтобы не только специалисты, но и начинающие пользователи могли легко ориентироваться в тексте.

Издательский отдел СП "Интерюнити"
Руководитель отдела А.А. Эйдес
Редакторы М.С. Гордон
О.В. Толкачева
Главный художник А.С. Широков
Обложка А.М. Ефремова
Рисунки Л.Х. Насырова
Фото В.Н. Кирюхина

Наш адрес: 121069, Москва,
ул. Чайковского, 20а
Телефон: 202-92-80
Телекс: 413932 NIDEL SU
Телефакс: 2302035

Издательство "Прометей" МГПИ им. В.И. Ленина Л13597 Зак.157
МПО "1-я Обр. тип." Объем 7,5 уч.-изд. л. 50 000 экз. Цена 3 руб.
© СП "Интерюнити", 1990

ВЫСТАВКА ПРОГРАММНЫХ ПРОДУКТОВ ФИРМЫ ASHTON-TATE

С 16 по 22 января 1990 г. в ГПНТБ СССР (Москва) проходила выставка программных продуктов фирмы Ashton-Tate. На выставке была также представлена имеющаяся в ГПНТБ литература по соответствующей тематике. Целью выставки являлось не только предоставление советским пользователям информации о деятельности фирмы, но и изучение советского рынка программного обеспечения. Создав версии своих основных пакетов на 16 языках, фирма Ashton-Tate начала проявлять серьезную заинтересованность в освоении восточно-европейских рынков программного обеспечения, в частности советского. На выставке фирма продемонстрировала следующие новые разработки:

- dBASE IV - одна из наиболее мощных и распространенных в мире систем управления базами данных (СУБД), реализующая широкие возможности создания файлов, выполнения запросов по образцу, разработки форм, генерации типовых бланков и отчетов. Принятый в системе язык программирования dBASE IV/SQL представляет собой расширенный вариант реализации стандарта языка SQL, содержащий более 310 команд и функций.
- dBASE IV Developer's Edition - интегрированная среда для профессиональных программистов, разрабатывающих собственные сложные системы на основе dBASE IV. Помимо самой СУБД в него входят функционально полный набор утилит и средств программирования, а также язык шаблонов.
- dBASE IV LAN - сетевой вариант dBASE IV, обеспечивающий

одновременную работу нескольких (до пяти) пользователей с базой данных при использовании развитых средств защиты данных от несанкционированного доступа.

- StepIV Ward - программа обмена данными между СУБД dBASE IV и системами Clipper, FoxBASE+, Quicksilver.
- Framework III - интегрированный пакет для работы с деловой документацией, содержащей текстовую и графическую информацию. Включает в себя редактор текста, СУБД и средства для обработки таблиц. Кроме того, пакет обладает расширенными графическими возможностями и в него встроен язык FRED, позволяющий создавать новые прикладные системы. От более ранних версий пакет отличаются возможность многоколоночного набора текста, наличие словаря синонимов и многоязыкового корректора орфографических ошибок, а также более высокая скорость обработки баз данных и таблиц размером до 32000 граф x 32 строки. Расширены средства сопряжения с системами dBASE и Lotus 1-2-3.
- Framework III LAN - сетевой вариант пакета Framework III.
- Framework III Run Time - версия пакета Framework III для профессиональных программистов. В состав пакета включена дополнительная программа, облегчающая разработку прикладных программ.
- RapidFile - быстродействующая и простая СУБД, обеспечивающая эффективное манипулирование данными при сравнительно простых запросах пользователя, работающего с малым числом файлов. В состав RapidFile версии 1.2 входят корректор орфографических ошибок,

рассчитанный на проверку 80 тыс. слов, и словарь синонимов на 470 тыс. терминов.

- TurboSearch - высокоскоростная информационно-поисковая система, работающая с текстовыми файлами, а также с файлами, информация в которых представлена в форматах систем dBASE III, Framework IV и MultiMate Advantage II.
- FileRecovery - средство восстановления поврежденных файлов для систем dBASE III+ и dBASE IV.
- FrontRunner - программная среда для разработки и выполнения резидентных программ пользователя. Подобные программы пока являются редкостью, однако каждый, кто пробовал писать собственные

СОБЫТИЯ

резидентные программы, оценит полезность такого инструмента для системного программиста.

- MultiMate Advantage II - простой и функционально полный пакет для обработки текстовой информации. В последнюю версию пакета входят расширенные средства для обработки документов: верстки страниц, печати типовых писем, многоколоночного набора, рисования линий и прямоугольников и т.д. Поддерживает шесть математических функций и прямой доступ к dBASE. Особенность данного пакета - наличие драйверов для более чем 400 принтеров.
- Applause - пакет деловой графики, позволяющий работать с текстами, строить графики и создавать рисунки, используя 256 цветов и большой набор специальных приемов.

Сознавая специфический характер процесса формирования цен на программные продукты для новых рынков, фирма Ashton-Tate не устанавливает там жестких цен на свои изделия, а ограничивается указанием рекомендованных цен (RRP - Recommended Retail Price). Для перечисленных выше продуктов такими рекомендованными ценами (в долларах) являются следующие:

dBASE IV	795,00
dBASE IV Developer's Edition	1295,00
dBASE IV LAN	995,00
StepIV Ward	99,95
Framework III	695,00
Framework III LAN	995,00
Framework III Run Time	995,00
RapidFile	295,00
TurboSearch	129,00
FileRecovery	99,95
FrontRunner	99,95
MultiMate Advantage II	565,00
Applause	495,00

Во время проведения выставки особенно подчеркивались преимущества "легального" приобретения продуктов фирмы Ashton-Tate. Достоинства официального взаимодействия с фирмой действительно весьма ощутимы. Во-первых, пользователь получает оригиналы систем, что дает гарантию их полноты и отсутствия вирусов. Во-вторых, зарегистрированным пользователям фирма гарантирует сервисное обслуживание, консультации и обучение. Наконец, "законные" владельцы программных продуктов получают все новые версии за минимальную дополнительную плату.

А.И. Масалович

ВЫСТАВКА-СЕМИНАР "ПРИКЛАДНЫЕ СИСТЕМЫ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА"

Выставка-семинар под таким названием прошла в Москве 14 марта. Среди ее организаторов - Научный совет по проблеме "Искусственный интеллект", Мос-

ковский дом научно-технической пропаганды им. Ф.Э. Дзержинского, Инновационное объединение АН СССР. Всех участников выставки в этом кратком сообщении перечислить не удастся; расскажу о том, что привлекло внимание.

Во-первых, были представлены различные секторы экономики. Государственные предприятия на выставке благополучно соседствовали с кооперативами и совместными предприятиями. С точки зрения посетителя, такое соседство было интересным - можно было непосредственно сравнить достижения тех, других и третьих. Вывод из этого сравнения неоригинален: все зависит не от того, в каком секторе работают разработчики, а от их квалификации. Например, на одинаково высоком профессиональном уровне выполнены пакет "ОРФО" для проверки русской орфографии, который представило Межреспубликанское научно-техническое объединение "Информатик", графический пакет Rainbow (СП "Диалог", Центр в МГУ) и автоматизированная система моделирования производства "АМИГО" (ВЦ АН СССР).

Во-вторых, в выставке приняли участие не только московские организации. Были представлены Ленинград, Ташкент, Днепропетровск. Популярностью пользовалась экспозиция кооперативной фирмы ПЛЮС/МИКРО (Караганда), предложившей в числе других пакетов систему автоматизации бухгалтерского учета, в основу которой положена монография венецианского купца XV века Луки Пачоли. Экстравагантность предложенных решений подчеркивалась рекламным плакатом, на котором милая девушка, прикрывая наготу 5-дюймовой дискетой, утверждала, что программное обеспечение фирмы ПЛЮС/МИКРО никогда не изменит.

В-третьих, значительное число экспонатов в той или иной степе-

ни имело отношение к медицине. Так что можно надеяться, что наше здоровье скоро улучшится. Круг разработчиков достаточно широк - от троцкого филиала Института атомной энергии им. И.В. Курчатова (АРМ кардиолога, компьютеризованный комплекс для лазерной хирургии, программы моделирования процессов генеза и нарушений ритма сердца) до СП "Диалог" (экспертные системы по траволечению и для диагностики методами тибетской медицины). Кстати, СП "Диалог" представило, пожалуй, наибольшее число экспонатов по самой разнообразной тематике.

Если добавить к сказанному, что работники ГПНТБ СССР развернули в фойе выставку литературы по вычислительной технике и организовали продажу периодических компьютерных изданий, то станет ясно, почему мероприятие, затейливо названное выставкой-семинаром, получилось достаточно интересным.

А.А. Аба

ВИЗИТ ПРЕДСТАВИТЕЛЯ ФИРМЫ INFORMIX SOFTWARE В МОСКВУ

Известная американская фирма Informix Software, специализирующаяся в области разработки систем управления базами данных (СУБД) и инструментальных средств для создания на их основе различных информационно-поисковых систем, функционирующих под управлением UNIX-совместимых ОС, начала "прощупывать" советский рынок. С этой целью представитель фирмы (точнее, ее Европейского бюро) дважды посетил СССР; во время последнего визита состоялась передача предлагаемого фирмой Informix Software программного обеспечения ряду организаций для тестирования и опытной экс-

платации. Так, Таллиннскому центру развития и СП "ИНТЕР-КВАДРО" были переданы следующие программные продукты, предназначенные для работы в среде OS/2: реляционная СУБД INFORMIX-SQL и инструментальные средства (инструментарий) INFORMIX-4GL для разработки информационно-поисковых систем.

Реляционная СУБД INFORMIX-SQL предназначена для решения таких задач пользователя, как:

- создание баз данных и электронных таблиц;
- проектирование экранных форм;
- ввод и модификация данных с помощью ранее созданных экранных форм;
- загрузка информации из файлов в базы данных;
- формирование запросов к базам данных с помощью ранее созданных экранных форм или интерактивного языка запросов;
- генерация и форматирование отчетов;
- проектирование меню.

Инструментарий INFORMIX-4GL для разработки информационно-поисковых систем содержит встроенные средства, обеспечивающие создание баз данных и формирование запросов к ним; проектирование интерактивных экранных форм, поддерживающих в разрабатываемых информационно-поисковых системах удобный для пользователя интерфейс; осуществление генерации отчетов.

В соответствии с достигнутой договоренностью тестирование и опытная эксплуатация предлагаемого программного обеспечения продлятся до 30 апреля 1990 г.

И.Г. Шестаков

"СОЦИАЛЬНАЯ ИНФОРМАТИКА-90"

Открытие всесоюзной школы молодых ученых и специалистов с таким названием состоится 26 апреля в п. Гурзуф Крымской области.

Основной целью школы является привлечение ее участников к обсуждению наиболее актуальных проблем информатизации общества, развития международных информационных связей, влияния ЭВМ на образ жизни и профессиональную деятельность.

В течение восьми дней работы школы предполагается обсудить результаты научных исследований и практических работ по следующим направлениям:

- международные информационные связи;
- социальные аспекты и последствия информатизации общества;
- информационно-вычислительные средства в процессе управления обществом;
- использование ЭВМ в научных исследованиях экономических и социальных процессов;
- молодежь в условиях информатизации общества (социально-философские, экономические и правовые проблемы);
- проблемы информатизации системы образования, досуга и быта;
- участие молодежи в разработке и освоении новых информационных технологий;
- организация компьютерного туризма;
- новые информационные технологии в полиграфической промышленности.

Организаторами и официальными спонсорами школы являются: Институт проблем информатики АН СССР; Всесоюзный институт научной и технической информации; Всесоюзное общество информатики и вычислительной техники; Всесоюзный совет молодых ученых и специалистов; Международный компьютерный клуб; Институт информационных и социально-экономических исследований; Издательско-полиграфическое объединение ЦК ВЛКСМ "Молодая гвардия"; Центр компьютерного творчества молодежи "Юность" ЦК ВЛКСМ; Бюро международного молодежного туризма "Спутник" ЦК ВЛКСМ; Высшая комсомольская школа при ЦК ВЛКСМ; СП "Диаг интернешнл"; Центр НТТМ

СОБЫТИЯ

"Матэна" (Минск); Компьютерный центр "Дельта" (Смоленск); Хозрасчетный центр научно-технических услуг "Информэкс".

Основные организационные хлопоты взяли на себя сотрудники Кафедры социальной информатики Высшей комсомольской школы при ЦК ВЛКСМ, которая к концу апреля, возможно, уже сменит свое название.

Согласие на участие в работе школы дали специалисты из Аризонского университета и Американского центра международных лидеров.

Повышенный интерес к школе позволяет строить планы относительно ее превращения в международную. Кроме того, мы надеемся, что проведение этой школы станет традиционным.

С.В. Кондратьев

МНЕНИЯ

А.Н. КОЗЫРЕВ



Соблазн информатизации витает над нашей страной и порождает множество новых сюжетов внутри старых проблем, включая главную нашу проблему - преодоление тоталитаризма. В преодолении тоталитаризма - общность исторических судеб информационного общества и демократии в СССР, если понимать демократию в соответствии с современными представлениями, характерными для большинства парламентских партий Европы, как право свободного выбора для индивида, обеспеченное механизмами представительской демократии. Тоталитаризм всячески ограничивает индивидуальную свободу, сводя ее в лучшем случае к осознанной необходимости, но именно в ограничении индивидуальных свобод - источник неэффективности тоталитарной общественно-экономической системы, что особенно отчетливо проявилось при переходе от индустриального общества к информационному. Внешне это выразилось в очевидной неспособности системы эффективно использовать современные информационные технологии и микропроцессорную технику, без которых бессильна современная промышленность, анемична власть и ненадежна оборона.

Информационное общество и ДЕМОКРАТИЯ

...Но это же сон волка в мерзлую ночь о том, "что за морем".

В.В. Розанов. Сахарна

Освобождение творческого труда как шанс на выживание

Парадоксальность ситуации - в том, что необходимая для укрепления власти и техническая по сути революция, именуемая информатизацией, невозможна без глубоких демократических преобразований во всех областях общественной деятельности, включая экономику, политику, право, идеологию и мораль. Технические проблемы не могут быть решены без преобразований, обеспечивающих максимальную свободу личной инициативы. Свобода инициативы требует соответствующего правового обеспечения, а это означает радикальное изменение идеологических и нравственных установок в таких вопросах, как собственность, эксплуатация, права и свобода личности.

С исчезновением частной собственности на средства производства эксплуатация вовсе не исчезает, как это представлялось нескольким поколениям русской демократической интеллигенции, - исчезает что-то совсем другое. "Можно с тем же правом, как говорят об эксплуатации рабочего, говорить об эксплуатации капиталистами и рабочими в совокупности творцов и создателей орудий производства", - писал В.И. Вернадский жене в письме от 21 июля 1917 г. Семьдесят лет спустя его мысль приобрела новое, более широкое звучание. Во-первых, в последние десятилетия, особенно с появлением современной вычислительной техники, стремительно возрастает капиталоемкость творческого труда. Зависимость творческого труда от капитала становится более явной и более похожей на обычную зависимость нанимаемого от нанимателя. По прогнозам большинства западных футурологов, в информационном обществе

творческий труд станет преобладающим. Следовательно, резко расширится слой эксплуатируемых и возрастет объем эксплуатации. Во-вторых, современное представление о менеджменте однозначно определяет труд менеджеров как творческий, следовательно, есть основания уточнить представления об эксплуатации наемных рабочих частным бизнесом. В третьих, широкое распространение коллективных форм собственности, среди которых наиболее известными являются кооперативная и акционерная, ведет к постепенному стиранию грани между собственниками и наемными рабочими. Применительно к СССР эти три обстоятельства можно дополнить еще одним. Нещадная эксплуатация активной части общества его пассивной частью столь болезненно сказывается на эффективности общественного производства в нашей стране, что поставлено под сомнение будущее существующего общественного строя. Из всех пере-

численных обстоятельств реально осознается самим обществом только последнее, не требующее для понимания ничего, кроме страха перед будущим. Гораздо важнее понять, что освобождение творческого труда и демократия по отдельности неосуществимы: свобода от эксплуатации очень тесно переплетается с правом на труд, интеллектуальной свободой и свободой слова.

Появление новых технических средств обработки и распространения информации избавило интеллектуальный труд от многих рутинных операций и ограничений чисто физического характера. В результате обнажились и обострились противоречия, обусловленные экономическими, идеологическими и правовыми ограничениями. Относительно благополучно существовавшая в нашей стране на протяжении более полувека тоталитарная система явно обнаружила неприятие широкого спектра новых способов информационного общения и очень сдержанное отношение к поддерживающим их техническим средствам. Соответствующая традиция идет из времен, когда перед каждым праздником печатались пишущие машинки, а множительная техника рассматривалась почти исключительно как орудие контрреволюционной агитации. Хотя в современных условиях все это оборачивается неприятием технического прогресса и грозит системе гибелью, изменение такой политики идет очень медленно, что можно проиллюстрировать на примере распространения персональных компьютеров (ПК) и другой микропроцессорной техники.

Если в США один ПК приходится на 7 человек, в Австрии - на 9, то в СССР - на 560 (по американским данным). Такое положение можно попытаться объяснить более низким уровнем жизни в СССР и другими объективными причинами, однако в очень близкой к СССР по уровню жизни Польше один ПК приходится на 46 человек, в Венгрии - на 26, а в более богатой Чехословакии - на 74. В каждом конкретном случае можно найти свое объяснение ситуации, но в целом определенная тенденция очевидна. Еще более резкие различия имеют место в степени доступности модемов (которых в СССР почти нет), телефаксов и другой техники, ориентированной преимущественно на обработку и распространение текстовой информации.

Ошеломляюще выглядят различия в экономической доступности компьютерной техники. По моим собственным подсчетам, отношение средней цены ПК к средней зарплате в США примерно в 400 раз ниже, чем в СССР.

Освобождение творческого труда и демократия по отдельности неосуществимы: свобода от эксплуатации очень тесно переплетается с правом на труд, интеллектуальной свободой и свободой слова.

ФАКТЫ

■ Фирма Future разработала программу FAST!, которая с помощью специального метода кэширования жесткого диска позволяет увеличить скорость обмена с ним данными в десятки раз. Эта программа была опробована в работе с 650 прикладными программами. Испытания показали, что минимальный коэффициент ускорения обмена данными с жестким диском равен 5, максимальный - 60. Цена программы FAST! 99,95 дол. Объем занимаемой ею оперативной памяти 8 Кбайт.

Для обычных потребительских товаров это отношение меньше в 50 - 100 раз, т.е. наблюдается та же самая тенденция. Картина будет полнее, если к этому добавить наличие чисто административных ограничений на доступ к информации и копировальной технике, а также различного рода дефицитов и лимитов (например, лимита на бумагу). По совокупности причин условия для интеллектуальной деятельности в СССР оказываются существенно хуже, чем в любой развитой капиталистической стране и даже в большинстве развивающихся стран. Парадоксальная ситуация заключается в том, что страна продолжает тратить значительные усилия и средства на создание препятствий к распространению информации и технических средств ее обработки. Наличием искусственных препятствий в значительной степени объясняется и крайне слабая оснащенность СССР компьютерной техникой, что особенно отчетливо видно при сравнении с Польшей, относительно высокая степень компьютеризации которой достигнута исключительно благодаря устранению препятствий свободному передвижению граждан, свободной продаже долларов за злотые и предоставлению польским гражданам права работать в западных фирмах.

Индивидуальная свобода и персональные компьютеры

Сила воздействия ПК на уклад жизни такова, что их появление американцы сравнивают с появлением индивидуального автомобильного транспорта. По словам Г. Форда, автомобиль создал Америку. С таким же основанием можно утверждать, что ПК создал информационное общество американского образца, ос-

новные контуры которого видны уже сейчас. Персональный компьютер, появившийся как игрушка профессиональных программистов и средство для первоначальной отладки программ, очень быстро был применен для обработки текстовой информации, передачи информации внутри организации и создания электронной почты. Вскоре ПК стал приспосабливаться к самым разнообразным видам интеллектуальной деятельности, обрел фанатичных почитателей и не менее фанатичных противников, распространился по всему миру и проник в СССР.

Если индивидуальный автомобиль обеспечил американцу независимость передвижения, то ПК подарил ему свободу и независимость при взаимодействии с нарастающим потоком информации. Получение и распространение самой разнообразной информации с помощью ПК постепенно вытесняет традиционные способы либо успешно сочетается с ними, привнося в информационные процессы новые возможности. Более активное поведение каждой отдельной личности в информационном потоке неизбежно сказывается на ее самооценке, мировосприятии и мировоззрении. В обществе неизбежно происходит переоценка ценностей.

С развитием и распространением ПК, модемов, телефаксов и другой микропроцессорной техники связан новый виток в материализации американской мечты о неограниченной индивидуальной свободе. Этой мечтой пронизаны описания информационного общества в сочинениях А. Тоффлера. Та же подоплека, по видимому, и у мрачных прогнозов о возможном порабощении западного общества всемогущими средствами массовой информации или технократией с помощью этих средств. Реальная опасность, скорее всего, сильно преувеличивается и будоражит вольнолюбивые умы. Забавна роль в этом хоре наших отечественных защитников демократии от информационного террора. Ведь Запад не знал ничего, подобного идеологическому давлению, сопровождавшему сталинский террор, и не испытал счастья воскрешения из исторического небытия. Представление о тоталитарном информационном обществе современный американец или западный европеец может почерпнуть только в литературе, подобной роману Дж. Оруэлла "1984", либо в достаточно специальных работах советологов. Советскому человеку проще - достаточно иметь не совсем исковерканную память, толику образования и капелку воображения. И все же мечта о неограниченной свободе, ставшая основой русского анархизма и одним из основных мотивов русской православной философии, сохраняет в нашем обществе удивительную жизнеспособность. Может быть, именно здесь скрыты источник почти религиозной любви советского программиста к ПК и причины появления своих хэкеров и чемпиона мира по игре "Бой в памяти" (Е.П. Лилитко) практически при отсутствии ПК.

Принято считать (и писать), что использование ПК повышает производительность ин-

теллектуального труда в два - три раза. На этом предположении строятся даже методики определения потребности нашей страны в ПК до 2005 г. Действительность чуточку сложнее. В одних руках ПК становится мощнейшим инструментом, многократно увеличивающим интеллектуальные возможности, в других - усовершенствованной пишущей машинкой или дорогой игрушкой, а то и просто обузой. Даже у профессиональных программистов производительность труда различается в десятки раз. Не имея

возможности провести точные измерения, я могу лишь экстраполировать результаты своих личных наблюдений за работой относительно небольшого числа профессионалов. Мой скромный опыт наблюдений и обобщений свидетельствует, что примерно 10 % наиболее квалифицированных программистов производят более 90 % всего программного продукта. Вот почему правильнее говорить не об увели-

чении производительности труда во столько-то раз, а о возведении интеллектуальных возможностей в квадрат, если арифметические сравнения здесь вообще уместны. Следовательно, неизбежным спутником компьютеризации и информатизации общества является резкое социальное расслоение, не имеющее ничего общего ни с кастовым (номенклатурным), ни с имущественным расслоениями. Это расслоение общества по способностям к общению с ПК, овладению информационными технологиями.

Социальные последствия расслоения по способностям уже заметны по обе стороны океана. Так, в США, где ПК имеются в огромном количестве семей и доступны практически всем их членам, многие дети, создающие компьютерные игры, зарабатывают больше родителей. В СССР таких проблем пока нет, но уже появился слой программистов, очень хорошо зарабатывающих в свободное от основной работы время. Нормальной стала ситуация, когда инженер-программист или младший научный сотрудник академического НИИ имеет приработок, превышающий основную зарплату директора того же института. Соблазн запретить и лишить доступа к компьютерам велик, но чреват последствиями. Уже сейчас для советских программистов открылась возможность работать в западных фирмах, где квалифицированному программисту ежемесячно платят 4500 дол и больше. Пока

этой возможностью воспользовались единицы (если не считать тех, кто сменил работу вместе с гражданством), однако запрет нормально зарабатывать в своем отечестве в сочетании с предполагаемой в недалеком будущем свободой выезда неизбежно станет концом советского программирования. Нечто подобное происходит и в других областях интеллектуальной деятельности, но именно в программировании конфликт между реальностью и потенциальными возможностями достиг сегодня

предельной остроты. Использование энергии этого конфликта и интеллектуальный потенциал всего нашего общества на его же благо - задача более трудная и благородная, чем удовлетворение сиюминутных желаний люмпенизированной общности и амбиций высокочинных недоучек. Поэтому необходимо предложить способным людям те организационные формы и условия работы, которые не выталкивали бы их ни из общественного производства, ни тем более из страны.

Наиболее естественная для высококлассного и самостоятельно мыслящего программиста форма организации работы - собственная фирма, в которой он делает основную работу, а наемные служащие выполняют второстепенную. Кто кого при этом эксплуатирует - большой вопрос,

однозначно ответить на который сегодня может только самонадеянный дурак. Появление относительно дешевых ПК практически снимает вопрос об эксплуатации наемного труда на основе имущественного неравенства. Гораздо важнее при этом неравенство творческих возможностей, которое на каком-то этапе оборачивается имущественным неравенством. Аналогичная ситуация возможна и в других интеллектуальных видах деятельности, не обязательно связанных с использованием ПК в качестве основного рабочего инструмента. Неравенство Шарикова и профессора Преображенского в недавно реабилитированной повести М. Булгакова - прекрасная тому иллюстрация. Кооператив программистов не является адекватной альтернативой частной фирме того же профиля, поскольку предполагает равноправие в решении ключевых производственных вопросов всех членов кооператива, т.е. заведомо неравных по интеллектуальным возможностям людей, а это обычно приводит к эксплуатации лидера более слабыми партнерами. Совершенно фантастические возможности для эксплуатации чужого таланта дает существующая система государственных НИИ и КБ. Избегать нещадной эксплуатации со стороны администраторов от науки удается главным образом полным бездарностям. Методы, которыми осуществляется эксплуатация талантов и просто способных людей, многооб-

разны и во многом осквернены традицией, что, увы, не способствует ослаблению их разрушительного воздействия. Уменьшить его можно, убрав надуманные запреты из закона о социалистическом предприятии и других законодательных актов, регулирующих производственную деятельность. Предоставление каждому возможности создать собственную независимую фирму - наиболее радикальное и предельно естественное решение вопроса, объективно ведущее к уменьшению эксплуатации в обществе. К позору ортодоксов, ослабление эксплуатации при этом связано не с уничтожением, а с возникновением частной собственности на средства производства.

Более глубокий слой проблем в деле освобождения творческого труда связан с охраной интеллектуальной собственности. К ним относятся проблема правовой незащищенности

ФАКТЫ

■ Фирма Microsoft разработала новый пакет для пользователей, программирующих на Бейсике, - BASIC Professional Development System (сокращенно PDS или BASIC 7.0). В состав пакета входит интегрированная система управления данными ISAM (Indexed Sequential Access Method), в которой имеются набор встроенных библиотек и база данных. По заверениям разработчиков, пакет BASIC 7.0 позволит составлять большие и работающие с высокой скоростью программы, так как в нем отражены не только все лучшие качества среды QuickBASIC, но и предусмотрены возможности работать с расширенной памятью, вызывать команды DOS и выполнять многие другие полезные для программиста операции. Цена пакета 495 дол.

■ Как уже сообщалось в нашем бюллетене, новые версии пакета Turbo Basic больше появляться не будут. Ему на смену пришел пакет PowerBasic, который распространяет уже не фирма Borland, а фирма Spectra Publishing. Новый пакет позволяет работать с несколькими дополнительными типами данных, например с данными "расширенной точности" (80 бит) и "учетверенными" словами (64 бит). Цена пакета PowerBasic 109.95 дол.

компьютерных программ в СССР и тесно связанная с ней проблема защиты наиболее та-

Запрет нормально зарабатывать в своем отечестве в сочетании с предполагаемой в недалеком будущем свободой выезда неизбежно станет концом советского программирования.

инфраструктура в большей степени приспособлены к созданию препятствий для распространения информации, чем к охране личности. Отсюда растущее недовольство пишущей публики деятельностью ВААП и сомнения многих борцов за права программистов в целесообразности распространения авторского права на компьютерные программы.

Весьма популярна идея создания для охраны программ совершенно нового правового института, не несущего в себе недостатков авторского права. Я считаю эту идею утопической и даже вредной, поскольку она объединяет совершенно несовместимые между собой точки зрения, прикрытые пока общим лозунгом-маской; т.е. популярность идеи достигается за счет своеобразного "лукавства", суть которого - в сокрытии противоречий. В случае же попытки провести этот лозунг в жизнь противоречия неизбежно обнаружатся.

Негативное отношение ортодоксальных марксистов к идее разгосударствления интеллектуальной собственности в значительной мере основано на недоразумении, поскольку классики об интеллектуальной собственности не высказались, а термины "интеллектуальный" и "собственность" чаще всего встречались в контексте, требующем негативной реакции. Однако сходство между собственностью и интеллектуальной собственностью примерно такое же, как между стулом и электрическим стулом.

Здесь мы сталкиваемся с явлением достаточно сложным и трудно узнаваемым, а потому опасным. В конечном счете самым прочным барьером на пути к свободе личности, информационному обществу и демократии может оказаться совокупность идеологических предрассудков, составляющих массовую культуру тоталитарного общества. Среди них представления о собственности как о единственном источ-

нике эксплуатации человека человеком играют совершенно особую роль. Ядовитая сила за-

вести сосредоточена в этом предрассудке подобно тому, как в русской сказке смерть Кощея сосредоточена на кончике иглы. Развеешь предрассудок такого ранга - почти то же, что сломать иглу со смертью Кощеевой на острие.

Оборотной стороной того же предрассудка является отрицание или игнорирование какой-либо общественно полезной роли частной собственности. Традиция эта имеет глубокие и прочные корни в нашей истории.

Истоки и феноменология русского тоталитаризма

В своей работе "Русская идея, основные проблемы русской мысли 19 в. и начала 20 в." (Париж, 1971) Н.А. Бердяев в качестве отличительного признака русской интеллигенции выделяет ее стремление "выработать себе тоталитарное, целостное мирозерцание, в котором правда-истина будет соединена с правдой-справедливостью". Из этого соединения, продиктованного стремлением к совершенной жизни, с достаточно четкой логической последовательностью можно вывести как "пещерное неприятие компромиссов", что, по словам А.И. Солженицина (Нобелевская лекция), "возведено в теоретический принцип и считается добродетелью ортодоксии", так и патологическую лживость идеологии, статистики, обществоведения,

Фактически все крупные русские мыслители XIX века, в том числе Л.Н. Толстой и Ф.М. Достоевский, видели в частной собственности лишь средство эксплуатации человека человеком. Поэтому утопическая и глубоко тоталитарная по сути идея построения общества одновременно справедливого и экономически эффективного очень легко связалась с конкретным способом ее воплощения в реальность через уничтожение частной собственности. Первоначально за таким способом воплощения идеи помимо теоретической поддержки со стороны демократической интеллигенции стоял лишь интерес люмпенов всех разновидностей и мастей, но не более того. Однако после ее материализации интерес люмпенов сменился интересом бюрократии, т.е. идея получила надежное место в АС. С тех пор в нашей политэкономии собственнику всегда противопоставляется труженик, а не бюрократ; хотя тружеником может быть и собственник, и бюрократ, но бюрократом собственник быть не может в принципе, поскольку распоряжается собственным имуществом, а бюрократ - чужим.

В стремлении к единству несоединимого, подкрепленном фанатичной преданностью идее, - наиболее характерная черта тоталитарного сознания. Проявления ее многообразны. В науке это может быть поиск философского камня или универсальной компьютерной программы для решения задач из всех областей жизни, в экономике - наиболее прогрессивная форма собственности, в национальной политике - единое для всех республик внутреннее устройство. В любом случае тоталитарное сознание стремится избежать многообразия форм и рассмотрения их особенностей, решить все проблемы разом, охватить всех и навсегда. С этой точки зрения,

все виды уравниловки - проявления тоталитаризма и тоталитарного сознания, имеющие с демократией достаточно мало общего. Превращение демократии в религию или универсальный инструмент для решения всех общественных проблем - такое же порождение тоталитарного сознания, как религия-революция или попытка построить эффективную экономику путем уничтожения всех форм собственности на средства производства, кроме одной.

Демократия ограничивает возможности чиновных недоучек, поскольку обеспечивает равное участие в принятии решений представителям всех социальных слоев. Интеллектуалам, обремененным знани-

ем и способным предвидеть последствия принимаемых решений, в этой игре не предоставляется каких-либо преимуществ, что справедливо. В то же время объективно это создает почву для конфликта между обще-

ФАКТЫ

■ Многие специалисты считают, что будущее за интегрированными (многофункциональными) пакетами. Фирма Microsoft выпустила очередную версию (2.0) интегрированного пакета Works. В состав пакета входят: текстовый процессор, база данных, программа для обработки электронных таблиц, программа для обеспечения связи между компьютерами.

■ Фирма AutoDesk, известная всем как разработчик пакета AutoCAD, вышла на рынок с новым программным продуктом - пакетом Animator, предназначенным для создания движущихся изображений (мультипликации). Фирма AutoDesk разработала для этого пакета специальный графический интерфейс для пользователя. Пойти по пути использования испытанного пакета, подобного Windows или GEM, в данном случае было нельзя из-за жестких ограничений по быстродействию. Пакет Animator ориентирован на видеоадаптер VGA.

обслуживающих Административную Систему (АС). Логика рассуждений и действий основывается здесь на последовательном отсечении всего, что затрудняет или замедляет движение к "лучшей жизни".

ством и интеллектуальной элитой. Одновременно с демократизацией общества набирает силу популизм.

Тоталитарная сущность популизма проявляется в преднамеренном подыгрывании массовому сознанию, упрощении действительности и связанного с ее оценкой выбора. Единая и неделимая власть диктатора заменяется властью стереотипа. В условиях нашей страны ситуация усугубляется тем, что стереотипы массового сознания порождены тоталитарной АС и четко вписаны в систему ее идеологического обеспечения, т.е. ориентированы на сохранение жизненно важных для АС институтов и принципов. Для идеологического обеспечения АС выработана целостная система преднамеренно неточных, или, выражаясь современным языком, "лукавых" терминов, лозунгов и научных понятий, прочно вросших в массовое сознание и не менее опасных, чем "лукавые" цифры.

Примерами лукавых терминов могут служить "интернациональная помощь" (интервенция), "социальная справедливость" (уравниловка), а также неперебиваемые на нормальный русский язык "моральная устойчивость" и "идеологическая выдержанность". Рынку в лукавой политэкономии противопоставляется план, хотя реальным антиподом рынка является натуральное распределение, а план и рынок прекрасно уживаются в экономике всех развитых стран. Аналогичным образом собственнику противопоставляется труженик, а не бюрократ, о чем уже говорилось выше. Эксплуатация неразрывно связывается с частной собственностью, инфляция — с отсутствием государственного контроля за ценами, хотя действительность буквально вопиет о невиданной со времен крепостного права эксплуатации при полном отсутствии частной собственности и о гигантской "скрытой" инфляции, буквально разъедающей хозяйственные связи и потому несоизмеримо более опасной, чем обычная "открытая" инфляция. Демократизация экономической жизни, безусловно, должна сопровождаться открытой инфляцией. Это не только следует из теоретических рассуждений в рамках кейнсианской и неоклассической теорий, но и подтверждается практикой развития всех социалистических стран, вступивших на путь демократизации. Однако АС и порожденное ею массовое сознание не приемлют открытой инфляции, на поводу у них идут официальная и популистская экономические науки. Происходит постоянное отравление тоталитарной системы собственными идеологическими выхлопами. Информатизация может только обострить этот процесс.

С неадекватным восприятием действительности тесно связана еще одна характерная черта тоталитарного сознания — ученически серьезное отношение к философии. Проявляется оно двояко. Во-первых, философия начинает вытеснять науку из сфер, уже доступных точному знанию, причем сама при этом постепенно превращается в религию. Во-вторых, всякая абстрактная идея, в том числе и фило-

софская, огрубляется быденным сознанием и обращается в команду фельдфебеля. Каждый раз, когда философское учение возводится в ранг государственной политики и внедряется в повседневную жизнь, результат оказывается плачевным, причем совершенно независимо от содержания учения. В этом смысле учение Ницше не хуже учения Маркса, хотя и не лучше. Впрочем, хотя для немецких нацистов и наших сталинистов цитата неизменно значила больше, чем научно установленный факт, но и цитата выбиралась сообразно требованиям текущего момента и добросовестно обслуживала творимый произвол. В "Капитале" К. Маркса философия несколько потеснила экономическую науку (т.е. науку о рациональном ведении хозяйства), сильно усложнив изложение, но не придя пока к явному противоречию с наблюдаемой действительностью. В расхождении В.И. Ленина с легальными марксистами ясно обозначились признаки превращения марксистской философии в религию. Традиция была сохранена в трудах его последователей и эпигонов, осыяна и насильно вбита в головы нескольких поколений советских экономистов. В результате возникла лукавая политэкономия, абсолютно бесполезная при ведении хозяйства, но чрезвычайно полезная для АС. В большей или меньшей степени та же судьба постигла все общественные и некоторые естественные науки, что было отмечено наиболее проницательными умами уже в конце 20-х годов: "В ряде государств (Япония, СССР) ограничение свободной научной мысли явно или скрыто принимает характер государственной религии" (Вернадский В.И. Философские мысли натуралиста. М.: Наука, 1988).

Тоталитарная и демократическая модели информационного общества

Эффективность творческого труда в современном мире обусловлена широким набором обязательных условий, в который входят интеллектуальная свобода, доступность и достоверность самой разнообразной информации, обеспеченность современными техническими средствами и информационными технологиями. Отсутствие хотя бы одной из перечисленных составляющих обрекает страну на хроническое технологическое отставание. По существу те же условия необходимы для реального обеспечения свободы слова.

По определению А.П. Ершова, информатизация общества — это "комплекс мер, направленных на обеспечение полного использования достоверного, исчерпывающего и своевременного знания во всех общественно значимых видах человеческой деятельности...". Коль скоро определение явно содержит требование достоверности информации, оно в принципе исключает возможность существования тоталитарного информационного общества. Однако модель такого общества довольно основательно иссле-

довалась американским советологом С. Гудманом, а в художественной форме — Дж. Оруэллом.

Видимое отличие тоталитарной модели информационного общества от демократической состоит в явном преобладании средств централизованного вещания и контроля за распространением информации над средствами межличностного общения. Демократическая, или "западная", модель характеризуется возникающими тенденциями, тоталитарная — централизованно сформулированными целями. Соответственно различаются движущие силы, системные условия, сферы и способы применения вычислительной техники.

В первом приближении информационное общество западного типа — это США сегодня, СССР не является информационным обществом даже в первом приближении, но есть тенденция его развития в сторону тоталитарной модели. В частности, для СССР характерно преобладание средств централизованного вещания в традиционных информационных секторах. Уже к 1965 г. СССР обладал мощными станциями радио- и телевизионного вещания в крупных городах. В настоящее время функционирует около дюжины спутников, обеспечивающих трансляцию радио- и телевизионных программ по всей территории страны. В то же время в крайне жалком состоянии находятся почта и телефонная сеть. К середине 80-х годов в СССР насчитывалось около 28 млн телефонных аппаратов по сравнению со 170 млн в США. Еще разительнее несоответствие между этими странами в числе междугородных и международных переговоров: примерно 2 млрд междугородных переговоров ежегодно в СССР против 45 млрд в США и примерно 2 млн международных вызовов ежегодно в СССР против 400 млн в США.

Отличаясь жестким централизованным планированием и политическим администрированием, СССР всегда рассматривался как кандидат номер один на широкое внедрение и использование государственной иерархической компьютерной сети. Разговоры об этом ведутся с середины 50-х годов по сей день. Провозглашение экономических реформ в середине 80-х годов не только не похоронило идею такой сети, но придало ей особую привлекательность. Аргументами в ее защиту служат принципиальная возможность избежать дублирования информации, экономическая эффективность и т.д. Негативный опыт прошлых попыток централизованного управления явно недооценивается, совершенно неадекватно оцениваются неудачи с общегосударственной автоматизированной системой и автоматизированными системами управления (АСУ), т.е. игнорируются причины, породившие патологическую лживость информации, циркулирующей в системах подобного рода. Причины эти в основном сводятся к незаинтересованности общества (в его нынешнем виде) в достоверной информации. Следовательно, они неизбежно приведут к аналогичному лукавству и новых источников информации, построенных на основе совре-

ФАКТЫ

■ Фирма Borland начала продажу пакета Paradox Engine - библиотеки, в состав которой входят 70 функций на языке Си. Таким образом, "раскрыта тайна" архитектуры СУБД Paradox, которую до сих пор держали в секрете. Новая библиотека позволит создавать прикладные программы, имеющие доступ к данным СУБД Paradox, и обрабатывать эти данные в одно- и многопользовательских средах.

менных и сколь угодно совершенных информационных технологий.

Подтверждением высказанной мысли может служить опыт эксплуатации АСУ. Предназначенные для повышения качества управленческих решений и вобравшие в себя многие достижения отечественной кибернетики, АСУ не оправдали возлагавшихся на них надежд. Это вовсе не означает, что АСУ совсем не нашли применения. Напротив, они активнейшим образом использовались, но часто совсем не для решения тех задач, на которые ориентировались их создатели. В частности, широкое распространение получило использование АСУ для введения в заблуждение деловых партнеров и, прежде всего, начальства. Добиваясь снижения плана и увеличения капитальных вложений, министр мог предъявить в Госплане или Совете Министров "оптимальный" перспективный план с нужными ему цифрами, якобы рассчитанный на ЭВМ, а в действительности заботливо введенный в ее память догадливыми подчиненными. Большое впечатление на руководителей всех уровней, не считая руководителей групп программирования, производила печать на алфавитно-цифровом печатающем устройстве большими буквами заголовков и эмблем. Поэтому психологическая обработка в условиях работающего вычислительного центра обычно проходила особенно успешно. Что касается реальных данных об управляемом объекте, то введение их в АСУ всегда было крайне рискованно, так как после этого данными могли воспользоваться ревизионные службы или вышестоящие инстанции.

Неудача с АСУ имеет еще один не вполне оцененный, на мой взгляд, аспект. Автоматизация конторского труда в США стала экономически целесообразной только после появления относительно дешевых ПК (сто-

имостью, не превышающей двух месячных зарплат служащего). До этого компьютеризация контор проводилась по мотивам, существование которых у нас принято игнорировать: ради удобства служащих, из престижных соображений и т.п. Наши АСУ не внесли удобств в деятельность тех, кто выполнял реальную работу. Говорить об экономической целесообразности компьютеризации конторского труда в СССР при существующих ценах на компьютеры просто дико, тем более дико было говорить об этом в эпоху АСУ. Следовательно, преследовались какие-то совершенно иные цели. Очень может быть, что рыцари АС искренне надеялись получить таким путем доступ к реальной экономической информации, но "народ разгадал их маневр и успешно отбил атаку". Аналогичная судьба неизбежно постигнет и следующую волну автоматизации контор в СССР, поскольку все основные компоненты предыдущих неудач пока сохраняются.

Элементы демократии в информационной сфере СССР представлены в основном так называемой "параллельной культурой", проявляющейся в традиционном самиздате и его новых разновидностях, порожденных развитием техники. Феномен такого рода создан В. Высоцким и появлением магнитофонов. Совершенно новые возможности открываются в связи с развитием видеодокументалистики, ставшей мощнейшим оружием против лживости печатного слова и воплотившей известный принцип "Лучше один раз увидеть...", а также появлением относительно дешевых текстовых процессоров. Войну с параллельной культурой тоталитаризм ведет не только с помощью прямых запретов, но и с помощью разного рода лимитов и дефицитов, в том числе лимитов на бумагу, дефицита магнитных носителей и т.д. Тем не менее параллельная культура стремительно развивается, причем процесс этот будет ускоряться по мере насыщения страны современной информационной техникой. Прав Артур Кларк, заявивший в недавнем интервью (НТР. - 1990, N 1-2), что устранение ряда диктаторских режимов в последнее время - прямой результат развития мировых коммуникационных связей. Еще раньше на неизбежное падение тоталитарного режима при насыщении страны информационной техникой обратил внимание Л. Валенса, прогноз которого блестяще подтвердился.

"Новые возможности для воображения и неординарности"

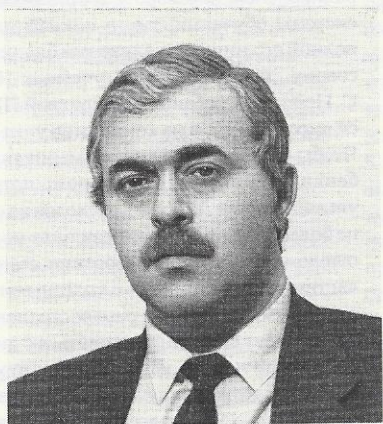
Сеймур Пейперт, признанный "живым классиком" среди специалистов по компьютеризации образования, связывает с массовым распространением ПК радикальные преобразования во всей человеческой культуре. Речь даже идет о делении ее на компьютерную и докомпьютерную, хотя "...компьютеры - это не замкнутая в себе культура, они служат

обоснованию весьма различных культурных и философских взглядов". В контексте обсуждаемого вопроса наиболее интересными представляются две высказанные С. Пейпертом мысли: о преодолении пропасти между гуманитарной наукой и естественнонаучным знанием (преодолении "матофобии"), а также о радикальном перевороте в организации всей системы обучения, т.е. о полной или почти полной его индивидуализации благодаря массовому использованию учебных ПК. По С. Пейперту, ребенок обучает свой ПК и учится вместе с ним, а не компьютер учит ребенка. Чтобы обеспечить каждого американского ребенка учебным компьютером, потребуется увеличить общую сумму расходов на обучение не более, чем на 5 %, но результат может быть ошеломляющим. "Образование станет более частным делом, и люди с полезными идеями, отличными от тех, что распространены сегодня, не будут больше сталкиваться с дилеммой: либо продать свои идеи консерваторам, либо положить их на полку. Они могут предложить их на открытом для всех рынке сбыта непосредственно потребителю. Появляются новые возможности для воображения и неординарности" (Пейперт С. Переворот в сознании и плодотворные идеи. М.: Мир, 1989).

Аналогичные возможности открываются и в организации науки. Речь идет, прежде всего, о компьютерных журналах, различных системах доступа к базам данных и знаний, следовательно, о достижении научными работниками относительной независимости от научных администраторов. Так техническая революция открывает ученым путь к индивидуальной свободе, но отнюдь не к демократии. Именно здесь заканчивается общность исторических судеб информационного общества и демократии, о которой говорилось в начале статьи, поскольку "свобода аристократична". Приходится признать, что путь к индивидуальной свободе наших ученых через массовую компьютеризацию не очень реален из-за своей дороговизны, как не реален массовый аристократизм в нищете. Более плодотворная роль компьютеров в преодолении тоталитаризма в науке через преодоление "матофобии". Можно надеяться на постепенное избавление общественных наук от догматизма и расширение сферы влияния точного знания, которое непосредственно связано с проникновением в гуманитарные науки математики и информатики. Разумеется, сама по себе компьютеризация не может компенсировать страшные деформации в научной этике, взаимоотношениях научных школ и их формировании, порожденные десятилетиями господства тоталитарной системы, но она вполне способна существенно ускорить восстановительные процессы.

Примерно то же - с демократизацией образования, провозгласить которую соблазнительно просто. Сложнее найти реальные пути демократизации, но начинать надо, безусловно, с самоизбавления от химер тоталитарного сознания.

МНЕНИЯ



С.П. ЛЬВОВ

ИДолы массовой КОМПЬЮТЕРИЗАЦИИ

Дело совершается орудиями и вспоможениями,
которые нужны разуму не меньше, чем руке.
Ф. Бэкон. "Новый органон, или истинные указания для
истолкования природы", 1620 г.

Глобальное распространение персональных компьютеров позволило перейти от компьютеризации отдельных функций профессиональной деятельности людей к компьютеризации всей деятельности в целом, т.е. к массовой компьютеризации. Однако на этом пути возникли достаточно серьезные проблемы.

Анализ сложившейся ситуации показывает, что в настоящее время отсутствуют общепринятые ответы на три главных вопроса:

1. Каковы основные цели массовой компьютеризации?
2. Каким образом ее эффективнее проводить?
3. Как оценивать изменения в деятельности людей и организаций, вызванные компьютеризацией?

Действительно, когда речь идет об установке персонального компьютера на конкретном рабочем месте, цель компьютеризации формулируется просто: автоматизировать одну или несколько функций, выполняемых человеком. Когда же объектом компьютеризации является согласованная профессиональная деятельность многих людей, функциональный подход неприемлем, так как невозможно даже просто перечислить все функциональные связи, возникающие между людьми в процессе этой деятельности.

Значительные трудности появляются и на этапе практической реализации идеи массовой компьютеризации, при создании большого числа автоматизированных рабочих мест (АРМ). Как правило, проектированием и внедрением АРМ занимается определенный коллектив разработчиков. Если число разработчиков невелико по сравнению с числом типов проектируемых АРМ, то трудно обеспечить создание АРМ всех требуемых типов. Если же число разработчиков сравнимо с числом типов АРМ, то возникают трудности, связанные с организацией работы самих разработчиков.

Массовая компьютеризация оказывает существенное влияние на деятельность людей и функционирование предприятий: меняются индивидуальное поведение сотрудников и их психология, распределение функций, прав и обязанностей среди персонала, организационная

структура предприятия, а в некоторых случаях даже глобальные цели всей его деятельности. Эти изменения, носящие в основном качественный характер, с большим трудом поддаются количественной оценке, к тому же они происходят на фоне других нововведений, выделить влияние которых от результатов собственно компьютеризации чрезвычайно сложно, поэтому получить достоверную количественную оценку результатов массовой компьютеризации практически невозможно.

По-видимому, аналогичные трудности возникают на начальной фазе любого этапа научно-технического прогресса. Эти трудности обусловлены тем, что при массовом распространении нового вида технических устройств начинается интенсивное взаимодействие и взаимопроникновение двух принципиально различных "миров": "мира человеческой деятельности" и "мира новой техники". В данном случае положение усугубляется тем, что "мир компьютеров" проникает в "мир мышления людей", ответная же реакция последнего и порождает многочисленные трудности.

Чтобы понять, в каком месте на стыке этих двух "миров" возникают основные проблемы, удобно воспользоваться введенным еще в XVII веке английским философом Ф. Бэконом образным представлением трудностей, с которыми сталкивается человеческое познание, в виде идолов (от греческого *idolum* - призрак, тень умершего, видение). Ф. Бэкон использовал этот термин в изначальном значении, имея в виду призрак, уводящий человеческое познание на ложный путь. Им было определено четыре вида идолов, мешающих сознанию правильно ото-бражать мир:

- идолы рода (племени), порожденные самой природой человека;
- идолы пещеры, определяемые ограниченностью индивидуального опыта;
- идолы площади (рынка), обусловленные природой языкового общения людей;
- идолы теорий (театра), связанные с социальной структурой общества.

По аналогии с этим определим четыре вида идолов, мешающих сознанию правильно воспринимать "компьютерный мир": идола компьютерного племени, идола производителя, идола коммуникаций и идола программного обеспечения.

Следует подчеркнуть, что идола всех перечисленных видов, в том числе и идола компьютерного племени, можно рассматривать как с точки зрения разработчиков системных и прикладных задач, так и с точки зрения их пользователей, т.е. необходимо различать два рода идолов: идола пользователя и идола разработчика.

Идола пользователя

Рассмотрим идола четырех перечисленных выше видов с точки зрения пользователя.

Идола компьютерного племени порождены принципами работы современных компьютеров, природой их "мышления". Абстрагируясь от физики процессов, можно считать, что функционирование компьютеров полностью описывается чисто математически, поэтому компьютерное "мышление" формально и рационально, оно способно порождать только тавтологии и сингулярные высказывания. Человеческое же мышление ассоциативно и иррационально, оно может порождать обобщающие, универсальные и синтетические высказывания. Хотя математика представляет собой продукт человеческого мышления, между творцом и его творением лежит непреодолимая пропасть, в которой и находят убежище идола компьютерного племени.

Идола производителя порождены огромным разнообразием как самих производителей, так и выпускаемых ими вычислительных средств и периферийного оборудования. Дело в том, что в любой организации, применяющей компьютеры, одновременно используется не менее нескольких десятков типов оборудования (процессоров, мониторов, принтеров, внешних запоминающих устройств, модемов и т.д.). Кроме того, несмотря на старания фирм-изготовителей обеспечить



идентичность однотипного оборудования, практически каждый экземпляр обладает собственным "характером", что обусловлено незначительными дефектами, допущенными при его изготовлении или установке.

В результате "миру индивидуальных пользователей" противостоит "мир индивидуального оборудования". И как человеку сложно поддерживать дружеские отношения одновременно с большим числом людей, так и пользователю трудно свободно овладеть всей вычислительной техникой, находящейся даже в пределах одного подразделения. Идолы производителя делают пользователя эгоцентристом, который боится выйти за пределы своего локального "компьютерного мира".

Идолы коммуникации порождены неограниченными возможностями современных компьютеров в отображении данных. В наше время для прекращения строительства Вавилонской башни Богу было бы достаточно дать каждому строителю по персональному компьютеру. Вскоре он обнаружил бы, что нет двух человек, у которых одни и те же данные были бы отображены в компьютерах идентичным образом, а сами компьютеры вместо решения конкретной задачи заняты только преобразованием данных.

Большинство имеющихся на рынке программных продуктов предоставляет пользователям столь широкий набор функций, что лишь очень немногие специалисты могут хорошо освоить более двух-трех пакетов или языков программирования. В результате возникают "удельные княжества" пользователей, границы которых надежно охраняют *идолы программного обеспечения*.

Идолы разработчика

Рассмотрим идолы четырех перечисленных выше видов с точки зрения разработчика системных и прикладных задач. В отличие от идов пользователей, эти идолы порождены особенностями мышления человека, который уверен в том, что уж в созданном-то им самим "компьютерном мире" профессиональный разработчик всегда сможет легко освоиться. Люди с такой точкой зрения убеждены, что глубоко разбираются в фундаментальных принципах работы современных компьютеров, могут легко научиться работать на любом оборудовании, с любыми типами данных и, наконец, свободно овладеть наиболее популярными языками и программными продуктами.

В определенном смысле идолы разработчика являются антиподами идов пользователей, или "антиидолами". Воздействуя на мышление не только самих разработчиков, но и определенной части пользователей, антиидолы обуславливают появление претенциозных, дорогостоящих и крайне неэффективных проектов.

Таким образом, основным препятствием на пути эффективного использования компьютеров является само человеческое мышление, отягощенное сонмом ложных представлений о себе и своем творении. Чтобы рассеять эти представления, недостаточно воздействовать на мышление стандартными методами пропаганды и обучения. Этого можно достичь только на определенном этапе специальным образом организованной деятельности, которую и будем называть массовой компьютеризацией.

К стратегии массовой компьютеризации

Основной способ борьбы с идолами, предложенный Ф. Бэконом, состоит в последовательном применении специального метода познания, краеугольным камнем которого является проверка следствий научных теорий опытным путем.

Как известно, в инженерной деятельности по созданию новых объектов реального мира (к которой относится и массовая компьютеризация) эмпирические критерии не используются. Вместо этого в ряде случаев применяется критерий реализуемости проектных решений. Однако в информатике и этот критерий неприменим, так как разработчики вычислительной техники и программного обеспечения располагают столь гибкими средствами, что практически любая идея потенциально реализуема. Поэтому отбраковка идей и методов осуществляется не по критерию реализуемости, а в процессе естественного отбора на рынке. При этом, как правило, побеждают те программные продукты, на разработку и рекламу которых затрачены наибольшие средства.

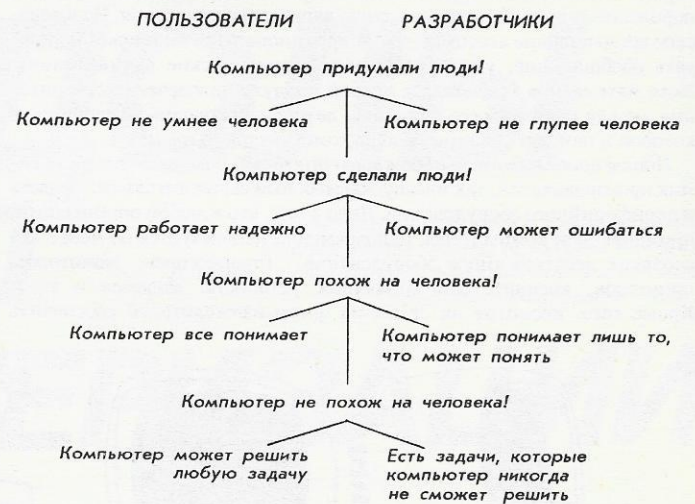
Применительно к методам массовой компьютеризации ситуация еще сложнее - в данном случае отсутствует и возможность естественного

отбора, поскольку нет самого рынка таких методов. Однако потребность в отборе методов массовой компьютеризации, причем на как можно более ранних ее стадиях, осознается все большим числом разработчиков и пользователей.

Хорошо известны выражения, образно определяющие сущность человеческого познания: знания - особо надежный вид веры, научные знания - особо надежный вид знаний. По аналогии можно сказать, что компьютеризация - наиболее эффективный способ использования компьютеров, массовая компьютеризация - наиболее эффективный способ компьютеризации.

Существует много способов организации коллективной деятельности людей. Однако, как показывает опыт, такая деятельность эффективна лишь в том случае, если ее участники руководствуются общими для всех принципами и правилами. Совокупность этих принципов и правил называется "философией деятельности". Таким образом функционируют, например, ведущие японские и американские корпорации.

По аналогии с этими примерами можно сформулировать "философию массовой компьютеризации" в виде следующей структурированной совокупности правил:



В контексте изложенного выше эти правила следует рассматривать как "орудия борьбы" с идолами всех видов. Однако для того, чтобы такая борьба была эффективна, помимо этих "орудий борьбы" следует также применять определенную стратегию массовой компьютеризации.

Суть предлагаемой стратегии состоит в следующем. До сих пор считалось, что пользователь и разработчик - это два различных лица. Однако в условиях массовой компьютеризации, когда компьютер становится доступным и широко распространенным средством, деление людей на пользователей и разработчиков приобретает все более условный характер.

Во время работы на современном компьютере происходит постоянное чередование процедур использования уже существующих объектов (программ, данных, средств коммуникации и т.п.) и создания новых объектов соответствующей природы. Образно говоря, граница раздела между пользователем и разработчиком проходит теперь в сознании каждого человека. С точки зрения пользователя, при массовой компьютеризации справедливы следующие высказывания:

Нет пользователя, кроме разработчика!

Нет разработчика, кроме пользователя!

Осознание этого феномена и является, по-моему, необходимым условием разработки стратегии массовой компьютеризации.

Может показаться, что в нашей стране, при всеобщем компьютерном голоде, разработка стратегии массовой компьютеризации по крайней мере преждевременна. Единственное правило, которому в настоящее время следует большинство потенциальных пользователей, формулируется чрезвычайно просто: "Дай компьютер". Однако с появлением персонального компьютера проблемы, о которых говорилось выше, встают перед пользователем в полный рост, и тут без стратегии массовой компьютеризации не обойтись.



ЛИКБЕЗ

А.В. СМОРОДИНСКИЙ

О.Ю. ПОСУДИНА

ТЕКСТОВЫЕ ПРОЦЕССОРЫ для IBM- совместимых ПК

Авторы, не претендуя на строгий научный анализ текстовых процессоров, делятся опытом практической работы с теми из них, которые наиболее популярны в СССР. Статья рассчитана на начинающих пользователей, желающих, составив представление о возможностях и характеристиках текстовых процессоров, остановить свой выбор на одном из них.

Перед выбором

Текстовые процессоры - это пакеты программ, предназначенные для ввода, редактирования и подготовки к выводу на принтер разнообразных текстов: книг, статей, отчетов, писем, технических описаний и т.д. (Так называемые редакторы текста мы рассматриваем как частный случай текстовых процессоров с ограниченными возможностями по обработке текста.)

По оценкам некоторых экспертов, пакеты программ для подготовки текстов составляют примерно 75 % от общего объема программного обеспечения ПК. Текстовый процессор является неотъемлемой частью таких про-

граммных продуктов, как интегрированные системы и оболочки экспертных систем. Практически не существует пользователей ПК, которые бы не применяли тот или иной текстовый процессор. Хотя теоретически каждый текстовый процессор можно использовать для подготовки любого текста, максимальная эффективность достигается при соответствии специфики этого процессора структуре и особенностям текста.

Текстовым процессорам посвящена довольно обширная литература. Сравнительные характеристики наиболее популярных за рубежом текстовых процессоров приведены, например, в статье *Современные текстовые процессоры и 38 программ для эффективного редактирования*, опубликованной во втором выпуске обзора "КомпьютерПресс" за 1989 г.

Необходимо отметить, что наряду с текстовыми процессорами в последнее время все большее распространение получают настольные издательские системы, позволяющие осуществлять профессиональную подготовку текстов к изданию. Эти системы составляют самостоятельный класс программного обес-

печения и в данной статье не рассматриваются.

Многообразие текстовых процессоров столь велико, что провести их исчерпывающий анализ практически невозможно.

Существующие в настоящее время текстовые процессоры значительно отличаются друг от друга характеристиками, возможностями по вводу и редактированию текста, его форматированию и выводу на печать, а также по степени сложности освоения пользователем.

Текстовые процессоры отличаются друг от друга и по требованиям, предъявляемым к ним пользователем в зависимости от его квалификации и вида создаваемых им объектов (исходные тексты программ, документы различной степени сложности, научные статьи, книги и т.д.). Так, для подготовки текстов программ целесообразно использовать редактор, входящий в состав программной среды (например такой, как Quick Basic и Turbo Pascal), поскольку при этом не требуется ни широких возможностей по редактированию текста, ни разнообразных наборов шрифтов. В то же время для подготовки научных статей или текста книги пользователю ну-

жен текстовый процессор с широким спектром возможностей, вплоть до конструирования шрифтов. Не последнюю роль при выборе текстового процессора играют объем занимаемой памяти (особенно при его использовании в составе систем автоматизированного проектирования) и цена.

По степени сложности все тексты, подготовливаемые с помощью текстовых процессоров, можно разделить на четыре типа:

- исходные тексты программ;
- деловая переписка и делопроизводство;
- научные статьи, содержащие формулы, и тексты небольшого объема;
- книги, содержащие формулы, или тексты большого объема и сложной структуры.

Выбор конкретного текстового процессора является весьма ответственным моментом, особенно при подготовке текстов больших объемов; при этом кроме объективных причин важную роль играют вкус и привычки пользователя.

Наиболее актуальным при описании процесса редактирования является понятие *интерфейса пользователя*, к которому в первую очередь относятся язык общения с текстовым процессором, а также устройства ввода-вывода (для ПК это клавиатура, манипулятор типа "мышь", экран дисплея и принтер).

Язык общения пользователя с текстовым процессором организован на основе диалоговых процедур, для реализации которых можно применять текстовые команды (неудобством командного языка является необходимость запоминания пользователем большого числа команд), функциональные клавиши (для реализации всех возможностей текстового процессора приходится применять множество комбинаций из двух, а иногда и трех клавиш, что не всегда удобно) и систему иерархических меню, которая также имеет свои плюсы и минусы. В стадии разработки и опытной эксплуатации находятся устройства речевого ввода информации, позволяющие преобразовывать произносимые слова в текст. В сложных текстовых процессорах используются различные комбинации перечисленных выше средств организации диалоговых процедур.

С точки зрения удобства для пользователя, одним из важнейших свойств текстовых процессоров является полное соответствие твердой копии образу документа на экране дисплея (к сожалению, далеко не все текстовые процессоры обеспечивают такое соответствие).

В силу целого ряда объективных и субъективных причин тенденции использования текстовых процессоров в СССР получили несколько иное развитие, чем на Западе.

Первыми отечественными текстовыми процессорами были: АБВ (ВЦ АН СССР), ЛЕКСИКОН (ВЦ АН СССР) и RT Print (ЦЭМИ АН СССР). Следует отметить, что редактор RT Print, уступающий двум первым по целому ряду характеристик, широкого распространения

не получил. Что касается текстовых процессоров АБВ и ЛЕКСИКОН, сопоставимых по характеристикам, то их судьбы сложились по-разному: АБВ мало известен рядовым пользователям, а ЛЕКСИКОН применяется в СССР достаточно широко.

Поскольку некоторые характеристики являются общими для разных текстовых процессоров, то подробно мы будем рассматривать их только один раз (применительно к тому текстовому процессору, для которого та или иная характеристика упоминается впервые). Отсюда существенная разница в объеме описания различных пакетов, причем более подробно описаны те из них, которые чаще используются.

✱ ФАКТЫ

■ Пакет Software Bridge, разработанный фирмой Systems Compatibility, позволяет преобразовывать файлы из одного формата в другой для таких популярных текстовых процессоров, как WordPerfect, Microsoft Word, WordStar, Sprint, MultiMate, Volkswriter. Тестирование показало, что пакет Software Bridge обрабатывает файлы безошибочно. Цена пакета 149 дол.

■ Фирма Microsoft разработала версию текстового процессора Word, предназначенную для работы в среде Windows. Новый текстовый процессор - Word for Windows - выполняет практически все функции настольной издательской системы, предоставляя пользователю, кроме того, расширенный набор графических средств. Цена пакета Word for Windows 495 дол.

■ Фирма WordPerfect выпустила версию 5.1 своего текстового процессора. Основными особенностями этой версии являются: специальные редакторы для создания сложных таблиц и ввода формул, возможность работы с "мышью", удобные выпадающие меню. Цена пакета 495 дол.

EDIT

Этот редактор, разработанный фирмой IBM, предназначен для создания, редактирования и подготовки к выводу на печать текстовых файлов (такими файлами могут быть исходные тексты программ, текстовая документация, данные для программ). Редактор EDIT позволяет создавать и модифицировать текстовые файлы, представленные в коде ASCII, при длине строки не более 140 символов. Не следует использовать этот редактор для редактирования файлов, представленных в ином коде (например, файлы типа COM или EXE, а также файлы, содержащие данные в двоичном коде), так как EDIT уничтожает любые "неправильные" с его точки зрения символы.

Процесс редактирования можно осуществлять с помощью функциональных клавиш или текстовых команд.

Функциональные клавиши позволяют выполнять многие операции редактирования, причем с помощью 10 функциональных клавиш и различных ключей их установки можно реализовать до 40 операций. Последней строкой на экране является строка подсказки, показывающая, какую функцию выполняет каждая из 10 задействованных в данный момент клавиш. Однако и такой сервис еще недостаточно удобен. Так как одну и ту же клавишу можно использовать для реализации до четырех различных функций, а в строке подсказки появляется название только той функции, которая соответствует клавише при данной установке ключа, то нахождение искомой функции требует некоторого дополнительного времени.

Осуществлять редактирование можно также, непосредственно вводя текстовые команды (в виде мнемонического кода на английском языке) либо в предпоследней строке экрана, либо в специально отведенной для этого левой части экрана.

Редактор EDIT позволяет выполнять следующие операции:

- вставлять и уничтожать символы и строки;
- осуществлять поиск последовательности символов и ее замену;
- просматривать текст, передвигаясь вверх, вниз, вправо, влево, в начало или в конец файла;
- производить перемещение, копирование, уничтожение, перекодировку и другие операции над отдельными строками или группами строк;
- вставлять текст из любого другого файла (место вставки помечается курсором);
- копировать часть текста редактируемого файла в любой другой файл;
- подготавливать к выводу с экрана на принтер текст или любую его часть;
- устанавливать границы строки и метки табуляции, центрировать строки;
- разбивать одну строку на несколько строк;
- объединять несколько следующих друг за

другом строк в одну строку, длина которой не превышает 140 байт;

- выравнивать текст по правой границе, вставляя между словами дополнительные пробелы;
- повторно выполнять четыре последние команды (с их модификацией или без нее);
- просматривать список файлов в каталоге.

Большинство этих операций выполняют все текстовые процессоры, поэтому ниже, при описании других текстовых процессоров, мы будем рассматривать только их отличительные особенности.

Важной характеристикой редактора EDIT является наличие в нем средства, позволяющего создавать макроопределения (макросы). Пользователь может присвоить любой из имеющихся функциональных или задействованных альтернативных клавиш макроопределения, что значительно упрощает и ускоряет выполнение повторяющихся операций.

Следует отметить возможности редактора EDIT по ликвидации последствий ошибочных действий, а именно:

- создание резервной копии файла в момент начала сеанса;
- восстановление данных при переполнении памяти на диске и обнаружении большинства ошибок ввода-вывода;
- восстановление пяти последних уничтоженных строк (благодаря наличию стековой памяти на пять строк по 28 байт каждая);
- восстановление 140 последних уничтоженных символов (благодаря наличию стековой памяти емкостью 140 байт).

Емкость буфера, в который считывается редактируемый текст, составляет 65 Кбайт, что, как правило, не накладывает ограничений на длину обрабатываемых файлов; если же емкость буфера оказывается недостаточной, файл разбивается на сегменты, каждый из которых редактируется отдельно. Имеется возможность считать в буфер заданное в явном виде число строк входного файла, а также подготовить к выводу на принтер весь редактируемый текст или его часть. Режим печати сочетается с режимом отображения текста на экране дисплея.

KEDIT

Этот редактор, разработанный фирмой Mansfield Software Group, предназначен для ввода, редактирования и подготовки к выводу на печать текстов программ, а также сравнительно простых текстовых файлов, представленных в коде ASCII. Размер редактируемых файлов ограничен лишь объемом свободной оперативной памяти.

Возможности этого редактора по обработке текста в основном аналогичны возможностям редактора EDIT, но существуют и небольшие отличия. В нем имеется несколько команд-вопросов, которые выводят на экран некото-

рую информацию, например текущее время, позицию курсора, последнюю ошибку.

Отличительными особенностями редактора KEDIT являются возможность работы с командами DOS без выхода из редактора, а также возможность выполнения таких сервисных операций, как смена каталога, копирование, переименование или удаление файлов.

К недостаткам этого редактора следует отнести то, что команды, управляющие его работой, не выбираются из меню, а либо вводятся с помощью специальных клавиш или их сочетаний (аккордов), либо осуществляется непосредственный ввод символов в командной строке. Это существенно усложняет освоение редактора, уменьшает наглядность выполняемых действий и снижает производительность труда пользователя. Поэтому редактор KEDIT не рекомендуется применять начинающим пользователям ПК.

ДОКУМЕНТ-16

Этот текстовый процессор, разработанный советско-болгарским институтом СБНИПИ "ИНТЕРПРОГРАММА", применяется для ввода, редактирования и подготовки к выводу на принтер текстовых документов.

Ниже перечислены компоненты текстового процессора ДОКУМЕНТ-16, называемого также Д-16, и указаны их основные функции.

1. Редактор, предназначенный для создания и редактирования текстов документов и текстовых заготовок ("шапок", стандартных текстов и т.д.) с последующим включением их в документы. Для защиты документа от несанкционированного доступа пользователь может присвоить ему пароль. При внесении изменений в документ редактор сохраняет и его предыдущую версию, что уменьшает вероятность полной потери всех версий. Файл документа кроме текста содержит управляющую информацию. Недостатком этого редактора является ограниченный набор возможностей: он позволяет при создании документа определить его формат, т.е. число строк на странице, длину строки, размеры полей слева, справа, сверху и снизу страницы.

2. Программа обслуживания адресного массива, позволяющая автоматизировать процесс вставки адресных данных в составленные документы при их печати. Эту программу можно использовать как совместно с остальными компонентами текстового процессора, так и самостоятельно.

3. Программа архивирования текстовых документов, предназначенная для сохранения параметров документа в каталогах и поиска документа по их заданным значениям.

4. Программа подготовки документа к выводу на принтер, осуществляющая печать документа в формате, определенном пользователем, и включение в текст документа текстовых заготовок. Эта программа позволяет автоматически составлять содержание (оглавление) документа, формировать указатели

терминов, а также выделять строки, группы слов или отдельные слова полужирным шрифтом или подчеркиванием.

Управление текстовым процессором Д-16 осуществляется с помощью функциональных клавиш и их сочетаний. В нижней части экрана размещается вспомогательная информация о возможности получения помощи. Это упрощает освоение текстового процессора и пользование им, в какой-то мере компенсируя ограниченность его возможностей.

Personal Editor II

Этот многооконный редактор, называемый также PE2, разработан фирмой IBM для создания, редактирования и подготовки к выводу на печать текстовых файлов, представленных в коде ASCII. Он выполняет все основные операции по обработке текста. В отличие от редактора EDIT, в нем отсутствует функция выравнивания строк по правому краю, но имеются некоторые дополнительные функции. Например, редактор PE2 позволяет осуществлять маркирование текста типа "прямоугольник" (часть строки и часть столбца); "знак" (знаки, слова и предложения); "строка" (отдельная строка или группы строк); заполнять маркированный блок любыми символами; разделять экран на два или четыре окна, в которые можно помещать части одного и того же файла.

По запросу пользователя редактор PE2 может выдавать некоторую справочную информацию, а именно: объем свободной оперативной памяти (в байтах), объем свободной памяти на текущем диске (в байтах), содержимое текущего диска и каталога, сведения о режиме горизонтальной прокрутки, границы строки и красной строки и другие данные.

Преимуществами редактора PE2 являются возможность выполнения команд DOS без выхода из редактора, а также высокая скорость печати при выводе текста на принтер.

Команды, необходимые для работы с редактором PE2, вводятся с помощью функциональных клавиш и их специальных комбинаций. Информация о многих из этих команд (но не обо всех!) выдается при обращении за помощью; пояснительный текст (HELP) занимает 19 экранных страниц, что существенно осложняет работу и снижает производительность труда пользователя. Предусмотрено также введение команд в специальной строке, расположенной в нижней части экрана.

Возможности редактора PE2 делают его особенно удобным для набора текстов программ и их "сборки" из заранее подготовленных фрагментов. Этот редактор можно также применять для создания документов и статей средней сложности (без рисунков и громоздких формул), однако и это не всегда удобно из-за отсутствия выравнивания строк по правому краю и автоматического разбиения текста на страницы.

ФАКТЫ

Благодаря усилиям разработчиков текстовые процессоры по своим возможностям все больше приближаются к настольным издательским системам. В свою очередь, разработчики издательских систем тоже не сидят, сложа руки. Свидетельством тому служит выход в свет версии 5.3 популярной издательской системы PageMaker фирмы Aldus. Новая версия располагает расширенными возможностями по обработке цветной графики. Пакет PageMaker 3.5 позволяет импортировать текстовые файлы в форматах Microsoft Word 4.0, Microsoft Works 2.0, Word Perfect 1.0, XyWrite III Plus и графические файлы в форматах TIFF, EPS, PICT.

ЛЕКСИКОН

Этот многооконный текстовый процессор, разработанный ВЦ АН СССР, предназначен для работы как с текстами программ, так и с различными текстовыми документами средней сложности. Ниже перечислены его основные наиболее интересные особенности:

1. **Многооконный интерфейс.** Пользователь может создавать на экране частично или полностью перекрывающиеся окна (одновременно до девяти окон), изменять их положение, размеры, цвет фона, переходить из одного окна в другое, а также сохранять отдельные созданный набор окон. В окнах можно размещать тексты из различных файлов и создавать новые тексты, что позволяет производить комплексную обработку сложной документации или совокупности программных модулей.

2. **Автоматический перенос слов** в соответствии с правилами орфографии при форматировании текста.

3. **Наличие механизма создания макроклавиш**, что позволяет включать в их определения произвольные символы или команды.

4. **Наличие встроенного экранного калькулятора**, с помощью которого можно производить операции над числами в различных системах счисления, выполнять арифметические действия и вычислять элементарные функции.

Текстовый процессор ЛЕКСИКОН предоставляет такие возможности по оформлению текстов, как форматирование абзацев с отступом вправо или влево в первой строке и выравниванием строк по правому и левому краям, центрирование заголовков, автоматическое разбиение текста на страницы. Имеется возможность задавать номер первой страницы документа и число строк в нем, а также по

требованию пользователя вставлять в текст так называемые жесткие разделители страниц. Входящая в ЛЕКСИКОН программа подготовки к выводу на принтер позволяет печатать документ с использованием нескольких шрифтов.

Для управления работой этого текстового процессора применяются система ниспадающих меню с подсказкой на каждое выбранное действие, расположенной в верхней строке экрана, а также команды, которые вводятся с помощью специальных клавиш и их сочетаний.

Недостаток текстового процессора ЛЕКСИКОН состоит в том, что подсказки организованы не лучшим образом: текст подсказки, занимающий весь экран, выдается не на конкретное действие, а на все случаи сразу. Перенос части текста из одного места в другое с использованием так называемого "кармана" также создает некоторые неудобства. Однако несмотря на отмеченные недостатки, ЛЕКСИКОН является наиболее популярным из отечественных текстовых процессоров.

WORD

Этот многооконный текстовый процессор, разработанный фирмой Microsoft, позволяет создавать, редактировать и подготавливать к выводу на печать текстовые файлы, представленные в коде ASCII. За рубежом этот текстовый процессор пользуется большой популярностью. Для управления его работой применяется система вложенных экранных меню, выбор из которых позволяет осуществлять такие операции, как маркировка, перемещение, копирование и удаление помеченного фрагмента текста; форматирование и выравнивание абзаца текста; создание окон и изменение их размеров. При работе с текстом перед его выводом на принтер пользователь может выделить для печати любой фрагмент текста, вставить в текст имя файла или время и дату его печати. Текстовый процессор WORD поддерживает драйвер манипулятора "мышь", что существенно повышает производительность труда пользователя.

Brief

Этот многооконный редактор, разработанный фирмой UnderWare, предназначен для ввода программ и данных, их модификации и вывода на печать, т.е. для редактирования любых файлов программ или данных, представленных в коде ASCII. Отличительными особенностями редактора Brief являются:

- возможность работы с очень длинными строками (до 512 символов);
- автоматическое сохранение текущего состояния буфера, если в течение определенного интервала времени содержимое буфера не изменялось;
- многооконность, обеспечивающая одновременную работу с многими файлами;

- поиск строки символов и ее замена;
- возможность аннулирования внесенных изменений;
- работа с очень большими текстами (размером до 1 Мбайт и более) без каких-либо дополнительных действий со стороны пользователя, быстрое перемещение курсора по очень большому файлу;
- работа с любым компилятором в режиме TURBO;
- возможность назначения и переопределения клавиш;
- программирование работы редактора в соответствии с возможностями и привычками пользователя.

Наиболее существенное преимущество редактора Brief - его гибкость, т.е. "перенастраиваемость" с помощью макросов в соответствии с потребностями пользователя. Вся работа редактора протекает в определенной среде, которая задается соответствующими макросами. Возможно выполнение команд DOS. Недостатком редактора Brief является отсутствие меню, поэтому все действия выполняются с помощью команд, задаваемых комбинациями функциональных клавиш, и ответов на соответствующие запросы.

Multi-Edit

Этот многофункциональный многооконный редактор, разработанный фирмой Amerigam Cybernetics, предоставляет пользователю такие нетрадиционные возможности, как:

- выполнение команд DOS без выхода из редактора;
- работа с калькулятором простейшего вида;
- настройка на редактирование текстов программ с учетом особенностей конкретного языка программирования (например, Си, Паскаль, Бейсик), т.е. задание стандартных конструкций языка типа IF (), FOR (), WHILE () и т.п.;
- трансляция текста программы без выхода из редактора.

К преимуществам этого редактора относятся: возможность использования символов псевдографики, что существенно упрощает представление данных в виде таблиц, а также наличие встроенного языка Macro для создания пользователем собственного набора макроопределений, обеспечивающего перепределение функциональных клавиш, создание окон на экране дисплея и занесение в них информации, изменение цветов окон и информации в них, перемещение окон на экране и т.п. Кроме того, язык Macro позволяет создавать свой собственный "сценарий" выполняемых действий (примером такого сценария является программа, демонстрирующая работу редактора).

При работе с файлами редактор использует многооконный интерфейс - до 100 окон, в каждое из которых можно выводить самые разнообразные объекты: текстовые файлы, меню, каталоги и подсказки. Размер редактируемого

файла ограничен только объемом свободной памяти; длина строки не должна превышать 254 символов.

Данный редактор поддерживает манипулятор типа "мышь", что существенно упрощает взаимодействие пользователя с редактором.

Наряду с нетрадиционными возможностями, о которых было сказано выше, Multi-Edit выполняет ряд таких функций обычного редактора, как подготовка к выводу на принтер файлов или помеченных блоков с заданием различных параметров печати (числа строк на странице, интервала между строками и т.п.); редактирование текста, в том числе его форма-

тирование и выравнивание, и др. Управление системой в режиме редактирования осуществляется с помощью меню либо команд, задаваемых нажатием комбинаций клавиш, что, конечно, не совсем удобно, однако в какой-то степени компенсируется наличием в редакторе режима контекстно-чувствительной подсказки.

По существу, Multi-Edit - это гибкая программная среда, настраиваемая с учетом специфики конкретного пользователя. В настоящее время в СССР применяются три различных варианта данного редактора: редактор, общение пользователя с которым осу-

ществляется по-английски; частично русифицированный редактор без переназначения функциональных клавиш и, наконец, полностью русифицированный редактор с переназначением этих клавиш.

В 1989 г. выпущена последняя версия редактора Multi-Edit - 4.0, которая имеет ряд дополнительных средств по сравнению с предыдущими версиями этого редактора, что существенно расширяет область ее применения. Например, с помощью встроенного языка Macro был создан набор макросов, позволяющих повысить производительность труда программиста при создании исходных текстов



Использование текстовых процессоров - лучший способ экономии бумаги!

программ на таких языках программирования, как Си, Паскаль, Бейсик, Модуль-2, Пролог и т.д. Кроме языка общения с пользователем претерпели изменения и внутренние структуры управления многофункционального редактора. Так, изменен интерфейс редактора с DOS, упрощен целый ряд процедур, применяемых для организации многооконного интерфейса, появилась возможность работы редактора с расширенной памятью. Однако "платой" за предоставленные возможности явилось увеличение требуемого объема памяти (как оперативной, так и на жестком диске). Версия 4.0 редактора Multi-Edit занимает около 850 Кбайт памяти на жестком диске и около 200 Кбайт оперативной памяти.

ChiWriter

Текстовый процессор ChiWriter (произносится "кирайтер"), разработанный фирмой Horstmann Software Design, предназначен для ввода, редактирования и подготовки к выводу на печать текстов книг, статей и т.п. В отличие от других текстовых процессоров, ChiWriter позволяет использовать в одном тексте практически произвольное (ограниченное лишь объемом памяти на диске) число наборов символов и шрифтов: это могут быть буквы русского и латинского алфавитов; шрифты различных кеглей (размеров) и гарнитур; математические символы, псевдографика; выделение подчеркиванием, курсивом, полужирным шрифтом и т.д. Особенностью этого текстового процессора является возможность конструирования нестандартных знаков, химических структур и рисунков. Предусмотрено также создание многоуровневых указателей, формирование заголовков, подстрочных примечаний и вставка в текст графических элементов, созданных с помощью других средств.

Управление работой текстового процессора осуществляется с помощью команд или иерархического меню. Возможна подготовка текста к выводу на принтер в двух режимах: для быстрой (черновой) печати и для печати, близкой по качеству к тому, которое обеспечивается пишущей машинкой; имеются драйверы для большого числа различных принтеров. Страницы формируются автоматически, возможно выравнивание текста по правому краю. Пожалуй, единственным существенным недостатком этого текстового процессора является использование кодировки (таблицы кодирования) символов кириллицы, отличающейся от используемых во всех известных нам текстовых процессорах. Это приводит к тому, что текст на русском языке, созданный с помощью этого текстового процессора, можно прочитать и подготовить к печати, только пользуясь им же самим.

T³

Текстовый процессор T³ (произносится "Тикуб"), разработанный фирмой TCI Software,

позволяет создавать, редактировать и готовить к выводу на принтер сложные технические документы, содержащие формулы, специальные символы и знаки, а также вставленные в текст чертежи, графики и рисунки. Это один из наиболее совершенных текстовых процессоров.

Интересно отметить, что в T³ реализованы концепции объектно-ориентированного подхода, который последнее время приобретает все большую популярность.

Текстовый процессор T³ размещается в двух каталогах (с именами T3 и T3VOLS) и имеет свой файл автовызова. Первый из этих каталогов содержит собственно текстовый процессор, а второй - его тома (томами называются файлы, в которых хранятся все используемые текстовым процессором средства: документы, макросы и т.д.).

T³ позволяет, используя восемь наборов символов (в том числе, наборы символов кода ASCII, математических и химических символов, кириллицы, латинского и греческого алфавитов, псевдографики и некоторые другие наборы символов), печатать текст с различными интервалами между словами и строками, используя шрифты разных кеглей и гарнитур. С помощью этого текстового процессора пользователь может даже создавать шрифты, комбинируя символы, входящие в различные уже существующие шрифты, а также изменять имеющиеся символы вплоть до создания новых. Используя эту особенность текстового процессора T³, можно получить символы кириллицы, по очертаниям полностью совпадающие со шрифтом пишущей машинки.

При редактировании документов T³ наряду с обычными для текстовых процессоров функциями позволяет выполнять целый ряд дополнительных:

- менять вид шрифта (кегель и гарнитуру), подключая в процессе редактирования любой из имеющихся в системе наборов шрифтов;
- делать шрифтовые выделения (курсив, полужирный шрифт и т.д.);
- создавать надстрочные и подстрочные индексы;
- центрировать заголовки;
- выравнивать символы по отмеченной позиции;
- использовать макроопределения, облегчающие набор сложных формул, задавать длины строк и интервалы между ними;
- создавать заголовки и сноски на печатаемых страницах;
- задавать формат страницы, т.е. выбирать для нее расположение заголовков, сносок и основного текста;
- разбивать документ на страницы вручную либо автоматически, нумеровать страницы в автоматическом или интерактивном режиме, менять параметры печати, т.е. устанавливать размер пробела между строками и задавать выравнивание строк по правому краю;
- печатать весь документ или отдельные его

страницы, задавая число копий для печати.

Такие разнообразные возможности реализуются с помощью разветвленной системы меню и специальных заготовок, а также большого числа специальных клавиш и их сочетаний, что крайне осложняет освоение текстового процессора T³. К тому же в этом текстовом процессоре нет системы подсказок HELP. Основным недостатком T³ является отсутствие доступа к его томам из других текстовых процессоров и из DOS; большим неудобством является отсутствие возможности работать с командами DOS без выхода из текстового процессора.

К достоинствам T³ относятся такие его возможности, как:

- конструирование новых символов и собственных шрифтов;
- автоматизация выполнения повторяющихся операций за счет создания макроопределений;
- смена шрифтов при редактировании;
- смещение курсора вверх и вниз на величину, кратную половине высоты шрифта, что позволяет создавать надстрочные и подстрочные индексы на нескольких уровнях.

Что же выбрать?

Все рассмотренные текстовые процессоры предназначены для работы в среде MS-DOS версии не ниже 3.20.

На наш взгляд, начинающему пользователю для создания исходных текстов программ следует применять редактор EDIT. По мере усложнения разрабатываемых программ и повышения степени освоения ПК пользователю стоит перейти к использованию редактора Multi-Edit, что позволит значительно повысить эффективность отладки программ. Для подготовки различных текстов, не содержащих формул, целесообразно применять текстовый процессор ЛЕКСИКОН, особенно в тех случаях, когда редактор используется нерегулярно. Наибольшей популярностью у советских пользователей ПК при подготовке сложных текстов с формулами пользуется ChiWriter (и, на наш взгляд, вполне заслуженно). Сотрудники издательств чаще других применяют WORD. Это, по-видимому, объясняется его совместимостью с настольной издательской системой Ventura Publisher. Мы считаем, что для подготовки текстов книг очень хорошо подходит T³.

А.С. БОГДАНОВ



Серия

VAX-11

VAX-11
VAX-11

*В серии
VAX-11
впервые
проявились
преимущества
архитектуры
32-разрядных
компьютеров
семейства
VAX.*

СЕМЕЙНЫЕ ХРОНИКИ

От PDP-11 к VAX-11

Успех фирмы DEC на рынке мини-компьютеров, связанный с выпуском серии 16-разрядных мини-компьютеров PDP-11, подтвердил, что архитектура компьютеров серии PDP-11 является весьма удачной. Самая производительная модель этой серии в 70-х гг. - PDP-11/70, уже могла конкурировать с некоторыми большими 32-разрядными вычислительными системами фирмы IBM. Тем не менее компьютерам серии PDP-11 был присущ существенный (с точки зрения пользователей) недостаток, свойственный, к сожалению, всем 16-разрядным мини-компьютерам, - относительно небольшое виртуальное адресное пространство, которое могла занимать программа пользователя, а именно 64 Кбайта. Этот недостаток в компьютерах серии PDP-11 отчасти компенсировался применением аппаратных средств управления памятью, хотя это и не позволяло расширить виртуальное адресное пространство. Кардинальным решением проблемы явилось создание фирмой DEC 32-разрядных компьютеров с новой архитектурой. Анонсирование первой модели нового семейства 32-разрядных компьютеров - VAX-11/780, состоялось в октябре 1977 г.

Архитектура

При разработке архитектуры компьютеров нового семейства VAX фирма DEC преследовала пять основных целей:

1. По возможности обеспечить максимальную совместимость с компьютерами серии PDP-11, но по сравнению с ними существенно расширить виртуальное адресное пространство и значительно улучшить функциональные характеристики.

2. Добиться высокоэффективной обработки данных благодаря введению новых режимов адресации и разнообразных типов данных.

3. Создать новый систематизированный набор инструкций, являющийся логическим расширением набора инструкций для компьютеров серии PDP-11. При этом инструкции для адресации операндов должны допускать применение любых режимов адресации независимо от используемых типов данных. Такой набор инструкций должен быть удобен при программировании, в частности при реализации конструкций и структур данных языков высокого уровня.

4. Предусмотреть возможность дальнейшего развития архитектуры, что должно допускать добавление новых инструкций и типов данных по единой схеме без нарушения общей структуры набора инструкций.

5. Обеспечить универсальность назначения нового семейства компьютеров, которые должны полностью перекрывать весь диапазон приложений компьютеров серии PDP-11.

Намеченные цели были достигнуты, а успех нового семейства компьютеров VAX превзошел все ожидания. Новые компьютеры по исполнению и стоимости были близки к классу мини-компьютеров, а по функциональным характеристикам уже мало уступали "большим" компьютерам и даже превосходили их по некоторым параметрам. Компьютеры семейства VAX стали одними из первых представителей класса супермини-компьютеров.

Наиболее привлекательной особенностью компьютеров семейства VAX явилось расширенное (по сравнению с компьютерами серии PDP-11) виртуальное адресное пространство - потенциально оно занимает 4 Гбайта, но фактически для программы пользователя ограничено 2 Гбайтами (для выпускаемых в то время компьютеров фирмы IBM его размер не превышал 16 Мбайт). Собственно, название VAX (Virtual Address eXtension) и отражает эту особенность. Такое существенное расширение виртуального адресного пространства в компьютерах семейства VAX обеспечивалось страничной организацией виртуальной памяти. Следует отметить, что функционирование виртуальной памяти поддерживается единым (и довольно сложным) комплексом технических и программных средств. Примечательно, что одновременно с разработкой аппаратуры для первых компьютеров семейства VAX велась и разработка предназначенной специально для них операционной системы VAX/VMS (Virtual Memory System) - системы виртуальной памяти. Поэтому не удивительно, что разработка системного программного обеспечения повлияла на архитектуру компьютеров семейства VAX. Хотя теперь они работают под управлением и других операционных систем, например системы ULTRIX-32 (адаптация ОС UNIX), все равно наиболее органичным сочетанием остается симбиоз VAX - VAX/VMS.

Рассмотрим некоторые элементы архитектуры, общие для всех компьютеров семейства VAX. Основным форматом данных для них является 32-битовое длинное слово. Процессор обеспечивает работу с 32-битовыми адресами, при этом минимальной адресуемой единицей является байт. В результате возможна адресация 2^{32} байт (4 Гбайт). Во всех компьютерах семейства VAX реализовано управление виртуальной памятью. Виртуальная память разбита на страницы длиной по 512 байт. С каждой страницей связан ряд атрибутов, регламентирующих права доступа к ней, например определяющих, разрешен ли программе доступ к той или иной странице с правом на чтение или запись.

Процессор имеет шестнадцать 32-разрядных регистров общего назначения, а также регистр PSL (Processor Status Longword - длинное слово состояния процессора); все эти регистры доступны для пользователей. Из 16 регистров общего назначения четыре зарезервированы

для специального использования. Это регистры PC, SP, FP, AP. В регистр PC (Program Counter), исполняющий роль счетчика инструкций, заносится адрес следующей выполняемой инструкции программы. Регистр SP (Stack Pointer - указатель стека) содержит текущий адрес вершины стека. Регистр FP (Frame Pointer - указатель блока вызова) содержит адрес начала некоторой структуры данных, помещенной в стек и используемой при вызове процедур (другое его название - указатель фрагмента стека). В регистр AP (Argument Pointer - указатель аргументов) помещается адрес начала списка аргументов, передаваемых процедуре.

Как и в компьютерах серии PDP-11, в компьютерах серии VAX-11 применяется векторная обработка прерываний. Всего имеется 32 уровня приоритетов аппаратных и программных прерываний. Защита контекста вычислительных процессов обеспечивается переключением процессора в один из четырех возможных режимов доступа: ядра, управления, супервизора, пользователя (указаны в порядке снижения уровня приоритета). В каждом из этих режимов на самом деле используется свой специальный указатель стека. Наличие нескольких регистров-указателей стека скрыто от пользователя.

Характерной особенностью компьютеров серии VAX-11 является возможность переключения процессора из основного режима работы в режим совместимости с компьютерами серии PDP-11, в котором используется 16-битовая адресация и может выполняться большинство инструкций из набора инструкций компьютеров серии PDP-11, причем их выполнение не является программной эмуляцией, а вместе с выполнением основного набора инструкций обеспечивается работой процессора. В базовый основной набор инструкций компьютеров семейства VAX входит 248 инструкций, а в полный основной набор - 304 инструкции. Такое разделение обусловлено тем, что в первых моделях компьютеров семейства VAX операции над числами с плавающей точкой, представленными с повышенной точностью, выполнялись не основным процессором, а дополнительным устройством - ускорителем выполнения операций над числами с плавающей точкой (FPA - Floating Point Accelerator).

В основной набор входят самые разнообразные инструкции, позволяющие вести обработку данных различного типа, а порой уникальные и экзотические инструкции, например инструкция POLY - вычисление значения полинома, или инструкция CRC (Cyclic Redundancy Check) - контроль с помощью циклического избыточного кода. Различаются и объемы памяти, требуемой для той или иной инструкции, - от 1 до 30 байт. Даже для одной и той же инструкции объем памяти может меняться в зависимости от используемых для операндов режимов адресации. Поэтому в компьютерах семейства VAX не требуется выравнивания начала инструкций по какой-либо иной границе, кроме границы байта.

Основными типами данных являются целые числа и числа с плавающей точкой. Целые числа независимо от

формата их представления могут интерпретироваться как целые со знаком и без знака. Для представления целых чисел имеются следующие форматы: 8-битовый байт, 16-битовое слово, 32-битовое длинное слово, 64-битовое квадрослово, 128-битовое октаслово. Для представления чисел с плавающей точкой используются четыре формата: F - одинарная точность (32 бита); D - двойная точность (64 бита); G - двойная точность с расширенным порядком (64 бита); H - учетверенная точность (128 бит). Помимо работы с указанными типами данных обеспечивается работа с упакованными десятичными числами, символьными строками, цепочками бит.

Для адресации операндов инструкций могут использоваться девять режимов адресации, в том числе семь основных, принципиально не отличающихся от применяемых в PDP-11, а именно: регистровый, регистровый косвенный, с автоуменьшением, с автоувеличением, с автоувеличением косвенный, со смещением (аналогичен индексному в PDP-11), со смещением косвенный. Кроме того, введены два новых режима адресации: литеральный и с индексацией. Последний применяется только в сочетании с любым из основных режимов адресации, за исключением регистрового. Возможны также различные модификации режимов адресации, например использование при адресации регистра PC (программного счетчика) вместо регистра общего назначения. Множество различных допустимых комбинаций режимов адресации облегчает программирование и позволяет повышать его эффективность.

VAX-11/780 и другие модели

Семейству компьютеров VAX положила начало первая серия компьютеров - VAX-11 (число 11 в названии серии должно было подчеркнуть преемственность архитектуры компьютеров серии PDP-11). Первая модель семейства - компьютер VAX-11/780 с быстродействием 1 млн инструкций/с, стал поистине эталоном. Интересно, что его выпуск продолжается и в настоящее время.

Конфигурация процессора модели VAX-11/780 являлась образцом при создании новых моделей компьютеров как серии VAX-11, так и более поздних серий. В нее входили: консольная подсистема на базе компьютера серии LSI-11; микропрограммная память WCS (Writable Control Store), в которую загружался управляющий микрокод; кэш-память емкостью 8 Кбайт. В качестве внутренней системной шины использовалась новая синхронная шина SBI (Synchronous Backplane Interconnect - интерфейсная шина межблочных соединений) с 32-разрядным трактом данных. Емкость реальной физической памяти могла достигать 12 Мбайт, в настоящее время она может быть увеличена до 64 Мбайт. Ускорение выполнения операций над числами с плавающей точкой, представленными с повышенной точностью, достигалось установкой ускорителя. Для

* ФАКТЫ

- На последней выставке IEEE Wescon (США) большое внимание было уделено системам автоматизированного проектирования. В экспозиции было представлено 75 фирм-разработчиков САПР радиоэлектронной аппаратуры, среди них такие известные, как Applied Micro-Systems, CAD Software, OrCAD Systems, Racal-Redac. Анализируя экспонаты, можно подумать, что в автоматизации проектирования, тестирования и производства электронных компонентов больше нет проблем. Даже ПК справляются с автоматической трассировкой, логическим синтезом и моделированием.
- Фирма Integrated Information Technology (США) сконструировала новый видеоадаптер - IGA (Integrated Graphics Array), представляющий собой комбинацию стандартных видеоадаптеров VGA и 8514/A, реализованную в виде одной микросхемы. Блок VGA нового видеоадаптера обеспечивает работу не только в режимах CGA, MDA, Hercules, EGA и VGA (16 цветов, разрешение до 640x380 точек или 256 цветов из палитры 256 000 цветов, разрешение 320x200 точек), но и в режиме Super VGA (16 цветов, разрешение 800x600 точек). Блок 8514/A позволяет отображать 256 цветов из палитры 256 000 цветов (разрешение 1024x768 точек). В настоящее время видеоадаптер проходит испытания. Предполагается, что он будет выпущен на рынок в следующем году.

подключения периферийных устройств можно было использовать шины MASSBUS и UNIBUS. К процессору могли быть подключены по четыре шины MASSBUS и UNIBUS. В стандартной конфигурации использовалась одна шина UNIBUS. Быстродействие процессора составляло 1,06 млн инструкций/с.

Для сравнения характеристик различных компьютеров семейства VAX фирма DEC приняла быстродействие процессора компьютера VAX-11/780 за единицу производительности (ЕП).

Стоимость компьютера VAX-11/780 в базовой конфигурации была все-таки довольно высокой по сравнению со стоимостью старших моделей серии PDP-11. Для расширения рынка сбыта фирма DEC решила выпустить ряд менее производительных, но соответственно и более дешевых компьютеров семейства VAX. В октябре 1980 г. была анонсирована вторая модель серии VAX-11 - VAX-11/750. Компьютер имел кэш-память емкостью 4 Кбайт. Максимальная емкость реальной памяти могла достигать 4 Мбайт (в дальнейшем 8 Мбайт). Можно было подключать две шины UNIBUS и три шины MASSBUS. Производительность компьютера составляла 0,65ЕП. Стоимость компьютера VAX-11/750 была почти на 40 % ниже стоимости компьютера VAX-11/780.

Младшая модель серии VAX-11 - VAX-11/730, была анонсирована в апреле 1982 г. В компьютер с базовой конфигурацией входили память емкостью 1 Мбайт и одна шина UNIBUS. Емкость памяти могла быть расширена до 5 Мбайт, можно было также подключать две дополнительные шины UNIBUS. Подключение шин MASSBUS не предусматривалось. Производительность компьютера составляла 0,3ЕП. Стоил компьютер VAX-11/730 почти в два раза меньше компьютера VAX-11/750.

Самым производительным в серии VAX-11 стал компьютер VAX-11/782, о выпуске которого было объявлено в феврале 1982 г. Он представлял собой многопроцессорную асимметричную систему, построенную на базе двух процессоров компьютера VAX-11/780. Процессоры разделяли общее поле памяти, что достигалось с помощью подсистемы памяти MA-780, имеющей двухпортовую схему включения. Все периферийные устройства подключались к одному из процессоров, выполняющему функцию ведущего. Емкость памяти этой вычислительной системы могла быть увеличена до 36 Мбайт. Ее производительность составляла 1,8ЕП.

В октябре 1983 г. было объявлено о выпуске двух новых, самых младших моделей семейства компьютеров VAX - VAX-11/725 и MicroVAX I (первом компьютере серии MicroVAX, о которой будет рассказано отдельно). Компьютер VAX-11/725 имел тот же процессор, что и компьютер VAX-11/730. По существу он представлял собой "урезанный" вариант своего предшественника, имея память емкостью 1-3 Мбайт и одну шину UNIBUS. Мало отличаясь по стоимости от модели VAX-11/730, этот компьютер быстро потерял спрос на рынке.

В апреле 1984 г. была анонсирована модель VAX-11/785, завершившая серию VAX-11. Она имела то же схемное решение, что и модель VAX-11/780, но была выполнена на более быстродействующей элементной базе. Производительность модели составляла (1,5-1,7)ЕП.

В серии VAX-11 впервые проявились преимущества архитектуры компьютеров семейства VAX. Успех этой серии не был случайным, и хотя ее сменило много других серий, до сих пор продолжается мелкосерийный выпуск трех наиболее удачных моделей: VAX-11/780, VAX-11/750, VAX-11/785.

СЛОВАРЬ

MASSBUS - высокоскоростная системная интерфейсная шина с 16-разрядным трактом данных. Передача служебной управляющей информации по шине осуществляется в асинхронном режиме, передача данных - в синхронном режиме. Первоначально шина была разработана для подключения к компьютеру PDP-11/70 быстродействующих внешних запоминающих устройств, таких как магнитные диски типа RP или RM. Скорость передачи данных по шине MASSBUS достигает 2 млн байт/с.

Q-BUS (LSI-11 Bus) - асинхронная интерфейсная шина с 16-разрядным совмещенным трактом данных, разработанная как системная шина для компьютеров серии LSI-11. Используется для тех же целей, что и шина UNIBUS, но дешевле нее благодаря использованию одних и тех же линий для передачи как данных, так и младших разрядов адреса. Поскольку при этом процессы передачи адреса и данных разнесены по времени, то быстродействие шины снижается почти вдвое по сравнению с быстродействием шины UNIBUS, но вместе с тем сокращается и число элементов в электронной схеме шины. В компьютерах различных серий семейства VAX шина

Q-BUS не используется (за исключением компьютеров серии LSI-11, применяемых в консольных подсистемах), однако для компьютеров серии MicroVAX она является основной системной шиной.

UNIBUS - асинхронная интерфейсная шина с 16-разрядным трактом данных. Является основной системной интерфейсной шиной для компьютеров серии PDP-11. В компьютерах семейства VAX используется для подключения внешних устройств, в том числе периферийных устройств, первоначально разработанных для компьютеров серии PDP-11. В последних сериях компьютеров семейства VAX практически не применяется. Скорость передачи данных по шине UNIBUS достигает 1,5 млн байт/с.

virtual address space (виртуальное адресное пространство) - диапазон адресов, которые могут использоваться внутри программы, независимо от их соответствия реальным адресам физической памяти. Верхняя граница виртуального адресного пространства обычно определяется максимальным числовым значением, которое может быть использовано при прямой адресации, а это значение зависит от информационной емкости регистров общего назначения (для мини-компьютеров, как правило, 16 бит). Максимальным значением, которое может быть представлено в 16-разрядном регистре, является 216, поэтому для мини-компьютеров (например, серии PDP-11), где минимальной адресуемой единицей является байт, виртуальное адресное пространство ограничено 64 Кбайтами.

СЕМЕЙНЫЕ ХРОНИКИ

В.В. ЛЕОНАС



UNIX

и микро-
процессорINTEL
80386

Появление
компьютеров
на основе
микропроцессора
Intel 80386
сделало
сочетание слов "ПК"
и "ОС UNIX"
неразрывным.

В настоящее время сочетание слов "персональный компьютер" и "операционная система (ОС) UNIX" воспринимается уже совсем не так фантастически, как лет 5-10 назад. Действительно, в те годы сообщение о разработке версий ОС UNIX для 8-разрядных персональных компьютеров (ПК) на основе микропроцессоров Intel 8080 и Z8000, а также для младших моделей 16-разрядных ПК серии LSI-11 могло вызвать в лучшем случае снисходительную усмешку. Появление версий ОС UNIX для ПК на основе микропроцессоров Intel 8086 и Intel 8088 (например, ОС VENIX) продемонстрировало лишь принципиальную возможность того, что ОС этого семейства могут работать (неважно, плохо или хорошо) на 16-разрядных ПК. Опыт разработки ПК на основе микропроцессора Intel 80286 и версий ОС UNIX для них (в первую очередь, речь идет, конечно, об ОС XENIX) показал реальность использования ОС этого семейства на 16-разрядных ПК. Однако только появление компьютеров на основе микропроцессора Intel 80386, по-прежнему называемых персональными, но которые язык уже (или, может быть, еще?) не поворачивается так назвать, сделало, по-видимому, сочетание слов "ПК" и "ОС UNIX" неразрывным на будущее. Этому также способствовало успешное завершение процесса стандартизации ОС UNIX (практически разработан набор стандартов POSIX).

Возникает законный вопрос: чем обусловлена необходимость создания новых ОС для ПК, если есть такая популярная ОС, как MS-DOS, для которой разработано фантастически большое количество пакетов прикладных программ из самых разных предметных областей? Не вдаваясь в обсуждение ситуации с "виртуальной" (она как бы есть, но ее как бы и нет) операционной системой OS/2, попытаемся ответить на поставленный вопрос.

Стандарты для подключения жестких дисков

На ПК, работающих под управлением ОС семейства UNIX, жесткие диски должны подключаться в соответствии с одним из следующих стандартов:

MFM - Modified Frequency Modulation; известны также другие названия этого стандарта: ST-506 или IBM PC/XT STANDART (17 секторов на дорожке);

RLL - Run Length Limited (26 секторов на дорожке);

ESDI - Enhanced Small Disk Interface;

SCSI - Small Computer Systems Interface.

Стандарт MFM (ST-506) реализует наиболее дешевый способ подключения жесткого диска, обладающий, однако, самым низким быстродействием; емкость диска при использовании этого стандарта составляет 20 Мбайт.

Подключение по стандарту RLL обеспечивает несколько большее быстродействие (в частности, благодаря увеличению емкости диска до 30 Мбайт), но реализация этого стандарта стоит немного дороже.

Значительно более дорогой стандарт ESDI позволяет получить существенный выигрыш в быстродействии, поскольку применяются диски с совершенно другим качеством покрытия.

Самым универсальным является стандарт SCSI, позволяющий подключать любые внешние устройства (а не только жесткие диски).

Даже простое сравнение ОС UNIX и MS-DOS показывает, что первая из них по своим возможностям намного богаче второй. Данное обстоятельство определяется, в первую очередь, различной природой этих двух операционных систем: если ОС UNIX является многопользовательской многопрограммной операционной системой, то MS-DOS представляет собой однопользовательскую однопрограммную операционную систему (строго говоря, ее и операционной системой-то назвать трудно - это, скорее, монитор). Поклонники MS-DOS, не знакомые с настоящими многопользовательскими многопрограммными ОС, обычно возражают против этого, ссылаясь в качестве примера на среду MS Windows, которая не является по-настоящему многопрограммной средой (о многопользовательской среде здесь речи вообще идти не может). Кроме того, в ОС UNIX максимальная длина программ практически не ограничена - она может достигать 16 Мбайт, в то время как в MS-DOS эта величина не превышает 640 Кбайт. Наконец, такие поддерживаемые ОС UNIX функциональные возможности, как электронная почта, сетевые средства, управление планированием выполнения программ, средства автоматического вызова программ на выполнение через заданные интервалы времени или в определенные моменты времени, контроль доступа к информации, различные многооконные системы (например, системы Multiview и Xsight, соответствующие стандарту X11) и другие возможности делают эту ОС привлекательной для пользователей.

Необходимо отметить, что среди фирм, занятых разработкой ОС UNIX для компьютеров на основе микропроцессора Intel 80386 (а таких фирм в настоящее время насчитывается не менее десятка), лидирующие

позиции занимают Interactive Systems Corp. (ISC) и Santa Cruz Operation (SCO).

В течение ряда последних лет разработанная фирмой SCO операционная система XENIX (версии которой были реализованы для компьютеров на основе практически всех типов микропроцессоров фирмы Intel: 8086, 8088, 80286 и 80386) считалась стандартом de facto для компьютеров, построенных на основе микропроцессоров этой фирмы, что позволило ей захватить к середине 1989 г. 80 - 90 % рынка ОС типа UNIX для ПК. Правда, в последнее время фирме ISC удалось потеснить фирму SCO, быстро реализовав ОС UNIX версии System V/386 Release 3.2, известную под названием 386/ix version 2.0 (в США цена 1995 - 2995 дол.). Аналогичный продукт фирмы SCO получил название SCO XENIX/UNIX System V/386 Release 2.3 (в США цена 3470 дол.). Помимо собственно операционной системы каждая из этих двух фирм предлагает ряд пакетов прикладных программ, среди которых хотелось бы отметить систему поддержки разработки программ на языке Си, пакет X Windows и пакет VP/ix для эмуляции среды MS-DOS.

Фирма SCO, кроме того, предлагает пакет для текстовой обработки SCO LYRIX 6.0, систему управления базами данных SCO FoxBASE (совместимую с dBASE II), электронную таблицу SCO Professional, совместимую с пакетом LOTUS 1-2-3, графический пакет SCO IMAGEBUILDER, системы для автоматизации учрежденческой деятельности SCO MANAGER Release 1.0, SCO Open Desktop и SCO Portfolio Solution, а также сетевой пакет SCO XENIX-NET.

К настоящему времени для перечисленных выше ОС разработано множество пакетов прикладных программ. Так, фирмами UNIFY, Informix, Ashton-Tate и DataAccess разработаны системы управления базами данных и инструментальные средства для создания информационных систем, а фирмой Autodesk создан пакет AutoCAD для автоматизации проектирования.

В заключение несколько слов о документации по рассмотренным выше ОС и ее качестве.

В комплект поставки SCO XENIX/UNIX System V/386 Release 2.3 входят:

- базовый комплект (собственно операционная система);
- система поддержки разработки программ;
- система обработки текстов.

Соответствующим образом организована и документация. В частности, документация на базовый комплект поставки оформлена в виде набора, состоящего из трех стандартных папок (байндеров) со сменными страницами и одной книги в мягком переплете (того же формата, что и у папок), страницы которой скреплены пружиной:

- Байндер 1 - "Release Notes"
- Installation Guide

- System Administrator's Guide
- System Administration (ADM)
- Hardware Dependent (HW)

Байндер 2 - "User's Guide"

- International Operating System Guide
- International Supplement. Release & Installation Notes

Байндер 3 - "User's Reference Manual"

- Commands (C)
- Miscellaneous (M)
- File Formats (F)
- Книга - "Xenix Tutorial"

На первый взгляд этот комплект документации производит весьма серьезное впечатление. Казалось бы,

Видеоадаптеры, поддерживаемые UNIX-совместимыми ОС

На ПК, работающих под управлением ОС семейства UNIX, могут использоваться практически любые видеоадаптеры: CGA (320x200 точек, 4 цвета), EGA (640x350 точек, 16 цветов), VGA (640x480 точек, 16 из 256 цветов), Enhanced VGA (1024x768 точек, 256 цветов), Hercules monochrome, IBM-compatible monochrome. Исключение составляют пакет SCO X11 и некоторые другие графические пакеты, для работы которых требуется использование видеоадаптеров EGA или VGA.

ни один из аспектов ОС UNIX не остался в ней неохваченным. На самом же деле здесь не хватает двух разделов: Programmer's Guide и Programmer's Reference Manual (эти разделы не входят в базовый комплект документации и являются частью комплекта документации по системе поддержки разработки программ).

Надо отдать должное фирме SCO, которая существенно переработала исходную документацию, подготовленную фирмой AT&T, улучшив ее удобочитаемость. В частности, использованы жесткие картонные разделители глав и разделов, снабженные выступающими "ушами", составлены подробные оглавления глав и разделов, на каждой странице проставлен колонтитул - справочный текст, позволяющий сразу определить, к какой главе или разделу она относится. Помимо наличия комплекта документации на бумаге имеется также возможность организации интерактивного доступа к документации с помощью команды tap с соответствующим "наполнением" информацией.

В состав документации по ОС 386/ix version 2.0 входят стандартный комплект документации по ОС UNIX System V Release 3.2 и дополнения, в которых описаны отличия от нее. Вся документация объединена в книги, страницы которых скреплены пружинами. Базовый комплект документации состоит из трех томов. К сожалению, документация не всегда содержит достаточно полную информацию.

* ФАКТЫ

- Норвежская фирма Dolphin Server Technology разрабатывает аналог микропроцессора Motorola 88000 на базе логических схем с эмиттерными связями. Новый микропроцессор будет работать на тактовой частоте 125 МГц и выполнять 1000 млн операций в секунду (для неоптимизированного кода - 300 млн операций в секунду). Микропроцессор параллельно выполняет восемь операций.
- Фирма Hewlett-Packard разработала новую технологию проектирования микропроцессоров, которая позволит реализовать на одном кристалле микропроцессор с RISC-ориентированной архитектурой Precision Architecture и по меньшей мере вдвое повысить быстродействие компьютеров, в которых он используется в качестве центрального процессора. Новая микросхема, реализованная на кристалле со стороной 2,5 см и содержащая около 1 млн транзисторов, будет работать на тактовой частоте свыше 48 МГц и потреблять мощность менее 10 Вт.
- Наконец и фирма Dell, известная высоким качеством своих настольных компьютеров, выпустила первую портативную модель - Dell 316LT (оперативная память емкостью от 1 до 8 Мбайт, жесткий диск емкостью 20 или 40 Мбайт, видеоадаптер VGA). Компьютер питается от батареек, которые можно заменять в процессе работы, поскольку специальный встроенный аккумулятор поддерживает работу процессора и памяти в течение нескольких минут.

СЛОВАРЬ

Первая попытка создания словаря по ОС UNIX была предпринята в 1983 - 1984 гг. коллективом разработчиков ОС ИНМОС (версия 0), совместимой с ОС UNIX v.6., в рамках работы группы по машинно-независимым ОС при Комиссии по языкам и системам программирования ГКНТ СССР. Настоящий словарь создан на основе вышеупомянутого словаря.

absolute full pathname (*абсолютное полное имя файла*) - последовательность имен каталогов, отделяемых друг от друга символом "/", начинающаяся от корневого каталога и заканчивающаяся собственно именем файла, которая однозначно идентифицирует путь от корневого каталога до файла (см. также *имя файла, полное имя файла, относительное полное имя файла, собственно имя файла*).

archive (*архив*) - содержимое архивного файла.

archive file (*архивный файл*) - файл определенной структуры, содержащий библиотеку объектных файлов.

block-oriented special file (*блок-ориентированный специальный файл*) - файл, соответствующий внешнему устройству, ориентированному на поблочный обмен информацией, например накопителю на жестком диске (см. также *специальный файл, байт-ориентированный специальный файл*).

byte-oriented special file (*байт-ориентированный специальный файл*) - файл, соответствующий внешнему устройству, ориентированному на побайтный обмен информацией, например терминалу (см. также *специальный файл, блок-ориентированный специальный файл*).

command (*команда*) - единица действия одного из интерпретаторов команд.

command file, shellscript (*командный файл*) - файл, содержащий командные строки.

command interpreter (*интерпретатор команд*) - программа, обеспечивающая интерпретацию команд, командных строк и командных файлов.

command line (*командная строка*) - строка, содержащая одну или более команд (разделяемых символами ";" или "|") и их аргументы.

command name, command (*имя команды*) - имя выполняемого файла, содержащего эту команду.

directory (*каталог*) - файл, содержащий список имен файлов и информацию о местоположении описателя каждого из этих файлов.

end of file, eof (*конец файла*) - ситуация, означающая, что указатель чтения-записи файла находится в этом файле после последнего байта.

executable file (*выполняемый файл*) - файл, содержащий выполняемую программу на языке машинных кодов или на языке одного из интерпретаторов команд, с установленным разрешением на доступ, по крайней мере, для одной из категорий пользователей (владелец, члены группы владельца, прочие пользователи).

execute (*вызов на выполнение*) - последовательность действий, приводящая к выполнению программы.

execute in background 1. (*вызов на выполнение в фоновом режиме*) - последовательность действий, приводящая к выполнению программы в фоновом режиме; 2. (*выполнение в фоновом режиме*) - выполнение программы в режиме, когда полностью отсутствует связь с терминалом.

file mode (*код защиты файла*) - целое число, хранящееся в описателе файла и содержащее информацию о виде файла (обычный файл, каталог, специальный файл (байт- или блок-ориентированный)) и правах доступа к этому файлу для различных категорий пользователей (владелец, члены группы владельца, прочие пользователи).

filename generation (*генерация имен файлов*) - формирование последовательности имен файлов в результате интерпретации метасимволов.

group identifier (*идентификатор группы*) - целое число, обеспечивающее однозначную идентификацию группы пользователей в ОС UNIX.

input from terminal (*ввод с терминала*) - ввод информации с клавиатуры терминала.

linker (*компоновщик*) - программа, обеспечивающая разрешение внешних ссылок, компоновку и подготовку выполняемых файлов.

login 1. (*войти в систему*) - осуществить последовательность действий по входу в систему; 2. (*выход в систему*) - последовательность действий пользователя, предшествующая началу работы и заключающаяся во вводе входного имени пользователя и, если это необходимо, его пароля.

login name (*входное имя пользователя*) - имя, под которым пользователь зарегистрирован в ОС UNIX, обеспечивающее однозначную идентификацию пользователя в процессе его входа в систему.

logout 1. (*выйти из системы*) - осуществить последовательность действий по выходу из системы; 2. (*выход из системы*) - последовательность действий пользователя, предшествующая завершению работы и заключающаяся во вводе команды logout.

metacharacter (*метасимвол*) - символ, имеющий для некоторой программы (например, для одного из интерпретаторов команд) специальное значение.

metanotation (*метанотация*) - строка символов, в которой используются метасимволы.

output to terminal (*вывод на терминал*) - вывод информации на экран терминала.

owner (*владелец*) - пользователь, идентификатор которого совпадает с идентификатором пользователя, хранящимся в описателе соответствующего файла.

owner name (*имя владельца*) - входное имя пользователя, являющегося владельцем файла.

pipe (*канал*) - средство обмена информацией между двумя процессами, используемое, в частности, при построении конвейеров.

pipeline (*конвейер*) - объединение двух или более команд в одну командную строку с помощью символа "|", обеспечивающее замыкание стандартного вывода левой команды на стандартный ввод правой команды в каждой паре команд, объединенных с помощью символа "|".

process context (*контекст процесса*) - информация о процессе, необходимая операционной системе, когда образ процесса находится в оперативной памяти.

process identifier, process id, pid (*идентификатор процесса*) - целое число, обеспечивающее однозначную идентификацию процесса в ОС UNIX.

root directory (*корневой каталог*) - корень древовидной структуры, соответствующей файловой системе.

system crash (*крах системы*) - аварийная ситуация, результатом которой является невозможность дальнейшего функционирования ОС UNIX.

user identifier, user id, uid (*идентификатор пользователя*) - целое число, обеспечивающее однозначную идентификацию пользователя в ОС UNIX.

СЕМЕЙНЫЕ ХРОНИКИ

Grep и awk - опыт сравнения

Статистические характеристики программы (например, число строк, число вхождений определенных символов или совокупностей символов) могут оказаться весьма полезными при решении ряда практических задач программирования. Одна из таких задач - оценка качества программы и квалификации написавшего ее программиста. При решении этой задачи могут пригодиться данные об использовании в программе (частоте вхождения в нее) различных языковых конструкций. Другая задача - выявление "сложных" программ, требующих повышенного внимания при отладке и сопровождении. Одним из признаков такой программы является, например, большое число управляющих операторов.

Определение статистических характеристик программы вручную или с помощью калькулятора утомительно и чревато ошибками. Специальные средства для автоматического вычисления таких статистических характеристик в системах разработки программ либо отсутствуют вообще, либо решают тривиальные задачи типа оценки "стиля" программы (под стилем программы понимается ее ясность, удобочитаемость и понимаемость).

При работе с операционной системой UNIX в распоряжении программиста имеются средства, позволяющие достаточно просто определить статистические характеристики программы.

Прежде всего, это язык awk для обработки структурированных текстов программ. Программа на языке awk представляет собой совокупность правил, каждое из которых содержит селектор, задающий некоторый критерий выбора записи в тексте, и описание действия, выполняемого в случае успешного применения критерия.

Пусть, например, в тексте программы file, написанной на языке Си, требуется определить число вхождений ключевых слов case, break, continue и т.д. Эту задачу можно решить с помощью следующей программы на языке awk:

```
# Программа содержится в файле prog
/case/ {r["case"]++;}
# При обнаружении слова "case" в
тексте программы соответствующий
элемент массива r увеличивается на 1
/break/      {r["break"]++;}
/continue/   {r["continue"]++;}
/default/    {r["default"]++;}
/do/         {r["do"]++;}
```

```
/else/       {r["else"]++;}
/for/        {r["for"]++;}
/goto/       {r["goto"]++;}
/if/         {r["if"]++;}
/return/     {r["return"]++;}
/switch/     {r["switch"]++;}
/while/      {r["while"]++;}
END          { for (i in r)
PRINT ("r["i"]=", r[i]);
```

Эта программа выполняется с помощью интерпретатора языка awk, который вызывается следующим образом:

```
awk -f prog file
```

Следует подчеркнуть, что язык awk - чрезвычайно гибкое, но довольно сложное средство. Его применение оправдано, когда требуется нетривиальный анализ структуры текста программы или его преобразование.

В ряде случаев для определения статистических характеристик программы можно воспользоваться командами grep, fgrep или egrep, осуществляющими поиск заданных конструкций в тексте по шаблону. Приведенная ниже последовательность команд grep, объединенных в командный файл интерпретатора команд c-shell, функционально эквивалентна рассмотренной выше программе на языке awk:

```
# csh
# Этот командный файл выполняется
интерпретатором команд c-shell
grep -c case      file r
grep -c break     file r
grep -c continue  file r
grep -c default   file r
grep -c do        file r
grep -c else      file r
grep -c for       file r
grep -c goto      file r
```

```
grep -c if      file r
grep -c return  file r
grep -c switch  file r
grep -c while   file r
```

Интересно отметить, что на выполнение последовательности команд grep, каждая из которых просматривает весь файл анализируемой программы, затрачивается меньше машинного времени, чем на интерпретацию программы на языке awk, в ходе которой текст просматривается однократно. В таблице приведены данные о времени анализа текста девяти программных модулей на языке Си, входящих в состав программы-администратора ОС ДЕМОС, с помощью программы на языке awk и командного файла грег (эксперимент проводился на компьютере СМ 1420).

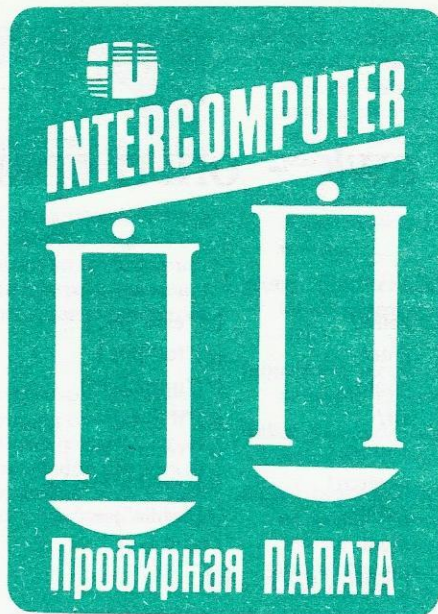
В скобках после имени модуля указывается число строк в нем, вычисленное с помощью команды wc. Эта команда может также использоваться для подсчета числа слов и символов в одном или более файлах, указанных в качестве ее аргументов.

Для поиска более сложных синтаксических конструкций, чем ключевые слова, можно применять команду egrep или специальные программы для требуемой обработки результатов, полученных с помощью команд grep. В ряде случаев такой способ оказывается эффективнее, чем применение программ на языке awk, благодаря экономии времени, необходимого для освоения языка awk, и памяти (для реализации программы, написанной на языке awk, требуется значительно больше памяти, чем для выполнения команды grep или egrep).

Комбинирование перечисленных инструментальных средств ОС UNIX и удачный выбор режимов их работы позволят программисту эффективно решать задачи обработки текстов программ.

Г.Г. Кольнер

Имя модуля	Время исполнения, с	
	Awk	Grep
adm.c (404)	12.0	7.5
adm1.c (134)	4.0	5.0
adm2.c (387)	10.9	6.5
adm3.c (318)	8.0	5.5
adm4.c (239)	7.0	5.0
adm5.c (309)	8.1	6.0
clbvd.c (64)	2.7	3.0
dostat.c (147)	4.9	4.5
timstat.c (292)	7.8	5.0



Начать тестирование наша "Пробирная палата" решила с программно-аппаратных комплексов, работающих под управлением столь интересного и приобретающего все большую популярность программного продукта, как операционная система (ОС) UNIX. В зарубежных компьютерных журналах, а также в отечественных публикациях на эту тему неоднократно объяснялись причины стремительного роста популярности ОС UNIX и совместимых с ней операционных систем; повторять эти рассуждения еще раз нам представляется излишним.

Мы провели тестирование нескольких доступных советскому покупателю UNIX-совместимых ОС. Следует заметить, что, с нашей точки зрения, бессмысленно исследовать эффективность аппаратных и программных средств отдельно. Это обусловлено тем, что в СССР число типов приобретенных легальным путем компьютеров, работающих под управлением UNIX-совместимых ОС, весьма невелико, а число официально купленных версий этих ОС для компьютеров одного типа еще меньше. Все это затрудняет получение достаточно представительной выборки.

UNIX- СОВМЕСТИМЫЕ ОС

Описание использованных методик тестирования

метсри - методика, позволяющая получить совокупную оценку вычислительной мощности процессора и эффективности использования ее конкретной операционной системой. Для получения этой оценки соответствующая программа тестирования резервирует в оперативной памяти компьютера область размером 100 Кбайт и заполняет ее блоками длиной 1 Кбайт. В последних байтах

каждого из указанных блоков содержится код ASCII порядкового номера блока. В процессе работы программа тестирования генерирует случайные целые числа в диапазоне 1 - 100, копирует содержимое блока с указанным порядковым номером из зарезервированной области оперативной памяти в буфер и сравнивает это содержимое с эталоном, т.е. с ожидаемым значением. Последняя операция повторяется 100 000 раз. Время выполнения описанной программы тестирования и принимается за искомую совокупную оценку.

memdisk - методика, позволяющая получить совокупную оценку эффективности файловой системы рассматриваемой ОС и соответствующих аппаратных средств, поддерживаемых ею. С точки зрения программной реализации, эта методика в основном похожа на методику **memcpu**. Одно из принципиальных отличий заключается в том, что вместо резервирования области оперативной памяти компьютера в данном случае резервируется область памяти на диске, т.е. на диске создается файл длиной 100 Кбайт. Далее этот файл заполняется блоками длиной 1 Кбайт, в последних байтах каждого из которых содержится код ASCII порядкового номера блока. Затем программа тестирования копирует содержимое случайно выбранного блока из созданного файла в буфер и сравнивает содержимое этого блока с ожидаемым значением. Последняя

мени на создание текстового файла длиной 1 Мбайт.

Sorting 1Mb - также традиционная для UNIX-совместимых ОС методика, позволяющая получить совокупную оценку эффективности файловой и операционной систем. Программная реализация этой чрезвычайно простой методики заключается в определении времени выполнения стандартной для всех UNIX-совместимых ОС утилитой **sort** операции лексикографического упорядочивания текстового файла длиной 1 Мбайт.

Результаты применения описанных выше методик приведены в таблице, где также содержатся дополнительные данные, очень важные для получения верного представления об эффективности рассматриваемой операционной системы.

эффективности разделения операционной системой вычислительных ресурсов компьютера, т.е. оценка эффективности планировщика процессов, состоит в определении эффективности внутренних механизмов ОС, поддерживающих режим разделения времени и обеспечивающих совместное функционирование нескольких процессов.

Для получения указанной оценки мы поступили следующим образом: в процессе тестирования одна из программ, реализующих перечисленные выше методики, вызывалась в асинхронном режиме пять раз подряд, в результате чего порождалось пять процессов, каждому из которых передавалась для выполнения одна и та же программа. Таким образом, в операционной среде одновременно функционировали пять активных процессов, каждый из которых являлся копией другого. Сравнивая

Методика тестирования	Программно-аппаратный комплекс							
	AT286	AT386-2MB	AT386-8MB	PS32	MC68020	VAX750	VAX780	VAX785
Время выполнения программы тестирования, с								
1*memcpu		25.3	25.6	337.0	55.2	210.7	123.0	68.3
5*memcpu		222.9	121.8	1671.2	269.9	1052.1	776.4	340.3
1*memdisk		54.8	17.4	214.0	32.8	61.1	194.0	27.9
5*memdisk		461.4	195.0	1954.4	235.8	270.6	802.2	246.3
1*(memcpu и memdisk)		32.4	33.4	433.0	68.0	253.0	168.0	85.3
и memdisk)		59.0	22.8	212.0	38.9	92.2	110.0	44.8
5*(memcpu и memdisk)		370.8	226.6	2281.2	361.4	1297.9	1032.0	442.0
и memdisk)		633.1	181.0	2101.2	322.8	425.6	788.0	238.0
Sorting 1Mb	1983.9	426.2	93.0		133.1	308.7	357.0	119.3
Время компиляции исходного текста программы, с								
mkfile	19.4	9.5	8.9	26.0	9.9	16.2	15.0	5.4
memcpu	23.7	6.2	3.7	35.0	12.4	15.1	19.0	5.4
memdisk	27.7	7.4	4.7	39.0	17.0	15.3	14.0	7.4
Sorting 1Mb	123.5	34.0	31.7	193.0	55.8	148.8	131.0	60.7

операция повторяется 100 000 раз. Время выполнения описанной программы тестирования и принимается за искомую совокупную оценку.

mkfile - традиционная для UNIX-совместимых ОС методика, позволяющая получить совокупную оценку эффективности файловой системы. Программная реализация этой методики чрезвычайно проста: она заключается в определении затрат вре-

Следует подчеркнуть, что программная реализация каждой методики представляет собой самостоятельную программу. Действительно, поскольку речь идет об ОС, поддерживающих режим разделения времени, целесообразно попытаться оценить это принципиальное отличие UNIX-совместимых ОС от однопользовательских однопрограммных ОС, например таких, как MS-DOS.

Простейшая методика оценки эф-

времени функционирования одного процесса и его пяти копий, можно получить представление об эффективности механизма разделения операционной системой вычислительных ресурсов.

Еще одной чрезвычайно важной совокупной оценкой эффективности вычислительной системы, является время компиляции исходного текста программы. Результаты его измерения также приведены в таблице.

КОНФИГУРАЦИЯ ТЕСТИРУЕМЫХ КОМПЛЕКСОВ

AT286

Компьютер IBM PC/AT на базе микропроцессора 80286, 10 МГц

ОЗУ емкостью 1 Мбайт, 12 МГц

Один диск типа ST251 (среднее время доступа 28 мс)

Операционная система SCO Xenix 2.1.3

AT386-2MB

Компьютер IBM PC/AT на базе микропроцессора 80386, 20 МГц

ОЗУ емкостью 2 Мбайт, 10 МГц (без состояния ожидания)

Один диск типа ST4096 (среднее время доступа 28 мс)

Операционная система SCO Xenix 2.3.1

AT386-8MB

Компьютер IBM PC/AT на базе микропроцессора 80386, 20 МГц

ОЗУ емкостью 8 Мбайт, 10 МГц (без состояния ожидания)

Два диска типа Maxtor XT1140 (среднее время доступа 16 мс)

Операционная система SCO Xenix 2.3.1

PS32

Супермикрокомпьютер фирмы Aniral UTEC (Франция) на базе микропроцессора MC68010, 10 МГц

ОЗУ емкостью 4 Мбайт, 12 МГц

Один диск типа ST251 (среднее время доступа 28 мс)

Операционная система QUIX 1.0 (разработка СП "ИНТЕРКВАДРО"), совместимая с ОС UNIX v.7

БЕСТА

Супермикрокомпьютер БЕСТА (разработка ЗИЛ) на базе микропроцессора MC68020, 16 МГц, шина VME

ОЗУ емкостью 8 Мбайт, 8 МГц

Один диск емкостью 320 Мбайт (среднее время доступа 28 мс)

Операционная система System V.3

VAX750

Компьютер VAX 11/750

ОЗУ емкостью 3 Мбайт

Два диска типа RA81

Операционная система UNIX BSD 4.1

VAX780

Компьютер VAX 11/780

ОЗУ емкостью 4 Мбайт

Четыре диска типа CDC 9766 (аналог RM05)

Операционная система UNIX BSD 4.1

VAX785

Компьютер VAX 11/785

ОЗУ емкостью 16 Мбайт

Один диск типа RA81

Операционная система Ultrix 1.0

ВОКРУГ САПР

А.И. МАСАЛОВИЧ



P-CAD

для любых плат

Если сравнивать системы автоматизированного проектирования (САПР) многослойных печатных плат (МПП) по уровню воодушевления, с которым о них отзываются пользователи, то одно из первых мест, вероятно, займет система P-CAD фирмы Personal CAD Systems (США). Это не означает, что система лишена недостатков, однако сложившийся круг ее сторонников с завидной изобретательностью находит способы их преодоления. Энтузиазм поклонников этой системы во многом оправдан: P-CAD действительно можно отнести к числу наиболее мощных и развитых САПР МПП.

Почему пользователи выбирают P-CAD?

Взяв в руки какой-либо каталог САПР МПП, можно убедиться, что набор предлагаемых систем чрезвычайно широк, а сами они весьма различаются как по возможностям, так и по ценам. Здесь можно найти и пакеты для ручного проектирования ценой до 500 дол (например, пакет Tango PCB фирмы ACCEL Technologies), и мощные полностью автоматические системы (такие, как Dash-PCB фирмы Future Net), цены на которые превышают 20 000 дол.

Такое разнообразие вызвано тем, что в соответствии с особенностями разрабатываемых МПП разные группы пользователей предъявляют к системам проектирования различ-

ные требования. Существуют ли системы, возможности и стоимость которых могли бы устроить большую часть потенциальных пользователей? Да, такие системы есть, и если провести опрос достаточно широкого круга советских разработчиков МПП, то одной из первых будет названа система P-CAD. Чем же вызвана широкая популярность этой системы у наших пользователей?

Во-первых, она позволяет проектировать МПП, параметры которых соответствуют возможностям отечественной технологии.

Во-вторых, система содержит большой набор драйверов периферийных устройств, что позволяет использовать ее на персональных компьютерах (ПК) с различными конфигурациями оборудования.

В-третьих, программы, входящие в состав системы, работают с общей интегрированной базой данных, в которой хранятся описания проектируемой схемы. Благодаря этому систему можно постоянно обновлять путем замены отдельных программ новыми версиями.

В-четвертых, поддерживаемая системой методология проектирования позволяет использовать отдельные программы и подсистемы как при проектировании МПП, так и при разработке полужаказных интегральных схем, что в настоящее время становится все более актуальным.

И наконец, широко распространенное среди программистов так называемое "несанкционированное

создание копий" и отсутствие эффективных барьеров для предотвращения этого приводят к резкому увеличению числа пользователей любой популярной системы, в том числе и системы P-CAD.

Состав системы

Прежде чем рассматривать состав системы P-CAD, необходимо разобраться с ее названием. P-CAD - это название первых программных систем, разработанных фирмой Personal CAD Systems. Так, в 1985-1987 гг. широкое распространение получили версии 2.0 и 3.0 системы P-CAD. Более поздние разработки данной фирмы имеют другие названия - Master Designer, Associate Designer и др., однако пользователи продолжают называть их версиями P-CAD, присваивая им новые номера по своему разумению. Например, систему Master Designer нередко называют "P-CAD версии 4.0". Кроме того, отдельно поставляемые программы и подсистемы пользователи также называют "P-CAD". Особенно часто подобные недоразумения происходят с системами PCB-2 и PCB-3, предназначенными для конструкторского проектирования МПП. В данной статье мы рассмотрим не только систему P-CAD в ее "чистом" виде, но и другие разработки фирмы.

Общая структура системы P-CAD версии 3.0 представлена на рисунке. Входящие в систему программы выполняют следующие функции:

PC-CAPS - создание и редактирование принципиальных схем для МПП и больших интегральных схем (БИС);

PC-NODES - формирование цепей;

PC-LOGS - интерактивное логическое моделирование;

PC-PACK - пополнение базы данных данными, представленными в различных форматах;

PC-LINK - объединение принципиальных схем;

PREPACK - предварительная обработка библиотечных файлов;

PC-CARDS - интерактивное проектирование МПП;

PC-PLACE - размещение элементов;

PC-ROUTE - трассировка соединений;

PC-BACK - дополнение базы данных для этапа логического проектирования по результатам этапа конструкторского проектирования;

PC-PRINT - вывод конструкторской документации на принтер;

PC-PLOTS - вывод конструкторской документации на графопостроитель;

PC-PHOTO - формирование управляющих лент для координатографа.

Кроме того, в состав системы входит несколько вспомогательных программ.

Исключив из рисунка программу интерактивного логического моделирования PC-LOGS, мы получим конфигурацию системы конструкторского проектирования PCB-3. Программы PC-LOGS, PC-CAPS, PC-NODES, PC-LINK, PC-PRINT, PC-PLOTS вместе с программой PC-FORM образуют систему логического проектирования CAE-2.

Новые разработки

Странные и запутанные маршруты, которыми программное обеспечение зачастую попадает к советским пользователям, не гарантируют своевременного получения ими новых качественных разработок. Как правило, к моменту достаточно широкого распространения и освоения пользователями какого-

либо пакета он успевает устареть, а разработавшая его фирма выходит на рынок с несколькими новыми изданиями. Поэтому пользователям системы P-CAD, вероятно, будет интересно узнать, какие новые разработки фирмы Personal CAD Systems представлены на рынке программного обеспечения в начале 1990 г.

Прежде всего, фирма предлагает две интегрированные системы проектирования печатных плат - Master Designer и Associate Designer. Обе системы ориентированы на ПК IBM PC/XT и IBM PC/AT и обеспечивают сквозное проектирование двусторонних и многослойных печатных плат, рассчитанных как на традиционную технологию, так и на технологию поверхностного монтажа. В обеих системах предусмотрен непосредственный контроль проектных норм. Системы различаются широтой реализуемых возможностей и диапазонами допустимых значений параметров.

Система Associate Designer проще и дешевле, чем Master Designer. Размещение компонентов с помощью этой системы выполняется вручную, а трассировка - в автоинтерактивном режиме. Средства оптимизации проекта в системе практически не предусмотрены, в частности не реализована возможность минимизации числа послойных переходов. Система позволяет проектировать платы размером до 50 x 50 дюймов с числом слоев не больше 50, содержащие до 500 компонентов и до 1000 логических цепей.

Система Master Designer обеспечивает как интерактивное, так и полностью автоматическое выполнение процессов размещения и трассировки. Она имеет некоторые принципиальные отличия от разработанных ранее систем, поэтому ниже будет рассмотрена отдельно.

Фирма предлагает также несколько пакетов программ, выполняющих отдельные стадии проектирования.

Для автоматизированной разработки схем предназначены пакеты PC-CAPS и Master Schematic. Включение новых средств управления данными в последнюю версию пакета PC-CAPS позволило вдвое увели-

чить скорость графической обработки логических схем высокой степени интеграции.

По сравнению с PC-CAPS возможности пакета Master Schematic расширены: предусмотрены ввод и вывод данных, представленных в формате EDIF, проверка дублирования имен цепей в подсхемах, ведение интегрированной базы данных. Благодаря широте предоставляемых возможностей оба пакета охотно используют как разработчики печатных плат, так и разработчики БИС.

Пакет PSpice расширяет возможности пакета PC-CAPS, позволяя выполнять моделирование как цифровых, так и аналоговых схем. В пакет включены средства статистического анализа, что дает возможность оценивать реализуемость схем и прогнозировать соблюдение температурных ограничений. Пакет интересен разработчикам БИС, поскольку гибкость заложенных в нем алгоритмов делает его пригодным для моделирования БИС всех типов.

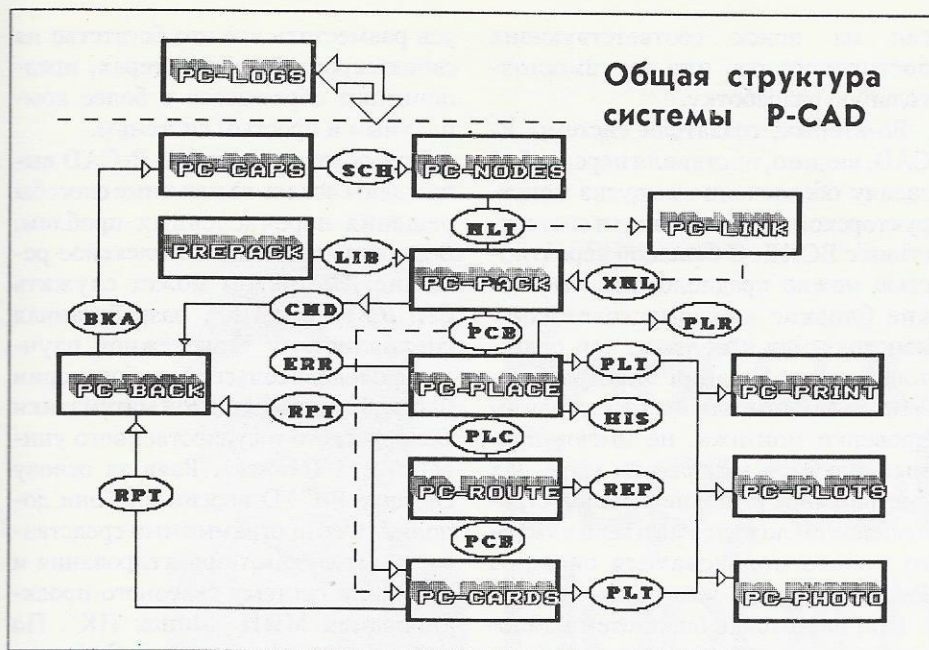
Пакет PC-LOGS предназначен для логического моделирования схем, содержащих до 500 и до 1200 логических вентилей, при использовании ПК IBM PC/XT и IBM PC/AT соответственно.

Кроме того, фирма отдельно представляет программы Master Router и Rip Up Writer для трассировки соединений, а также программу PC-Gerber для редактирования файлов в формате Gerber и программу G-Plot для получения контрольных изображений платы.

Подводя итог сказанному, можно сделать вывод, что при комплексном подходе к решению организационных и технических задач системы фирмы Personal CAD Systems могут быть с успехом применены для проектирования крупноформатных прецизионных МПП.

Master Designer II - приближение к возможностям САПР на рабочих станциях

САПР на базе IBM-совместимых ПК позволяют успешно проектиро-



23460 дол. В комплекте с графическим сопроцессором пакет стоит 30000 дол. Это почти вдвое превышает типичную для таких систем цену, однако все же значительно ниже уровня рыночных цен САПР на 32-разрядных рабочих станциях.

PDIF - средство для обмена данными между САПР

У пользователей часто появляется необходимость в переносе проектных данных из одной системы в другую. Поэтому каждая крупная современная система проектирования в той или иной форме содержит средства для обмена проектными данными с другими системами. В системе P-CAD для этой цели используется формат PDIF (P-CAD Database Interchange Format). Файл данных в формате PDIF состоит из двух основных разделов: SYMBOL - общее описание проекта или его части, и DETAIL - описание внутренней структуры проекта. В разделе SYMBOL содержатся описания контактов, параметров изображения, параметров конструкции и другие атрибуты. Основным содержимым раздела DETAIL является набор описаний логических цепей и компонентов схемы. Синтаксически файл в формате PDIF строится в виде последовательности ключевых слов, снабженных числовыми или символьными параметрами.

В состав системы P-CAD входят также программы PDIFIN и PDIFOUT, выполняющие преобразование данных из формата базы данных системы в формат PDIF и обратное преобразование.

PDIF - достаточно мощное и хорошо отлаженное средство для обмена проектными данными между различными САПР и их компонентами. Однако его сложность часто становится серьезным препятствием для программистов, занятых организацией информационного взаимодействия внутри САПР. Затраты сил и времени, необходимые для разработки полного двустороннего транслятора из формата PDIF в какой-либо другой формат САПР, часто оказы-

вать печатные платы малой и средней сложности. Однако при разработке схем высокой степени интеграции возможностей ПК становится недостаточно и проектировщики предпочитают использовать 32-разрядные рабочие станции фирм Daisy Systems, Mentor Graphics и Valid Logic Systems. Стремясь привлечь на свою сторону разработчиков сложных схем, фирмы-изготовители САПР предпринимают отчаянные усилия, чтобы приблизить возможности своих систем к возможностям САПР на рабочих станциях.

Для фирмы Personal CAD Systems результатом таких усилий явилось создание пакета Master Designer II. Советским пользователям знакома версия 4 этого пакета, созданная в конце 1988 г. Пакет позволяет проектировать печатные платы, содержащие до 1300 компонентов, 2500 цепей, 100 логических слоев и 32000 выводов. Кроме того, предусмотрены автоматическая перестановка выводов, вентилях и компонентов; поворот компонентов на любой угол; задание размеров контактных площадок и толщин проводников с точностью до 1 мкм; задание шага координатной сетки. Для повышения качества размещения и улучшения трассируемости платы разработчик может воспользоваться гистограммами и "силовыми" вектора-

ми, которые строятся автоматически.

Библиотека компонентов, доступная для пакета Master Designer II, содержит описания более 6000 компонентов и в настоящее время считается одной из наиболее полных среди существующих библиотек.

Какие же резервы использованы в пакете Master Designer II для того, чтобы приблизиться к возможностям САПР на 32-разрядных рабочих станциях? Во-первых, пакет допускает использование спецификации дополнительной памяти EMS, что позволяет обойти 640-Кбайтное ограничение емкости оперативной памяти, присущее PC-DOS. Во-вторых, пакет способен поддерживать специальный графический процессор, во много раз увеличивающий скорость формирования графических изображений на экране.

Дополнением к пакету Master Designer II являются четыре программы: Master Placer (автоматическое размещение), Master Router (интерактивная трассировка), Rip Up Writer (100%-ная автоматическая трассировка) и PC-Therm (контроль температурных ограничений).

Расширенные возможности пакета Master Designer II отразились на его цене. Базовый вариант пакета стоит 8495 дол, расширенный вариант с перечисленными выше дополнениями -

ваются чрезмерными и вынуждают программистов искать другие способы включения системы P-CAD в среду САПР.

Чего не может P-CAD?

Какой бы мощной и удобной ни была система проектирования, она не избавляет пользователя от необходимости самостоятельно решать целый ряд вопросов, связанных с организацией сквозного процесса разработки печатных плат. Не является исключением и P-CAD. Какие же проблемы остаются у проектировщика, остановившего свой выбор на системе P-CAD?

Во-первых, система P-CAD (как, впрочем, и другие зарубежные системы) не имеет постпроцессоров, ориентированных на выпуск управляющих лент для советского технологического оборудования. Оборудование, поддерживаемое системой, достаточно разнообразно, однако большая его часть мало распространена в СССР. Отечественное же оборудование для выпуска печатных плат (координатографы, сверлильные станки и станции контроля), в основном ориентированное на отечественные языки управления, несовместимо с импортным оборудованием. Поэтому после приобретения системы P-CAD проектировщикам приходится тратить много времени и

сил на поиск соответствующих постпроцессоров или их самостоятельную разработку.

Во-вторых, создатели системы P-CAD, видимо, не ставили перед собой задачу обеспечения выпуска конструкторской документации в соответствии с ЕСКД. С большой вероятностью можно предположить, что такие близкие каждому советскому конструктору проблемы, как подготовка спецификации типовых элементов замены или выпуск таблицы проверки монтажа, не имеют прямых аналогов в западном мире. Их программное решение в нашей стране целиком ложится на плечи каждого нового пользователя системы P-CAD.

При подготовке библиотек компонентов много хлопот доставляет ориентация системы на использование дюймов в качестве основной единицы измерения. Собственно, система позволяет работать как с дюймами, так и с миллиметрами, однако все задействованные библиотеки должны быть приведены к выбранной пользователем единице измерения.

Как ни парадоксально, недостатком системы P-CAD иногда оказываются ее мощность и широта возможностей. Известны версии системы, занимающие (вместе со вспомогательными программами и библиотеками) 60, 80, а то и более дискет. Владельцы ПК ЕС-1841, затрудня-

ясь разместить все это богатство на своих скромных винчестерах, предпочитают обращаться к более компактным и простым системам.

Пользователи системы P-CAD вынуждены искать те или иные способы решения перечисленных проблем. Возможно даже их комплексное решение. Примером может служить САПР "Минск ПК", разработанная специалистами Молодежной научно-исследовательской лаборатории Факультета прикладной математики Белорусского государственного университета (Минск). Взяв за основу систему P-CAD версии 3.0, они дополнили ее программными средствами и методиками проектирования и получили систему сквозного проектирования МПП "Минск ПК". По сравнению с системой P-CAD она имеет следующие дополнительные возможности:

- построение принципиальных схем;
- автоматическое формирование слоев питания и земли;
- автоматическое составление документации (перечней компонентов, таблиц контактов, таблиц соединений и т.д.);
- автоматическая генерация управляющих программ для координатографов ("Минск-2005", ЕММА-85), сверлильных станков (ABL-24, MARK-6), станций контроля (Mitra).

СЛОВАРЬ

component (компонент) - базовый элемент для построения схемы. Компонентом может быть микросхема, резистор, конденсатор и т.п. Компоненты должны быть заранее описаны в библиотеке компонентов.

cutlines (линии сечения) - горизонтальные линии, используемые программой автоматического размещения для анализа пересечений цепей и близости расположения компонентов (с учетом физических параметров трасс).

force vector ("силовой" вектор) - вектор, начало которого расположено в центре компонента, а направление указывают путь наиболее предпочтительного перемещения данного компонента.

histogram (гистограмма) - графическое представление относительной плотности заполнения платы, основанное на подсчете общего числа цепей, пересекающих каждую контрольную линию.

lattice points (точки размещения) - точки на плате, в которых могут быть размещены компоненты.

merit factor (коэффициент улучшения) - число, выводимое вместе с гистограммой и показывающее, как изменится качество размещения при перемещении, повороте или взаимной перестановке компонентов.

net (цепь) - множество контактов и их соединений, обеспечивающее получение определенного электрического сигнала.

outline (контур) - линия, показывающая габариты компонента.

pin (вывод) - входной или выходной контакт компонента.

placement (размещение) - процесс расположения компонентов на поверхности платы, а также результат этого процесса.

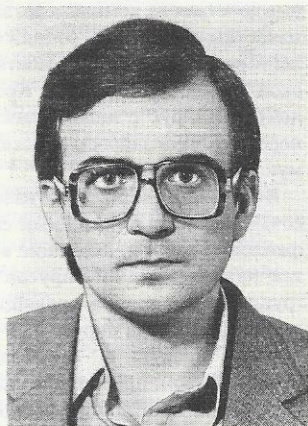
ratsnets ("крысиные норы") - графическое представление схемы в виде графа соединений контактов.

resource lines (контрольные линии) - группа вертикальных и горизонтальных линий, используемых для подсчета проходящих через них цепей и построения гистограмм.

routing (трассировка) - процесс прокладки проводников на слоях платы для реализации электрического соединения контактов, а также результат этого процесса.

routing channel (канал трассировки) - пространство, требуемое для прокладки проводников заданной толщины при обеспечении требуемого зазора между ними.

АНТИВИРУСЫ



Н.Н. БЕЗРУКОВ

Классификация Вирусов Попытка стандартизации

Актуальность проблемы защиты компьютера от вирусов обусловила необходимость организации оперативного обмена информацией и взаимодействия специалистов, работающих в данной области, что невозможно без разработки стандартной классификации компьютерных вирусов.

"Интеркомпьютер" начинает публикацию статьи, основанной на версии 3.0 первой части монографии Н.Н. Безрукова "Компьютерная вирусология" (ее предполагает выпустить в 1990 г. издательство "Наука"). Первая редакция этой работы, в которой сделана попытка систематизировать сведения о компьютерных вирусах, появилась в сентябре 1989 г. В дальнейшем текст монографии обновлялся ежемесячно. Очередные версии работы (на дискете) распространяются бесплатно на ежемесячном семинаре "Системное программирование", который проводится в Киеве под руководством автора (каждый второй четверг месяца в ауд. 4-205 Киевского института инженеров гражданской авиации, начало семинара в 15.00) и в Москве на аналогичном семинаре под руководством Л.Г. Бунича (каждый третий четверг месяца в помещении конференц-зала ЦНСБ ВАСХНИЛ, начало семинара в 11.00).

В этом номере рассматриваются принципы классификации; классификационные таблицы и пояснения к ним будут опубликованы в следующем номере "Интеркомпьютера".

Компьютерный вирус - это программа, обладающая способностью к скрытому саморазмножению в среде стандартной операционной системы компьютера с помощью включения в исполняемый код (загружаемые программы, компоненты операционной системы, пакетные файлы или компилируемый текст) своей, возможно, модифицированной копии, которая сохраняет способность к дальнейшему размножению. Такое определение является вариантом определения, данного Ф. Коуэном (F. Cohen) - автором первого серьезного математического исследования компьютерных вирусов, в статье *Computational Aspects of Computer Viruses*, опубликованной в августовском номере журнала "Computers & Security" за 1989 г. Пионером среди отечественных исследователей в этой области был А.А. Чижов, который в 1987 - 1988 гг. провел ряд экспериментов с компьютерными вирусами.

Необходимо подчеркнуть, что применение стандартной классификации существенно облегчает накопление и распространение знаний в любой области, позволяя, в частности, однозначно определять как известные вирусы и их разновидности, так и новые, только что появившиеся вирусы и их еще неисследованные разновидности (штаммы). При этом следует использовать ограниченный набор сравнительно простых и непротиворечивых признаков, для определения которых не требуется серьезный анализ зараженных про-

грамм и элементов операционной системы. Кроме того, переход к стандартной классификации облегчит пользователю выбор антивирусных средств (очень часто эти средства, разработанные программистами какой-то организации для решения конкретной проблемы, затем распространяются по всей стране, хотя и не имеют документации). В ряде случаев новые антивирусные средства разрабатываются только потому, что отсутствует информация о возможностях существующих антивирусных программ.

Если разработчики не сообщают типы вирусов, на которые ориентированы те или иные антивирусные программы, или сообщают лишь придуманные ими неформальные названия ("клички") этих программ, то пользователям приходится выяснять их применимость экспериментальным путем. Например, так называемый "Венский" вирус (С-648 по предлагаемой классификации) разработчиками программы-фага FAG_OM назван Omega, что, конечно, свидетельствует об их изобретательности, но ничего не говорит потенциальным пользователям этой антивирусной программы.

К сожалению, в настоящее время при классификации используются в основном именно такие "клички". Анализ этих названий позволил выявить три типа "кличек", учитывающих:

место обнаружения или разработки вируса

(например, вирусы Lehigh, Jerusalem, Vienna, Alameda);

содержащиеся в теле вируса текстовые строки или выдаваемые вирусом сообщения (например, вирусы Vacsina, Eddie, Dark Avenger, Disk Killer, sUMsDos);

эффект, вызываемый вирусом, (например, вирусы Time Bomb, DOS-62, Cascade, Black Friday).

При этом один и тот же вирус может иметь множество названий, и новое название вируса, использованное разработчиком той или иной антивирусной программы, далеко не всегда соответствует новому вирусу. Перечень названий многих широко известных вирусов напоминает список имен арабского шейха. Например, мне приходилось встречать не менее девяти названий вируса, обнаруженного в декабре 1987 г. в Иерусалимском университете (RCE-1813 по предлагаемой классификации), три из которых - Israeli virus ("Израильский вирус"), Jerusalem ("Иерусалим") и PLO ("ООП") - относятся к первому типу, два названия (sUMsDos и sU) - ко второму типу, и, наконец, еще четыре: Black Hole ("Черная дыра"), Black Friday ("Черная пятница"), Friday 13 ("Пятница, 13-е") и "Вирус замедления" - к третьему типу. Данный вирус "вырезает" в левом углу экрана "черную дыру", удаляет с диска все файлы, которые запускаются по пятницам, пришедшимся на 13-е число, и, кроме того, примерно через 20 мин после запуска зараженной программы искусственно замедляет работу компьютера в несколько сотен раз.

Конечно, такое многообразие названий создает определенные трудности, особенно если учесть, что рассматриваемый вирус имеет несколько штаммов, отличающихся особенно-стями функционирования.

Принципы построения классификации

Основным требованием к классификации вирусов является ее объективность - классификация должна основываться на фиксированном наборе непротиворечиво измеряемых или наблюдаемых признаков. В идеальном случае эти признаки следует выбирать так, чтобы, скажем, два разработчика антивирусных средств, независимо работающих в Киеве и Москве, использовали для одного и того же вируса одинаковое название и разные названия для разных вирусов. Это позволит быстро выявлять новые вирусы или их штаммы.

Очевидно, что объективная классификация вирусов существенно облегчит систематизацию, распространение и накопление знаний в области компьютерной вирусологии, а также

упростит выбор программных средств для борьбы с тем или иным вирусом.

Следует подчеркнуть важность не просто данной классификации как таковой, но и принятие ее в качестве *стандартной*. Уже сейчас отсутствие стандартной классификации вирусов приводит к ряду нежелательных явлений: каждый разработчик антивирусных программ использует свою уникальную классификацию вирусов, из которой часто неясно, о каком конкретно вирусе идет речь, тем более что помимо основного типа вируса обычно существует ряд его штаммов со сходными, но не идентичными свойствами;

у пользователей наблюдается тенденция аппроксимировать общее число вирусов количеством известных им антивирусных программ, и прежде всего, программ-фагов, которые, образно говоря, "выкусывают" тело вируса из зараженной программы, восстанавливая тем самым ее работоспособность в близком к первоначальному состоянию.

Такая аппроксимация приводит к существенной переоценке общего числа имеющихся компьютерных вирусов, однако человек быстро "рационализирует" этот факт, разбивая один реальный вирус на несколько "виртуальных" и приписывая каждому из них собственный набор признаков. Так, автору приходилось сталкиваться с "самодельной" классификацией, в которую вирусы C-648 и RCE-1813 входили в двух вариантах каждый, причем второму варианту вируса C-648 приписывались свойства вируса RCE-1813 (замедление работы компьютера).

В процессе чтения лекций и проведения семинаров по данной проблеме, а также в процессе работы над рукописью книги "Компьютерная вирусология" была выработана схема классификации вирусов, включающая в себя три таких основных элемента, как:

классификационный код вируса (напоминает код в классификации транзисторов);

дескриптор вируса (формализованный список его основных свойств);

сигнатура вируса (строка для контекстного поиска данного вируса в зараженной программе).

Классификационный код вируса

В предлагаемой классификации каждому вирусу присваивается классификационный код, состоящий из буквенного префикса, цифровой части (корня), которая содержит характеристику вируса, и необязательного буквенного суффикса. Например, в классификационном коде RC-1704f буквы RC - это префикс, 1704 - корень (характеристика), а f - суффикс.

Главным требованием к классификационному коду является возможность определения

большинства свойств вируса на незараженном компьютере. Выполнение каких-либо действий по исследованию вируса на зараженном компьютере является наиболее распространенной и одновременно наиболее грубой ошибкой, которую допускают неопытные пользователи. Следует подчеркнуть, что любые действия на компьютере, зараженном неизвестным вирусом, сопряжены с определенным риском вызвать срабатывание "троянской" компоненты вируса. Кроме того, резидентный вирус с целью маскировки может перехватывать запросы и искажать выдаваемую информацию.

В настоящий момент мне известен ряд вирусов, обладающих указанным свойством. Так, файловые вирусы, входящие в состав группы так называемых TP-вирусов (вирусы этой группы имеют номера, хранящиеся в предпоследнем байте кода вируса в 16-ричном виде), начиная с вируса TP-34, обладают интересным свойством: при попытке трассировать зараженную программу резидентный вирус "выкусывает" вирус из программы, "подсовывая" отладчику уже излеченную программу. Аналогично буттовые вирусы, известные под названием "Пакистанских" (Brain, Ashar и др.), при попытке просмотреть бутсектор на зараженном компьютере "подсовывают" пользователю оригинальный бутсектор, сохраненный вирусом в одном из секторов, отмеченных, как дефектные (и тем самым исключенных из распределения под файлы).

Буквенный префикс. Этот элемент классификации характеризует среду размножения вируса. Все вирусы, с этой точки зрения, можно разделить на пять основных типов:

файловые вирусы, заражающие файлы типа COM и/или EXE (префикс C, E или CE);

буттовые вирусы, или вирусы загрузчиков, заражающие бутсектор (boot sector) жесткого и гибкого дисков, т.е. загружаемый сектор, либо блок MBR - Master Boot Record (префикс B, D или M);

драйверные вирусы, заражающие драйверы устройств или запускающие себя включением в файл CONFIG.SYS дополнительной строки (префикс S);

пакетные вирусы, использующие для своего запуска язык управления заданиями операционной системы или написанные на этом языке (префикс J);

компилируемые вирусы, включающие свой код в исходный текст некоторых компилируемых программ (префикс T).

Следует упомянуть так называемые *сетевые вирусы* (точнее, *репликаторы*), обеспечивающие "рассылку" своего кода всем абонентам сети или их части. Среди них встречаются как пакетные вирусы (например, вирус Christmas Tree написан на достаточно хорошо известном в нашей стране языке управления заданиями REXX операционной системы VM - наиболее распространенной для компьютеров семейства IBM/370), так и компилируемые вирусы (например, вирус Р. Морриса, мл., имеющий компилируемую компоненту). В MS-DOS встречаются только два из упомя-

нутых выше типов вирусов: файловые и бутонные (вирусы общего назначения и вирусы загрузчиков по терминологии И.Ш. Карасика).

Буква R в префиксе показывает, что вирус является резидентным (после запуска зараженной программы такой вирус закрепляется в оперативной памяти и, перехватывая некоторые прерывания, активизируется при определенных действиях операционной системы).

Количественная характеристика. Эта часть классификационного кода вируса представляет собой достаточно просто определяемую количественную характеристику какого-либо свойства вируса; эти характеристики для большинства типов вирусов отличаются друг от друга. Например, для файловых вирусов в качестве характеристики можно использовать нормированное приращение длины файла при заражении.

Буквенный суффикс. В классификационном коде суффикс (необязательный элемент) отражает дополнительную информацию о вирусе, например наличие штаммов, неотличимых по префиксу и характеристике).

Конечно, рассмотренный выше классификационный код вируса не позволяет описать все основные свойства вируса. Однако предлагаемая систематизация свойств вирусов, надеюсь, представляет значительный интерес как для разработчиков антивирусных программ, так и для пользователей, поскольку позволяет объединять разнородные факты, относящиеся к поведению того или иного вируса, облегчая их запоминание и сопоставление.

Дескриптор вируса

В качестве второго элемента классификации выбран дескриптор вируса, отражающий систематизацию основных характеристик вируса в закодированном виде. Дескриптор вируса состоит из групп символов, начинающихся с прописной латинской буквы, за которой следуют строчные латинские буквы или цифры. При этом прописная латинская буква определяет вид дескриптора, а следующие за ней строчные буквы или цифры соответствуют значению дескриптора для конкретного вируса. Например, в дескрипторе XabYcZdmt отражены три свойства: X со значением "ab", Y со значением "c", и Z со значением "dmt".

Сигнатура

Поскольку все известные в настоящее время вирусы можно обнаружить с помощью контекстного поиска, применяя антивирусные программы-детекторы, одной из важных задач классификации является составление перечня сигнатур, т.е. списков строк для контекстного поиска. С помощью сигнатур осуществляется "детектирование" впервые используемого программного обеспечения на наличие вирусов, что позволяет повысить степень защищенности компьютера.

Хотя в дальнейшем в качестве сигнатур применяются только текстовые строки, для них

можно использовать и регулярные выражения. Последние существенно устойчивее к некоторым мутациям вирусов и, кроме того, при меньшей длине обеспечивают лучшее качество распознавания (меньшее количество ложных срабатываний). Все это делает их предпочтительнее простых текстовых строк. Версию классификационных таблиц с сигнатурами из регулярных выражений автор предполагает опубликовать несколько позднее.

Стандартизация сигнатур особенно важна для вирусов, имеющих много штаммов, поскольку формальные обозначения вируса, подобные описанным выше классификационному коду и дескриптору, обладают тем недостатком, что в заданном пространстве признаков некоторые штаммы нельзя отличить друг от друга. В то же время сравнительно легко обеспечить уникальность сигнатуры, по крайней мере, для известных типов вирусов (теоретически можно создать вирус, не имеющий ни одной сигнатуры, т.е. вирус, который принципиально нельзя обнаружить с помощью контекстного поиска).

Очевидно, что вероятность обнаружения вируса с помощью сигнатуры, соответствующей участку, содержащему команды, выше, чем у сигнатуры участка, содержащего данные, например текстовые строки (так как последние могут быть модифицированы). Поэтому выбор сигнатуры целесообразно выполнять на основе анализа реконструированного исходного текста вируса. Длина сигнатуры не должна быть слишком большой (длинную сигнатуру невозможно запомнить и труднее ввести вручную); в то же время при недостаточной длине сигнатуры или при выборе для нее нехарактерных участков кода вируса будет происходить много ложных срабатываний, что весьма нежелательно.

Строго говоря, список строк для контекстного поиска, выбранный в качестве сигнатуры, не должен содержаться ни в одной из наиболее распространенных в MS-DOS системных программах, включая, конечно, и компоненты MS-DOS. Таким образом, для выбора отвечающей указанным требованиям сигнатуры необходим ряд экспериментов, в ходе которых сами сигнатуры становятся предметом сравнения и анализа.

В настоящее время имеется ряд удачных программ-детекторов, позволяющих обнаружить вирус с помощью поиска соответствующих строк (сигнатур) в файлах; эти сигнатуры естественно "принять за основу" (например, сигнатуры, используемые в двух известных американских программах-детекторах: SCAN фирмы McAfee Associates и VIRSCAN фирмы IBM). Для определенности обозначим буквой M сигнатуру, используемую программой-детектором SCAN, а буквой I - сигнатуру, используемую программой-детектором VIRSCAN.

Необходимо отметить, что сигнатуры для ряда вирусов, распространяющихся в нашей стране (C-534, C-623, C-529 и др.) в существующих версиях этих программ отсутствуют, а сигнатуры для группы TP-вирусов неудач-

ны. В таких случаях в статье используются выбранные автором сигнатуры, которые обозначены буквой B.

Кроме того, в теле некоторых вирусов встречаются характерные текстовые строки. Такие сигнатуры, обозначенные буквой T, будем использовать как вспомогательные.

При наличии сигнатуры проверку зараженности файлов вирусом данного типа можно выполнять, применяя не только специализированные антивирусные программы (из них наиболее удачной, по-моему, является программа Virus Locator (VL), разработанная А.В. Шеховцовым, которая позволяет проводить поиск сигнатур в каталоге или заданных его ветвях), но и пакеты Norton Utilities (NU) или PCTools, благо у пользователя они всегда под рукой (в последнем случае для просмотра всех файлов можно использовать режим глобального поиска по диску). Следует отметить, что антивирусная программа Virus Locator получила первую премию по классу программ-детекторов и фагов на конкурсе бесплатных антивирусных программ, который проводился в феврале 1990 г. Киевским семинаром "Системное программирование".

Окончание статьи и классификационные таблицы будут опубликованы в следующем номере "Интеркомпьютера".

Читатели, желающие регулярно получать текущие версии монографии "Компьютерная вирусология" на дискетах, могут подписаться на электронный бюллетень СОФТПАНОРА-МА Киевского семинара "Системное программирование". По вопросам подписки обращаться по адресу: 252680, Киев-58, ГСП, Проект космонавта Комарова 1, Киевский институт инженеров гражданской авиации (КИИГА), корп. 3, Авиацентр НТТМ (тел. 484-94-63).

ФАКТЫ

■ Фирма Foresight Resources разработала версию САПР Drafix, работающую в среде Windows. - Drafix Windows CAD. Хотя этот пакет может работать только с двумерной графикой, но, выполняя большинство функций таких мощных пакетов, как AutoCAD и Cadkey, он стоит гораздо дешевле - 695 дол.

АНТИВИРУСЫ

Классификация антивирусных программ



И.Ш. КАРАСИК

В предыдущем номере "Интеркомпьютера" рассматривались вопросы, связанные с "анатомией" и "физиологией" вирусов различных типов. В этом номере я хочу познакомить вас с классификацией антивирусных программ, позволяющих защитить компьютер от проникновения вирусов или, по крайней мере, обезопасить его от их вредного воздействия.

Классификация антивирусных программ, приведенная ниже, основана на различных особенностях вирусов, проявляющихся в разные моменты их жизни. Следует подчеркнуть, что пользователю совершенно безразлично, кто испортил ему информацию: вирус или троянская программа, поэтому, говоря об антивирусных программах, я все время буду иметь в виду еще и оказание какого-то противодействия троянским программам (давно обещанный разговор о троянских программах состоится позднее).

При всей идейной простоте антивирусных средств разработка качественных антивирусных программ - дело достаточно тонкое и весьма трудоемкое. Во всяком случае, для создания хорошей антивирусной программы необходимо иметь богатую, постоянно пополняемую коллекцию вирусов, выделять действительно новые вирусы и изучать их, т.е. проводить собственные исследования в области тайного проникновения вирусов в систему.

Я уже не говорю о том, что надо знакомиться с новыми версиями операционной системы (ОС), следить за публикациями о подробностях ее функционирования и т.д.

Детекторы вирусов и фаги

Итак, что же выдает присутствие вируса в компьютере? В первую очередь, наличие ко-

дов вируса в теле некоторых исполняемых файлов либо в известных секторах жесткого или гибкого диска. На этом основано функционирование специализированных антивирусных программ-детекторов, предназначенных для обнаружения конкретных вирусов. Теоретически обнаружение вируса, заразившего файл, - дело относительно несложное. Нужно только знать, что ищешь, т.е. знать специфику конкретного вируса (ключевые поля, устойчивые коды и т.п.). Кроме этого, следует представлять механизм внедрения вируса в программу или, если это вирус загрузчика, знать, в каком месте на диске размещает он свои коды. Располагая такой информацией, можно написать антивирусную программу-детектор для обнаружения конкретного вируса. Детектор вирусов загрузчиков должен просматривать конкретные сектора жесткого или гибкого диска, детектор вирусов общего назначения - файлы типа COM или EXE по всему дереву каталогов или по его части.

Хочу отметить, что согласно последней информации об IBM-совместимых компьютерах вирусы общего назначения могут поражать не только файлы типа COM или EXE, но и оверлейные части исполняемых файлов (оверлей), а также драйверы, вставляемые в DOS во время загрузки ОС. Во всяком случае, в документации на программу SCAN49 ее разработчик Дж. Макафи (J. McAfee), известный специалист по борьбе с вирусами, упоминает о таких возможностях. Приходится верить ему

на слово, так как ни моим коллегам, ни мне такие вирусы еще не встречались. Сказанное выше означает, что даже для поиска исполняемых файлов по дереву каталогов необходимо писать специальные программы, поскольку для оверлейных частей исполняемых файлов нет стандарта на расширение (правда, в программе SCAN49 есть специальный ключ /E, с помощью которого можно задавать расширения оверлеев и драйверов).

Вообще, приведенное выше утверждение о том, что писать программы-детекторы достаточно просто, не следует понимать буквально. Прозрачна только идея, при написании же программ-детекторов для конкретных вирусов необходимо учитывать ряд особенностей, которые я и попытался описать ниже.

Первая особенность связана с выбором той части текста программы, по которой можно выявить наличие вируса. Несмотря на кажущуюся простоту, это довольно тонкий вопрос. Не стоит идентифицировать вирус по каким-то заметным цепочкам литер или кодов, и уж совсем наивно пытаться обнаружить вирус, используя для поиска видимые глазом текстовые поля сообщений - такие поля не влияют на работу вируса, и их изменить проще всего. При существующем сервисе на IBM-совместимых компьютерах внести изменения в исполняемый файл очень просто - это может сделать и разработчик вируса, и даже любой малограмотный пользователь. Программа-детектор, основанная на идентификации

Название	Разработчик	Характеристика
AIDSTEST	Д. Лозинский	Фаг. Известная мне версия распознает 21 вирус. Первая премия по классу фагов на Киевском семинаре (руководитель Н. Безруков).
ANTI-KOT	О. Котик	Фаг. Последняя версия распознает и удаляет 13 вирусов.
DIAGLOT	Г. Агасандян	Детектор. Версия DIAGLOT-12 позволяет распознавать 12 вирусов.
DOCTOR	А. Чижов	Фаг. Версия 1.52 распознает 22 вируса. Вторая премия по классу фагов на Киевском семинаре (руководитель Н. Безруков).
SCAN49	Дж. Макафи	Детектор. Версия 1.7V49 распознает 51 вирус.
VR	А. Осипенко	Фаг. Версия 3.50 распознает 20 вирусов и некоторые необычные файлы.

вируса по таким признакам, уже не узнает его, хотя вирус практически не изменился. Так и рождаются слухи о новых штаммах вирусов, вызванные, как правило, написанными "в лоб" программами-детекторами.

Ориентироваться, конечно, следует на какие-то характерные для данного вируса шаблоны, т.е. последовательности (цепочки) машинных команд. Достаточно разумным представляется искать эти последовательности в окрестности точки входа в вирус или в той части вируса (для вирусов с резидентной частью), где проверяется активность его резидентной части. В любом случае выбор цепочки машинных команд в настоящий момент - вопрос, скорее, искусства, чем науки (особенно если вспомнить, что интеллектуальные вирусы обладают свойством мутации, а также способностью к интерференции, т.е. могут в ограниченной степени менять свой код или коды каких-то других вирусов). Не забудьте, что, в конце концов, вирус - изделие рук человеческих, и при необходимости разработчик может изменить любую его часть. Правда, история знает случаи чрезвычайной устойчивости формулировок: так, после Зингера никому еще не удалось получить патент на швейную машину, иглолка которой имела бы ушко, расположенное в нижней части. Возможно, и для вирусов удастся найти такие инвариантные последовательности кодов. Во всяком случае, Дж. Макафи в документации к своей программе SCAN49 говорит именно о последовательностях команд (исследования И. Митюрина позволили выявить эти последовательности). Правильный выбор цепочек команд для поиска позволяет значительно уменьшить число шаблонов, по которым ведется поиск: несколько штаммов одного вируса могут быть идентифицированы по одной, характерной для этих штаммов цепочке.

Следующая особенность, которую мне хочется отметить, - это алгоритм поиска инвари-

антных последовательностей кодов вирусов в исполняемых файлах. При некорректной организации поиска кодов вирусов или при недостаточно жестких дополнительных проверках подозрительных программ применение детектора может дать странные результаты. Так, одна из первых диагностических программ - DIAG, разработанная польским специалистом Е. Собчиком (J. Sobczyk), утверждала, что инфицирована вирусом, хотя, конечно, это было не так. Просто из-за некорректной реализации поиска кодов вирусов программа DIAG, обнаружив в своем теле шаблон для поиска вируса, принимала его за часть кода вируса. Например, одна из версий известной антивирусной программы ANTI-KOT сообщала, что эта программа длиной менее 600 байт поражена вирусом длиной 648 байт. Организация поиска характерных кодов вирусов важна еще и потому, что программа может быть одновременно заражена несколькими вирусами или одним вирусом несколько раз.

Качественная программа-детектор, безусловно, должна правильно реагировать на такие ситуации, сообщая, какими вирусами или сколько раз инфицирован исполняемый файл.

Последняя из особенностей, связанная с реализацией качественных программ-детекторов, относится к эффективности поиска инвариантных последовательностей кодов вирусов. Детекторам довольно часто приходится просматривать значительные объемы данных; для того чтобы эта работа могла быть выполнена за приемлемое время, необходимы эффективные алгоритмы поиска инвариантных последовательностей команд.

Некоторые разработчики антивирусных программ-детекторов детально исследуют вирусы и реализуют в своих программах возможности настоящих антивирусных программ-фагов, т.е. программ, которые умеют не только обнаруживать, но и удалять из инфицированной программы код вируса, восста-

навливая тем самым исходное состояние программы (правда, не всегда точно).

На мой взгляд, важнее провести "детектирование" программ, так как при обнаружении вируса почти всегда можно отыскать неинфицированную копию исходной программы в архиве. Конечно, никто не откажется заодно и "вылечить" пораженную программу, если только лечение будет выполнено корректно. Совершенно очевидно, что это значительно более сложная задача, чем просто обнаружение вируса в программе. Я бы сказал, что по этой причине доверия к программам-фагам соответственно меньше.

В настоящее время разработано множество программ-детекторов и фагов. В таблице перечислены некоторые наиболее популярные программы-детекторы и фаги, за качество которых я могу поручиться.

В документации на антивирусную программу SCAN49 упоминается созданная фирмой IBM программа-детектор, которая по ассортименту диагностируемых вирусов и по виду цепочек команд для поиска не совпадает с программой SCAN49. Существует мнение, что цепочки команд для поиска эта программа берет из специального текстового файла, содержащего описания вирусов, и поэтому для нее добавление нового вируса к списку обнаруживаемых - тривиальная операция.

Достоинства качественных программ-де-

Возможна ли идеальная классификация?

Прежде чем начать разговор об антивирусных программах, хотелось бы отметить, что антивирусные программы делятся на универсальные (реагирующие на заражение любым вирусом) и специализированные (реагирующие на заражение лишь определенными вирусами и абсолютно нечувствительные к присутствию любых других вирусов). Идеалом, конечно, является универсальная программа, которая четко обнаруживает и надежно защищает от любого вируса.

Приведенная в статье классификация антивирусных программ не является исчерпывающей и точной.

Во-первых, постоянно появляются все более изощренные вирусы, разрабатываются все более совершенные программы противодействия им и, соответственно, размываются границы между классами антивирусных программ.

Во-вторых, после прочтения статьи вам станет ясно, что качественная антивирусная программа должна реализовывать функции антивирусных программ сразу нескольких классов (так, собственно, и происходит на деле). Наконец, разработано множество программ для проверки различных эвристических соображений. Эти программы практически не отражены в приведенной классификации.

текторов и фагов совершенно очевидны: они позволяют точно и недвусмысленно определить наличие в компьютере вируса, а в некоторых случаях и оперативно излечить инфицированные программы.

Принципиальными недостатками даже самых совершенных программ-детекторов являются: их неуниверсальность (программа-детектор рассчитана на обнаружение вируса конкретного типа) и запаздывание по отношению к появлению новых вирусов (детекторы всегда появляются с опозданием, поскольку, чтобы научить детектор распознавать вирус, нужно предварительно выделить и исследовать его).

При написании детекторов и фагов следует помнить, что эти программы, имеющие дело с самыми разнообразными опасными вирусами, должны быть защищены (по крайней мере, должны уметь обнаруживать свое заражение!), иначе они очень быстро превращаются в источник заражения вирусами. Последнее замечание относится также к архивированным программам, т.е. к программам, которые хранятся в архиве (в сжатом виде). Дело в том, что исполняемые файлы часто помещают в архивы, а после архивации, конечно, ни один детектор не может обнаружить вирус в зараженной программе. Правда, в документации на программу-детектор SCAN49 упоминаются какие-то интерфейсные программы, которые позволяют сканировать исполняемые файлы в архивированном виде.

Программы-вакцины

Этот класс универсальных антивирусных программ применяется только для борьбы с вирусами общего назначения.

Учитывая опасность вирусного заражения, можно позаботиться о защите своей программы еще во время ее написания, встроив в нее некоторую диагностическую часть, реагирующую на внедрение вируса. Можно, например, запомнить в каком-нибудь месте программы длину некоторых исполняемых модулей или последовательность машинных кодов в окрестности точки входа в программу; вычислить и запомнить какой-нибудь вариант контрольной суммы программы и т.д. В процессе работы ваша программа должна проверять все эти характеристики, определяемые при создании исполняемого файла. Несовпадение текущих и запомненных (эталонных) значений этих характеристик свидетельствует о несанкционированном изменении программы, а это прямое указание на ее инфицирование. Не стоит, однако, думать о возможности вирусного заражения при написании любой программы; существует, наконец, множество уже написанных программ, которые никто и не предполагал защищать.

Можно ли защитить ранее написанные программы? Конечно, можно. Мне известны, по крайней мере, три отечественные программы-вакцины: STAMPER А. Чижова, PROTECT Д. Стефанкова и СТРАЖ Ф. Шерстюка.

Принцип действия программы-вакцины состоит в том, что она внедряется в защищаемую программу подобно вирусу общего назначения и запоминает перечисленные выше характеристики последней. При запуске защищенной, но не зараженной программы управление получает программа-вакцина, проверяющая состояние защищенной программы примерно так же, как это сделал бы программист.

Допустим, что после установки защиты исполняемую программу заражает какой-ни-

Не только обнаружение, но и борьба

По моему твердому убеждению, самое важное требование пользователя к антивирусной программе — быстро и надежно установить факт заражения компьютера вирусом. Этой информации достаточно для того, чтобы начать борьбу с незваным пришельцем. Конечно, если есть возможность использовать специально сконструированный для такой борьбы инструмент, результата можно достичь намного быстрее и с меньшими затратами, но это, по выражению математиков, не является необходимым условием.

У разработчиков антивирусных программ рано или поздно появляется соблазн использовать свои знания о вирусах для того, чтобы предоставить в распоряжение пользователя инструмент не только для обнаружения вирусов, но и для борьбы с ними. Для некоторых классов антивирусных программ это — естественное развитие их возможностей. Для программ, занимающихся поиском вирусов в исполняемых файлах, такое усовершенствование состоит в том, чтобы научить эти программы не только обнаруживать, но и удалять вирусы из зараженных файлов. Так программы-детекторы вирусов превращаются в программы-фаги. Для программ-мониторов проверка ситуаций, характерных для конкретных вирусов, пожалуй, тоже оправдана. Возможно, что при этом программа-монитор в какой-то степени теряет свою универсальность, но приобретает при этом дополнительный «запас прочности».

Для программ других классов такое расширение поля их деятельности едва ли разумно: программа разбухает, становится аморфной и неудобной в употреблении.

будь вирус. Как работает такая дважды модифицированная программа? Понятно, что сначала управление получает вирус, который делает свое «черное» дело, и помешать этому программа-вакцина, конечно, не может. После завершения инициализации вирус должен передать управление исполняемой программой; на самом же деле в процессе установки

защиты исполняемая программа была так изменена, что управление получает программа-вакцина. Она проверяет текущее состояние исполняемой программы, которое, естественно, отличается от эталонного (изменения внес вирус при инфицировании этой программы), и немедленно поднимает тревогу, обнаружив отличие. В некоторых случаях, например, если повреждения программы при ее инфицировании не слишком велики или объем запомненной программой-вакциной информации достаточно велик, программа-вакцина может «вылечить» зараженную программу (примерами могут быть такие вакцины, как СТРАЖ, PROTECT и т.п.).

Вирусы загрузчиков заражают не исполняемые файлы, а файловую систему в целом, поэтому для борьбы с ними изложенный выше способ защиты использовать непосредственно нельзя, однако в модифицированном виде он годится и для этой цели.

Достоинство программ-вакцин состоит в том, что с их помощью можно защитить любые программы, а также их оверлейные части от всех вирусов общего назначения, в том числе и от еще не разработанных вирусов.

Недостатки рассмотренного выше способа борьбы тоже понятны:

- программы-вакцины не могут обнаружить заражение программы, если оно произошло до момента включения защиты;
- длина защищенной программы возрастает;
- время загрузки каждой защищенной программы увеличивается;
- вирус, зная «устройство» программы-вакцины, может обойти защиту;
- программы-вакцины не помогают в случае троянских программ.

Программы слежения за состоянием файловой системы

Программы этого класса, очень похожие на программы-вакцины, описанные выше, реагируют на попытки заражения исполняемых файлов, следя за их состоянием: запоминают длину и контрольные суммы избранных исполняемых файлов, дату их создания и другие характеристики. Отличие рассматриваемых программ от программ-вакцин состоит в том, что характеристики исполняемых файлов запоминаются в отдельных файлах. При этом длины исполняемых файлов не увеличиваются, и процесс слежения за их состоянием для вируса остается незамеченным, т.е. вирус в принципе не может узнать о том, что состояние каких-то файлов контролируется, и поэтому не может имитировать запомненное состояние.

Программы слежения за состоянием файловой системы универсальны, т.е. реагируют на попытку заражения любым вирусом; отслеживания различных версий вирусов не требу-

Название	Разработчик	Характеристика
CRCDOS	Р. Фейт (R. Faith)	Работает в двух режимах: сбор информации о файлах, перечисленных в списке, и проверка актуальной информации на совпадение с запомненной в момент ее сбора.
Vaccine 1.3	Фирма Art Hill	Запоминает информацию о системных файлах, создает каталог с копиями этих файлов под странными именами.
DLI	В. Герасимов	При первом запуске запоминает состояния файлов на логическом диске; при повторном запуске проверяет совпадение текущего и запомненного состояний файлов, сообщая об изменениях, уничтожении и создании новых файлов.

ется. Эти программы можно модифицировать так, чтобы они обнаруживали и заражения вирусами загрузчиков.

Однако при всей своей привлекательности полностью решить проблему борьбы с вирусами эти программы не могут, так как обладают рядом недостатков:

1. Эффективность программ слежения зависит от частоты их запуска - если программой слежения пользоваться раз в год, лучше этого не делать вообще.

Регулярно приходится тратить значительное время на просмотр файловой системы (затраты времени зависят от объема информации, записанной на ваших дисках, числа просматриваемых файлов, характера проводимых программой слежения проверок и их качества). Правда, это проблемы решаемые - запускает же операционная система UNIX при каждой перезагрузке программу FSCK, предназначенную для просмотра и "ремонта" файловой системы, тем более, что реально нужно следить за состоянием только наиболее часто используемых программ.

2. Программы слежения за состоянием файловой системы большого практического значения не имеют, так как они бессильны против троянских программ, разрушающих файловую систему (когда файловая система сильно разрушена, следить, пожалуй, уже не за чем).

3. Программы слежения, подобно программам-вакцинам, не в состоянии обнаружить заражение программы, если оно произошло до момента включения защиты, т.е. они не реагируют на уже зараженные программы.

4. Применение программ слежения требует от пользователя определенной компьютерной грамотности.

К настоящему времени разработано множество программ слежения за состоянием системных файлов (возможности этих программ по отслеживанию и даже восстановлению системных файлов патологически гипертрофированы, но, к сожалению, ничего, кроме слежения за состоянием этих файлов, подобные программы делать не умеют). Наличие такого

большого числа программ слежения за состоянием системных файлов тем более странно, что пока известен лишь один специализированный вирус - Lechigh, поражающий только системные файлы.

В таблице приведены программы слежения, с которыми мне приходилось часто работать.

Программы-мониторы

В настоящее время можно, наверное, говорить как о классических программах-мониторах, которые являются универсальными анти-вирусными программами и позволяют в какой-то степени активно противодействовать вирусам и троянским программам, так и о более совершенных программах-мониторах, которые, частично утратив свойства универсальности, позволяют более эффективно (избирательно) противостоять вирусам.

Классическая программа-монитор старается заблокировать не только процесс размножения (репликации) вирусов, который является неотъемлемой частью жизни вирусов, но и вообще любые опасные, по мнению программы-монитора, попытки доступа к жесткому или гибкому диску. Такая программа-монитор пытается перлюстрировать запросы на доступ к дискам: если она считает, что запросы на модификацию данных опасны, то поднимает тревогу и при отсутствии разрешения на модификацию блокирует исполнение запросов. Для проверки правильности запросов на изменение данных эти программы используют различные алгоритмы, отслеживая попутно те или иные события, имеющие, возможно, отношение к деятельности вирусов.

Следует отметить, что программам этого класса присущи два недостатка, сильно снижающих их ценность:

1. Наличие на компьютерах BIOS и возможность непосредственно выдавать команды ввода-вывода в DOS делают надежды на качественную защиту дисков довольно призрачными. Я готов поверить в то, что вирус не будет

выдавать команды ввода-вывода из соображений экономии длины кода, но почему бы ему не передавать управление непосредственно в точку входа прерывания int 13? Найти эту точку - не такая уж неразрешимая задача; обнаружить же и, тем более, заблокировать прямую передачу управления в BIOS в незащищенном режиме (real mode) невозможно.

2. Действия, которые программы-мониторы должны отслеживать и блокировать, по существу не являются криминальными. Многие прикладные программы при обращении к данным на дисках используют те же приемы, что и вирусы. Например, в редакторе Brief при модификации файлов часто применяется доступ по абсолютному адресу сектора. Поэтому на программы-мониторы обрушивается целый "шквал" запросов ввода-вывода, среди которых трудно выделить запросы, действительно обусловленные деятельностью вирусов. Дело осложняется тем, что вирусы, как правило, не пользуются штатными средствами DOS, а модифицируют управляющие блоки непосредственно. Аппаратные же средства защиты памяти в DOS не работают, поэтому программы-мониторы, чтобы не пропустить момент запуска вируса, вынуждены обеспечивать защиту постоянно, а не только в моменты реальной опасности.

Хорошая программа-монитор обеспечивает защиту дисков на разных уровнях:

- на системном уровне работы с файлами (при их создании, открытии, позиционировании, чтении и записи) с помощью блока FCB (File Control Block) или с помощью доступа к файлам (File Handle), подобного осуществляемому при использовании ОС UNIX;
- на уровне доступа к секторам по абсолютным адресам и на уровне физической адресации (при выборе номера цилиндра, дорожки и сектора).

Известно, что один системный запрос вызывает серию запросов на уровне физической адресации, и программе-монитору приходится разбираться, чем вызваны эти запросы на доступ: действительно системными запросами или "творчеством" вируса либо троянской программы. Выше было показано, что сделать это корректно не представляется возможным, поэтому программы-мониторы часто поднимают тревогу в совершенно безобидных ситуациях. Это не только создает неприятности для пользователя, но быстро приводит к тому, что у него вырабатывается "соглашательский" стереотип, в результате чего применение программы-монитора становится практически бессмысленным.

Следует подчеркнуть, что несмотря на такие серьезные недостатки, программы-мониторы позволяют активно бороться и с вирусами (почти с момента их появления в компьютере, блокируя процесс репродукции), и с троянскими программами (в том случае, если они начинают уж очень существенно разрушать файловую систему).

Ниже в качестве примера перечислены не-

которые наиболее популярные программы-мониторы:

- VIRBLK - простая программа-монитор, которая поднимает тревогу при попытке изменить или создать файл типа COM или EXE;
- ANTI4US - более сложная программа-монитор, отслеживающая разные события и обладающая возможностью перенастройки в процессе работы;
- FSP - наиболее совершенная программа-монитор, проверяющая законность доступа к файлам на всех уровнях; эта программа минимально "шумит", отслеживая различные "полезные" события, например изменения в КМОП-памяти. Имеется возможность настроить программу FSP на защиту определенных логических частей диска и осуществить таким образом отслеживание состояния файловой системы (при своем старте или при запуске избранных исполняемых файлов программа FSP проверяет контрольные суммы этих файлов; можно указать и имена исполняемых файлов, и способ проверки контрольной суммы). К сожалению, версия 1.41 программы FSP и версия 2.0 пакета Norton Commander сильно взаимодействуют друг с другом.

К программам-мониторам тесно примыкает ряд других программ, разрабатывавшихся не для борьбы с вирусами, а для других целей, но неожиданно оказавшихся подспорьем в борьбе с вирусами. К числу таких программ относятся программы обслуживания диска, разделения доступа (например, ADM, WATCHDOG), а также некоторые программы кэширования диска, позволяющие защищать от записи какие-то из обслуживаемых ими логических устройств (например, программа SC, разработанная фирмой MAXTOR). В ряде случаев программы обслуживания или кэширования диска позволяют получить лучшие по сравнению с обеспечиваемыми программами-мониторами результаты в борьбе с вирусами хотя бы потому, что, в отличие от последних, реализуют какие-то варианты защиты с помощью пароля, а не простое подтверждение запроса. Программы обслуживания диска, часто встраиваемые в операционную систему с помощью драйверов, имеют больше возможностей для корректного разделения запросов на *правильные* (обусловленные нормальным функционированием системы) и *вирусные* (связанные с наличием вирусов). Однако без использования режима защиты процессора (protection mode) ни программы кэширования, ни программы обслуживания диска, ни пакеты разделения доступа, ни даже программы-мониторы не обеспечивают надежной защиты компьютера от вредного воздействия вирусов.

Современные программы-мониторы отслеживают более специфические события, обусловленные действиями вирусов, чем классические программы-мониторы. Поэтому для них не характерен высокий уровень ложных тревог, являющийся бичем классических мониторов.

Например, SCANRES - программа-монитор, разработанная тем же Макафи, сканирует исполняемый файл в процессе загрузки, пытаясь обнаружить в этом файле вирус. Конечно, универсальность у такого монитора полностью потеряна, и в смысле вирусной специализации эта программа-монитор эквивалентна программе-детектору (точнее, это резидентный детектор); от монитора, однако, осталась оперативность реагирования.

Программа-монитор ANTI-KOR, разработанная О. Котиком, построена исходя из эв-

Вниманию пользователей пакетов Lotus, Framework, Brief, Multi-Edit и других пакетов прикладных программ, имеющих достаточно мощный аппарат макрокоманд и допускающих создание пользовательских макросов!

В августовском номере журнала "Computers & Security" за 1989 г. один из его редакторов Х. Дж. Хайленд (H. J. Highland) в рубрике *Random Bits & Bytes* описал душераздирающую историю конструирования макровируса в среде Lotus. Насколько я понял, автор рассматривает все-таки не вирус, поскольку о размножении вредной макрокоманды речь не идет, а троянскую (в соответствии с принятой нами терминологией) макрокоманду - макрос; он утверждает, что "аналогичные приемы можно использовать для создания троянских макрокоманд в любой другой программе, позволяющей пользователю писать свои макросы".

Я хочу обратить внимание читателей на то, что такие троянские макрокоманды очень специфичны. Вероятно, для обнаружения этих макрокоманд ни одна из ранее разработанных программ-детекторов не годится, так как проявляют они себя только во время работы в той программной среде, для которой написаны. Тот способ обнаружения троянских макросов, который предлагает Х. Дж. Хайленд, специфичен для пакета программ Lotus. Кстати, этот способ похож на те простые способы обнаружения вирусов, которые описаны в статье об устройстве вирусов (см. "Интеркомпьютер", 1989, N 2). Во всяком случае, будьте внимательны и обращайтесь внимание на любые несуразности, которые могут возникать при работе с данными. Помните, что троянские программы появились уже и в среде прикладных пакетов.

В книге *Computer Viruses: A High-Tech Decease* Р. Бепрепа (R. Burger) рассмотрен вирус, написанный на языке пакетных файлов DOS.

Результаты серьезного математического исследования вирусов, выполненного знаменитым Ф. Коуэном (F. Cohen), приведены в статье *Computational Aspects of Computer Viruses*, опубликованной в том же августовском номере журнала "Computers & Security".

ристических соображений: после завершения исполнения каждой программы монитор просматривает таблицу векторов прерываний и состояние блоков управления памятью. По мнению разработчика, вирус общего назначения с резидентной частью или вирус загрузчика внесет какие-то изменения в распределение векторов прерываний и памяти. Рассматриваемая программа-монитор сохраняет свойство универсальности по отношению к вирусам с резидентной частью и вирусам загрузчиков, но не реагирует на вирусы без резидентной части и троянские программы. Следует отметить, что это относительно новая программа, поэтому информация об ее свойствах носит предварительный характер.

Антивирусные программы SBM (разработчики Б. Мостовой и В. Еременко) и CHECK21 (разработчик В. Двоглазов) - призеры по классу мониторов Киевского семинара (руководитель Н. Безруков). Эти программы, используя изысканные приемы работы с прерываниями во времени, не позволяют вирусам перехватывать прерывание int 21 и в большинстве случаев успешно подавляют репликационную активность вирусов (к сожалению, подобные их действия не распространяются на троянские программы).

Мне хотелось бы предложить свой вариант программы-монитора, которая годится для борьбы с вирусами общего назначения, имеющими резидентную часть. Особенность этих вирусов состоит в том, что во время старта они вынуждены проверять, активна ли уже их резидентная часть. Программа-монитор, эмулирующая правильный ответ на запрос вируса об активности его резидентной части, защищает компьютер не хуже программы-монитора, которая отслеживает запросы на доступ к диску, сообщая пользователю о "сакраментальном" запросе и ставя его в известность о попытке заражения вирусом. Образно говоря, такая программа очень точно моделирует иммунитет живых организмов по отношению к перенесенной инфекции; конечно, прежде чем научиться программе эмулировать активность резидентной части нового вируса, нужно обнаружить вирус и как-то исследовать его. В этом и состоит один из основных недостатков предлагаемой программы-монитора. Этот недостаток присущ и детекторам, однако следует помнить, что для написания программ-монитора и программы-детектора необходимо исследовать различные аспекты функционирования вируса.

Хочу подчеркнуть, что пессимизм, прозвучавший при описании программ-мониторов, не стоит понимать слишком буквально, так как до настоящего времени эти программы являются единственным средством активного противодействия вирусам или троянским программам (особенно это относится к различным не классическим, а более совершенным программам-мониторам, поскольку разработчики вирусов еще не обратили на них внимания).

При просмотре вновь полученных потенциально зараженных или, возможно, троянских программ с помощью мониторов программы с

аномальными свойствами удастся выявить еще до того, как разрушительные действия вирусов или троянских программ проявятся в полной мере; это позволяет активно противостоять натиску инфекции.

Завершая обзор антивирусных программ, мне бы хотелось упомянуть еще об одном весьма распространенном классе программ, которые, строго говоря, нельзя назвать антивирусными. Речь идет о программах, предназначенных для демонстрации содержимого оперативной памяти компьютера и определения состояния векторов прерываний. Эти программы очень полезны, а в ряде случаев просто необходимы при сражениях с вирусами, так как позволяют с разной степенью детализации (правда, иногда не совсем корректно) просмотреть, как используется оперативная память: вирусы загрузчиков и вирусы общего назначения с резидентной частью постоянно находятся в оперативной памяти, и эффективные программы, предназначенные для демонстрации содержимого памяти, позволяют быстро установить факт заражения компьютера вирусом. К настоящему времени разработа-

но множество подобных программ: MEM, MAPMEM, VTSR, MI, PCSTAT, PCMAP.

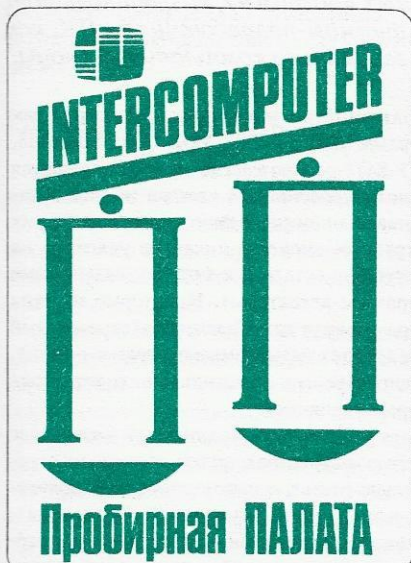
Особенность этих программ состоит в том, что вирусы могут отлично маскировать свое присутствие в памяти, и лишь одна программа из перечисленных выше - PCMAP - позволяет установить сам факт заражения компьютера вирусом загрузчика. Большинство резидентных частей кодов вирусов можно разместить в буферах DOS, где обнаружить их очень сложно. Например, вирус, инфицировавший системный драйвер, формально становится частью DOS, поэтому его весьма трудно обнаружить.

Мне хотелось бы обратить внимание читателей на программу PCMAP, разработанную Д. Стефанковым, которая подробно и с максимальной точностью демонстрирует "анатомию" использования оперативной памяти и векторов прерывания компонентами DOS и резидентными программами. Эта программа рассчитана на профессионалов, которым она предоставляет уникальные средства демонстрации содержимого оперативной памяти как для обнаружения вирусов, так и для исследования функционирования отладчиков и паке-

тов прикладных программ. Еще одной приятной особенностью программы PCMAP является "вирусоустойчивость": при старте она проверяет свою длину и длины некоторых других кодов, сообщая пользователю о факте своего заражения; в некоторых случаях эта программа способна самовосстанавливаться.

Хочу сообщить читателям, что в апрельском номере журнала "PC Magazine" за 1989 г. (с. 193 - 228) помещен достаточно подробный обзор западных антивирусных программ и пакетов. Еще больше антивирусных программ рассмотрено в книге *Computer Viruses* Р. Робертса (R. Roberts), но там информация более старая и менее подробная.

В одном из следующих номеров "Интеркомпьютера" предполагается привести список троянских программ с минимальными комментариями, обсудить способы "лечения" зараженного компьютера, а также поговорить о тривиальных, но весьма эффективных приемах "техники безопасности" при работе с зараженным компьютером.



Вниманию граждан и организаций!

«ИНТЕРКОМПЬЮТЕР»

открыл

«ПРОБИРНУЮ ПАЛАТУ» -

лабораторию тестирования программного обеспечения

Лучшие пакеты будут удостоены сертификата

«ВЫСШАЯ ПРОБА»

«ВЫСШАЯ ПРОБА» -

прекрасная реклама ваших программ

В одном из следующих номеров "Интеркомпьютера" будут опубликованы результаты тестирования антивирусных программ

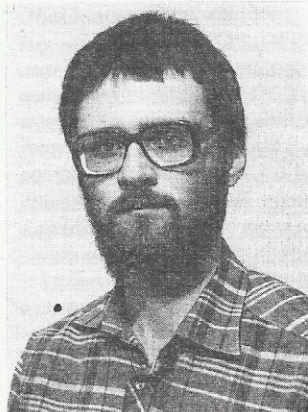
Наш адрес:

121069, Москва

ул. Чайковского, 20 а

Тел. 202-92-80

АНТИВИРУСЫ



Ф.Н. ШЕРСТЮК

Вirusы и антивирусны на IBM-совместимых ПК

Большинству пользователей персональных компьютеров (ПК) приходилось сталкиваться с компьютерными вирусами, а тот, кто с ними еще не знаком, к сожалению, скоро познакомится: вирусы все чаще поражают ПК. Происхождение их самое разнообразное: вирусы пишут хулиганствующие программисты для того, чтобы потешить свое самолюбие или заработать деньги на продаже антивирусов (антивирусных программ), разрабатывают даже в кругах, близких к таким крупным фирмам-разработчикам ПК, как IBM, в целях борьбы с фирмами, производящими компьютеры-клоны.

Несколько советов

Известно, что в программировании, как и в медицине, лучший способ не заразиться - это профилактика. Для предотвращения заражения компьютера вирусами рекомендуется следовать приведенным ниже советам.

Пользуйтесь только проверенными программами (абсолютной гарантии, конечно, никто дать не может, но программа с фирменной дискеты или программа, которой ваш хороший знакомый регулярно пользовался не менее полугода, вероятно, не инфицирована).

Избегайте "коллекционирования" программ, которыми вы не пользуетесь.

Помните: самыми распространенными источниками вирусов являются игры и антивирусные программы.

Компьютеры пользователей, которые следуют этим советам, крайне редко заражаются (компьютер применяется для реальной работы, выполняемой с помощью сложившегося инструментария, для расширения которого нужны очень веские основания; на диске нет места для "коллекционирования" ненужных программ; не хватает времени для компьютерных игр).

Необходимо иметь на защищенных от записи дискетах копии всех файлов, с которыми вы работаете, в том числе и текстовых (существуют вирусы, которые вносят не сразу

заметные изменения в текстовые файлы). Особое внимание следует обратить на резервные копии операционной системы (их должно быть не менее двух).

Если вирус все же проник в ваш компьютер, то самый простой и надежный (хотя и наиболее трудоемкий) способ избавиться от него, не "подцепив" при этом новый вирус, состоит в следующем: перезагрузите операционную систему (нажав кнопку RESET или выключив/включив питание, а не с помощью комбинации клавиш Ctrl+Alt+Del) с дискеты, содержащей незараженную копию операционной системы, отформатируйте все незащищенные диски и дискеты (существуют вирусы, которые заражают не файлы, а диски), затем восстановите на дисках всю информацию с резервных копий. Пусть этот совет также не покажется утопией - я знаю организацию, где подобная процедура проводилась на 10 компьютерах приблизительно раз в неделю (так часто там появлялись новые вирусы).

Способы "лечения"

Скорее всего, советы, изложенные выше, для вас не вполне приемлемы, и вам, вероятно, придется воспользоваться антивирусными программами. Все антивирусные программы можно разделить на специализированные и универсальные.

Большинство известных антивирусных программ (AIDSTEST, VDEATH, SERUM3, ANTI-KOT и множество других) являются специализированными: каждая из них обнаруживает наличие одного или нескольких конкретных вирусов, никак не реагируя на присутствие остальных (это так называемые программы-детекторы). Некоторые из этих программ могут даже "вылечить" зараженный файл (это так называемые программы-фаги).

Недостатками описанных антивирусных программ являются:

слишком узкая специализация: уже сейчас известно множество вирусов, и число их стремительно растет, в частности за счет появления новых разновидностей (штаммов), вызванных мутациями - изменениями, которые хулиганствующий программист может легко внести в вирус, после чего соответствующий антивирус перестанет опознавать этот вирус;

обилие ошибок из-за того, что такие программы, как правило, пишутся в спешке;

большая вероятность заражения при использовании (практика показывает, что именно специализированные антивирусные программы чаще всего являются разносчиками вирусов).

Универсальные антивирусные программы предназначены для борьбы с целыми классами вирусов. В отличие от специализированных

антивирусных программ, предназначенных для борьбы с уже исследованными вирусами, универсальные антивирусные программы можно использовать для борьбы как с известными, так и с неизвестными (еще ненаписанными) вирусами. Мне приходилось сталкиваться с универсальными антивирусами двух типов: резидентными антивирусами и программами-ревизорами.

К резидентным антивирусам относятся ANTI4US, FLUSHOT, CDM и несколько других программ, которые препятствуют "размножению" вирусов, для чего при обнаружении подозрительных действий запрашивают у пользователя их подтверждения.

Основными недостатками этих программ являются назойливость (действия, требующие подтверждения, в подавляющем большинстве случаев совершаются вовсе не вирусами, а вполне невинными программами) и низкая надежность (разработчикам вирусов известно несколько простых и весьма эффективных способов обхода таких антивирусных программ). Кроме того, резидентные антивирусные программы "съедают" много оперативной памяти и вступают в клавиатурные конфликты с другими программами.

Программа-ревизор сравнивает некоторые параметры файлов (длину, контрольную сумму и т. п.) с эталонными, что позволяет обнаружить, подвергался ли файл несанкционированным изменениям. Однако в силу того, что такая программа работает довольно долго, ревизии проводятся не чаще одного раза за сеанс работы с компьютером либо не над всеми файлами.

Печальные выводы

Из вышесказанного можно (и нужно) сделать вывод, что в настоящий момент антивирусные программы либо неэффективны, либо опасны, либо и то, и другое вместе. На мой взгляд, причины возникновения этой печальной ситуации таковы:

1. Программисты высокой квалификации уделяют разработке антивирусов недостаточное внимание. Это естественно: вирусы у них практически не заводятся; кроме того, у таких программистов есть значительно более серьезные задачи, чем написание антивирусов.

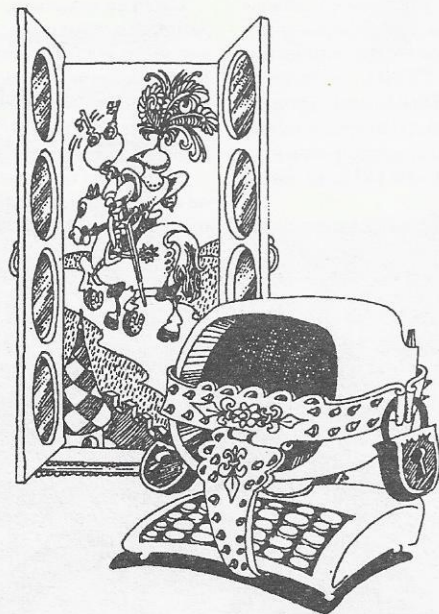
2. Центр тяжести в борьбе с вирусами в настоящее время, к сожалению, перенесен на "индивидуальный террор", т.е. на разработку специализированных антивирусных программ. Применение этого метода, проистекающего от неверия в полную победу над вирусами, с ростом числа вирусов неизбежно приведет к полному поражению. По количеству и качеству универсальные антивирусные программы в настоящее время серьезно уступают специализированным, что является серьезной методологической ошибкой.

3. Другая ошибка, регулярно допускаемая при создании универсальных антивирусных программ, состоит в непонимании того факта, что никакая антивирусная программа не должна сковывать свободу действий пользовате-

ля, поскольку она предназначена для облегчения его деятельности.

4. К сожалению, сведения об "анатомии" вирусов в настоящее время получили чрезвычайно широкое распространение (они даже публикуются в печати). Результат, на мой взгляд, может быть только один: резкий рост числа штаммов имеющихся вирусов и появления новых отечественных вирусов.

5. Некоторым людям наличие вирусов выгодно, например тем, кто продает антивирус-



ные программы или предлагает свои услуги по обеззараживанию компьютеров, либо тем, кто хочет какие-то свои огрехи списать на действия вирусов, и т.д.

Что же делать?

На мой взгляд, недопустимо упускать инициативу из своих рук, отдавая ее разработчикам вирусов. Следует подчеркнуть, что несмотря на обилие самих вирусов, идей, положенных в их основу, совсем немного, причем существенного роста числа этих идей пока не предвидится. Все имеющиеся на сегодняшний день методы борьбы с вирусами практически исчерпали себя, необходимо разработать принципиально новые. В связи с этим я хочу поделиться некоторыми соображениями.

Все известные мне вирусы можно разделить на три типа:

1. *Специализированные системные вирусы.* Эти вирусы портят системные файлы (IO.SYS, MSDOS.SYS, IBMBIO.COM, IBMDOS.COM, COMMAND.COM), не удлиняя их (наименее распространенный тип вирусов).

2. *Вирусы загрузчиков.* Они портят загрузчик и перехватывают прерывание int 13. Вирусы этого типа достаточно широко распространены.

3. *Вирусы общего назначения.* Эти вирусы

инфицируют как прикладные, так и системные программы, т.е. удлиняют файлы типа EXE и COM. Это наиболее распространенный тип вирусов.

Борьба с вирусами каждого из этих типов требует индивидуального подхода, но общая идеология разработки соответствующих антивирусов состоит в том, что если нельзя предотвратить размножение вирусов (опыт применения резидентных универсальных антивирусов показывает, что это действительно так), то не стоит и пытаться это делать, но выявить факт заражения надо на возможно более ранней стадии. Желательно также попробовать удалить вирус из зараженного файла или диска. Обе эти задачи можно решить, лишь располагая достаточной информацией о параметрах эталона - незараженного файла или диска, и достаточно часто сравнивая объект возможного заражения с эталоном. Посмотрим, как эту идеологию применить для борьбы с вирусами каждого из перечисленных выше типов.

Поскольку число системных файлов невелико, то имея их эталонные копии можно из файла AUTOEXEC.BAT запускать программу сравнения системных файлов с эталонной копией.

Вирус загрузчика перехватывает прерывание int 13 еще до начала загрузки DOS. Поэтому для обнаружения такого вируса необходимо знать, указывает вектор прерывания int 13 в момент начала загрузки на ПЗУ (в этом случае все нормально) или на ОЗУ (это означает, что вектор прерывания кем-то перехвачен - вероятно, вирусом). Для обнаружения вируса достаточно написать сравнительно простой драйвер и поместить его первым среди драйверов, расположенных в файле CONFIG.SYS. Если хранить копию загрузчика и адрес ПЗУ, на который должен указывать вектор прерывания int 13, то легко "вылечить" компьютер от вируса загрузчика.

В конец каждого файла типа EXE или COM можно дописать программу-"стража", которой в момент запуска файла передается управление; этот "страж" проверяет, не удлинился ли файл, к которому он "прицеплен", т.е. каждый файл, защищенный "стражем", в момент запуска сам проверяет, не заражен ли он. Если в теле этого "стража" хранить также контрольную сумму файла и копии его "уязвимых мест", т.е. тех областей файла, которые скорее всего будут попорчены вирусом, то, основываясь на этой информации, зараженный файл можно восстанавливать (это возможно только в том случае, когда вирус не повредил никаких мест, кроме "уязвимых").

Говоря о реализуемости изложенного выше, следует иметь в виду, что к настоящему времени разработано множество программ для сравнения длины файла с эталоном; драйвер для защиты от вирусов загрузчиков, по-видимому, написать несложно; программа-"страж", которую я уже написал, успешно эксплуатируется в нескольких организациях, "страж" во всех случаях обнаруживает факт заражения и в подавляющем большинстве случаев позволяет "вылечить" зараженный файл.

СОВЕТЫ БОРИСА АЛЕКСЕЕВИЧА

Напомним, что утилита SysInfo (SI) из пакета Norton Utilities позволяет также проверить установленный в компьютере жесткий диск (винчестер), предварительно подготовленный для работы с операционной системой DOS (ниже мы рассмотрим способы его подготовки и необходимые для этого инструменты). С помощью утилиты SI можно узнать, что винчестер работоспособен и действует быстрее (или медленнее) винчестера компьютера типа IBM PC/XT (обычно это винчестер ST225 фирмы Seagate), работающего на частоте 4,77 МГц, во сколько раз, сколько составляет значением параметра DI (Disk Index). Однако эта слишком поверхностная проверка не позволяет оценить качество винчестера, а если винчестер предварительно не инициализирован и не отформатирован для работы с операционной системой DOS, то такая проверка вообще не даст результатов.

Грубо оценить качество винчестера можно и без всяких инструментов: достаточно внимательно прислушаться к издаваемым винчестером звукам (это очень похоже на аналогичную проверку автомобильного двигателя): чистые звуки без скрежета и скрипа говорят об относительно хорошем качестве винчестера (конечно, лишь по сравнению с поврежденным дисководом).

Для детальной проверки качества винчестера разработаны специальные программы, применение которых не требует особой подготовки - достаточно знать, что контроллер винчестера служит для управления перемещением головок дисков и передачей данных, а термин "цилиндр" обозначает группу расположенных друг под другом дорожек на поверхностях жестких дисков (винчестер может состоять из нескольких дисков, собранных в пакет, причем каждый диск имеет две магнитные поверхности).

Тестирование винчестера проще всего выполнять с помощью программы SSTOR, которая входит в состав пакета SpeedStor, разработанного фирмой Storage Dimensions (или программы HARDRIVE, входящей в состав предыдущих версий того же пакета).

После вызова одной из этих программ в меню нужно выбрать режим Diagnostics, а в появившемся после его установки меню последовательно выбрать режимы Controllor, SeekTest и ReedTest. В режиме Diagnostics на экране появляется строка, содержащая такие сведения о винчестере, как его тип, емкость, число цилиндров и т.д. Наиболее полные сведения о качестве работы винчестера можно получить, установив режим MediaAnalysis, в котором магнитные поверхности дисков, образующих винчестер, тщательно анализируются и все обнаруженные на них дефекты заносятся в специальную таблицу, позволяющую в дальнейшем избежать использования дефектных дорожек. К сожалению, в автоматическом режиме пакет SpeedStor не позволяет отбраковать дорожки, которые после нескольких попыток все же удается прочитать. Однако, наблюдая за процессом тестирования, можно записать дефектные до-

Сегодня мы продолжаем разговор о проверке работоспособности персонального компьютера, начатый в предыдущем номере "Интеркомпьютера".

рожки (номера соответствующих цилиндров и головок появятся на экране в процессе тестирования) и затем занести их в ту же таблицу. Если этого не сделать, то при большом числе таких "ненадежных" дорожек быстродействие винчестера падает, что отмечается утилитой SI.

Следует отметить, что в первых трех режимах ранее записанные данные сохраняются, а в режиме MediaAnalysis при записи полученных после его установки данных ранее записанная информация стирается. Кроме проверки качества работы винчестера пакет SpeedStor позволяет подготовить его для работы с операционной системой DOS (владельцы машин семейства PS/2 фирмы IBM могут использовать пакет SpeedStor только для тестирования).

Начальная проверка работоспособности и качества винчестера вполне возможна при наличии уже упоминавшихся выше средств. Однако рано или поздно вам придется пережить ужас при обнаружении потери данных, записанных на винчестере вашего компьютера. Причин для этого более чем достаточно: появление дефектов на магнитной поверхности, вирусы, процедуры уплотнения диска, случайное удаление нужных файлов заодно с ненужными и т.д. (Поэтому не выключайте компьютер, не сделав резервных копий файлов, содержимое которых вы меняли, - такая привычка убережет вас от бессмысленной работы.)

На наше счастье, существует программа, которая в состоянии не только проверить качество работы винчестера, но и предотвратить потерю данных. По разным причинам со временем качество магнитной поверхности диска ухудшается. Проявляется такое ухудшение прежде всего в том, что некоторые попытки прочитать с диска элемент данных оказываются неудачными. Приближение катастрофы маскируется еще и схемой коррекции ошибок, применяемой как последнее средство, когда многократные (до 30-50 раз) попытки прочитать элемент данных с диска оказались неудачными и появилось сообщение

ABORT, RETRY, IGNORE, FAIL?

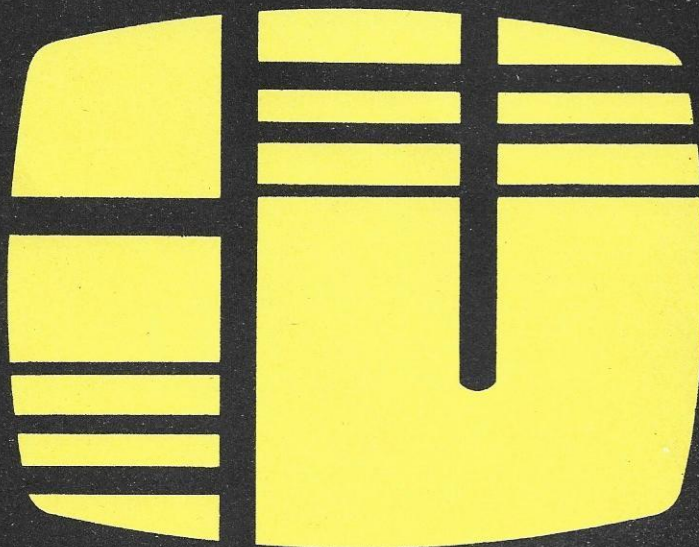
Обычно в этот момент спасти данные удастся далеко не всегда. Следовательно, чем раньше будут обнаружены сбои на диске, тем будет легче предпринять что-либо для спасения данных. Такую работу может выполнить программа

Disk Technician Advanced фирмы Prime Solutions. Любой однократный сбой при обращении к диску эта программа запоминает в специальной базе данных. Она анализирует частоту сбоев и предсказывает возможные потери данных, сообщая человеку, что нужно предпринять для предотвращения их потери. Более того, программа Disk Technician Advanced может выполнить эту работу автоматически, заново формируя ненадежную дорожку диска и перезаписывая данные в другое место.

Конечно, можно работать на компьютере, не используя специальных программ обслуживания, но в таком случае спокойная жизнь вам обеспечена только до первой аварии и потери информации.



Приобретайте, изучайте и применяйте программы обслуживания жестких дисков!



INTERUNITY

ИНТЕРКОМПЬЮТЕР познакомит вас
с программными продуктами
и техническими средствами, разрабатываемыми в СССР и за рубежом.
Вы узнаете о новостях мира вычислительной техники
и тенденциях его развития,
получите оперативную информацию
о последних разработках
и результатах тестирования
новейших программ и устройств.
Рекомендации специалистов
помогут вам сориентироваться
на компьютерном рынке
и выбрать именно те программы и устройства,
которые в наибольшей степени
подойдут для решения
стоящих перед вами задач.
ИНТЕРКОМПЬЮТЕР будет держать вас
в курсе важных событий
и расскажет о достижениях,
которые также находят отражение
на страницах ведущих компьютерных изданий
Byte, PC Magazine, PC World, Dr. Dobb's Journal, InfoWorld, PC Week.
Не менее актуальной
будет и информация о состоянии
компьютерного рынка в СССР.

СП << ИНТЕРЮНИТИ >>
121069, Москва, ул. Чайковского, 20а
Телефон: 202-92-80
Телекс: 413932 NIDEL SU
Телефакс: 2302035

An ADVERTISEMENT
in INTERCOMPUTER



will help your company
to achieve the best position
at the Soviet
computer market
and acquire partners among
thousands of enterprises
and organizations receiving
INTERCOMPUTER
at various areas
of the Soviet Union.

IT MIGHT BE WISE
TO MAKE A STEP
INTO THE FUTURE NOW!

PLACING
an ADVERTISEMENT
in INTERCOMPUTER
IS a PROSPECTIVE
INVESTMENT.

All questions, insertion orders,
and ad materials
should be directed to:
INTERUNITY Joint Venture
USSR, 121069, Moscow
ul. Tchaikovskogo, 20a
Telephone: 202-9280
Telex: 413932 NIDEL SU
Telefax: 2302035