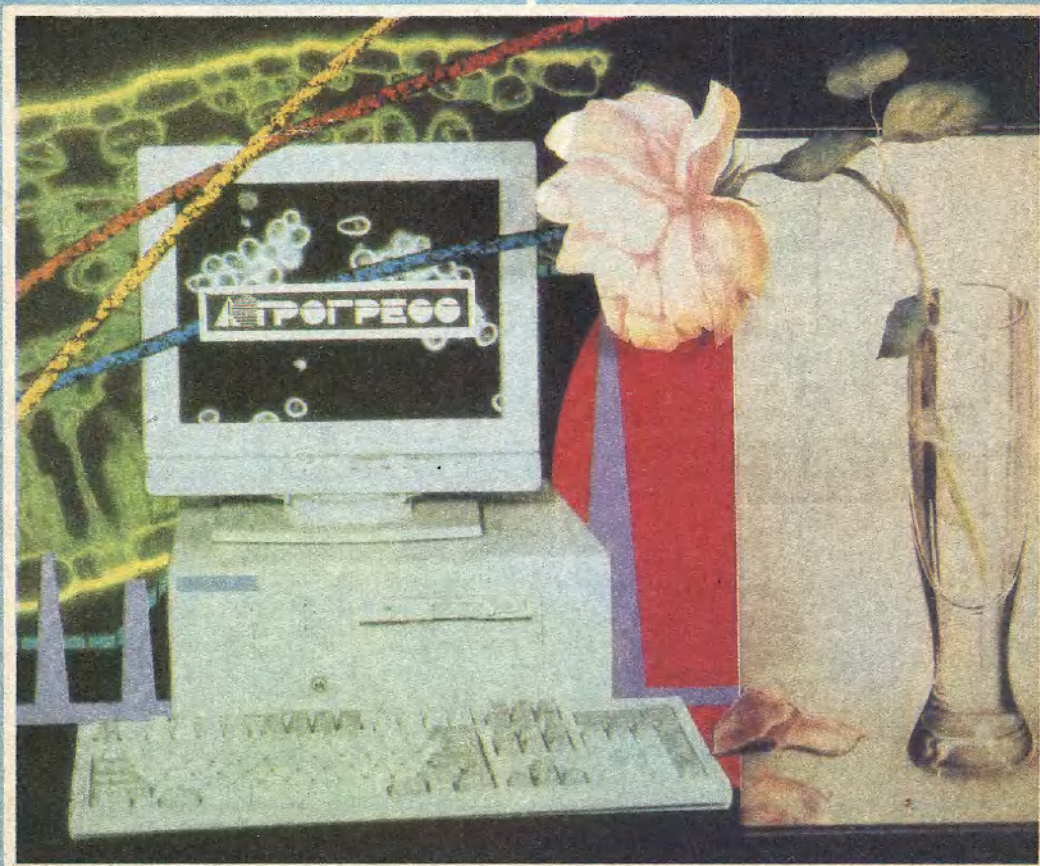
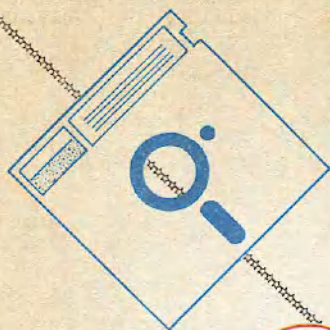


ИНФОРМАТИКА И ОБРАЗОВАНИЕ

5 1991





УК-НЦ



МАЛОЕ ПРЕДПРИЯТИЕ "ТЕХНОКОМ"

предлагает:

- СИСТЕМА ПОСТРОЕНИЯ КУРСОВ ОБУЧЕНИЯ И КОНТРОЛЯ ЗНАНИЙ УЧАЩИХСЯ;
- СИСТЕМЫ ПОДГОТОВКИ ТЕКСТОВОЙ И ГРАФИЧЕСКОЙ ИНФОРМАЦИИ;
- СИСТЕМА СОЗДАНИЯ НАЦИОНАЛЬНЫХ АЛФАВИТОВ С ВОЗМОЖНОСТЬЮ ПЕЧАТИ;
- ЭФФЕКТИВНЫЕ СЕТЕВЫЕ ПРОГРАММНЫЕ СРЕДСТВА;
- КУРСЫ ПОДГОТОВКИ ПРЕПОДАВАТЕЛЕЙ ПО ЭКСПЛУАТАЦИИ УЧЕБНЫХ КЛАССОВ;
- КУРСЫ ПО ОБСЛУЖИВАНИЮ И РЕМОНТУ УК — НЦ;
- РЕМОНТ МС 0511;
- СТЕНД ДИАГНОСТИЧЕСКИЙ ПО РЕМОНТУ МС 0511;
- ПОСТАВКА РАЗЛИЧНЫХ МОДИФИКАЦИЙ УЧЕБНЫХ КЛАССОВ С Ч/Б И ЦВЕТНЫМИ МОНИТОРАМИ;
- РАСШИРЕНИЕ ПАМЯТИ — ЭЛЕКТРОННЫЙ ДИСК (1 Мбайт);
- КАССЕТА ПЗУ (БЕЙСИК);
- АДАПТЕР ТЕЛЕВИЗИОННЫЙ (МОДУЛЯТОР) ДЛЯ ПОДКЛЮЧЕНИЯ УКНЦ К ЦВЕТНОМУ ТЕЛЕВИЗОРУ.



ЗАЯВКИ НАПРАВЛЯТЬ ПО АДРЕСУ :
103460 г. Москва (з-д "Квант") МНПП "Техноком"



Тел. 536-60-29
536-85-84
536-67-10



ИНФОРМАТИКА И ОБРАЗОВАНИЕ

Содержание

Методика обучения

Гольцман М., Первин Ю., Первина Н. Элементы музыкальной грамоты в курсе раннего обучения информатике	3
Федюшин Д. Парадигмы программирования	11
Паттурина Н. Общение учителя и учеников на уроках информатики	15
Система курсов информатики	17
Международный семинар в Москве	24
Зубченко А. НЦПСО — современная индустрия программных педагогических средств	25
TeachCAD — союз ученика и учителя	29
«Юниор-1» — решение проблемы индивидуализации обучения	34

Кабинет ВТ

Артамонов В. КУВТ «Корвет»	37
Филиппский Ю., Пиунов С. «Корнет» для «Нейвы»	41
Кривошеев Г., Козаренко С. Функциональные возможности ОС локальной сети для ПЭВМ «Агат»	43
Кривоногов В. Каталог-меню для УКНЦ	47
Розенфельд А. Использование NED	48
Калейдоскоп УКНЦ	49
Баранов А. Использование псевдографики в обучающих программах	50
Мальцев Д., Китайкин В. NETSY — инструментальное средство для КУВТ-86	53
Канивец В. Активное средство защиты авторских прав	56
Волков Е. Беседы с «Сократом»	59
Здравствуй, «Русич»!	60

Клуб БК

Зальцман Ю. Архитектура и ассемблер БК	65
Улучшение сопряжения БК и ТВ	73
Упрощение организации движения изображений в Бейсике	74
БК-0010.01 в роли осциллографа	74
Калейдоскоп БК	77

Педагогический опыт

Левшин Н., Рижняк Р. Математический задачник для V—VI классов	77
Алипов Н., Сергеева О. Программы-репетиторы в обучении	80
Зинченко И. ЭВМ в начальной школе	82
Гейдар А. Из опыта преподавания информатики в V классе	84

Серда Ю. Разработчикам педагогических программных средств 85

Внеклассная работа

Переход И., Касаткин В. Арифметика фибоначчисвой системы счисления 87

НИТ в дошкольном образовании

Рязанова И. Английский язык и компьютер для дошкольников 91

Точка зрения

«Альтернатива» расширяет деятельность 95

Нам пишут

Вербальный компонент и стратегия школьной информатики (реплика гуманитария) 99

ППП «Телец» 100

Блох А., Ящук С. Информатика в школе на основе программируемых микрокалькуляторов 100

УКНЦ в Малой академии наук 101

«Агат»? 102

Откликнитесь, биокibernетики! 102

4

Информация

Всесоюзный конкурс «Юный программист» («Старшие для младших») 104

Всесоюзный семинар в Москве 105

Омская конференция по проблемам информатизации образования 106

Конференция в Нальчике 107

Главный редактор академик

В. А. МЕЛЬНИКОВ

Редакционная коллегия

И. Н. АНТИПОВ

В. Н. АФАНАСЬЕВ

И. М. БОБКО

Г. В. ГОДЖЕЛЛО

С. А. ЖДАНОВ

Б. В. ЛОМОВ

Ю. В. ЛУИЗО

(зам. главного редактора)

Н. Г. МЕЛЬДИАНОВ

И. С. ОРЕШКОВ

О. К. ПАВЛОВА

А. Ю. УВАРОВ

А. И. ФУРСЕНКО

В. О. ХОРОШИЛОВ

К. В. ШЕХОВЦЕВ

(редактор отдела)

Обложка *С. Плющ*

Редактор отдела *А. Кравцова*

Научный редактор *Н. Копыгина*

Зав. редакцией *Н. Игнатова*

Художественный редактор *Л. Коновалова*

Корректор *В. Антонова, Е. Морозова*

Сдано в набор 22.07.91. Подписано в печать 04.09.91.

Формат 70×100 1/16. Бумага офсетная. Печать офсетная.

Усл. печ. л. 10,40. Усл. кр.-отг. 42,88. Уч.-изд. л. 11,08.

Тираж 55 800 экз. Заказ 1159. Цена 1 руб. 20 коп.

Издательство «Педагогика» Академии педагогических наук СССР и Государственного комитета СССР по печати.

Почту направлять по адресу: 119034, Москва, Смоленский б-р, д. 4. Издательство «Педагогика». Журнал «Информатика и образование». Адрес редакции: Лефортовский пер., д. 8. Телефон: 261-11-29.

Ордена Трудового Красного Знамени Чеховский полиграфический комбинат Государственной ассоциации предприятий, объединений и организаций полиграфической промышленности «АСПОЛ» 142300, г. Чехов, Московской обл.

© «Педагогика», «Информатика и образование», 1991

М. ГОЛЬЦМАН, Ю. ПЕРВИН, Н. ПЕРВИНА

Элементы музыкальной грамоты в курсе раннего обучения информатике

Урок 2. Тема урока: Запись нот в Шарман-
щике.

Цель урока: освоение навыков записи нот.
План урока: проверка домашнего задания;
знание названий нот и их размещения
на нотном стане.

Длительность нот.

Работа с музыкальными текстами.

Правила записи мелодий. Окно экрана.

Темп музыкального произведения.

Лабораторная работа: набор простой мелодии.
Хоровое исполнение песни.

Домашнее задание: написать названия нот
и слова заданной песни («Песенка кро-
кодила Гены» В. Шаинского).

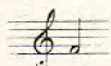
§ 8.4. Длинные и короткие ноты

Звуки различаются между собой не только по высоте, но и по длительности. Послушайте, как звучит, например, очень короткое ДО первой октавы, затем звучание того же недолгого звука, наконец, весьма продолжительное звучание ДО. Различие услышать легко.

Звуки гаммы (§ 8.3) записаны длин-
ми нотами. Ноты такой длительности назы-
вают целыми.

Нота, звучащая вдвое короче целой, на-
зывается половинной. Половинная нота от-
личается от целой не только длительностью
звучания, но и обозначением. В обозначение
половинной ноты кроме белого кружка-ноты
входит еще и присоединенная к нему верти-
кальная черта, называемая штилем.

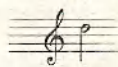
11



На рис. 11 показана половинная нота ФА пер-
вой октавы. Обратите внимание, как распо-
ложен штиль: черточка штиля направлена вверх

от кружочка ноты; штиль нарисован справа от кружка. А вот как выглядит половинная нота РЕ второй октавы (рис. 12): штиль направлен вниз, примыкая к кружочку слева.

12 5



Запомните и соблюдайте правило рисо-
вания штилей: ноты ниже СИ первой окта-
вы (расположенной ровно посередине нотного
стана, на третьей линейке) имеют штили,
направленные вверх и примыкающие к кру-
жочку справа, а ноты выше СИ — примыкаю-
щие слева штили вниз. Штиль у ноты СИ
разрешается рисовать любым из этих двух
способов.

Даже половинная нота звучит столь дол-
го, что, как правило, не встречаются музы-
кальные произведения, которые были бы на-
писаны только половинными нотами. Требу-
ются более короткие звуки. Вдвое короче по-
ловинной ноты является четвертная, или, ко-
роче, **четверть**.

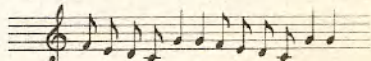
Для четвертной ноты принято обозначе-
ние в виде маленького черного кружочка
со штилем. Четвертями начинается мелодия
знаменитого «Чижика». Вот как записывает-
ся начало этой мелодии на нотном стане
(рис. 13): все ноты здесь, кроме самой по-
следней, — четвертные.

13



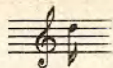
В быстрых песнях бывают нужны и еще
более короткие звуки. Нота, вдвое короче
четверти, имеет название **восьмая** (или одна
восьмая). Восьмая обозначается черным кру-
жочком с фигурным хвостом на конце штиля.
В основном из восьмых составлено начало
песни про двух гусей, живших у бабуся
(рис. 14).

14



Существуют и другие длительности звуков. Вдвое короче восьмой звучит шестнадцатая (или одна шестнадцатая), в изображении которой заметен удвоенный фигурный хвостик на штиле (рис. 15). Такие ноты встречаются в произведениях, исполняемых очень быстро.

15

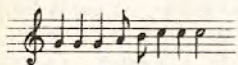


Вопросы и упражнения:

1. Одна мелодия записана половинными нотами, а другая — восьмыми. Какая из этих двух мелодий звучит быстрее?

2. Вот запись второй половины песенки о чижике (рис. 16):

16



6 Перепишите ноты этой мелодии в тетрадь. Над каждой нотой напишите ее название (т. е. высоту звука), а под нотой — ее длительность. Длительности нот обозначайте так: половинная — $1/2$, четвертная — $1/4$, восьмая — $1/8$.

§ 8.5. Музыкальный редактор (первое знакомство)

Музыкальный редактор помогает писать ноты на экране компьютера. В этом редакторе работают два исполнителя. Один из них — Попугай; он расставляет ноты на нотном стане, повышает и понижает звуки, меняет их длительность. Другой исполнитель — Шарманщик; он исполняет мелодию, превращая ноты в «живые» звуки. Его именем назван и весь редактор — Шарманщик.

В былые времена бродили по городам и деревням, селениям и ярмаркам старые шарманщики-музыканты, которые распевали песни под звуки старинного инструмента — шарманки. Шарманку можно сравнить с магнитофоном, на котором записана (встроена) одна-единственная мелодия: на шарманке нельзя «сменить кассету». Чтобы встроена в шарманку мелодия звучала, шарманщик должен был без устали крутить ручку. Чаще всего шарманщик путешествовал вместе с неразлучными помощниками — собачкой, обезьяной или попугаем. Они помогали старику шарманщику «предсказывать» людям судьбу: вытаскивали из ящика билетки, на которых было написано будущее легковерного слушателя. Наш музыкальный редактор, конечно, имеет большие возможности, чем старинный инструмент: на нем можно сыграть самые разные мелодии. Но все же он очень прост и потому назван Шарманщиком.

Из двух исполнителей музыкального редактора Попугая приходится работать больше. Он, словно курсор, перемещается стрелками ВПРАВО и ВЛЕВО вдоль нотного стана. Остановите Попугая и нажмите пробел. Тогда Попугай уронит нотку на нотный стан. Нотка всегда укладывается между второй и третьей линейками — она становится нотой ЛЯ. Не беда, если для составления мелодии требуется какая-нибудь другая нота. Стрелкой ВВЕРХ можно поднять ноту на любую линейку, стрелкой ВНИЗ — опустить.

Обычно Попугай бросает только целые ноты. Но в любой момент длительность ноты можно изменить: достаточно нажать пробел — и целая нота превращается в половинную; еще одно нажатие — и нота становится четвертной; после еще одного нажатия пробела на том же месте появляется восьмая, после восьмой — шестнадцатая, затем тридцать вторая, а после нее снова целая. Таким образом, за несколько нажатий пробела можно выбрать для ноты любую длительность.

Целые ноты обычно встречаются в мелодиях сравнительно редко. Можно попросить Попугая всегда бросать, например, четвертные ноты, а затем, если это необходимо, превращать их в ноты любой другой длительности. Для этого надо сделать четвертной одиночную ноту, стоящую в окошечке в левом верхнем углу экрана (это делается одновременным нажатием пробела и *SHIFT*, быть может, несколько раз). После этого Попугай будет бросать на нотный стан только четвертные ноты.

С помощью Попугая можно написать на нотном стане любую мелодию.

Каждую записанную ноту можно услышать. Достаточно нажать клавишу *TAB*, и та нота, над которой стоит курсор-Попугай, прозвучит в ваших наушниках.

Когда мелодия готова (записана нотами) и хочется прослушать ее целиком, надо надеть наушники (или подсоединить усилитель звука) и нажать клавишу выполнения. Тогда раздвигается занавес под нотным станом, на середину экрана выходит старичок Шарманщик и начинает накручивать ручку своей шарманки.

А в это время либо для каждого из учеников в наушники, либо на весь класс через усилитель раздаются звуки записанной на нотном стане песенки.

Если одновременно с нажатием клавиши выполнения нажать *SHIFT*, то мелодия начинает звучать не с начала, а с той ноты, над которой стоит Попугай. В то время, когда Шарманщик исполняет мелодию, Попугай перемещается вдоль по нотному стану, находясь в любой момент над звучащей нотой.

Если песню компьютера захотелось приостановить, не дослушав ее до конца, надо нажать *STOP*. После следующего нажатия *STOP* песенка будет продолжена с того места, где она была приостановлена. Если же в середине исполнения песни надо остановиться и не продолжать ее, то нажимают *SHIFT+STOP*.

Вопросы и упражнения:

1. Для чего в редакторе Шарманщик существуют два исполнителя?
2. Какова форма курсора в редакторе Шарманщик?
3. Как набираются ноты разной длительности?
4. Наберите и исполните гамму.
5. Чем отличается исполнение музыкального произведения с помощью наушников и посредством магнитофона?
6. Чем отличается остановка звучащей песни от ее приостановки?

§ 8.6. Песенка в окошке

Песенку из нот можно составить таким же образом, как из букв составляется слово.

При наборе текста нажимают буквенную клавишу, и буква появляется на месте квадратика-курсора. Курсор же сам продвигается вправо, показывая тем самым, где будет стоять следующая буква слова.

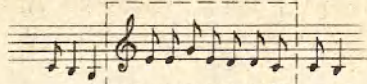
Курсор-Попугай показывает на нотной строке ту позицию, в которой находится сейчас обрабатываемая нота. Ноту, упавшую на нотный стан (это всегда СИ первой октавы), можно поднять или опустить (стрелками *ВВЕРХ* или *ВНИЗ*), удлинить или сократить (пробелом), а только затем записывать следующую ноту. Но для этого надо сначала подвинуть Попугая на место новой ноты (стрелкой *ВПРАВО*), на потому что сам Попугай не прыгает к следующей ноте.

Когда текст заполняет строчка за строчкой весь экран, то самая верхняя строчка уходит за край экрана, а на свободной строчке, появившейся из-за нижнего края, можно продолжать записи. Экран компьютера напоминает окно, через которое можно рассматривать длинный текст. Стрелками *ВВЕРХ* и *ВНИЗ* окно перемещается вдоль текста.

Точно так же ноты длинной песенки рано или поздно заполнят всю видимую на экране часть нотного стана. С появлением следующей ноты весь нотный стан отодвигается назад влево, так что новая нота занимает опять крайнюю правую позицию экрана.

И опять экран становится похожим на окно, перемещаемое вдоль нотного стана стрелками *ВЛЕВО* и *ВПРАВО* (рис. 18).

18



Хотя через окно видно только часть нотной записи песни, можно всегда прочитать ее всю, передвигая окно.

Когда песня столь длинна, что ее можно видеть только с помощью перемещаемого окна, полезны такие действия:

вернуть Попугая одним прыжком в самое начало нотного стана, к скрипичному ключу;

поместить Попугая в начало окна (начало нотного стана и левый край экрана могут не совпадать!);

поставить Попугая одним прыжком над самой последней нотой нотного стана.

Нотную запись длинной песни на нотном стане можно сравнить с альбомом, на каждой странице которого умещается ровно одно окно экрана. Такой «альбом» легко листать вперед и назад.

Вопросы:

1. Чем запись текста отличается от записи нот? Чем они похожи?
2. Как прочитать нотную запись длинной песни, не помещающейся на экране?
3. Как можно быстро переместить курсор в начало нотного стана? А как в начало окна? Чем отличаются эти перемещения?

§ 8.7. Темп

Одну и ту же мелодию можно сыграть с разной скоростью. Так, в веселом танце звуки очень быстро сменяют друг друга, а в протяжной народной песне, записанной нотами той же длительности, — медленно. Быстроту исполнения мелодии музыканты называют темпом.

Написав музыкальное произведение, композитор над началом нотного стана записывает по-русски или по-итальянски название темпа: «Протяжно», «Умеренно», «Оживленно», «Быстро», «Очень быстро» и т. д. В Шарманщике тоже можно установить темп перед исполнением мелодии.

Под нотным станом видна шкала темпа. Слева от нее написано слово «Медленнее», справа — «Быстрее». На середине шкалы стоит курсор-ползунок. Его можно двигать по шкале. При смещении ползунок вправо устанавливается более быстрый темп. Когда ползунок смещается влево, темп исполнения становится медленным.

Вопросы и упражнения:

1. Взгляните на ноты песни композитора В. Шаинского про крокодила Гену и народной песни «Во поле березка стояла». Какие темпы указаны в этих песнях?

2. Напишите ноты знакомого вам «Чижика». Исполните песню в высоком темпе. Повторите песенку, сыграв ее медленно.

Лабораторная работа № 2: набор простых мелодий.

1. Запишите ноты песни «В лесу родилась елочка».

2. Исполните эту песню в нескольких разных темпах.

Методический комментарий.

Урок начинается с проверки домашних работ. Задание было несложным, но сегодняшняя проверка необходима. Она не формальна. Прошлый урок погрузил учащихся в слишком необычную для них обстановку. Необходимо проверить усвоение новых понятий и закрепить их.

На доске — гамма ДО мажор во второй октаве, материал домашнего задания. Учитель спрашивает вразбивку названия нот. Новый материал — рассказ о длительности нот (§ 8.4).

Этот материал не сложен. Иллюстрируется он с помощью Шарманщика: дети сами убеждаются в возможности рисования нот любой длительности путем нажатия пробела.

С появлением понятия штиля надо обратить внимание детей на правило вычерчивания штилей (§ 8.4).

Учитель объясняет необходимость эталонной длительности нот, устанавливаемых Попугаем. Прием фиксирования длительности важен в практике машинной записи нот. Поэтому, показав его, учитель просит детей поупражняться в установке эталонных длительностей.

Все дети получают заготовленные до урока листы с записанными на них нотами и словами известной детской песенки «Маленькой елочке холодно зимой».

Школьники должны под каждой нотой написать ее название и длительность. Обозначения длительностей (учитель показывает эти обозначения на доске или на плакате):

1 — целая, $1/2$ — половинная,

$1/4$ — четверть, $1/8$ — восьмая.

Дети выполняют задание самостоятельно. Учитель бегло просматривает завершённые работы. Листы остаются у детей. С записанными на них нотами еще предстоит работа на уроке.

Прежде чем записывать ноты, надо ввести понятие окна нотного стана (§ 8.6) — по аналогии с окном текстового редактора. Это позволит не отвлекаться в ходе лабораторной работы.

Следующий этап урока — лабораторная работа № 2. Выполняя ее, дети знакомятся с новым для них понятием темпа.

Первое задание работы состоит в переписывании нот с листков на экраны. Выполняя это задание, дети убеждаются в существовании экрана-окна.

Прежде чем приступить к прослушиванию, учитель исполняет мелодию на «Соловухе» (на случай, если в записях учеников допущены ошибки) и на центральном компьютере. Затем дети прослушивают в наушниках мелодию песни в стандартном варианте (с неявно заданным стандартным темпом). Наконец, учитель просит детей спеть песню под аккомпанемент компьютера, выпевая название нот. Учитель поет с детьми.

Некоторое отвлечение от лабораторной работы: дети записывают над нотами по слогам слова первого куплета песни. Оригинал служит плакат или запись на кодоскопе. Ту или другую заготовку учитель сделал до начала урока. Следует важное объяснение, начинающееся с вопроса: «Почему слова над нотами записываются по слогам?» Возможно, дети сами смогут сформулировать ответ: «Петь можно только гласные буквы, а поскольку в каждом слоге есть только одна гласная, то выделение гласных в слове совпадает с его делением на слоги».

Когда дети переписали слова песни, ее можно спеть хором снова, выпевая на этот раз не названия нот, а слова песни. Чтобы получился слаженный хор под аккомпанемент ученических компьютеров, дети в наушниках одновременно по взмаху руки учителя нажимают клавиши выполнения на своих компьютерах.

После этого следует рассказ о тем-

пе музыкального произведения. В § 8.7 учебника упоминается, что названия темпов иногда пишут по-итальянски. Можно объяснить детям (или попросить высказаться по этому поводу учеников музыкальной школы), что родины многих видов музыкального искусства является Италия и потому итальянская терминология до сих пор сохраняется в записях музыкальных произведений. Отвлечение о темпе непосредственно предшествует выполнению второго задания лабораторной работы: дети индивидуально прослушивают через наушники мелодию в разных темпах.

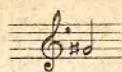
Продвинутые школьники во время лабораторной работы набирают и исполняют мелодии своих домашних заданий. Закончив лабораторную работу, учитель при наличии времени может предложить классу послушать записи юных музыкантов. С этой целью усилитель последовательно подключается к каждой из машин-исполнителей.

В конце урока резервируется время для домашнего задания. На листах, которые дети получают от учителя, — ноты и слова припева песни В. Шаинского про крокодила Гену. Надо под нотами написать их названия и длительности, а над нотами — слоги слов припева.

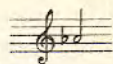
Три следующих параграфа учебника и сопровождающая их лабораторная работа открыты для всех и не выделены шрифтом, но обязательны они только для детей, обучающихся в музыкальной школе, которые способны прочитать эти параграфы самостоятельно.

§ 8.8. Бемоли и диезы

Звуки, которые издают черные клавиши фортепианной клавиатуры, не похожи ни на один из семи основных звуков. И все-таки у них нет специальных обозначений. Для обозначений черных клавиш используются названия основных звуков и некоторые дополнительные знаки. Например, есть на фортепианной клавиатуре черная клавиша справа от ноты СОЛЬ. Все ноты, находящиеся правее ФА на клавиатуре, записываются выше ФА на нотном стане. Черная клавиша справа от ФА — ее ближайшая соседка. Поэтому обозначают ее с помощью той же ФА, ставя перед нею знак повышения — диез (\sharp). Точно так же следующую черную клавишу, расположенную правее СОЛЬ, называют СОЛЬ диез, а изображают так, как показано на рис. 19.



На ту черную клавишу, что находится правее СОЛЬ, можно посмотреть и с другой стороны: она находится левее ЛЯ, т. е. ниже ее. Поэтому эту черную клавишу можно обозначить по имени ее ближайшей правой (верхней) соседки с добавлением впереди знака понижения — бемоля (b). В таком обозначении она называется ЛЯ бемоль (рис. 20).



Таким же образом звук черной клавиши, которая является левой соседкой СИ, можно называть СИ бемоль.

Значит, каждая черная клавиша может иметь любое из двух обозначений — либо понижение правой основной ноты с помощью бемоля, либо повышение левой основной ноты с помощью диеза. Оба они обозначают один и тот же звук. А какое из двух обозначений выбрать, определяется по специальным правилам, которые установлены в музыке.

Чтобы Шарманщик поставил повышающий знак диеза перед нотой, надо поставить Попугая над этой нотой и нажать SHIFT+ВВЕРХ. Установку понижающего бемоля делают с помощью SHIFT+ВНИЗ.

Вопросы и упражнения:

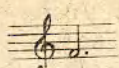
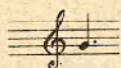
1. Какие названия имеет первая из черных клавиш октавы?
2. Назовите обозначения всех пяти черных клавиш октавы, которые строятся с помощью диезов. А как те же клавиши обозначаются с помощью бемолей?
3. Перепишите мелодию, показанную на рис. 21, так, чтобы в ней не было ни одного диеза.



§ 8.9 Точка

Чтобы указать длительность ноты, надо выбрать одну из нескольких возможностей: целая, половина, четверть, восьмая... А как быть, если нужно задать длительность ноты, которая должна звучать дольше, чем четверть, но короче, чем половинная?

Точка, поставленная рядом с нотой, справа от нее, увеличивает длительность ноты на ее половину. Если точка стоит около четверти (рис. 22), то такая нота имеет длительность $1/4 + 1/8$. А когда точка поставлена около половинной ноты (рис. 23), это означает длительность $1/2 + 1/4$.



Попугай, расставляющий ноты и устанавливающий их длительности, может нарисовать и точку. Для этого надо поставить Попугая над нотой, требующей удлинения, а затем нажать клавишу с точкой на клавиатуре компьютера.

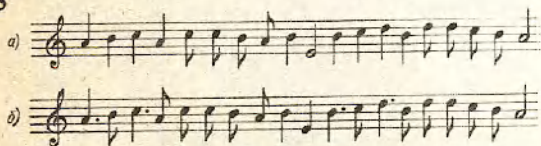
Вопросы:

1. Какая из трех нот на рис. 24 самая долгая? Какая самая короткая?

24



2. Послушав две мелодии (рис. 25 а и б), скажите, какая из них кажется вам более точной записью начала известной песни «Катюша». Почему?

10
25

§ 8.10. Пауза

Чтобы отделить одну часть мелодии от другой или чтобы выразить ее прерывистый характер, композитор иногда предусматривает временные приостановки в звучании мелодии. Это паузы — знаки молчания. Во время паузы музыка не звучит. Разумная расстановка пауз часто очень украшает мелодию.

Паузы также имеют длительность, как и ноты: они могут быть целыми, половинными, четвертными, восьмыми, шестнадцатыми, ... Их обозначения показаны на рис. 26.

26



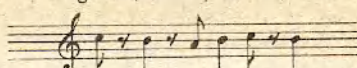
Паузы имеют очень строгое начертание: целая пауза — черный прямоугольник под четвертой линейкой, половинная — такой же прямоугольник на третьей линейке.

В редакторе Шарманщик паузу (так же как и ноту) ставит на нотный стан наш старый знакомый Попугай. Чтобы Попугай «бросил» паузу, надо нажать клавишу с надписью **GRAPH**, которая расположена слева от пробела. Тогда на нотный стан падает пауза такой же длительности, как нота в окошечке над нотным станом. Многократным нажатием клавиши **GRAPH** можно превратить целую паузу в паузу любой длительности (так же как многократное нажатие пробела при установке ноты позволяет назначать любую ее длительность).

Вопросы и упражнения:

1. Какой из двух отрывков (рис. 27) звучит дольше?

27



2. В гамме (рис. 27) сократите наполовину длительность звучания каждой отдельной ноты, не меняя длительности звучания всей гаммы.

Лабораторная работа № 3: паузы и знаки альтерации.

1. Запишите предложенную мелодию.
2. Прослушайте ее.
3. Удлините вдвое каждую из нот и пауз.
4. Прослушайте измененную мелодию.

5. Замените все ноты с диезами на одинаково звучащие ноты с бемолями. Урок 3. Тема урока: Простейшее редактирование мелодий.

Цель урока: отработать навыки корректировки музыкальной информации с использованием внутреннего слуха учащихся.

План урока: запись и прослушивание домашнего задания.

Правила корректировки ошибок в Шарманщике.

Упражнения по исправлению ошибок в нотных записях на слух: замена, удаление, вставка.

Лабораторная работа: редактирование.

Домашнее задание: рассказ на тему «Песня, которую я люблю».

§ 8.11. Редактирование мелодии

При записи нот возможны ошибки. Поэтому очень важно научиться их исправлять.

Ошибки бывают трех видов:

вместо правильной ноты написана неверная (или, как говорят музыканты, фальшивая);

написана лишняя нота;

пропущена нота.

Обратите внимание, что при записи текстов указывались такие же типы ошибок. Столь же похожими являются и способы исправления ошибок.

Исправление ошибок первого вида делается заменой. Над ошибочной нотой ставится курсор-Попугай. Теперь, меняя высоту и длительность ноты, можно на этом месте поставить верную ноту. Заметьте, что нота

правильной высоты (имеющая название правильной ноты) с неверной длительностью тоже считается ошибочной и ее надо заменять, исправляя длительность.

Операция исправления второго вида ошибок — удаление. Удаление, как в тексте, выполняется клавишей *DEL*, которую надо нажать в тот момент, когда Попугай встал над лишней нотой.

В некоторых случаях можно использовать для удалений клавишу *BS*. Она действует так же, как в текстовом редакторе: стирает ноту, стоящую слева от Попугая.

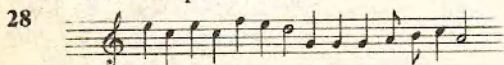
Еще одна операция редактирования в Шарманчике напоминает редактор Микрон: удалить конец нотной строки можно с помощью *CTRL+E*. Тогда с нотного стана стираются все ноты после той, над которой стоит Попугай. Нота под Попугаем тоже удаляется.

Когда нужно стереть все, что написано на нотном стане, нажимают привычную комбинацию клавиш — *SHIFT+HOME*.

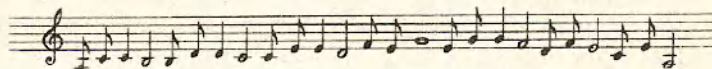
Чтобы восстановить пропущенную ноту, нужна вставка. Надо подвести Попугая к ноте, перед которой должна быть вставлена пропущенная нота, и нажать клавишу *INS*. Тогда вся нотная строка справа от Попугая подвинется вправо, освободив место для вставки, и можно вставлять требуемую ноту.

Вопросы и упражнения:

1. На рис. 28 изображены ноты «Чижика». Какая здесь допущена ошибка? Как исправить ее?



2. В ошибочной мелодии песенки крокодила Гены, ноты которой показаны на рис. 29, пропущены сразу две ноты подряд. Где ошибка? Какие это ноты? Расскажите, как исправить ошибку.



3. В нотах песни «Во поле березка стояла» (рис. 30) записан лишний звук. Найдите его.



Лабораторная работа № 4: редактирование.

1. Напишите гамму целыми нотами.

2. Прослушайте гамму.

3. Вслед за каждой целой нотой поставьте четвертную ноту той же высоты.

4. Уберите с нотного стана все целые ноты.

5. Прослушайте новый вариант гаммы.

Методический комментарий.

В уроках, посвященных Шарманчику, вообще говоря, могут быть выделены компоненты музыки и информатики. На сегодняшнем уроке такое разделение сделать трудно: с одной стороны, учащиеся оперируют аппаратом информатики — типовыми средствами редактирования информации, с другой стороны, здесь учитель впервые затрагивает глубинные вопросы музыкальной педагогики: если на двух предшествующих уроках музыкальная информация главным образом предьявлялась ученику извне, то на нынешнем уроке его внутренний слух — умение слышать музыку — служит основой для осуществления формальных операций корректировки ошибок.

После упражнений прошлого урока домашнее задание, по-видимому, не представит для детей больших трудностей, хотя займет немалое время: дети записывают мелодию припева в песне В. Шаинского про крокодила Гену. Учитель может задать вопросы, рассчитанные в первую очередь на учащихся музыкальной школы: кто такой Шаинский? Какие песни он написал? Какие еще композиторы пишут песни для детей? Вместе с тем надо быть готовым самому ответить на эти вопросы, назвав И. Дунаевского, Д. Кабалевского, А. Пахмутову, А. Островского, Я. Френкеля, Е. Крылатова, Ю. Чичкова...

Это небольшое отклонение, посвященное роли композиторов в создании песен, представляется важным даже в воспитательном плане. Практика показывает, что большинство третьеклассников общеобразовательной школы ассоциируют песню не с ее создателем, а с исполнителем.

Проверка работы — сравнение эталонного звучания и записей школьников. Сначала центральная машина исполняет песню, затем дети запускают мелодию на своих машинах и сообща-

ют о наличии ошибок. Информация о количестве ошибок нужна для завершения работы с песней крокодила Генны: если ошибок много, то ее хоровое исполнение идет под аккомпанемент центрального компьютера; когда ошибок мало, дети поют, не снимая наушников, — слушают свои машины. Хор поет дважды — первый раз выпеваются названия нот, второй раз — слова песни. Как всегда, личное участие учителя — стимул активности детей.

Если имела место хотя бы одна ошибка в ученических записях, это обстоятельство надо использовать в качестве стартовой позиции для объяснения нового материала — приемов исправления ошибок на нотном стане (§ 8.11. «Редактирование мелодий»). Здесь тоже максимально используется аналогия с ранее изученными редакторами, в первую очередь с текстовым редактором Микрон. Поэтому обсуждение формальных правил корректировки не должно вызывать трудностей.

Более трудными могут оказаться практические упражнения на обнаружение ошибок, поскольку они опираются на способности детей слышать музыку. Эти способности по-разному развиты у различных детей.

Учитель сначала исполняет на центральном компьютере правильно звучащую мелодию «Чижика», а затем рассылает на ученические машины ноты песни, в которой допущена ошибка — фальшивая нота. Для облегчения поиска ошибки фальшивая нота поставлена не в начале мелодии, а в конце ее, на самом сильном звуке — тонике. Обмену музыкальными текстами по сети будет посвящен следующий урок. Сейчас для получения нот от учителя достаточно выполнить формальные операции, хорошо знакомые им, впрочем, по Микрону: вызвать меню режимов общения с учителем и войти в режим приема.

После обнаружения ошибки дети самостоятельно делают исправления. Первую корректировку, выполненную детьми, учитель должен проверить индивидуально, послушав наушники каждой машины. Эта проверка позволит сразу

выделить группу детей (возможно, пустую) с пониженными способностями восприятия музыкальной информации. Таких детей надо будет в большей мере привлекать к формально-синтаксическим операциям с символьным представлением музыкальной информации, в меньшей — к работе с «живым» звуком. Тем, кому самостоятельная корректировка не удалась, помогает учитель. На удалении первой ошибки нельзя экономить время!

Таким же образом выполняется упражнение по обнаружению пропущенной ноты. Эта ошибка тоже размещается ближе к концу текста (но не последней нотой!), тоже по возможности близко к тонике и на сравнительно долгом звуке. Технология демонстрации ошибки та же: сначала звучит на весь класс правильная мелодия песни про гусей, которые жили у бабуси, а затем — рассылка по сети музыкального текста с ошибкой. На этот раз учитель проводит проверку выборочно только у тех школьников, которые на первом упражнении заставили его выполнять их исправления.

Третье упражнение выполняется так же. Оно проверяет умение услышать лишний звук в мелодии: сравнивается эталонно звучащая песня «Маленькой елочке холодно зимой» с мелодией, разосланной по сети, и воспроизводимый в наушниках музыкальный текст с ошибкой. Здесь учитель проверяет только тех учеников, кто просит его об этом.

Теперь лабораторная работа «Редактирование» не покажется сложной: она не предъявляет никаких требований к слуховым способностям учащихся. Каждый ребенок должен показать в конце работы учителю отредактированный музыкальный текст — гамму ДО мажор, записанную четвертными нотами.

Домашнее задание факультативно: можно попросить детей написать сочинение «Песня, которую я люблю». Оно не только раскроет ученика учителю, но и, возможно, позволит последнему с большей индивидуальностью построить завершающий урок по теме «Концерт».

Окончание следует.

Парадигмы программирования

*Все время «да» и «нет» — это
тоже мало радости.*

В. Шекспир

В вопросе применения языка Пролог в курсе информатики есть два аспекта: первый — использование языка Пролог как учебного языка и второй — в качестве инструментального, для разработки программ учебного назначения.

История применения языка Пролог в образовании началась еще в 1978 г. в Англии в виде эксперимента в одной из средних школ Уимблдона. В дальнейшем был создан проект «Логика как язык ЭВМ для детей». Этот проект использовал версию языка Микро-Пролог, особенностью синтаксиса которой является не предикатная, а списочная форма записи фактов и правил [19, 24]. В проекте сначала участвовали дети 10—12 лет, а затем, после упрощения общения с системой, подключились и младшие школьники в возрасте 7—9 лет. Обучение строилось на постоянном усложнении баз данных и баз знаний Пролога. Сначала рассматривались фактографические базы с простейшими запросами с одной подцелью, потом вводились базы с правилами и сложными запросами, затем базы с рекурсиями. Восприятию материала способствовало наглядное изображение объектов и отношений между ними в виде графа (семантической сети). Аналогичные эксперименты, но уже со старшеклассниками были проведены в 1984—1986 гг. в Дании, причем в курсе информатики на первом этапе там изучался Пролог и параллельно с ним Лого, а на втором этапе — Паскаль. Задачи по Прологу в основном подбирались из области литературы и лингви-

стики, строились программы синтаксического анализа для английского и немецкого языков. Эксперименты в Дании и английский проект прошли успешно, были получены интересные методические результаты.

Автор статьи ведет эксперименты по применению языка Пролог в курсах информатики и программирования с 1987 г. в физматшколе и на факультете прикладной математики МИЭМ. В экспериментах со школьниками использовались версии языка Микро-Пролог и Пролог-Д для ПЭВМ «Ямаха», а в курсах программирования для студентов кроме перечисленных версий версии МПролог [40], Турбо-Пролог [27], Эрити Пролог для IBM PC/XT/AT. Подобные эксперименты проводили и проводят и другие авторы учебника [2], а также некоторые учителя информатики, работающие по этому учебнику.

Некоторые выводы из проведенных экспериментов:

1. Язык Пролог вполне доступен школьникам даже младших классов.

2. При изучении языка Пролог школьник практически с первого же занятия начинает работать с «осмысленными» программами, которые можно рассматривать как информационно-логические модели, проводить над ними компьютерные эксперименты. Возможность непосредственного моделирования на языке Пролог стимулирует творческую активность учащихся при решении «реальных» задач.

3. Парадигма логического программирования тесно связана с логическим стилем мышления, который включает в себя умения рассуждать, доказывать, подбирать факты, формулировать утверждения и обосновывать предлагаемые решения. Поэтому освоение языка

Пролог и логического программирования неразрывно связано с изучением элементов логики.

4. Курс информатики должен знакомить как с процедурной, так и с декларативной парадигмой, причем язык Пролог должен идти перед процедурным языком. Знание Пролога не мешает усвоению процедурной парадигмы. В качестве процедурного языка на втором этапе курса рекомендуется использовать для младших школьников Лого, а для старшеклассников язык Паскаль.

5. Наблюдаются значительные трудности при переходе от процедурной парадигмы к декларативной (логической). Например, если в X классе школьники изучали алгоритмизацию и язык Бейсик, то в XI классе у сильных учащихся, хорошо освоивших процедурный подход, могут возникнуть сложности с освоением логической парадигмы, связанные с изменением стиля мышления и невозможностью переноса освоенных методов процедурного программирования в логическое, зато слабые учащиеся, не освоившие алгоритмизацию, могут сделать значительные успехи в логическом программировании. Подобные сложности даже еще в большей степени, наблюдаются у учителей информатики, студентов и профессиональных программистов.

6. Из различных версий языка Пролог в школе лучше использовать русскоязычную версию Пролог-Д, написанную специально для учебных целей и реализованную для «Ямахи» (MSX2), КУВТ «Русич» УКНЦ, БК-0010 «АГАТ», БК-0011, IBM PC/AT/XT [30—37]. Немаловажным для преподавания начального курса информатики является то, что Пролог-Д имеет графические средства [35]. Для КУВТ «Корвет» в НИИ СЧЕТМАШ разработана версия языка Пролог, близкая к описанной в книге [20] и также имеющая средства работы с графикой.

7. При изучении языка Пролог практически не наблюдается явление «пустого экспериментирования», когда школьник настолько увлекается бессмысленными изменениями внутри программы, наблюдая «эффекты», что полностью теряет связь с исходной задачей.

8. Вузовский курс информатики лучше строить как трехэтапный: на первом этапе изучается логическое программирование (язык Пролог), на втором этапе — процедурное (например, на базе языка Турбо-Паскаль или Модула-2), на третьем этапе — объектно-ориентированное программирование (в качестве языков «поддержки» можно использовать объектно-ориентированное расширение Турбо-Паскаля или Модулы-2, язык СИ++). Дело в том, что смена процедурной парадигмы на объектную — процесс столь же болезненный, как и на логическую, но при цепочке «логическая — процедурная — объектная» смена парадигм у студентов значительно облегчается.

Если рассматривать язык Пролог как инструментальный язык, то он прекрасное средство для создания баз знаний по различным предметам, в том числе и гуманитарным, экспертных систем, интеллектуальных обучающих систем. Распространение языка Пролог в качестве инструментального, как и других новых информационных технологий, сдерживается отсутствием в настоящее время в системе образования адекватной вычислительной техники с достаточно большой оперативной памятью и винчестерскими дисками. Отдельные учебные классы IBM-совместимых компьютеров общей погоды не делают, но могут подготовить плацдарм для последующего распространения НИТ в образовании.

На пути к задачку,
или О классификации задач
по языку Пролог

Exemplis discimus

Преподаватель, впервые начинающий работать с языком Пролог, испытывает значительные затруднения при подборе задач и упражнений, наиболее эффективных для освоения языка и формирования у учащихся парадигмы логического программирования. Задачки для традиционных процедурных языков оказываются практически непригодны. Язык Пролог не предназначен для решения вычислительных задач и задач по своей сути алгоритмических, хотя конкретные версии языка Пролог имеют процедурные компоненты, в част-

ности встроенные арифметические предикаты, предикаты сравнения чисел, ввода-вывода и т. д. Сфера его применения — решение нечисловых задач, задач символьной обработки. Особенно он эффективен для решения информационно-логических задач, где главное значение имеют сложноструктурированные нечисловые данные, а также для таких, где важную роль играет поиск решений среди множества вариантов. У языка Пролог простой синтаксис, практически не требующий для освоения специальных, направленных только на синтаксис, задач и упражнений.

Основной упор в обучении делается на логическую компоненту и в меньшей степени, обычно в углубленном курсе, — на управляющую компоненту.

Можно выделить следующие классы задач, построенные по принципу усложнения логической компоненты.

Класс 1. Фактографическая база данных без альтернатив.

Содержит только факты, причем все имена предикатов в базе различны.

Пример: A;
B;
C;

Класс 2. Фактографическая база данных с альтернативами.

Содержит факты с одинаковыми именами предикатов.

Пример: A₁;
A₂;
A₃;

здесь и далее A_i — предикаты с одинаковыми именами и числом аргументов.

Замечание. Рассматривая фактографические базы данных, школьники знакомятся с типами данных Пролога: числовыми и символьными константами, переменными, структурами, списками — и с основной операцией над данными в Прологе — операцией сопоставления (унификации). Постепенно возрастает сложность запросов: от запросов, содержащих один предикат (подцель), без аргументов-переменных, к запросам с переменными и к сложным запросам с несколькими подцелями. Одновременно объясняется механизм прямого перебора и прямого перебора с возвратом для используемой версии Пролога.

Класс 3. База знаний без альтернатив.

Содержит факты и правила, причем имена предикатов у всех фактов и в левых частях правил различны.

Пример: A;
B;
C;
D ← A, C;

Класс 4. База знаний с альтернативами.

Здесь имена предикатов у некоторых фактов и в левых частях правил совпадают.

Пример: A₁;
A₂;
A₃ ← B, C;

Класс 5. База знаний с рекурсией.

В языке Пролог возможна рекурсия двух видов: алгоритмическая рекурсия, когда правило логического вывода ссылается на само себя, и рекурсия по данным, когда составной объект является частью этого же составного объекта.

База знаний с алгоритмической рекурсией содержит правило вида

$R \leftarrow A, U, B, R, C;$

где R — предикат, определяющий рекурсивный вызов, U — предикат, определяющий условие выхода, A, B, C — предикаты, не влияющие на рекурсию. Рассматривая рекурсивные базы, отдельно выделяется подкласс задач на рекурсивную обработку списков.

Класс 6. Базы знаний со встроенными процедурными предикатами.

В разных версиях Пролога имеется различный набор встроенных предикатов. Учебная версия Пролога должна содержать графические средства, арифметику, предикаты сравнения, ввода-вывода и некоторые другие.

Класс 7. База знаний с управлением.

В большинстве версий Пролога существуют два предиката для управления механизмом перебора: предикат отсечения (cut, «!») и предикат отказа (fail, ЛОЖЬ).

Предикат отказа обычно используется для организации повторения выполнения некоторых целей.

Предикат отсечения применяется, когда надо ограничить поиск по базе данных программы, т. е. при выполне-

нии определенных условий запретить возврат.

Пример: $A1 \leftarrow B, I, C;$
 $A2;$

При выполнении предиката отсечения предикаты, стоящие левее его, «замораживаются», т. е. прекращается поиск альтернативных решений для B , а для A — использование альтернативных утверждений ($A2$), лежащих ниже правила.

Особенности выполнения этих встроенных предикатов зависят от конкретной реализации Пролога.

Класс 8. База знаний с отрицанием.

Предикат NE (not) занимает особое место среди других встроенных предикатов и имеет определенное, отличающееся от принятого в логике. Предикат NE трактуется как отрицание по невыполнимости, т. е. отрицание некоторой цели истинно, если не существует возможности доказательства истинности самой цели.

Класс 9. Динамические базы знаний.

Пролог-программа может изменять сама себя в процессе выполнения. Существуют специальные предикаты, добавляющие или убирающие из базы указанные факты или правила.

Класс 10. Сложные базы знаний.

Сложные базы представляют комбинацию баз из предыдущих классов, а если позволяет версия, имеют средства поддержки модульности Пролог-программы.

Замечание. Приведенная классификация носит методический характер и не претендует на универсальность и единственность.

Литература

1. Советский энциклопедический словарь. М.: Советская энциклопедия. 1980 и последующие изд.

2. Основы информатики и вычислительной техники: Пробное учебное пособие для X—XI кл. ср. шк. / В. А. Каймин, А. Г. Щеголев, Е. А. Ерохина, Д. П. Федюшин. М.: Просвещение, 1989.

3. Кушниренко А. Г. и др. Основы информатики и вычислительной техники: Пробный учебник для ср. уч. заведений / А. Г. Кушниренко, Т. В. Лебедев, Р. А. Сворень. М.: Просвещение, 1990.

4. Дал О., Дейкстра Э., Хоор К. Структурное программирование. М.: Мир, 1975.

5. Вирт Н. Систематическое программирование. Введение. М.: Мир, 1977.

6. Йодан Э. Структурное проектирование и конструирование программ. М.: Мир, 1979.

7. Вирт Н. Алгоритмы и структуры данных. М.: Мир, 1989.

8. Хендерсон П. Функциональное программирование. Применение и реализация. М.: Мир, 1983.

9. Хювонен Е., Селиянен Й. Мир Лиспа: Методы и системы программирования на языке Лисп. М.: Мир, 1990.

10. Фути К., Судзуки Н. Языки программирования и схемотехника СБИС. М.: Мир, 1988.

11. ЭВМ пятого поколения: Концепции, проблемы, перспективы / Под ред. Т. Мото-ока. М.: Финансы и статистика, 1984.

12. Симон Дж. ЭВМ пятого поколения: компьютеры 90-х годов. М.: Финансы и статистика, 1985.

13. Осуга С. Обработка знаний. М.: Мир, 1989.

14. Представление и использование знаний / Под ред. Х. Уэно, М. Исудзука. М.: Мир, 1989.

15. Приобретение знаний / Под ред. С. Осуги, Ю. Сазки. М.: Мир, 1990.

16. Чень Ч., Ли Р. Математическая логика и автоматическое доказательство теорем. М.: Наука, 1983.

17. Хоггер К. Введение в логическое программирование. М.: Мир, 1988.

18. Логическое программирование. М.: Мир, 1988.

19. Кларк К., Маккейб Ф. Введение в логическое программирование на микро-Прологе. М.: Радио и связь, 1987.

20. Клоксин У., Меллиш К. Программирование на языке Пролог. М.: Мир, 1987.

21. Язык Пролог в пятом поколении ЭВМ. М.: Мир, 1988.

22. Стерлинг Л., Шапиро Э. Искусство программирования на языке Пролог. М.: Мир, 1989.

23. Доорс Дж. и др. Пролог — язык программирования будущего. М.: Финансы и статистика, 1990.

24. Макаллистер Дж. Искусственный интеллект и Пролог на микро-ЭВМ. М.: Машиностроение, 1990.

25. Братко И. Программирование на языке Пролог для искусственного интеллекта. М.: Мир, 1990.

26. Грей П. Логика, алгебра и базы данных. М.: Машиностроение, 1989.

27. Ин Ц., Соломон Д. Использование Турбо-Пролога. М.: Мир, 1990.

28. Ковальски Р. Логика в решении проблем. М.: Наука, 1990.

29. Маллас Дж. Реляционный язык Пролог и его применение. М.: Наука, 1990.

30. Григорьев С. Г., Морозов В. Н. Знакомьтесь — язык Пролог // Информатика и образование. 1987, № 4.

31. Каймин В., Григорьев С. и др. Элементы логики и начала языка Пролог // Информатика и образование. 1989, № 4.

32. Каймин В., Григорьев С. Пролог в школьной информатике // Информатика и образование, 1990, № 2.

33. Григорьев С. Работа системы Пролог-Д // Информатика и образование, 1990, № 4.

34. Григорьев С. Программирование на Прологе-Д // Информатика и образование. 1990. № 5.
35. Григорьев С. Графические средства системы Пролог-Д // Информатика и образование. 1990. № 6.
36. Григорьев С. Обработка списков на Прологе-Д // Информатика и образование, 1991, № 1.
37. Алексеев М. Пролог-Д на «Агате» // Ин-

форматика и образование. 1991. № 3.

38. Логическое программирование. М.: Знание, 1988.

39. Логический подход к искусственному интеллекту: от классической логики к логическому программированию. М.: Мир, 1990.

40. Иванова Г. С., Тихонов Ю. В. Введение в язык Пролог. М.: Изд-во МГТУ, 1990.

Н. ПАТТУРИНА

Общение учителя и учеников на уроках информатики

На Всесоюзной конференции по проблемам образования в области информатики, вычислительной техники и автоматизации, состоявшейся в начале декабря 1990 г. в Ленинграде, на секции информатики в средних учебных заведениях и вузах возникла дискуссия об организации общения учителя и учеников на уроках информатики по результатам социологического опроса школьников, обучающихся ОИВТ.

Оказалось, что одной из особенностей уроков информатики, положительно оцененной школьниками, является то, что учитель информатики реже общается с ними на уроке, чем остальные учителя. Возникает вопрос: как следует отнестись к этому факту? Рассмотрим его с психологической точки зрения.

Личность в процессе общения формирует свои ценности, позиции, оценки относительно конкретных жизненных ситуаций. Однако, процесс их формирования не прост. Личностная позиция не формируется в одноразовом общении. Динамика межличностного общения предполагает, как минимум, два типа деятельности — опосредствованных взаимоотношений. Они складываются у личности с наиболее значимыми для нее группами (или отдельными лицами). Один из них — определяющий, а другой — зависимый [1, 2].

«Определяющий» тип взаимоотношений — это сложившийся тип взаимоотношений субъекта с референтной для него группой (или лицом), который детерминирует характер его (субъекта) взаимоотношений с другими людьми. Позиция, ценности, оценки относитель-

но конкретных жизненных ситуаций, выработанные в этом типе взаимоотношений, переносятся субъектом в другие важные для него ситуации общения и во многом определяют их развитие. Для «определяющего» типа взаимоотношений характерно присвоение личностью референтных отношений, которые становятся для нее впоследствии субъективными факторами поведения.

«Зависимый» тип взаимоотношений — это такие взаимоотношения субъекта с референтной для него группой (или лицом), где совместно выработанные позиции, оценки относительно конкретной жизненной ситуации в большинстве случаев либо не переносятся субъектом во взаимоотношения с другими людьми, либо предъявляются значимому лицу, включенному в «определяющий» тип взаимоотношений для их оценки. В случае принятия последним этих позиций и ценностей они становятся для субъекта личностно значимыми и, наоборот, в случае непринятия утрачивают для него значение.

Формируясь в ситуации общения, обусловленной «определяющим» типом взаимоотношений, отношение личности к своей жизнедеятельности (ее позиция относительно конкретной жизненной ситуации) отстаивается ею затем в «зависимом» типе взаимоотношений, где оно корректируется, дополняется и становится значимым свойством личности. В дальнейшем оно может служить личности средством персонализации в общении и деятельности с другими людьми, оказывая влияние на формирование отношения других людей к своей жизнедеятельности.

Итак, подчеркнем, процесс общения предполагает, как минимум, два типа взаимоотношений: один — «определяющий», при котором формируются позиция, ценности; другой — «зависимый», где личность отстаивает эти ценности. «Зависимый» тип взаимоотношений необходим для личности, для защиты ею выработанных в «определяющем» типе взаимоотношений позиций и ценностей. Только в этом случае они (позиция, ценности) становятся для личности значимыми, своими.

Следовательно, когда на уроке учитель прекрасно объясняет материал, затем дает интересные задачи или создает проблемные ситуации, чтобы закрепить этот материал, он прекрасно организует только один тип взаимоотношений — «определяющий» — и упускает из внимания другой — «зависимый», когда ученикам крайне необходимо обсудить этот материал между собой.

Так и получается, что знания, приобретенные на уроке, не становятся для учеников личностно значимыми, своими, а остаются необходимыми только в учебном процессе; в обыденной жизни они не используются.

Отсюда и ответ на ставший уже традиционным вопрос, почему школьник хорошо знает, как следует себя вести, но не ведет себя так. Нравственные знания просто не стали для него значимыми, своими, так как за пределами школы идет обсуждение других позиций и ценностей.

Вышесказанное позволяет отметить огромную важность организации учителем на уроке обсуждения между учениками новой учебной темы или нравственного вопроса, когда ученики беседуют не с учителем, а друг с другом на «заданную тему». Это возможно в процессе группового поиска наилучшего ответа, спора между двумя (или большим количеством) группами учеников класса или с учениками из параллельного класса и т. д. Здесь учитель играет роль направляющего, но не активного участника общения школьников.

Следует отметить, что в многочисленных психолого-педагогических исследованиях доказано, что даже в процессе помощи друг другу ученики лучше усваивают новый материал.

Таким образом, тот факт, что учитель информатики меньше общается с учениками на уроке, чем учителя других предметов, свидетельствует о том, что сама ситуация преподавания предмета ставит учителя в такие условия, когда ему необходимо использовать знания динамики общения школьников.

Урок информатики должен состоять, как минимум, из двух основных частей.

В первой части учителю необходимо ясно, просто и доступно объяснить материал и проверить правильность усвоения учениками его основных моментов.

Вторую часть урока нужно посвятить обсуждению нового материала между учениками. При этом они могут уже начать выполнять работу на персональных компьютерах. Важно, чтобы школьники самостоятельно использовали новые сведения и имели бы право советоваться друг с другом. Учитель же в этой части урока — координатор, консультант по сложным вопросам, но не активный участник общения. «Деловой шум» в этой части урока просто необходим.

При такой организации общения учителя и учеников на уроках информатики создаются условия наилучшего усвоения учениками новых знаний.

Знания, полученные таким образом, становятся для ученика личностно значимыми, своими. Это позволит создать потенциал для развития творческих способностей школьников, самостоятельности суждений, которые так необходимы современному человеку.

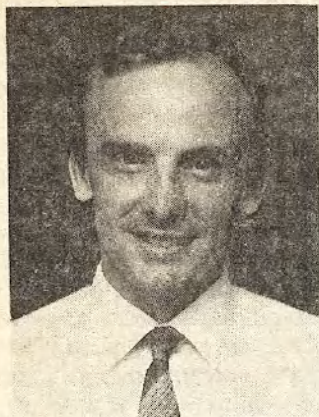
Л и т е р а т у р а

1. Воробьева Н. П. Возрастные особенности общения школьников с позиций концепции персонализации // Психологические проблемы взаимодействия коллектива и личности // Под ред. А. В. Петровского. М. 1988. С. 122—129.

2. Воробьева Н. П. Возрастная динамика общения школьников: Автореферат дис. ... канд. психол. наук (19.00.07). М., 1989. 21 с.

3. Петровский А. В. Проблема развития личности с позиций социальной психологии // Вопр. психологии. 1984. № 4. С. 15—29.

Система курсов информатики



— Борис Григорьевич, расскажите нашим читателям какие идеи лежат в основе проекта? Каковы его отличительные черты? Что нового принесет он в школы?

— Прежде чем отвечать на эти вопросы, я хотел бы несколько слов сказать о специфике того этапа информатизации образования в нашей стране, в начале которого мы сейчас находимся. I этап — 60—85 гг., II — с 1985 г. по настоящее время, а сейчас абсолютно четко можно выделить признаки, позволяющие утверждать, что наступает III этап. Во-первых, в значимых масштабах в школы начинает поступать вычислительная техника другого поколения, я имею в виду IBM — совместимую. Во-вторых, в обществе и в образовании сменилась парадигма, ориентирующая на использование компьютеров. Если на I и II этапах работала парадигма: хочешь использовать компьютер — освой язык программирования и отсюда тезис: «Программирование — вторая грамотность», то сейчас во всем мире и у нас ситуация резко изменилась, а именно: «Хочешь добиться успеха в профессиональной деятельности — используй информационные технологии!»

Проект Гособразования СССР и фирмы IBM «Пилотные школы» разворачивается во всех республиках. В редакцию приходят письма, в которых читатели просят подробнее рассказать о проекте, о той экспериментальной работе, которая ведется в «Пилотных школах». Мы обратились с этими вопросами к научному руководителю проекта Б. Г. Киселеву.

191

Это принципиально другой подход, при котором на первое место выдвигается не компьютер, и тем более, не программирование, а НИТ в различных областях человеческой деятельности. И этот подход окажет, на мой взгляд, существенно большее влияние на процессы, происходящие в нашей школе, чем качественное изменение технических возможностей.

В-третьих, для нового этапа характерно, что в школах не просто устанавливаются компьютеры — вместе с ними поставляются и другие элементы НИТ, которые позволяют учителям и администрации сразу использовать их эффективно.

Я хотел бы подробнее остановиться на том, что именно я понимаю под НИТ, так как проект ориентирован на использование НИТ в средней общеобразовательной школе.

Обычно под понятием НИТ подразумевают совокупность методов и средств обработки информации, доступных самому широкому кругу пользователей. НИТ ликвидирует барьер между пользователем и системой переработки информации. Следует отметить, что по-

нятие НИТ на замыкается только на компьютер, а включает в себя всю совокупность методов и средств, ориентированных на сбор, хранение, передачу, обработку и представление информации. Можно сказать, что НИТ концентрируют в себе значительную часть достижений науки и техники последних десятилетий и делают эти достижения доступными практически для каждого человека.

Реальными воплощениями НИТ, которые оказывают влияние на сферу образования, являются интегрированные системы обработки информации (например: WORKS, FRAMEWORK, MASTER) и многоканальные системы сбора и представления информации или системы МУЛЬТИМЕДИА. Как правило, подобные системы совместимы друг с другом по информационному интерфейсу, что позволяет при решении практических задач подобрать средства, соответствующие реальной сложности и трудоемкости процессов переработки информации.

Идеология использования НИТ в образовании может быть рассмотрена в трех плоскостях: использование НИТ непосредственно в информатике, в других предметах и для подготовки и переподготовки педагогических кадров.

Рассмотрим НИТ в информатике. Основная направленность содержания курса, реализованного в проекте, — это освоение знаний и навыков, дающих возможность уже в школе использовать НИТ при изучении других дисциплин, с одной стороны, а с другой — подготовка ребенка к жизни в сложном, быстро меняющемся мире, обучение его современным методам решения задач и подходам к их решению, знакомство с уровнем современных технологий. У нас разработана программа, с которой можно познакомиться.

— С какого класса следует начинать изучение такого курса?

— Мое мнение, чем раньше — тем лучше. Я не вижу принципиальных препятствий, устанавливающих нижнюю возрастную границу освоения элементов НИТ. Как правило, речь идет о конкретных системах и методах их использования при решении практических задач. Если мы говорим о компьютере —

то это устройство, с которым человек общается на пользовательском уровне. Работа на компьютере и работа с простейшими интегрированными пакетами, как показывает опыт, доступна с самого младшего возраста. Но главное, что наиболее характерно, интегрированные пакеты — это не отдельно взятые текстовой или графический редакторы, базы данных, электронные таблицы, а именно целостные системы, позволяющие переходить от работы с графикой к работе с электронными таблицами, включать полученные результаты в отчет или доклад, подготовленный с помощью текстового редактора и т. д.

Здесь напрашивается уже принципиально иная организация учебной деятельности, я бы ее назвал проблемно ориентированной. Кроме того, в таких условиях ребята могут организовать совместную работу, реализовать коллективный проект. В младших классах это может быть стенгазета, в старших — исследования, связанные с экологией, с социальными проблемами.

В плане решения задач содержание курса должно быть ориентировано на моделирование в широком смысле. Возможность моделирования я ставлю на первое место, потому что именно это позволяет использовать компьютер как принципиально новый инструмент познания окружающего ребенка мира. Фактически у школьника первый раз появляется реальная возможность ответить на вопрос: а что будет, если...? Он может рассчитать и посмотреть, что будет, если взорвется атомная бомба, если повернуть реки вспять... В нашем проекте есть система, моделирующая на доступном школьнику уровне проблему озера Арал. Он может наблюдать процесс высыхания озера, имея возможность вмешаться. Он может регулировать объем воды, выделяемый на полив, и наблюдать, к чему это приводит: если воды на полив меньше, снижается урожай хлопка, и неизбежны экономические потери и т. п. Фактически компьютер позволяет ему заглянуть в прошлое и из этого прошлого моделировать настоящее и будущее. Это то новое, что принес компьютер в школу. Современные информационные технологии в образовании делают подобные задачи

доступными на определенном уровне, конечно, для школьников, что очень важно для их нравственного и социального воспитания.

Итак, в курсе информатики, реализованном в проекте, есть три основных компонента: *знакомство с компьютером, знакомство и работа с интегрированными типовыми системами и моделирование*. И чем раньше эти три компонента будут освоены школьниками, тем лучше, потому что такой подход позволяет дать ребенку методологию. По отношению к ребенку это не очень благозвучное сочетание, тем не менее таков единственный цивилизованный путь подготовить его к деятельности в современном обществе. Пока в рамках проекта этот базовый, как мы его называем, курс информатики изучается в VIII—IX классах.

Описанный базовый курс на самом деле является введением, в котором ребенок знакомится с методами, подходами, принципами. Использовать их он должен в рамках конкретных предметов, с учетом их специфики. С одной стороны, углубляясь в изучение других предметов на новой основе, ребенок более полно овладевает НИТ, т. е. эти процессы тесно взаимосвязаны. Но эффективно они пойдут только тогда, когда будут реально пересмотрены традиционные содержание, методы, формы среднего образования. В настоящее время эффективно реализовать эту связь мы можем только с VIII класса, потому что еще мало техники.

— *На мой взгляд, для этого не хватает подготовленных педагогов, да и школа не готова к таким глубоким реформам...*

— Проблему подготовки кадров мы рассмотрим ниже. А сейчас я хотел бы добавить, что кроме базового курса информатики в школе, вообще говоря, должна быть представлена система курсов информатики, в которую курс, о котором мы говорим, входит одним из компонентов. А кроме него — курс углубленного изучения информатики, ориентированный на ребят, интересующихся, что внутри компьютера, как заставить его делать все, что захочешь? В школе всегда есть любознательные, и им не достаточно

пользовательского подхода. Углубленный курс можно использовать в спецшколах, лицеях, гимназиях, во внеклассных формах работы и т. п. Дополненный изучением языков программирования, он должен обеспечить ребенку знакомство с тем, что мы называем основами НИТ и в области техники, и в области математики. Это основа для будущих специалистов в этих областях.

Теперь о проблеме педагогических кадров. Мое глубокое убеждение, что наряду с описанными выше курсами информатики в школы должен поставаться курс информатики, ориентированный на учителя, для учителя. Это должен быть не изолированный курс, а подерживаю, и это очень важно, а курс, включенный в эту систему курсов, чтобы система представлений о НИТ в базовом курсе для учеников перекликалась с системой представлений, знаний и навыков, которые даются учителям. Курс для учителей тоже должен иметь практическую направленность. Это не курс подготовки к ведению курса информатики для учеников, а курс, ориентирующий учителя на использование НИТ в своей повседневной деятельности.

— *Борис Григорьевич, не кажется ли Вам, что такой курс нужно изучать в педагогических вузах, а не в школе?*

— Отчасти я с этим согласен, и безусловно, такая подготовка обязательно должна осуществляться в педагогическом вузе, но с другой стороны, НИТ — это ведь не нечто статическое. У нас за пять лет ситуация в корне изменилась, а что будет еще через пять лет и представить трудно! Прогнозы — дело сложное. Я говорю о постоянной поддержке современного уровня информационной культуры учителя. Именно о постоянной, потому что остановка губительна: человек безнадежно отстает. Другое дело как организовать преподавание такого курса методически, технически и т. п.? Здесь можно обсуждать различные варианты, например, в живописной обстановке собираются единомышленники, обсуждают эти вопросы. Но в рамках проекта мы эту проблему решаем.

Я должен отметить, что у учителей

очень сильна «внешняя» мотивация, она «подталкивается» самой жизнью. И во всем мире зачастую информатизация образования идет не благодаря учителям, а из-за требований родителей.

Если вернуться к вопросу об использовании НИТ в других предметах, то в рамках проекта мы развиваем следующие направления.

Во-первых, непрерывность. В использовании НИТ в учебном процессе не должно быть перерывов. Причем в рамках большого совместного проекта с фирмой IBM НИТ будут использоваться не только в школе. Непрерывность на самом деле — очень актуальная проблема. Если человек не поддерживает постоянно достаточно высокий темп собственного развития, успеха можно и не добиться. Есть проекты по детским дошкольным учреждениям и по вузам, и конечно хотелось бы, чтобы непрерывность и преемственность были обеспечены.

— *Может быть из таких детей надо формировать отдельные группы и обучать их по специальным программам?*

— Да, такие группы надо создавать, но здесь возникают другие проблемы — требуются совсем другие педагогические кадры. На самом деле проблемы, связанные с непрерывностью очень сложны, и хорошо, что ими сейчас начинают заниматься. Но мы стремимся к тому, чтобы это не было очередной кампанией с раздачей наград, а чтобы был результат.

Во-вторых, мы работаем над новыми курсами, в частности, над математикой. Сейчас этот курс имеет ярко выраженную алгоритмическую направленность. Хорошо известно, что существуют пакеты программ, которые эффективно решают не только задачи школьного курса математики, но и вузовского. У ребенка возникает справедливый вопрос: почему я работаю как машина и трачу свое золотое время? Простого ответа на этот вопрос, вообще говоря, нет. Но изменение курсов — это предмет особого разговора...

— *Можно надеяться, что наши читатели смогут в публикациях о деятельности центра подробно познакомиться с этими работами?*

— Да, конечно. Новые информационные технологии приходят в школу — и это девиз проекта. Пересматривается содержание практически всех предметов, выявляются очень интересные межпредметные связи, формируются новые предметы, например, экология, социология, экономика, в которых знания, умения, навыки, полученные ребятами при изучении классических дисциплин, активно используются для исследования сложных современных проблем. Это уже качественный скачок в образовании.

— *Я думаю, что изменение содержания школьного образования должно происходить в сторону расширения гуманитарной составляющей, ориентации на духовное и нравственное развитие личности.*

— Это отдельный вопрос, и я много над ним думал. Ведь учебный и воспитательный процессы неразрывно связаны, о воспитательной стороне никогда нельзя забывать. Нужно помнить, что если технология, особенно в современных условиях, стремительно меняется и эволюционирует, то культура — это, вообще говоря, некая инварианта в развитии человеческого общества, и наша задача в том, чтобы технологии не приводили к деградации культуры. Включение НИТ в образование ни в коем случае не должно идти за счет воспитания, а наоборот, должно способствовать духовному развитию ребенка, знакомству с общечеловеческими культурными ценностями. НИТ позволяют это, формируя широкий кругозор, глобальное мышление, разносторонние подходы к решению задач, и это, на мой взгляд, очень существенно. НИТ позволяют подходить к решению задач не только с технической, но и с нравственной стороны, с точки зрения общечеловеческой культуры. Очень хорошо, что Вы затронули эту проблему, так как здесь есть еще один аспект, очень важный лично для меня.

Одна из проблем, стоящих перед нашей школой — стандартизация подхода к учебному процессу. Но нельзя обучать личность по какому-то стандарту. Понятия личность и стандарт — несовместимы. Перспектива развития школы — возможность дать каждому ребенку проявить себя наиболее полно, а это

можно реализовать только в том случае, если у ребенка есть возможность выбора — он может выбрать школу, курсы, учителей. Это возможно только при наличии многообразия. Сегодня это не представляется возможным. Так вот, одним из средств, позволяющих помочь в решении этих проблем, и являются, на мой взгляд, новые информационные технологии.

В проекте существенное внимание уделяется концепции технологии разработки программно-методических комплексов (ПМК) — ядра НИТ. Опыт показывает, что в большинстве случаев, когда речь идет о компьютерах, срабатывает стереотип: раз используем компьютеры, значит в конце концов делаем программу. Отличительной чертой III этапа информатизации образования является осознание того факта, что методика использования программных средств, методическое обеспечение существенно отстают от возможностей, которые предоставляют для образования НИТ. Это факт, что методик практически нет. Мы считаем, что должны разрабатываться программно-методические комплексы, которые бы давали ориентиры и помогали учителю использовать НИТ в учебном процессе.

Отдельный вопрос — разработка инструментальных средств для учителя, в частности для учителя-предметника. Это должны быть средства, ориентированные на подготовку курсов, тем, уроков. Нами разработан пакет по математике, который демонстрировался на ВДНХ СССР. Он очень легко и быстро осваивается. При создании подобных средств можно решить в первом приближении задачу многообразия. Мы начинаем с разработки инструментальных средств и планируем разработать ПМК по всем предметам.

В ПМК должны, естественно, входить некое программное средство, или элемент собственно НИТ и так называемая методическая оболочка, включающая материалы, предназначенные как для ученика, так и для учителя. Я вкладываю следующий смысл в понятие «методическая оболочка» — это то, что определяет или составляет среду. В современной информатике синонимом тер-



мина «оболочка» является термин «среда». Мы говорим, например, о среде Турбо-Паскаля. Оболочка — то, во что «завернуто» базовое программное обеспечение, сам компьютер. В этом смысле пользователь общается именно с оболочкой, а не с компьютером. Пользователь-учитель должен общаться не непосредственно с программой, а с методической оболочкой, содержание которой должно пробуждать и стимулировать его творческую инициативу, педагогическую фантазию, побуждать к новым методам и формам работы. Один интерфейс этой оболочки, как я уже говорил выше, должен быть ориентирован на ученика, а другой — на учителя. Учитель, используя эту методическую оболочку, и должен адаптировать программное средство к своей личности, к личности ученика, к конкретным условиям учебного процесса. К сожалению, сегодня мы сталкиваемся в основном с жесткими инструкциями по использованию программных средств, и не более того.

На мой взгляд, чтобы решить эту проблему, а мы именно так и поступаем в проекте, нужно ввести понятие «технологический цикл» разработки программно-методического комплекса.

Что я понимаю под технологическим циклом? Любая разработка, связанная с внедрением НИТ в обучение, должна инициироваться, исходя из требований учебного процесса, и заканчиваться апробацией в учебном процессе. Получается как бы итерационный цикл, потому что в процессе апробации возникают новые требования и процесс совершенствования, в принципе, может быть достаточно долгов.

Технологический цикл поэтому должен строиться таким образом, чтобы уже на первом этапе разработки ПМК ставились и формулировались бы методические и дидактические цели.

— *Кем, на Ваш взгляд, они должны быть сформулированы?*

— Вообще говоря, Учителем с большой буквы или опытным методистом.

К сожалению, очень часто подобного рода разработки ведутся, исходя из представлений программистов о содержании и методах обучения, и, может быть, этим объясняется, что многие разработки не получают поддержки. Из 10 разработок 9 оказываются практически не нужными, хотя на них затрачено много сил, времени и средств. Эти вопросы тесно связаны с вопросом об экспертизе. На мой взгляд, она должна быть многоступенчатой и включать начальный этап — оценку и анализ заявки, т. е. начальную экспертизу.

В таком технологическом цикле акцент должен быть сделан на разработке методического обеспечения, которая является более трудоемкой, чем разработка программного средства, но, как я уже говорил, именно эта сторона определяет жизнеспособность и полезность ПМК. Если заявку пишет учитель, то методику должен разрабатывать методист, который представляет сложность проблем, с которыми сталкивается школа, и обладает достаточно широким кругозором, чтобы определить, как именно должен и может использоваться данный программный продукт в городской, сельской школе и т. п. Это и есть методическая оболочка, которая позволит адаптировать программу к личности каждого конкретного учителя. Это очень важный момент.

— *И очень дорогой.*

— А все то, о чем мы говорим, вообще дорого. И это подчеркивал Альфред Борк, одна из статей которого была опубликована в «ИНФО». В 1985 г. он говорил, что создание программно-методического комплекса стоит около 300 000 долларов, а на I-ом Международном симпозиуме, ГК НО СССР и ИВМ в 1990 г., который предварял наш проект, он уже называл цифру 1 млн. долларов. Вообще говоря, делать надо только то, что действительно необ-

ходимо. Да, это стоит дорого! Поэтому прежде, чем приступать к подобным проектам, нужно тщательно проработать их технико-экономическое обоснование. Надо реально представлять, что будет вложено и что получено. «По дешевке» хорошее образование наладить очень трудно. В этом нужно отдавать себе отчет, тем более что современные информационные технологии сами по себе капиталоемки.

Разработку методики и программных средств надо вести параллельно, и мы стараемся этот принцип выдерживать. Реально это трудно, потому что еще нет достаточного числа квалифицированных методистов. На выходе технологического цикла обязательно должен быть проведен психолого-педагогический эксперимент. Должна пройти апробация ПМК в условиях реального учебного процесса плюс адаптация ПМК к конкретным условиям его эксплуатации. Лично для меня, а я с уважением отношусь к экспертным комиссиям, важно мнение ученика и учителя, работающего с ПМК. Мой опыт показывает, что положительные оценки комиссии школы совпадают только для относительно хороших программных продуктов. Более жесткими являются оценки учеников и учителей. В конце концов все становится видно в классе: нужно — не нужно, удобно — неудобно, работает — не работает. После апробации в условиях эксперимента ПМК можно тиражировать в реальную школу.

Касаясь такого важного вопроса, как содержание методической оболочки (а это предмет отдельной статьи и дискуссии), я считаю, что в нашем проекте реализуется новый подход.

Эффективность той или иной системы определяется тем, насколько она интегрирует возможности, которые необходимы для передачи ученику соответствующих знаний, умений, навыков. Ведь ученик в процессе обучения погружается в определенную среду, в которой есть различные каналы передачи информации: зрительный, слуховой и т. д. Ученик работает в среде с реальным объектом изучения (в лабораториях), с базой знаний (книга, знания учителя и т. д.). Современные информационные техно-

логии еще не позволяют интегрировать все элементы учебного процесса, пока таким интегрирующим началом является сам учитель. Если на I и II этапах информатизации образования элементы НИТ использовались в основном как хороший канал передачи информации, причем достаточно мощный, эффективный и с обратной связью (классические обучающие программы), то сейчас можно интегрировать, например, модель и базу знаний. Степень интегрированности — это тот параметр или критерий, если хотите, по которому можно сравнивать, на мой взгляд, различные разработки. Создать метасистему, которая бы интегрировала все элементы среды обучения, не представляется пока возможным, но то, что люди будут упорно пытаться создавать такие системы — бесспорно.

В этой проблеме есть еще один, очень важный исследовательский аспект. Сейчас фактически открылся новый Космос. Если раньше человек познавал окружающий мир, то сейчас, ис-

следуя процессы познания, мы на самом деле исследуем прежде всего человека, его личность, процессы, проходящие в нем самом. И этот Космос с одной стороны, неисчерпаем, а с другой — абсолютно непознан. Ведь когда создаешь подобного рода интегрирующие системы, ПМК — основываешься на определенных концепциях, а каких-то выверенных, адекватных концепций познания я, по крайней мере, не знаю. Есть только теории, подчас взаимоисключающие друг друга: когнитивная, бихевиористская, теория поэтапного формирования умственных действий и др. Задачи, связанные с процессом обучения — прежде всего исследовательские задачи. Поэтому я думаю, что люди будут стремиться создавать такие системы, чтобы понять процесс познания, и компьютер — существенная веха на этом пути. Чем больше мы будем об этом знать — тем эффективнее будет система образования в целом.

25

Беседу провела А. Кравцова

Учебно-методический центр «Информвидео»

при Московском городском институте усовершенствования учителей

приглашает учителей информатики

на курсы по научно-методическим основам преподавания информатики по машинному варианту.

Курсы в объеме 72 ч проводятся в каникулярное время в течение 10 дней: ноябрь 1991 г., январь, март, июнь, август, ноябрь 1992 г. и т. д.

Обучение проводится на различных видах ВТ: УКНЦ, КУВТ-86 с Бейсином и Фокалом, БК-0011, «Корвет», «Агат», «Ямаха», IBM-совместимые компьютеры.

Слушателям курсов передаются программно-методический комплекс, включающий новые сетевые программные средства для УКНЦ, КУВТ-86, БК-0011, «Ямахи» и методические пособия по преподаванию информатики.

Стоимость курсов — 820—870 рублей (обучение, программно-методический комплекс на дискетах УМЦ).

Слушатели обеспечиваются общежитием.

Желающие пройти обучение должны прислать гарантийное письмо об уплате с указанием желательного времени обучения и вида ВТ. Учебно-методический центр вышлет вам приглашение и счет. Оплата производится до начала курсов по безналичному расчету.

Адрес УМЦ «Информвидео»:
125167, Москва, Авиационный пер., д. 6, к. 205.
Телефон для справок: 151-59-31.

Международный семинар в Москве

Чтобы преуспеть на международном рынке, очень важно вовремя и умело скооперироваться.

(Из выступления г-на Б. Сундеби, президента фирмы IST)

26 В июне 1991 г. в Московском городском комитете по народному образованию состоялся международный семинар «Информатизация 91—92: интеграция информационных технологий». Организаторы семинара — МГК по НО, НЦПСО, Министерство образования Швеции, шведская фирма IST и американская фирма «PROXIMA» — предложили интересную программу. Фактически на обсуждение собравшихся, а в их число вошли представители системы дошкольного воспитания, профессионального образования, районных отделов народного образования, творчески работающие учителя, представители Государственного комитета по народному образованию СССР и Министерства народного образования РСФСР, была представлена концепция информатизации образования и сферы управления образованием Москвы. В ее основе лежит совместный международный проект, осуществляемый организаторами семинара.

Обозначая проблемы, стоящие перед МГК по НО, заместитель председателя комитета Л. Е. Курнешова отметила, что в ближайшие 2—2,5 года в Москве будет осуществляться программа информатизации образования, в ходе реализации которой эти проблемы будут решены. Главные положения программы: техника, используемая в системах общего и профессионального образования, дошкольного воспитания и сферы управления, должна отвечать современному уровню, быть совместимой на аппаратном уровне и способствовать осуществлению единой политики в области технологии изгото-

товления и распространения программных средств.

Проект имеет статус экспериментального. На первом этапе в московские школы поступит 130 комплектов IBM PC-совместимой техники и несколько комплектов — в систему управления. Школы и учреждения, участвующие в проекте, будут определены МГК, и не исключено, что в некоторые школы поступит более одного комплекта. Проект рассчитан первоначально на два года и одобрен комиссией по образованию РСФСР. В конце 1991 г. закончится автоматизация управленческой деятельности МГК и сеть объединит компьютеры для создания информационно-аналитического банка ведения делопроизводства, анализа статистических данных и т. п.

О работе над проектом шведской стороны подробно рассказал президент фирмы International Software Trading (IST) г-н Б. Сундеби.

Фирма IST — одна из ведущих фирм в Скандинавии по созданию и распространению программных средств для целей образования, включая управленческую деятельность, отвечающих самым современным технологиям и требованиям. Фирма имеет филиалы во многих странах Европы. Г-н Б. Сундеби подчеркнул, что в сотрудничестве с советскими коллегами из НЦПСО его привлекает общность на философском и концептуальном уровнях понимания проблем развития современных информационных технологий в образовании. Результаты сотрудничества соответствуют международным стандартам, перспективное направление совместных исследований — создание и использование средств мультимедиа.

Аппаратную поддержку второго этапа проекта на уровне периферийных устройств (жидкокристаллические дисп-

лей для демонстрации изображения с компьютера на экраны различной площади с управлением от лазерной указки и т. п.) будет, по-видимому, осуществлять американская фирма «PROXIMA».

Как отметил член совета фирмы IST г-н Б. Викстрем, настало время и для СССР, в масштабе государства или отдельных регионов, кооперироваться с другими странами для осуществления прорыва в сфере образования с использованием того, что мы называем новыми информационными технологиями. Только таким образом мы можем войти в международное сообщество,

а совместный опыт работы IST и НЦПСО (фирмы уже вложили в проект немало средств) вселяет надежду на успех.

О концепции создания пакетов программных средств рассказали генеральный директор НЦПСО А. А. Зубченко и зам. генерального директора В. А. Урнов. Сотрудники центра продемонстрировали присутствующим ряд разработок. В 1991/92 учебном году эту работу смогут оценить учителя и ученики московских школ — участники экспериментального «Московского проекта».

А. ЗУБЧЕНКО

генеральный директор НЦПСО (CAIRC)

НЦПСО — современная индустрия программных педагогических средств

Постоянные читатели ИНФО, судя по объему переписки, превратившей наш секретариат в почтовое отделение, очевидно, заметили регулярную рекламу пакетов программных средств Научного центра программных средств обучения (НЦПСО). Интерес непосредственного потребителя и пользователя — факт, отрадный для любого производителя, а для нас он отраден вдвойне, но не только по коммерческим или каким-либо иным прагматическим соображениям. Другие обстоятельства подкрепляют нашу уверенность в правильности главных направлений деятельности НЦПСО.

Потребность в организации индустрии педагогических программных средств (ППС) проявилась с первых шагов информатизации образования, но и сейчас еще рано говорить о том, что такая индустрия существует где-либо в мире, и тем более приходится только надеяться на то, что она появится в нашей стране. Чаще всего в качестве главного препятствия к созданию индустрии ППС называют чрезвычайную дороговизну технологий, обеспечивающих раз-

работку программных систем для образовательных целей. И это действительно так. Например, интерактивная многосредовая система Info Window (разработка фирмы IBM, 1984 г.) адекватно отвечает потребностям образования, но она очень дорога и пока еще недоступна для большинства школ США. Кроме того, трудно разрешимы организационные задачи объединения усилий специалистов сопредельных областей — от электроники до гуманитарных наук, участие которых совершенно необходимо на разных этапах разработки и реализации ППС и технологий, лежащих в их основе.

Тот, кто постоянно на протяжении последних 5—6 лет следит за ходом информатизации образования, уже должен был привыкнуть к шуму бурного потока созидания новых и «самых новых» центров новых информационных технологий (НИТ). Безусловно, этот процесс, как и всякий процесс вообще, имеет свою положительную сторону. Она состоит в том, что чем больше будет создано центров НИТ по той методике,

по которой они создаются, тем быстрее выводится полная несовместимость производственных и организационных структур поддержки НИТ и принципиально потребительской структуры научных и учебных организаций, на базе которых они размещаются. Лучший, на наш взгляд, выход из этого положения может состоять только в том, что, избавляясь в целях самосохранения от указанного противоречия, научные и учебные организации сферы образования получат достаточный опыт освоения НИТ и станут грамотными и подготовленными пользователями и потребителями, участниками и регуляторами рынка информационных технологий, систем и ППС. На сегодняшний день скорее всего именно такой результат введения НИТ в сферу образования может и должен служить показателем качественных изменений собственно образования.

Таким образом, общая ситуация в информатизации образования (при минимальном финансировании и отсутствии широко осознанной конкретной потребности в конкретной информационной технологии для целей образования) вовсе не служит стимулом для развития не только целостной индустрии ППС, но даже и отдельного предприятия. И тем не менее уже в течение двух лет учителя информатики Москвы и других регионов страны активно пользуются целым рядом ППС и технологий, созданных от начала и до конца в виде законченного продукта в НЦПСО. Следовательно, и в тех условиях, о которых говорилось выше, хотя и не должен, но может существовать разумный промышленный механизм производства ППС.

В 1988 г., к моменту создания НЦПСО, информатизация школ проходила по очень простой схеме: 99 % успеха определялись фактом приобретения техники, если это был КУВТ-86, УКНЦ или «Агат»; и 100 % — если это был класс «Ямаха». Проблема ППС зависела от творческих способностей учителя информатики формировать «компьютерную грамотность», обучая школьников программированию на любом доступном учителю языке, будь то Фокал, Бейсик или школьный алгоритмический

язык. Такое положение не могло сохраняться долго, и без всякой критики понятно, что «программистский вариант» информатики не решает задач общего среднего образования.

В качестве первого шага, предварившего создание НЦПСО, в Москве была сделана удачная попытка объединения усилий групп разработчиков, которые при естественном интересе к новому, на чистом энтузиазме, достаточно профессионально занимались информатикой на разных видах учебной техники, в том числе и непосредственно в школах. Эти группы разработчиков и школы, в которых они работали, стали базовыми точками инфраструктуры информатизации Москвы. Используя хозрасчетную основу финансирования этой инфраструктуры, руководители НЦПСО к концу 1989 г. отчитались перед Московским комитетом по народному образованию 15 пакетами ППС, поставленными в школы города, и действующей на хозрасчете городской службой сервиса и ремонта учебной ВТ. Последнее обстоятельство очень важно, поскольку отсутствие оперативно контролируемой службы сервиса и ремонта при известном качестве и надежности отечественной техники сводило бы почти на нет эффект применения новых ППС.

Совместная работа многих групп разработчиков, которые на собственном опыте осваивали всю глубину афоризма «не программа работает в машине, а машина работает в программе», позволила довести в 1991 г. количество ППС до 30 для всех видов отечественной техники. К стандартным пакетам поддержки курса информатики добавились пакеты обучающих программ по некоторым предметам, а также аппаратные средства широкого назначения. Среди последних следует отметить долгожданный программно-аппаратный комплекс «Локальная сеть «Агат» и игровой пакет с аппаратной поддержкой джойстик/контроллер для всех видов отечественной учебной ВТ. На Всесоюзной выставке «Новые педагогические программные средства для средней школы», проведенной Госкомнаробразом СССР на ВДНХ СССР в декабре 1990 — январе 1991 г., два указанных продукта, а также ППС

«Интегрированная среда ЛОГО» для УК-НЦ, текстовый редактор «WRITER» (УК-НЦ), ППС «ИНФОРМАТИКА-90/VG» (УК-НЦ) и ППС «НАДЕЖДА» — знакомственный графический редактор для MSX-1,2 — отмечены серебряными медалями. Золотой медалью отмечен конструктор графических интегрированных сред «Дизайнер» для IBM PC/XT/AT, PS-2. Целый ряд авторских разработок, вошедших в состав того или иного ППС, за оригинальность и нестандартность решений получили свидетельства участников ВДНХ СССР, а деятельность НЦПСО отмечена дипломом I степени за разработку, создание, документирование, производство и тиражирование педагогических программных и программно-аппаратных средств.

Вместе с тем более существенным результатом деятельности НЦПСО, по нашим собственным оценкам, является принятая для разработчиков НЦПСО идеология создания информационных технологий для образования. Первый, уже реализованный, этап содержится в технологиях открытых переходных сред. Примером могут служить: система исполнителей «КуМир», разработанная в МГУ (А. Г. Кушниренко и др.); система исполнителей «Роботландия» (Ю. А. Первин и др.); среда обучения «ЛОГО» для УК-НЦ (НЦПСО). Отличительной особенностью переходных сред является их ориентация на формирование средств, при помощи которых учащиеся могут самостоятельно добывать новые знания, систематизировать уже полученные и конструировать новые интеллектуальные орудия, осваивая не только информатику, но и другие предметы на основе информационных технологий.

Второй этап развития информационных технологий, к которому были готовы разработчики НЦПСО, приобрел реальные очертания с реализацией проекта IBM — СССР «Пилотные школы». Проект открыл качественно новую перспективу информатизации образования, по крайней мере за счет более совершенной техники. Под этот проект для «пилотных» школ Москвы уже рас-

пространена одна из версий технологии «Дизайнер», представляющая комплексный инструментальный для конструирования графических интегрированных сред. На основе высокоуровневых пользовательских интерфейсов и за счет максимального использования «мыши» и графических возможностей компьютера каждый пользователь «Дизайнера» может построить собственную «субъективную» интегрированную среду.

Таким образом, принципиальное направление в развитии информационных технологий для целей образования определяется степенью открытости систем этих технологий. Действительно, открытость переходных сред ограничена системой команд того или иного исполнителя (СКИ), а возможности переходной среды исчерпываются полной совокупностью исполнителей и набором их свойств, которые могут быть сконструированы внутри этой среды. Любый активно работающий пользователь — учитель или ученик — может в этом случае решать достаточно широкий круг задач в разных предметных областях, начиная прежде всего с информатики.

В отличие от переходных сред, технологии типа «Дизайнер» не требуют освоения промежуточных СКИ. Это открытая система инструментов с «прозрачным» пользовательским интерфейсом, которая позволяет программировать в какой-либо содержательной области без написания единой строчки программы. Но и у таких, на первый взгляд универсальных технологий есть свои специфические ограничения. Пользователь такой системы должен очень хорошо владеть предметной областью и иметь ясные намерения и четко сформулированные цели применения информационной технологии для решения собственных содержательных задач.

Естественно, что только многообразный практический опыт работы с системами типа «Дизайнер», позволит определить место этих систем в образовании, однако для этого потребуется провести вполне определенную и трудоемкую работу по превращению вновь разрабатываемых систем и технологий в стандартное обеспечение для персонального компьютера. Какой именно

персональный компьютер должен применяться в образовании — это вопрос перманентный, но, видимо, в настоящее время IBM-подобная техника является лучшим выбором в плане стандартизации информационных технологий и организации совершенно необходимого в этом случае широкого международного сотрудничества в области разработок ППС.

30 В течение 1,5 лет НЦПСО (в международных проектах наша организация называется CAIRC (Computer-Assisted Instruction Research Centre) работает в тесном контакте со скандинавской фирмой IST (International Software Trading AB), которая сотрудничает с фирмой IBM в Европе в разработке программных средств для образования. Близость позиций НЦПСО и IST по техническим и технологическим вопросам, полное взаимопонимание в определении идеологии и направлений развития ППС позволили за короткое время получить существенные практические результаты. Дело в том, что в 1991 г. по инициативе Московского городского комитета по народному образованию и при поддержке исполкома Моссовета параллельно проекту «Пилотные школы» стартует региональный «Московский проект», по которому в московские школы будет поставлено 130 классов IBM PC/AT/XT. И впервые за весь период активной информатизации школы народное образование встречает технику нового уровня не с пустыми руками. Во многом благодаря сотрудничеству НЦПСО и IST помимо стандартного базового и сетевого обеспечения в поставку классов IBM включены ППС универсального и прикладного назначения фирмы IST, разработанные в международном стандарте SAA.

В составе ППС представлены оба уровня информационных технологий пакет «JUNIOR-1» — переходная

среда, по нашему определению, и пакет «TeachCAD» — открытая инструментальная среда. Мы надеемся, что их совместное использование в едином цикле обучения и на единой технической основе позволит получить практическое обоснование для определения новых направлений в развитии информационных технологий. Одно из таких направлений, подтверждающих эффективность открытых систем типа «Дизайнер» и «TeachCAD», прорабатывается во взаимодействии НЦПСО, IST и американской фирмы «PROXIMA». 3 июня 1991 г. на Международном семинаре по информатизации образования в Московском городском комитете по народному образованию фирма «PROXIMA» уже продемонстрировала комплект IBM-совместимого оборудования, работа которого в среде «TeachCAD» позволила увидеть значительные преимущества МУЛЬТИМЕДИА, полностью основанных на информационных технологиях. Этот тип МУЛЬТИМЕДИА открывает самые широкие возможности для информатизации любого учебного предмета в школе и комплексного использования новейших информационных аппаратных и программных средств.

Короткая, но чрезвычайно насыщенная деловая история НЦПСО доказывает, что при ориентации на возможное и достижимое, умение работать на любом «подручном» материале, правильном анализе и учете мирового опыта, высоком профессионализме «команды», состоящей из самых разных специалистов, можно не только быстро получить практически значимые результаты, но и научиться вести эффективные опережающие разработки новейших технологий обучения, адекватно формулируя как текущие, так и долгосрочные проекты информатизации образования.

SAA — пользовательский интерфейс, удовлетворяющий требованиям стандарта SAA (System Application Architecture).

TeachCAD — союз ученика и учителя

Одним из основных программных средств, с которым предстоит работать ученикам и учителям в рамках «Московского проекта» информатизации народного образования, будет TeachCAD — новый пакет для IBM PC-совместимых компьютеров.

Однако рынок программных средств для PC-компьютеров так богат и разнообразен, так заполнен высококачественной продукцией, что появление еще одного пакета должно быть по крайней мере объяснено. Что же собой представляет TeachCAD?

Ответить на этот вопрос нужно в двух аспектах: ЧТО позволяет пользователю делать TeachCAD и КАК.

Итак, TeachCAD — это мощный инструмент создания графических сред, демонстрационных и обучающих программ, использующий ряд нестандартных подходов и решений, разработанный с очень высоким качеством. Особого внимания заслуживает пользовательский интерфейс системы. Он дает возможность непрофессионалу создавать программы профессионального качества.

Как нам представляется, при разработке TeachCAD авторам удалось сделать шаг вперед в развитии идеологии и практики создания программных средств. Остановимся на этом немного подробнее.

Начнем с того, что существующие многочисленные программные средства в большинстве своем ориентированы на традиционные способы обработки информации. Они далеко не в полной мере используют возможности компьютера как принципиально нового средства работы с информацией и человеческого общения.

Действительно, текстовые процессоры лишь количественно изменяют традиционный процесс письма на бумаге, не изменившегося качественно со временем своего возникновения. Точно так же базы данных являются всего лишь электронным эквивалентом архива со множеством пыльных папок или клинописных



глиняных табличек. Разнообразные графические редакторы тоже имитируют традиционное рисование, сменив лист бумаги на экран компьютера.

Что-то совершенно новое, не имеющее аналогов в истории культуры, было создано лишь, пожалуй, в сфере индустрии развлечений (здесь на экране создается управляемая другая реальность).

Впрочем, попытки выйти на качественно иной уровень освоения возможностей техники делались неоднократно. Например, помочь человеку облечь свои мысли в насыщенную графикой форму живого отчета, гипертекста, рекламного ролика, чтобы донести их до других людей. Из наиболее известных программ в этой области можно назвать Link Way и StoryBoard, созданные по заказу фирмы IBM. Это же направление продолжает и развивает и TeachCAD.

Поиски исторически новых форм обработки информации тесно связаны и со стремлением превратить компьютер в полезное и привычное для большинства людей информационное средство, каким, скажем, стал автомобиль в качестве средства передвижения. Но этому препятствует ряд обстоятельств, которые постарались учесть авторы TeachCAD.

Первое из них — это проблема дизайна. Устройство должно быть хорошо знакомым и простым в управлении, чтобы любой человек мог им воспользоваться. Современный графический САА-стандартный интерфейс TeachCAD,

ориентированный на мышь, позволяет эффективно использовать опыт работы со стандартным программным обеспечением.

Второе обстоятельство, мешающее компьютеру стать «информационным автомобилем», — проблема быстродействия компьютера. Но это не столько техническая, сколько финансовая проблема. Распространено мнение, что только очень дорогой компьютер, с быстрым процессором и большим объемом оперативной памяти может поддержать современный графический пользовательский интерфейс. Сверхбыстрая графика высокого разрешения системы TeachCAD открывает доступ к созданию мощных графических программ даже на обычном PC/XT компьютере.

32

Трудность в нахождении общего языка с компьютером — еще одно серьезное препятствие на пути всеобщей компьютеризации. Язык программирования — нечто большее, чем система обозначений для команд, которые воспринимает компьютер. Язык вместе с программным обеспечением, которое понимает его, может полностью преобразить компьютер. И хотя за последние десятилетия процесс изготовления программ несколько удешевился и перестал быть делом немногочисленной касты компьютерных жрецов, но сложность и объем программ выросли в десятки и сотни раз. По-прежнему для занятия программированием требуются немалые усилия, знания и опыт, позволяющие увидеть будущую программу сквозь строки ее исходного текста.

TeachCAD последовательно придерживается принципа WYSIWYG (What You See Is What You Get — что вы видите, то вы и получаете). Это означает, что пользователь создает программу сразу в терминах конечного результата. Такой подход позволяет непрограммисту создавать программы профессионального уровня, широко использующие возможности современных компьютеров (в первую очередь графические), и без единой строчки на каком-либо языке программирования.

При традиционном подходе разработчик должен был бы написать на одном из языков программу, описываю-

щую процесс появления картинки, текстового окна или другого объекта в нужном месте и в нужном виде, и команды, связывающие объект с соответствующим действием пользователя, что заставляет мыслить о создаваемой среде не в непосредственно составляющих ее элементах, а в процедурных понятиях, описывающих структуру и поведение этих объектов. Так, например, расставляя мебель в помещении, человек описывает его на языке эскизов и чертежей.

Работая же в системе TeachCAD, вы, желая увидеть изображение кресла на экране в определенном месте, не задаете его координаты и не описываете это действие словами или в символическом каком-нибудь языке программирования, а просто берете кресло с помощью мыши и ставите его на желаемое место. Вы видите и воспринимаете образы именно так, как будете видеть и воспринимать их во время реальной работы вашей будущей программы.

В самом общем виде, TeachCAD позволяет создавать графические среды, легко управляемые движением мыши и нажатием на ее кнопки. Скажем, вы расставили на экране мебель, а затем распорядились, чтобы при наезде маркера мыши на кресло в шкафу открывалась дверца (т. е. изображение шкафа с закрытой дверцей заменялось на изображение шкафа с открытой дверцей), при нажатии на правую кнопку при этом появлялся список вещей, которые хранятся в этом шкафу, при нажатии же на левую кнопку, допустим, запускалась бы программа, демонстрирующая по очереди крупным планом сами вещи, указанные в списке, и т. п.

Несложно представить себе сюжет программы другого, в том числе учебного, содержания. Так, авторы системы для демонстрации ее возможностей за несколько часов подготовили обучающую программу по анатомии и физиологии человека. Скользя маркером мыши по изображению внутренних органов человека, пользователь получает на экране их названия, а нажав на кнопку мыши — более детальное их изображение и дополнительную информацию,



а также, простым перемещением мыши, — имитацию их функционирования.

Фантазия преподавателя любого предмета при создании такого, да и не только такого, рода обучающих программ с помощью TeachCAD получает очень большой простор. Учитель может как сам без большого труда создавать обучающие программы, так и поручить частично эту работу ученикам.

Можно проконтролировать и знания ученика, предложив ему сделать подобную программу по заданной теме, скажем по географии. А можно считать эту работу формой не контроля, но активного обучения. В самом деле, пока ученик — один или с друзьями — разработает свою программу, он досконально разберется в предмете — обучая других, обучаюсь сам (Learning Through Teaching). При этом он не только лучше узнаёт географию, но и продвигается по учебному материалу в своем темпе и в той мере, в которой это ему интересно и доступно, т. е. обучение индивидуализируется.

Кроме того, такую работу ученик будет, естественно, делать не только вместе с одноклассниками, но и вместе с учителем. Возникают почва для союза ученика и учителя, их новые отношения друг к другу. От противостояния по разные стороны учительского стола-баррикады, заложенного традиционной классно-урочной системой, они переходят к сотрудничеству на общей территории совместного созидания.

Это воспитывает в ученике инициативу и самоуважение, умение работать

не рядом, а вместе — качества, необходимые в современных условиях.

Основные возможности TeachCAD настолько просты и естественны, что доступны человеку любого возраста и профессии. Работая в TeachCAD, пользователь забывает о технических проблемах и думает лишь о конечном результате. TeachCAD дает возможность заняться изготовлением программ, не утруждаясь изучением программирования и других скучных вещей, ограничивающих фантазию.

Конечно, для многих людей рисовать ничуть не проще, чем программировать. Обширные библиотеки графических образов (так называемые clip-arts), входящие в состав пакета TeachCAD, позволяют взять быстрый старт: несколько движений мышью — и первая программа готова.

TeachCAD может заинтересовать и профессионального разработчика. Внешний вид интерфейса — один из ключевых моментов, определяющих качество всей программы. Однако эксперименты над сложным графическим интерфейсом требуют большого количества квалифицированного труда программиста. TeachCAD позволит быстро изготовить макет интерфейса, «живьем» увидеть взаимодействие его элементов на самых ранних этапах разработки. Более того, благодаря возможности вызывать из готового сценария произвольные .BAT и .EXE файлы, можно полностью построить интерфейс вашей программы с помощью TeachCAD, собрать пакет программ в единую среду.

TeachCAD выводит информацию в зависимости от логического уровня объекта, с которым производятся манипуляции. Логических уровней четыре:

- 1 — уровень фона;
- 2 — уровень зон активации;
- 3 — уровень маркеров движения мыши;
- 4 — уровень маркеров кнопок мыши.

Следует также отметить, что более высокий уровень отображает информацию, относящуюся ко всем более низким уровням. Таким образом, уровень зон отображает не только все зоны активации, но и основной фон с текущей

библиотекой образов, если она есть.

Первый логический уровень содержит несколько оригинальных и весьма полезных дополнительных функций. TeachCAD позволяет оформить загруженный фон (полноэкранное изображение в .PIC, .PCX или в собственном .PCI графических форматах) с помощью библиотеки образов (небольших изображений в .CUT-формате). Возможность манипулировать графическими образами также максимально удобна благодаря таким оригинальным особенностям, как прозрачный цвет — способность перемещать и располагать образы любой формы на исходном фоне; возможность тасовать образы, как колоду карт.

В результате работы на первом уровне вы можете создать и сохранить совершенно новый фон будущего сценария.

Второй и третий уровни (уровень зон активации и уровень маркеров движения) соответствуют «уровню движения мыши». Таким образом, используя зоны активации для указания на некоторый объект на фоне (объект задается по зоне активации) и вызывая к каждой зоне соответствующий образ — маркер, мы получаем своеобразный анимационный эффект, состоящий в следующем: когда курсор мыши попадает внутрь зоны активации — над предварительно указанным (но любым) местом экрана возникает соответствующий маркер.

Четвертый логический уровень — уровень кнопочных маркеров — это «уровень нажатия кнопки мыши».

При работе с информацией различной природы вам необходимо иметь

под рукой инструменты для работы с ней. Необходимость переключаться между компилятором, DOS, текстовым и графическим редактором, мучительно ждать завершения компиляции требует стальных нервов и может многократно замедлить работу.

Чтобы решить эту проблему, TeachCAD, кроме собственно среды разработки программ, называемой Designer, включает в себя:

мощный графический редактор Painter, поддерживающий чтение изображений из популярных форматов;

специализированный текстовый процессор Writer, ориентированный на оформление текстов, выводимых на экран;

не имеющее аналогов средство Imager, позволяющее вести библиотеки графических образов и оформлять экран методом аппликации, тасуя графические образы, как колоду карт;

встроенный интерпретатор, позволяющий в любой момент посмотреть, как выполняется разрабатываемая программа;

систему व्यюеров, позволяющую мгновенно увидеть содержимое любого используемого файла в максимально релевантном виде.

Таким образом, TeachCAD представляет собой интегрированную среду, содержащую мгновенно вызываемые встроенные редакторы и व्यюеры для всех типов данных, использующихся в системе; встроенный интерпретатор избавляет от ненужных простоев.

Кроме того, TeachCAD представляет собой настоящий компилятор. Результат его работы — единый .EXE файл, не требующий для выполнения ничего, кроме DOS.

Так как система TeachCAD была разработана в русле основных идей систем LinkWay и StoryBoard (хотя совершенно независимо от них), поясним, в чем их различия.

TeachCAD, будучи компилятором, выгодно отличается от LinkWay и StoryBoard, которые позволяют подготавливать программы, для исполнения которых необходимо иметь сами эти системы, а не только DOS, к тому же



в Story Board создается россыпь файлов, да еще используется некий собственный внутренний язык системы.

Далее, в процессе создания программы, чтобы просмотреть конфигурацию, вам не надо выходить из TeachCAD, он является и интерпретатором конфигурации, что очень удобно на стадии подготовки программы, а также, сохраняя в памяти конфигурацию и после ее компиляции и создания. EXE файла, оставляет возможность дальнейшей работы с данной конфигурацией, т. е. степень интерактивности очень высока. LinkWay же для исполнения созданной в нем конфигурации требует возвращения в DOS, что, безусловно, замедляет работу да и неудобно.

Одними из главных отличий являются, кроме того, несравнимо более мощный инструментарий (редакторы — они, кстати, могут быть использованы и сами по себе, вьюеры, имиджер) полностью «мышиный» интерфейс, сверхбыстрая графика высокого разрешения, ряд интересных принципиальных и технических решений вроде «прозрачного» цвета, работы с CUT'ами и др.

Итак, с помощью системы TeachCAD, не написав ни одной строчки на языке программирования, вы сможете успешно, с большим комфортом и лучшим качеством, чем работая в других системах, создавать программы профессионального качества, которые будут работать практически на любом IBM-совместимом компьютере и охватят многие области человеческой деятельности:

1. Обучающие программы.
2. Базы графических образов.
3. Рекламные ролики.
4. Обмен информацией.
5. Гипертексты.
6. Электронные учебники.
7. Живые отчеты.

8. Learning Through Teaching (обучая других, обучаюсь сам).

9. Разработка интерфейсов программ.

Работа над пакетом TeachCAD, потребовавшая сравнительно немного времени, но очень много напряженного труда, прошла ряд этапов. Первый Designer был спроектирован в 1989—1990 гг. в Научном центре программных средств обучения в Москве как среда для построения интерфейсов из готовых элементов. Немногочисленные совершенно необходимые средства редактирования (меблирование фона и прозрачный цвет) были добавлены лишь вследствие совершенного их отсутствия в графических редакторах.

В процессе работы над проектом в НЦПСО в 1990 г. была создана графическая библиотека, обладающая уникальным быстродействием.

В качестве базового графического редактора для первой версии Designer был выбран распространенный в СССР графический редактор Dr. Halo. В соответствии с этим был разработан и пользовательский интерфейс системы.

IST, продолжая развивать основные идеи вышеупомянутых инструментальных сред фирмы IBM — LinkWay и StoryBoard создала собственную систему — TeachCAD. Дополнив TeachCAD предложенными НЦПСО средствами редактирования, доработав интерфейс и использовав возможности графической библиотеки, представленной НЦПСО, фирма IST и разработала последнюю версию TeachCAD, которая была представлена выше.

Мы полагаем, что наступило время изменения роли компьютеров в школе как объекта изучения на уроках информатики на роль средства познания различных наук и социализации личностей школьников. И TeachCAD открывает поистине захватывающие перспективы для эффективного использования компьютеров в школе.

«Юниор-1» — решение проблемы индивидуализации обучения

В нашей стране и за рубежом стремятся познакомиться подрастающее поколение с компьютером, ставя при этом следующие задачи.

1. Развитие психических функций ребенка: мышления (в том числе алгоритмического), внимания, воображения, воли и т. д.

2. Формирование основных навыков пользователя ЭВМ.

3. Знакомство с информационными процессами в современном обществе.

4. Развитие творческих способностей учащихся.

5. Индивидуализация обучения.

6. Использование компьютера как средства познания.

Для решения этих задач как в целом, так и в частности разработан целый ряд курсов и отдельных программ. Характерной особенностью большинства из них является то, что, будучи созданы для школы, они придерживаются сложившихся традиций классно-урочной системы. В частности, в них широко распространены метод фронтальной работы преподавателя с группой учащихся, разработанный еще триста лет назад чешским педагогом Яном Амосом Коменским, основателем классно-урочной системы.

Сам Я. А. Коменский, прекрасно сознавал, что созданная им система не считается с различиями в способностях детей, необходимостью их поддержки и развития. В своем главном труде он писал: «...наш метод приспособлен для посредственных способностей (какие встречаются чаще всего), чтобы не было недостатка в тормозах для сдерживания более тонких натур... равно как в шпорах... для побуждения более медленных».

Очевидно, что для раскрытия индивидуальности ребенка, развития его способностей нужен иной подход, который позволил бы каждому продвигаться в своем развитии в соответствии с собственными возможностями, интересами и в своем темпе.



В этом ключе шведской фирмой IST на базе опыта НЦПСО и разработок для ПЭВМ «Ямаха» предприятия «Роботландия» был создан пакет «Юниор-1» для IBM PC-совместимых компьютеров, предназначенный для знакомства детей с основами компьютерной грамотности.

Пакет, в который входят комплект программ и книга для ученика («Руководство пользователя»), состоит из трех частей. В первой части объединены программы, связанные с понятиями «исполнитель» и «алгоритм»: «Словарь», «Перевозчик», «Ханойские башни», «Переливашка», «Конюх», «Чепуха». Программы второй части пакета связаны с теми же понятиями, что и программы первой части, но рассматриваются в ином аспекте — математика и принципы работы компьютера: «Угадайка», «Автомат», «Плюсик», «Байтик», «Логика». В третью часть пакета вошли программы «Правилка», «Фокусник», «Морской бой», «Мудрый кролик», «Опыт И. П. Павлова». Программы, входящие во все три части пакета «Юниор-1», объединены единым интерфейсом, соответствующим международным стандартам.

Книга для ученика устроена таким образом, что позволяет ему начать свою работу с любой программы любой части пакета. Иными словами, ребенку предоставлено право самостоятельно выбирать порядок знакомства с программами. Работа с самой книгой также не регламентирована: ребенок может про-

честь ее полностью, может обращаться к ней по мере необходимости, а может и вовсе не заглядывать в нее. Все это решается им самим в зависимости от индивидуальных вкусов и желаний. Книга насыщена иллюстрациями. В ней много вопросов к ребенку, непосредственных обращений, побуждающих его к действиям с компьютером. Есть и «подсказки», причем они «дозированы». Какой из них воспользоваться — более или менее подробной — решать самому ребенку. В книге содержатся исторические экскурсы, поясняющие происхождение некоторых терминов и задач. Возможно, они привлекут к работе с компьютером тех детей, чьи интересы лежат в областях гуманитарных знаний.

Старинные алгоритмические этюды, предлагаемые в первой части пакета, могут быть решены и без компьютера. Решая их с помощью компьютера, можно убедиться в тех возможностях, которыми он обладает: возможность повторить (проиграть) найденное решение, еще раз осмыслив его, начать решение сначала, отыскать оптимальное решение, найти ошибку и скорректировать алгоритм решения задачи на любой стадии и т. д. Программы первой и второй частей пакета способствуют развитию алгоритмического мышления и формированию некоторых навыков пользователя ЭВМ.

Возможность использования компьютера как средства познания реализуется, в частности, при моделировании опыта И. П. Павлова (программа «Опыт И. П. Павлова» из третьей части пакета).



Мы намеренно не указываем возраст, для которого предназначен «Юниор-1». Круг его адресатов очень широк, поскольку поработать с ним полезно всем, кто делает лишь первые шаги в знакомстве с компьютером. Задачи подобраны так, что решать их с помощью компьютера интересно и дошкольникам, и школьникам, и взрослым. Этому способствует выразительность и яркость экранного видового ряда, предложенного художниками, оформившими программы. Такая красочность создает положительный эмоциональный настрой, что немаловажно для продуктивной интеллектуальной работы. Для дошкольников и младших школьников это имеет особое значение, поскольку позволяет дольше удерживать внимание детей, что в этом возрасте является для них весьма трудной задачей.

Пакет «Юниор-1» задуман как средство индивидуального обучения ребенка — его собственно знакомства с компьютером. Иными словами, пакет предназначен не столько для обучения, сколько для самостоятельного овладения действиями, необходимыми пользователю ЭВМ.

Тем не менее индивидуальное, самостоятельное обучение детей отнюдь не противоречит современным принципам построения урока, хотя по ряду причин вписывается в него с большим трудом. Систематичность и успешность использования его возможностей во многом зависят от преподавателя: глубины его знакомства с индивидуальными особенностями своих учеников, принципов, которыми он руководствуется в предоставлении им свободы при овладении программным материалом, его мастерства в целом.

Таким образом, кроме того, что предлагаемый пакет «Юниор-1» можно использовать во внеклассной работе, он может быть полезен учителю и на уроке. Его применение предоставляет ученикам возможность работать на уроке самостоятельно, в удобном для каждого темпе. У учителя благодаря этому появляется время для индивидуальной коррекции работы наиболее сильных и наиболее слабых учеников — он может дать дополнительные, более сложные,

задания одним и помочь разобраться в материале другим. В связи с этим, возможно, удастся преодолеть те недостатки классно-урочной системы, которые были сформулированы еще Гегелем: «...мнение, что наставник должен тщательно изучать индивидуальность каждого ученика, сообразовываться с нею и развивать ее, является совершенно пустым и ни на чем не основанным. Для этого у него нет времени. Своеобразие детей терпима в семейном кругу, но в школе начинается жизнь... по общим для всех правилам. Тут приходится заботиться о том, чтобы дети отвыкли от своей оригинальности».

Мы специально подробно не рассматривали программы, входящие в па-

кет, не разбирали их достоинства и недостатки, перспективы их совершенствования и расширения пакета, что позволит увеличить круг задач, решаемых с его помощью. Это предмет особой дискуссии. Вы сможете сами судить об этом, приобретя пакет «Юниор-1». В данной статье нам было важно познакомить вас с самой концепцией индивидуализации обучения, в русле которой сделан пакет.

Литература

1. Коменский Я. А. Великая дидактика. М., 1939. С. 133.
2. Хрестоматия по истории педагогики. Т. 2. Ч. 1. М., 1940. С. 122.

ВЕРНОСТЬ ТРАДИЦИИ —

ЗАЛОГ КАЧЕСТВА И ГАРАНТИРОВАННЫЙ УСПЕХ

Предприятие СБИС — это 5 лет успеха на рынке программного обеспечения в СССР.

**Частным лицам, школам, организациям и предприятиям
предлагаем**

Как новые, так и лучшие из старых программ для «Электроники БК-0010» (0010.01) по вашему выбору, высокое качество и невысокие цены.

Договоры с авторами программ для БК-0010 и «Поиска» на тиражирование программ с выплатой до 20 % от реализации.

Только у нас — информационное обеспечение пользователей и программистов: бесплатные каталоги программного обеспечения, информационный бюллетень клубного типа «БИС» для владельцев БК и «Поиска», методические рекомендации по использованию БК.

Компьютеры «Поиск» (см.: Наука и жизнь. 1991. № 2) с пакетом программ, адаптированных для магнитофона.

Классы «Поиск» на базе IBM-совместимых компьютеров в комплекте с эргономической мебелью и программным обеспечением, с поставкой «под ключ», гарантийным и послегарантийным обслуживанием.

РАДЫ БУДЕМ ВИДЕТЬ ВАС В ЧИСЛЕ НАШИХ ЗАКАЗЧИКОВ!

**167024, Сыктывкар, а/я 430,
региональный центр киевского ПО «Электронмаш» СБИС.
Тел.: [821-22]-2-64-36.**

В. АРТАМОНОВ
НОВОСИБИРСК

КУВТ «Корвет»

Введение

Корвет (франц. *corvette*) — 1) в парусном военном флоте XVIII—XIX вв. легкий военный корабль, предназначавшийся для разведки и выполнения вспомогательных задач; 2) в британском и американском флотах периода II мировой войны корветом назывались сторожевые корабли.

Советский энциклопедический словарь

Думается, что и те, кто только начал работать с «Корветом», и те, кто уже давно его использует в своей деятельности, могут согласиться с тем, что (в соответствии с морской терминологией) этот компьютер — легкий и изящный военный корабль во флотилии персональных ЭВМ, на котором можно решать и отнюдь не вспомогательные задачи в отличие от настоящего корвета. Все зависит от того, насколько глубоко изучена эта машина. Конечно, поскольку «Корвет» является одним из первых в нашей стране массовых персональных компьютеров, он не свободен от недостатков как с технической точки зрения, так и в области программного обеспечения. Но в целом, думается, открывает весьма реальный путь создания в нашей стране конкурентоспособного образца ПЭВМ.

В отличие от своих собратьев — больших ЭВМ, требующих для нормальной работы специальных помещений, — «Корвет» неприхотлив и в принципе может работать в следующих климатических условиях:

температура окружающего воздуха 5—40 °С;
относительная влажность воздуха 40—80 % при температуре 25 °С;
атмосферное давление 630—800 мм рт. ст.

При температуре 25 °С и выше «Корвет» начинает давать сбои, которые выражаются в порче информации на дисках. Если температуру в помещении понизить не удастся, то можно

попробовать подготовить диски (отформатировать, записать ОС и т. д.) в условиях повышенной температуры. Это даст возможность в новых условиях продолжить работу с классом дальше. По-видимому, причину указанных явлений следует искать не в электронике, а в изменении геометрии элементов дисковода под действием высокой температуры.

КУВТ «Корвет» состоит из рабочего места преподавателя (РМП) на основе ПК8020, рабочих мест учащихся (РМУ) на основе ПК8010, локальной информационной сети и системы электропитания. Персональные компьютеры ПК8010 и ПК8020 имеют следующие характеристики:

восьмиразрядный процессор
КР580ВМ80А с быстродействием 625 тыс. коротких операций в секунду;

ПЗУ — 32К байт;

ОЗУ пользователя — 64К байт;

видеоконтрольное устройство с экраном 16 строк по 64 знака или 512×256 точек с 8 градациями яркости (цветами).

Возможно одновременное подключение двух видеоконтрольных устройств — монохромного и цветного, а также специализированного или бытового магнитофона.

ПК8020 имеет, кроме того, средства для подключения печатающего устройства матричного типа (например, «Epson» или «Robotron») и накопителей на гибких магнитных дисках.

Локальная сеть КУВТ представляет собой моноканальную сеть с топологией «общая шина» и предусматривает подключение до 16 рабочих мест: одного РМП и до 15 РМУ. В качестве соединителя используется витая пара проводов длиной до 50 м. Скорость передачи информации — не менее 19500 бит/с. В качестве адреса передачи используется индивидуальный номер каждого РМУ (поскольку РМП одно, ему номер не присваивается), который жестко задается с помощью аппаратных средств при оборудовании класса. Поэтому рекомен-

дуются после установки КУВТ узнать у специалистов; производивших пусконаладочные работы, адрес каждого РМУ и сделать метки с помощью полоски бумаги и клейкой ленты. Это поможет в дальнейшем избежать ошибок при обмене информацией.

Если все рабочие места включены в локальную сеть, то последовательность включения питания должна быть следующей: вначале на всех РМУ, а только после этого на РМП. Если же хотя бы одно РМУ включить во время работы РМП, то РМП «зависает», и требуется перезагрузка операционной системы, интерпретатора языка Бейсик, прикладной программы и т. д.

Это конструктивная недоработка, которую своими силами исправить достаточно сложно.

Документация

Когда вес документов достигнет веса самолета, самолет начнет летать.

Дональд Дуглас.

Известно, что люди не любят составлять документацию на созданную ими продукцию. Между тем документация — самое важное из того, что они должны сделать.

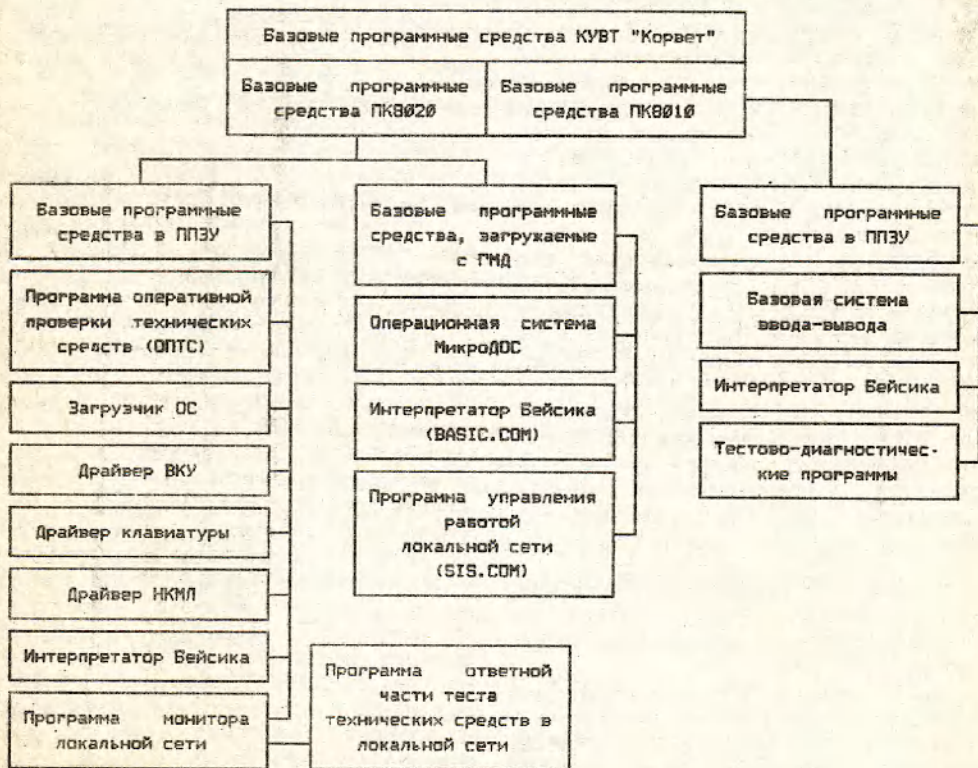
Тот, кто разрабатывал достаточно сложные программные комплексы, знает, какой внутренний протест вызывает необходимость написания документации на них, что необхо-

димо, даже если этим комплексом предстоит пользоваться самому разработчику через несколько месяцев. А уж если документация пишется для абстрактного пользователя!.. К сожалению, эту человеческую слабость не перебороли и составители документации на КУВТ «Корвет». Она изобилует повторениями, неточностями, туманными местами и т. п. Как же извлечь из нее рациональное зерно?

Если вы хотите всерьез заниматься программированием, то рекомендуем для начала обратиться к документу, который называется «Единая система программной документации» (его можно купить в любом магазине стандартов). В соответствии с ЕСПД на каждую самостоятельную программу или программный комплекс должны быть составлены следующие эксплуатационные документы:

- ведомость эксплуатационных документов;
- формуляр;
- описание применения;
- руководство системного программиста;
- руководство программиста;
- руководство оператора;
- описание языка (если программа — транслятор с какого-либо языка программирования);

40



руководство по техническому обслуживанию.

Посмотрим теперь, из каких компонентов состоит программное обеспечение, поставляемое вместе с КУВТ «Корвет». Структура базовых программных средств КУВТ «Корвет» приведена на рисунке.

Конечно, не на все эти средства необходим полный комплект документации, но как минимум на ОС МикроДОС и интерпретатор языка Бейсик он должен быть. Это избавило бы пользователей от многих ошибок во время эксплуатации комплекса.

Какие же документы на программное обеспечение поставляются с «Корветом»? Из 24 книг 9 (с 16-й по 24-ю) содержат собственно программную документацию. Однако 23-я и 24-я книги описывают работу на языке Форт, демонстрационные и игровые программы, что на практике используется мало; книга 17-я, содержащая описание базовых программных средств КУВТ «Корвет», ничем не отличается от 18-й и 19-й, где рассказывается о базовых программных средствах ПК8010 и ПК8020 соответственно; в свою очередь 18-я полностью повторяет 19-ю в части, касающейся базовых программных средств, размещаемых в ПЗУ.

Таким образом, можно рекомендовать следующую последовательность изучения оставшихся книг, связанных с программным обеспечением КУВТ «Корвет»:

«Комплекс вычислительный «Корвет»» (книга 1);

«Комплекс вычислительный «Корвет». Базовые программные средства. Комплект программных средств» (книга 16);

«Базовые программные средства ПК8020» (книга 22);

«Базовые программные средства ПК8020. Интерпретатор языка Бейсик» (книга 21);

«Базовые программные средства ПК8020» (книга 19).

Всю остальную программную документацию можно использовать для уточнения некоторых неясных мест.

Операционная система МикроДОС

Программы — это посредники между пользователем и машиной.

Гарлен Д. Миллс

Сердцем любого компьютера (не только персонального) является операционная система, приводящая в движение тонкие механизмы его взаимодействия с пользователем. В конечном счете вся работа, начиная от проверки состояния технических средств компьютера до выполнения вашей

программы на Бейсике или любом другом языке программирования, происходит под управлением операционной системы (ОС). Поэтому изучение программного обеспечения целесообразно начинать с изучения ОС. В «Корвете» используется обычно СР/М и программно совместимая с ней МикроДОС. Читателю, получившему «Корвет» с СР/М, рекомендуем интересную книгу «Знакомьтесь: персональная ЭВМ «Корвет»» С. А. Ахманова (мл.) и др. Здесь же речь пойдет об основах работы с другой ОС — МикроДОС (в документации она описана в книге 22).

МикроДОС — адаптированный (приспособленный к КУВТ «Корвет») вариант известной микрокомпьютерной дисковой ОС СР/М, разработанный в 1986 г. в Международном научно-исследовательском институте проблем управления. Основу системы составляют два модуля (программы): базовая дисковая операционная система (БДОС), обеспечивающая взаимодействие пользователя и ПК8020 (интерпретирует команды, вводимые оператором; организует обмен информацией между различными устройствами), и базовая система ввода-вывода (БСВВ), представляющая собой набор программ, связывающих МикроДОС со стандартными устройствами ввода-вывода (клавиатура, экран ВКУ, принтер, дисковые накопители).

Вся МикроДОС размещается на трех дорожках (нулевой, первой, второй — системных дорожках) ГМД и загружается в ОЗУ с помощью записанного в ПЗУ загрузчика.

Для хранения данных и программ МикроДОС использует дисковую систему из двух дисководов (А: и В:), одному из которых присваивается статус системного (он используется для операций загрузки и перезагрузки ОС). Первоначально статус системного имеет верхний дисковод — А:. Носителем информации является гибкий магнитный диск диаметром 5,25 дюйма (113 мм), который получает имя работающего с ним дисковода.

Начальная загрузка МикроДОС. После включения тумблера питания и при нажатии на кнопку «Сброс» подается звуковой сигнал, а на экран ВКУ выдается сообщение «ОПТС 1.1». ОПТС — это оперативная проверка технических средств, программа, в течение 10—12 с проверяющая аппаратуру компьютера. До конца проверки в дисковод А: должен быть установлен ГМД с ОС.

При нормальной загрузке ОС на экране появляется сообщение вида

Здесь 48К — объем (в килобайтах) области ОЗУ, представляемой для программ пользователя. Подсказка «A>» означает, что система готова принимать и выполнять команды пользователя, при этом дисковод А: является системным.

Процедура начальной загрузки начнет выполняться только тогда, когда вы отпустите кнопку «Сброс».

МикроDOS может выполнять достаточно большой набор команд, который делится на две группы:

резидентные команды, реализующие ряд основных функций ОС (программные модули, соответствующие этим командам, постоянно находятся в ОЗУ);

транзитные, используемые как вспомогательные средства и загружаемые в ОЗУ по мере необходимости с ГМД (соответствующие этим командам программы должны быть записаны в виде файлов на диск и иметь расширение имени .COM или .SPR).

Рассмотрим резидентные команды.

S — назначить системный дисковод с указанным именем. Появляющаяся после первоначальной загрузки ОС на экране подсказка «A>» указывает на то, что верхний дисковод (A:) является системным. Признак системности дисковода учитывается при работе с дисками. Дав команду

S B:<BK>

вы сделаете системным дисковод B:, командой же

S A:<BK>

— снова дисковод A:.

После выполнения команды

B B:<BK>

на экране появится подсказка «A>», но для МикроDOS системным будет уже дисковод B! Это обстоятельство следует учитывать при перезагрузке системы и при модификации информации в ОЗУ или на дисках.

Следует отличать системный дисковод от текущего. На системном дисковде обязательно должен находиться диск с ОС. На текущем же может находиться любой диск. Ввод в качестве команды указателя дисковода (A: или B:) делает текущим нужный дисковод. Первоначально дисковод A: является системным и текущим одновременно.

U — изменение кода пользователя. Это очень удобная команда в том случае, если с одним и тем же диском приходится работать поочередно нескольким пользователям.

Весь диск делится на 16 непересекающихся подобластей (номера от 0 до 15). Подобласть с номером 0 называется системной, системные файлы из нее доступны всем пользователям данного диска. Остальные подобласти доступны только тем пользователям, которые имеют код, совпадающий с номером подобласти. Так, если после загрузки системы вы наберете команду

U 3<BK>

то подсказка на экране примет вид «3 A>». Это означает, что вы имеете доступ к программам и данным в подобласти с номером 3. Команда U позволяет иметь на одном диске программы с одинаковыми именами без опасности затереть их друг другом. По умолчанию код пользователя равен 0.

Не следует путать системную область диска (дорожки 0—2), где расположена МикроDOS, с системной подобластью. В документации здесь некоторая путаница.

O — опрос числа системных дорожек. Как было сказано, МикроDOS может занимать до трех дорожек диска. С помощью команды O можно установить, сколько конкретно занимает ваша ОС (для сравнения: CP/M занимает две первые дорожки). Команда выводит информацию для обоих дисководов.

D — вывод на экран консоли таблицы оглавления ГМД для текущей области пользователя. Рассмотрим особенности организации файловой системы в МикроDOS. Файлом здесь называется какая-либо совокупность данных, снабженная специальным именем и хранимая в виде записей на ГМД. Имя файла должно состоять из названия файла (не более восьми алфавитно-цифровых символов, исключая «>», «<», «», «>», «<», «:», «*», «?», «[», «]», «—», «^», «!»), точки и, возможно, типа файла (расширения). Тип файла должен состоять не более чем из трех алфавитно-цифровых символов (ограничения такие же, как и в имени файла). В отличие, скажем, от CP/M пользователь может выбрать любой тип файла или оставить имя файла вообще без типа.

Исключение составляют типы COM и SPR. Если вы написали программу, которая будет работать непосредственно под управлением МикроDOS, то ее имя должно обязательно иметь тип COM (SPR), иначе программа не будет выполняться системой.

С каждым файлом могут быть связаны некоторые характеристики — атрибуты файла. Они указывают на то, какие операции можно выполнять над файлом. Атрибут задается одной из букв D, S, R, W и записывается в конце типа файла с помощью транзитной команды ATTR, которую мы в дальнейшем рассмотрим. Возможны следующие сочетания атрибутов:

D и W — над файлом разрешено вы-

полнять все операции, файл не системный; D и R — над файлом разрешено выполнять только операцию чтения, файл не системный;

S и R (или W) — файл системный и не доступен для модификации (вспомните системные файлы из нулевой подобласти).

Во многих командах МикроДОС имя файла можно задавать неявно. Для этого используются символы «?» (замена любой литеры имени) и «*» (замена группы литер в имени). Так, неявное имя A?BC1.BAS определяет все файлы с названиями, отличающимися второй литерой, а неявное имя *.COM идентифицирует все файлы с произвольным именем и типом COM.

Команда D в общем виде записывается так:

D *дискковод* *имя*,

где параметр «дискковод» задает дискковод (A: или B:), параметр «имя» — либо явное, либо неявное имя файла. Например, команда D B: *.COM<BK>

означает: вывести на экран имена всех командных файлов из оглавления текущей подобласти ГМД дисквода B:.

Заметим, что параметры «дискковод» и «имя» могут быть опущены, тогда на консоль осуществляется вывод оглавления текущей подобласти ГМД текущего дисквода.

Команда D осуществляет вывод оглавления той подобласти, в которой вы находитесь в данный момент.

УПР+С — реинициализация системы («теплый старт»). В результате выполнения этой команды с системного диска копируется только часть МикроДОС. При этом прекращается выполнение текущей программы, закрывается файл консоли вывода (т. е. файл, в который выводится информация с экрана консоли), перезагружается БДОС без изменения текущего кода пользователя и дисквода.

Ю. ФИЛИПСКИЙ, С. ПИУНОВ

«Корнет» для «Нейвы»

КУВТ «Нейва» (аналог «Корвета») состоит из одной ПЭВМ ПК8020 (РМП) и 12 ПК8010 (РМУ), связанных в локальную сеть (ЛС).

Логическая связь между устройствами осуществляется под управлением программы STS (базовый вариант).

Работа ЛС под управлением STS обеспечивает лишь изучение различных учебных материалов (подготовленных учителем или программистом) и программирование на языке

«Теплый старт» можно использовать для выхода из выполняемой в МикроДОС программы, не дожидаясь конца программы, когда системы должна реинициализировать себя автоматически. Смена системного ГМД требует обязательного выполнения «теплого старта»!

E — удаление файла или группы файлов из текущей подобласти пользователя на ГМД. Вызывается команда строкой вида

E *дискковод* *имя*,

где параметры «дискковод» и «имя» задаются аналогично предыдущей команде.

Поскольку E очень ответственная операция, то МикроДОС переспрашивает:

удалить - Y/N?

При вводе одного из символов Y, y, D, d происходит удаление. Если вы передумали производить данную операцию, нажмите клавишу N (n).

Пример: команда вида

E B: *.BAS<BK>

в случае вашего подтверждения удалит с ГМД на дискводе B: все файлы типа BAS.

При попытке удаления системного файла (с атрибутом S) выдается сообщение об ошибке и команда заканчивает работу. Если встречается файл только для чтения (с атрибутом R), то система просит подтверждения на него удаление.

Будьте особенно внимательны при удалении с помощью этой команды группы файлов, поскольку система запрашивает подтверждение на удаление только вначале! Для избирательного удаления файлов есть специальная транзитная команда — ERAQ.

Не производите удаление файлов с единственной копии важного для вас диска!

Рассмотренные выше резидентные команды МикроДОС объединяются в одну группу — команды управления работой дисковой системы.

Окончание следует

Бейсик. Многие программы (текстовые и графические редакторы, базы данных, игры и т. д.) недоступны для РМУ из-за отсутствия в них операционной системы (ОС) — программного ядра любой ПЭВМ. Это большой недостаток базового варианта управления ЛС, так как полноценное изучение и использование ПЭВМ возможны только на РМП.

Для преодоления этих недостатков была

разработана сетевая операционная система (СОС) «Корнет».

Организация работы ЛС под управлением СОС «Корнет» принципиально отличается от базового варианта. «Корнет» позволяет использовать РМУ и РМП как равноценные ПЭВМ, связанные друг с другом.

Основные различия STS и «Корнета» с точки зрения пользователя приведены в таблице.

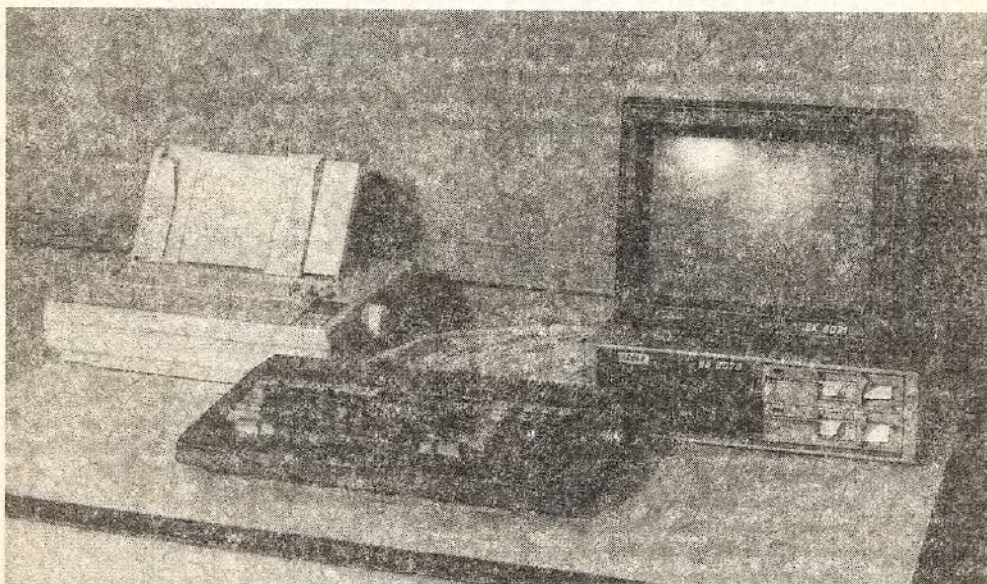
STS	СОС «Корнет»
Инициатором обмена является учитель (РМП). Он определяет, кому и что передать и что принять	Инициатором обмена является ученик. Он решает, что ему нужно загрузить (с любого диска), и может сохранить файл в своем разделе на диске
Для участия РМУ в обмене необходимо, чтобы РМУ к моменту запуска STS было включено	РМУ в любой момент может войти в СОС «Корнет» и выйти из нее (при наличии ПЗУ с загрузчиком СОС)
ЛС настроена на передачу файлов для интерпретатора Бейсика или специально подготовленных программ	На РМУ можно запустить любые программы, работающие корректно в ОС CP/M или МикроДос. Ограничение — вывод на печать
Для инициализации «электронного диска» необходима специальная программа Возможен прямой вывод информации на экран от РМП к РМУ	Формирование «электронного диска» встроено в СОС как на РМП, так и на РМУ Реализован только обмен файлами

СОС «Корнет» обеспечивает доступ с РМУ к дисководам РМП. Работа как на РМУ, так и на РМП практически не отличается от работы в среде широко распространенной ОС CP/M.

«Корнет» состоит из резидентной части (RESID) и оболочки (SHELL) и располагается в верхних адресах памяти. Резидентная часть служит для управления ресурсами системы, организации файловых операций и обмена по сети. Она постоянно находится в памяти компьютера. Оболочка обеспечивает взаимодействие пользователя с системой, ввод и исполнение встроенных и транзитных команд. Область памяти, занимаемая оболочкой, может использоваться транзитными программами.

СОС поддерживает 16 логических дисковых устройств, обозначаемых буквами латинского алфавита от А: до P:. Каждое логическое устройство (кроме P:) доступно для чтения всем ПЭВМ, а для записи только одной. ПЭВМ, которой разрешена запись на данное дисковое устройство, является его собственником.

Дисководы А и В на РМП распределяются следующим образом: дисковод А совпадает с логическим устройством А:, его собственником является РМП; дисковод В распределяется на несколько логических устройств (В:, С:, D: и т. д. до N:), которые распределяются между РМУ. Максимальное количество логических устройств на В — тринадцать, что определяет максимальное количество РМУ в системе. Разбивка на логические устройства и определение их размера производится программой создания сетевого



диска DISNET. Сетевой диск помещается в дисковод В на РМП и должен находиться в нем постоянно во время работы СОС.

Логическое устройство О: — электронный диск РМП. Оно становится доступным для РМУ после инициализации со стороны РМП. Если инициализация не производилась, обращение к устройству О: со стороны РМУ вызывает сообщение об ошибке выбора диска.

Логическое устройство Р: — электронный диск РМУ. Каждое РМУ имеет свое индивидуальное устройство Р:, недоступное как другим РМУ, так и РМП.

В «Корнете» имеются те же встроенные команды, что и в СР/М (за исключением команды USER), однако они имеют некоторые особенности, облегчающие работу.

Если при запуске транзитной программы перед ней не указано имя дискового устройства, система производит поиск файла с этой программой в следующей последовательности: на текущем логическом устройстве; на логическом устройстве А:; на логическом устройстве О:.

Загрузка СОС «Корнет» на РМУ реализована в трех вариантах (все они предполагают, что СОС «Корнет» загружена на РМП).

1. КУВТ «Нейва» настроен на работу ЛС только под управлением СОС «Корнет», в ПЗУ каждого устройства ЛС записан загрузчик. Благодаря этому загрузка СОС с РМП на РМУ происходит автоматически при включе-

нии РМУ. Этот вариант не позволяет работать в ЛС под управлением STS.

2. Загрузчик записан в отдельный блок ПЗУ, подключаемый к разьему «Расширение» РМУ. Процесс загрузки аналогичен первому варианту.

3. Загрузка СОС «Корнет» на РМУ происходит после запуска с РМП специальной программы, передающей загрузчик на включенные РМУ. При включении впоследствии дополнительных РМУ они могут быть догружены «Корнетом».

В настоящее время основным вариантом поставки является третий.

Производственное объединение «Октябрь» (623400, г. Каменск-Уральский, ГСП 79; тел.: 6-53-48) и малое предприятие «Хорда» (623400, г. Каменск-Уральский, а/я 1; тел.: 3-62-55) поставляют всем заинтересованным организациям СОС «Корнет» по цене 333 рубля. СОС работает на всех КУВТ «Корвет» и «Нейва» производства различных предприятий и с различными версиями встроенного ПО.

«Октябрь» может поставлять КУВТ «Нейва» (вариант исполнения ПТ 1.320.167-03) с видеоОЗУ на РМП объемом 192К байт, что обеспечивает формирование на РМП электронного диска емкостью 144К байта. Обмен РМУ с электронным диском О: на РМП позволяет увеличить скорость передачи информации в 1,5 раза и повысить срок службы дисков и дисководов.

Г. КРИВОШЕЕВ, С. КОЗАРЕНКО
НИИВК, Москва

Функциональные возможности ОС локальной сети для ПЭВМ «Агат»

Разработанная в 1990 г. в Научно-исследовательском институте вычислительных комплексов Москвы операционная система локальной сети (ОС ЛС) для ПЭВМ «Агат» в определенном смысле завершает разработку системного программного обеспечения для многопользовательских учебно-тренажерных комплексов и открывает новые возможности использования ПЭВМ «Агат» в качестве терминальных станций и простых локальных сетей делопроизводства. В 1990 г. ОС ЛС получила поощрительную премию в отраслевом конкурсе программных средств Министерства радиопромышленности и серебряную медаль ВДНХ СССР.

Функционально ОС ЛС позволяет объединить работу по моноканалу до 128 ПЭВМ «Агат-7» и «Агат-9» со скоростью

передачи информации до 62,5 Кбод на расстоянии не менее 3 км;

обеспечить одновременное независимое функционирование в сети систем программирования: «Бейсик», «Ассемблер», «Школьница» и пакетов прикладных программ: «Система Численного Моделирования», «Система Подготовки Текстов».

Необходимыми техническими средствами для функционирования ОС ЛС являются контроллеры локальной сети, имеющие уникальные номера и установленные в каждую ПЭВМ, и хотя бы один дисковод типа ЕС 5023 с ПЭВМ «Агат», работающие под управлением ОС ЛС, которые условно делятся на главные и терминальные. Главная — это ПЭВМ, на которой загружен сетевой программный комплекс (СПК).

В этом случае ПЭВМ может выполнять начальную загрузку одной из систем программирования или пакетов прикладных программ на другие (терминальные) ПЭВМ и обеспечить доступ к своим дисковым запоминающим устройствам (ДЗУ) загруженным ею ПЭВМ. Главная ПЭВМ в каждый момент времени находится в автономном режиме или в режиме «работа по сети». При работе в автономном режиме главная ПЭВМ поддерживает функционирование СП «Ассемблера» и «Отладочный комплекс» СП «Школьника». В режиме «работа по сети» (см. рис. 1) управление находится у ОС ЛС.

*** ОС ЛС НИИВК 1990 г. Москва ***
Внимание! Работа по сети!!!

Собственный номер 3

Текущее состояние:
ДЗУ текущее 1
ДЗУ загрузки 2
Тип абонента «Агат 9»
Текущий абонент 21

Изменить состояние

Выполнить загрузку
Тесты
Тип файла А
Передать файл
Выполнить файл
Передать сообщение
Выполнить команду

ОС ЛС главной ПЭВМ постоянно «слушает сеть» и при появлении запросов выполняет операции по передаче или приему файлов с терминальными ПЭВМ.

«Собственный номер» ПЭВМ — номер установленного в нее сетевого контроллера. Этот номер является «адресом», который используется в запросах при обмене информацией между ПЭВМ сети. Обработка запросов на главной ПЭВМ организована по прерываниям, которые формируются контроллером сети, если в запросе он обнаруживает собственный адрес. ОС ЛС хранит в специальных таблицах номера всех загруженных с нее ПЭВМ и отвечает только на те запросы, которые поступают с них. Если запрос поступил от ПЭВМ, которая была загружена не по сети или чья загрузка была выполнена другой главной ПЭВМ, то на экране появится сообщение «Чужой абонент» и ОС ЛС не будет его обрабатывать. Для того чтобы разрешить взаимодействие между ПЭВМ, в этом случае достаточно задать на главной ПЭВМ номер «Чужого абонента» в качестве текущего. Как только номер текущего абонента будет изменен, взаимодействие тут же будет вновь прекращено.

ОС ЛС осуществляет связь между ПЭВМ сети на двух уровнях. Первый — уровень файловой системы. Второй — уровень

пользовательских интерфейсов.

На файловом уровне ДЗУ главной ПЭВМ рассматривается просто как дополнительный дисковод терминальной ПЭВМ. Файловый уровень позволяет на терминальной ПЭВМ выполнить любую операцию дисковой операционной системы (ДОС) соответствующей системы программирования с ДЗУ главной ПЭВМ (например, CATALOG, LOAD, SAVE, OPEN и т. д., для СП Бейсик; ОТКРЫТЬ, ПРОЧИТАТЬ, КАТАЛОГ и т. д. для СП «Школьника» и прочих прикладных пакетов).

На этом уровне для работы с ОС ЛС в системы программирования и пакеты прикладных программ не добавлено ни одной дополнительной операции. Переход от работы с собственными ДЗУ на работу с удаленным ДЗУ главной ПЭВМ осуществляется стандартным переходом на новое ДЗУ, предусмотренным в системах.

Например, в СП Бейсик, для того чтобы просмотреть каталог диска, установленного в ДЗУ главной ПЭВМ, достаточно выполнить команду CATALOG, SN, где N — номер разъема, в который установлен контроллер сети. Фактически главная ПЭВМ, с которой установлена связь, рассматривается терминальной как дополнительный ДЗУ.

Файловый уровень ОС ЛС «Агат» обеспечивает выполнение следующих основных функций:

начальную групповую и индивидуальную загрузку объединенных сетью ПЭВМ одной или различными системами программирования или пакетами прикладных программ; обеспечение доступа терминальных ПЭВМ к файлам ДЗУ главной ПЭВМ;

выполнение операций просмотра каталогов, чтения, записи и выполнения файлов и т. д.

ОС ЛС «не видна» из ДОС систем программирования или пакетов прикладных программ. Достигается это за счет того, что все ДОС для работы с ДЗУ используют программу Чтения/Записи Сектора (RWTS). ОС ЛС после загрузки в память соответствующей системы программирования или пакета прикладных программ подменяет вход в эту программу на собственный фильтр. Функциями этого фильтра является распознавание того, к какому ДЗУ пришло обращение. Если ДОС обращается к собственным ДЗУ ПЭВМ, управление передается RWTS, если же в параметрах обращения к RWTS указан разъем, в котором установлен контроллер сети, то формируется и передается по сети запрос к ОС ЛС главной ПЭВМ. На ней и выполняется требуемая операция чтения или записи из ДЗУ. В случае выполнения записи сектора сам

сектор принимается по сети с терминальной ПЭВМ и записывается на ДЗУ главной; в случае чтения сектор читается из ДЗУ и передается на терминальную. При этом для ДОС создается полная «иллюзия» того, что он работает с собственной программой RWTS. На терминальных ПЭВМ все программы файлового уровня находятся в постоянной памяти (ПЗУ) контроллера сети и фактически не используют памяти ПЭВМ. Единственными ячейками оперативной памяти, которые используются для работы ОС ЛС на терминальной ЭВМ, являются ячейка, в которой хранится номер главной ПЭВМ, и ячейка, в которой хранится номер разъема контроллера сети.

ПЗУ контроллера сети содержит: программу автоматической начальной загрузки;

программы обработки и установки прерываний по локальной сети;

программу приема и передачи сектора данных по локальной сети;

служебные программы управления и контроля за состоянием локальной сети.

Как известно, ПЭВМ «Агат» имеет несколько разъемов, в которые могут устанавливаться контроллеры различных устройств. Такие устройства, как ДЗУ и контроллер сети, относятся к типу устройств, с которыми может выполняться начальная загрузка системы. Сразу после включения питания ПЭВМ, системный монитор анализирует конфигурацию ПЭВМ и, найдя контроллер устройства, с которого может быть выполнена загрузка, передает ему управление и пытается выполнить начальную загрузку. Если в конфигурации присутствуют несколько устройств такого типа, то делается попытка выполнить загрузку с устройства, установленного в разьеме с наименьшим номером. Если в ПЭВМ одновременно присутствует и контроллер сети, и контроллер ДЗУ, то представляется целесообразным устанавливать сетевой контроллер в разьеме с номером большим, чем контроллер ДЗУ. В этом случае у ПЭВМ сохраняется возможность как обычной загрузки с ГМД, так и загрузки по сети. Для загрузки по сети в этом случае после включения питания необходимо выполнить сброс, а затем команду CNOOG, где N — номер разъема контроллера сети. После выполнения указанных действий управление получит программа автоматической начальной загрузки и на экране появится сообщение

Сеть Агат N...
Жду загрузки!

Сама загрузка начнет выполняться после соответствующей директивы «Выполнить загрузку» с главной ПЭВМ. Если

в ПЭВМ установлен только контроллер сети, или этот контроллер установлен в разьеме с номером меньшим, чем контроллер ДЗУ, то после включения питания ПЭВМ выйдет в состояние ожидания загрузки по сети автоматически.

В ходе выполнения начальной загрузки главная ПЭВМ читает необходимую информацию с собственного ДЗУ и передает по секторно в сеть. В заголовке передаваемого по сети сектора указывается, как конкретно ПЭВМ он передается, физическая конфигурация памяти и логический адрес, по которому должен быть помещен этот сектор на терминальной ПЭВМ. Программа автоматической начальной загрузки принимает адресованную ей информацию и размещает ее в памяти. Последним главная ПЭВМ передает так называемый стартовый сектор. В нем хранится стартовая программа, которая формирует конфигурацию системы и запускает ее.

Следует отметить, что протокол обмена по сети, который соблюдается при выполнении начальной загрузки, существенно отличается от того протокола, по которому работает ОС ЛС с уже загруженными ПЭВМ. В частности, при выполнении начальной загрузки отсутствует схема обработки конфликтов. Начавшийся процесс загрузки не должен прерываться другими ПЭВМ до полной передачи всех секторов. Программа автоматической загрузки просто «слушает» линию и забирает с нее каждый адресованный ей сектор, не предпринимая никаких активных действий. Главная ПЭВМ при этом не знает, на какой из терминальных ПЭВМ загрузка оказывается выполнена нормально, а на какой она завершилась аварийно.

На загруженных ПЭВМ обмен ведется с многократным повторением запросов в режиме конкуренции за каждый сектор. В этом случае терминальная ПЭВМ «слушает» линию, и если она оказывается занятой, то ПЭВМ выходит в состояние ожидания на время, кратное собственному номеру. Затем прослушивание возобновляется, если линия вновь занята, следует новая пауза с новым интервалом ожидания. Этот процесс повторяется до 50 раз. Если тем не менее линия оказывается занятой, то следует уже запрос пользователю, который может либо продолжить попытки, либо завершить их ошибкой обмена. Такой алгоритм позволяет избежать клинчей в сети в случае повреждения линии или выхода главной ПЭВМ из режима «работы по сети». Как только терминальная ПЭВМ обнаруживает, что линия свободна, она передает запрос главной, а та в свою очередь принимает запрос, выдает в сеть подтверждение приема запроса и

начинает выполнять запрошенную операцию.

Реализованный на файловом уровне секторный обмен информацией позволил переложить все управление файловыми операциями на ДОС терминальных ПЭВМ. Именно они начинают, контролируют и завершают операции выполнения, чтения и записи файлов. Притом что в целом в сети могут одновременно работать несколько систем программирования и пакетов прикладных программ и при этом каждая из них одновременно с другими запрашивать с главной ПЭВМ собственные файлы — главная ПЭВМ воспринимает все эти операции просто как запрос чтения или записи очередного сектора диска.

Несмотря на достаточную функциональную полноту, простоту использования и высокую надежность файлового уровня ОС ЛС, у пользователя может возникнуть естественное желание в расширении своих возможностей в работе с сетью. Так, например, принцип «прозрачности» ОС ЛС для ДОС систем программирования делает невозможным непосредственное вмешательство в работу терминальных ПЭВМ со стороны главной, а подобное искушение у изощренного пользователя может появиться (справедливости ради, правда, следует отметить: подобный пользователь такую возможность мог бы найти, просто контролируя соответствующие файлы). Следует отметить, что уровень файловой системы замкнут и не допускает своей модификации пользователем.

Для реализации дополнительных требований пользователь может воспользоваться следующим уровнем — уровнем пользовательских интерфейсов. Для того чтобы воспользоваться этим уровнем, пользователь загружает в оперативную память ПЭВМ специальные сетевые драйверы. Эти драйверы могут быть загружены как автоматически — со стартовыми программами систем программирования (например, с файлом HELLO для СП Бейсик или ЗАПУСК в СП «Школьника») или в процессе работы системы программирования. Так, например, типовой драйвер для СП Бейсик загружается файлом HELLO в ходе сетевой загрузки и помещается в не используемую СП Бейсик память начиная с адреса 800 и инициализируется.

Драйвер включает 4 подпрограммы:
установить обработку прерывания;
принять команду;
принять сообщение;
передать сообщение.

Программа «установить обработку прерываний» устанавливает контроллер сети в ждущий режим, и при появлении адресованного ему сообщения управление передается программе «принять сообщение» или

«принять команду». Если сообщение оказывается ошибочным, оно игнорируется. В противном случае, если принятая по сети информация оказалась сообщением, оно выдается на экран и выполнение прерванной программы продолжается, если же информация оказалась командой (например, NEW, STOP, LIST, CATALOG, LOAD и т. д.), то она обрабатывается СП Бейсик.

Для того чтобы передать сообщение по сети, необходимо записать текст этого сообщения по адресу 200 и затем передать управление программе «передать сообщение».

Сообщения и команды терминальной ПЭВМ всегда принимаются и передаются только от ПЭВМ, которая для нее считается главной, но если сменить этот номер, а он хранится по адресу \$200, то можно «услышать» любую ПЭВМ сети. Аналогично можно изменить и номер ПЭВМ, для которой выполняется передача. Фактически указанный драйвер позволяет организовать обмен любой информацией между любыми ПЭВМ сети.

Драйвер написан на языке ассемблера СП Бейсик, и поэтому пользователь его легко может модернизировать для собственных нужд.

На главной ПЭВМ режим передачи команд и сообщений поддерживается соответствующими пунктами меню: «передать файл», «выполнить файл», «передать сообщение», «выполнить команду». Выполнение пунктов «передать сообщение» и «выполнить команду» позволяет набрать с клавиатуры главной ПЭВМ и передать одной или нескольким терминальным ПЭВМ любое сообщение или команду. Например, пункт «выполнить команду» позволяет набрать с главной машины и выполнить на терминальной произвольную программу. Пункты меню «передать и выполнить файл» просто передают на терминальные ПЭВМ, в зависимости от типа файлов команды LOAD, BLOAD, RUN, BRUN и EXEC с именем требуемого файла. Собственно выполнение этих команд осуществляется ОС ЛС файлового уровня. Перечисленные выше пункты меню не выполняются на терминальной ПЭВМ, на которой не инициализирован сетевой драйвер.

Суммируя сказанное, можно отметить следующее: использование ОС ЛС для ПЭВМ «Агат» позволяет в настоящий момент создавать сложные и гибкие вычислительные комплексы, сочетающие разнообразие и простоту применения широкого круга стандартных программных средств ПЭВМ «Агат», широкие возможности для разработки собственного пользовательского программного обеспечения.

Телефоны для справок: 330-11-38, 272-48-83, 489-00-91.

Каталог-меню для УКНЦ

Самое неудобное для УКНЦ — то, что, когда выходишь в Бейсик-систему, становишься «слепым», так как не знаешь, какие Бейсик-программы есть на диске. Предлагаю программу, совмещающую каталог и меню.

Сначала с помощью редактора текстов нужно на системном диске немного изменить командный файл BASIC.COM, сделав его таким:

```
ASS MZ0: DK:
SET TT QUIET
UNLOAD MC
SET MC ADDR=0
LOAD MC
F7S9
BASIC
UNLOAD MC
SET MC ADDR=77
LOAD MC
ASS MZ1: DK:
R SY:NETUK
```

Затем нужно создать командный файл BASIC.COM, полностью повторяющий BASIC.COM за исключением последней команды: вместо R SY:NETUK в нем должно быть @SY:NETUK.

Создадим еще два командных файла.

```
NETUK.COM:
R SY:DIR
DK:CAT=DK:X.ASC/A/F/C:5
^C
R SY:NETUK
```

```
DBAS.COM:
ASS MZ1: DK:
R SY:DIR
DK:CAT=DK:X.ASC/A/F/C:5
^C
R SY:DBAS
```

Наконец, нужно записать на рабочий диск приведенную ниже программу CAT. Она позволяет выводить с рабочего диска на терминал РМУ (или РМП) имена файлов с расширением ASC в пять колонок. Удобно иметь ее и на системном диске, чтобы при необходимости можно было копировать на вновь создаваемый рабочий диск.

Рекомендую на одну из функциональных клавиш записать команду

```
LOAD"CAT",R
```

чтобы не вызывать каталог-меню всякий раз «вручную».

Работать с программой несложно. В левом кармане дисковода — системный диск, а в первый карман вставляйте ваши рабочие диски (с CAT). Если вы хотите изменить

каталог (или создать его), загружайтесь с помощью команды

```
@SY:BASIC
дисконный Бейсик запускайте командой
@SY:DBAS
программу связи —
командой
@SY:NETUK
```

Если нет необходимости изменять содержание каталога, то работайте, как обычно, командами

```
@SY:BASIC, R SY:NETUK, R SY:DBAS
```

Каталог при этом будет выводиться старый. После загрузки Бейсик-системы на РМУ (или РМП) вызовите каталог-меню — и имена всех программ перед вами; с помощью клавиш-стрелок выберите нужную программу, загрузите и работайте с ней.

49

```
10 REM KАТАЛОГ - МЕНЮ ДЛЯ УКНЦ
20 REM В. В. КРИВОНОГОВ
30 REM *****
40 WIDTH 80,0
50 POKE &0177560,0
60 CLEAR 5000
70 COLOR 8,2,2
80 NX=5
90 POKE &0177560,0
100 DIM J$(24),F$(NX+1,20)
110 CLS
120 KX=1
130 OPEN "CAT.DIR" FOR INPUT
140 INPUT #J$(KX)
150 KX=KX+1
160 IF NOT (EOF) THEN 160
170 CLOSE
180 DX=LEN(J$(KX-3))\14
190 ? J$(1)
200 FOR IX=2 TO KX-3
210 Q$=""
220 FOR MX=1 TO LEN(J$(IX)) STEP 14
230 F$(MX\14+1,IX-1)=" "+
MID$(J$(IX),MX,6)+" "
240 Q$=Q$+F$(MX\14+1,IX-1)
250 NEXT MX
260 NEXT IX
270 ? Q$
280 ? Q$
290 NEXT IX
300 ? J$(KX-2)
310 ? J$(KX-1)
320 ? AT(0,22) "Выберите файл и нажмите
клавишу <ВВОД>"
330 GOSUB 420
340 POKE &0177560,&0100
350 COLOR 8,2,2
360 CLS
370 F$=MID$(F$(1+X\8,Y%),2,6)
380 ? AT(15,8) "Ж Д И Т Е !"
```



```

390 ? AT(4,10) "Вам загружается
                               файл ";FB0
400 LOAD FB0,R
410 END
420 X%=0
430 Y%=1
440 A%=X%
450 B%=Y%
460 COLOR 2,2,8
470 LOCATE X%,Y%,1
480 ? AT(X%,Y%)F0(1+X%\8,Y%)
490 CX=1+X%\8
500 FOR TX=1 TO 2
510 GOSUB 720
520 IF W%=13 AND TX=1 THEN RETURN
530 IF W%=27 AND TX=1 THEN K%(1)=W%
                               ELSE K%(2)=W%
540 IF W%<>27 AND TX=1 THEN TX=2
550 NEXT TX
560 IF K%(1)=27 AND K%(2)=65 THEN 610
570 IF K%(1)=27 AND K%(2)=67 THEN 630
580 IF K%(1)=27 AND K%(2)=66 THEN 650
590 IF K%(1)=27 AND K%(2)=68 THEN 670
600 GOTO 490
610 IF Y%=1 THEN IF F0(CX,K%-4)=""
                               THEN Y%=K%-5 ELSE Y%=K%-4
                               ELSE Y%=Y%-1
620 GOTO 680
630 IF F0(CX+1,Y%)="" THEN X%=0
                               ELSE X%=(X%+8) MOD 8*NX%
640 GOTO 680
650 IF F0(CX,Y%+1)="" THEN Y%=1
                               ELSE Y%=(Y%+1) MOD (K%-3)
660 GOTO 680
670 IF X%=0 THEN IF Y%=K%-4 THEN
                               X%=(X%-1)*8 ELSE X%=8*(NX%-1)
                               ELSE X%=X%-8
680 COLOR 8,2,2

```

```

690 LOCATE AX,B%,1
700 ? AT(AX,B%)F0(1+AX\8,B%)
710 GOTO 440
720 IF INP(&0177560,&0200) THEN 730
                               ELSE 720
730 W%=PEEK(&0177562)
740 RETURN

```

Используя этот прием, можно создавать тематические каталоги, каталоги программ с разными расширениями (.COD, .BIN, .ASC), каталоги программ по классам, в которых вы работаете (придется только повозиться, наводя порядок в именах программ и соответственно редактируя NETUK.COM, не забывая изменять строку 130 программы CAT, чтобы получать программы различных каталогов). Можно также формировать пакеты из созданных вами программ, составляя их таким образом, чтобы всякий раз по окончании работы с программой вновь загружался каталог. Такие пакеты легко дополнять новыми программами в процессе работы.

Использование всего лишь одного системного диска позволяет экономить дискиеты и записывать, например, программы промышленного пакета, состоящего из нескольких дискет, на одной рабочей дискете. Так, например, Бейсик-практикум (основные средства и графика — всего 8 дискет) размещается у меня на 4 дискетах: 1 системной и 3 рабочих. Экономия в два раза! А удобства в работе ничуть не хуже, скорее наоборот.

А. РОЗЕНФЕЛЬД

Использование NED

Среди пользователей УКНЦ получила распространение Нортон-подобная система Smirnow Commander (SC) — надстройка над RT-11, создающая дополнительные удобства. Она полезна, в частности, при проведении занятий по изучению текстового редактора.

Для УКНЦ есть несколько текстовых редакторов: Микромир, NED, WRITER, K-13, K-52 и т. д. На мой вкус, предпочтительным является NED. Аргументы в его пользу: сравнительная простота системы команд; богатые функциональные возможности; легкость перехода в дальнейшем на более мощный, хотя и более сложный, редактор WRITER (системы их команд почти одинаковы);

пользующиеся принтером D100M могут задавать управляющие печатью коды из текста (что невозможно при работе с WRITERом);

средствами NED можно создавать музыкальные программы, например для «проигрывателя» MUSIC.SAV;

возможны создание и распечатка LOGO-программ.

Однако учитель, особенно начинающий, выбравший для изучения NED, столкнется с трудностями при пересылке и записи текстовых файлов, да и самого NED. О том, как можно преодолеть эти трудности, я и расскажу.

Прежде всего нужно иметь системный диск с программой-пересыльщиком на РМУ монитора (SJ!) RT-11, желательнее с SC, и программами KEYS.SAV и NED.SAV. К сожалению, при пересылке NED широко распространенным сетевым пересыльщиком BAM (KLNET) утрачиваются значения ключей при работе в режиме PUC, кроме того, некоторые символы пробиваются как псевдо-

графические (хотя на печать вывод идет нормально).

Итак, порядок работы с NED при использовании его на нескольких РМУ.

1. Системная дискета вставляется в верхний дисковод (MZ0:), и с нее на все РМУ загружается монитор SJ системы RT-11 (лучше грузить машины по очереди и то урока, на 12 машин уходит минут 10—12). Программа связи с классом остается.

2. Каждый ученик на РМУ вызывает поочередно KEYS.SAV и NED.SAV, непосредственно обращаясь к дисководу (т. е. набирает имя+ВВОД). На это уходит около 2 мин.

3. Если все ученики будут работать с каким-то одним текстовым файлом, например TEXT.LST, заранее помещенным на рабочую дискету в устройство DK: (по умолчанию), то все они набирают после появления приглашения на экране

```
M<номер машины>.LST=TEXT.LST
```

например

```
M12.LST=TEXT.LST
```

после чего начинают работать с пересланным им файлом TEXT.LST.

4. Если же каждый работает индивидуально, со вновь создаваемым файлом, то набирается, например

```
M12.LST/C
```

после чего в каталоге рабочего диска отводится место для последующей записи этого файла.

Беда в том, что при записи с РМУ все эти файлы «валятся» друг на друга, поэтому в конечном счете на диске останется

только результат работы последнего ученика, сделавшего запись. С этим можно бороться путем создания квазидиска.

Действия учителя при записи файлов с РМУ.

1. После того как учащиеся приступили к работе с NED, учитель выходит из программы связи (т. е. из меню «ж, з, ф, к» выбирает «к» — конец работы).

2. В меню «Работа с диском» выбирает «Создание квазидиска» и помещает на левую панель SC пока еще пустой его каталог, а на правую — каталог рабочего диска, после чего вновь устанавливает программу связи с классом (т. е. выбирает в меню «Рассылка OS на РМУ»).

3. Сбор файлов с машин идет по очереди! Для каждого РМУ:

ученик записывает на рабочий диск результаты своей работы (т. е. нажимает K1+E и на запрос EXIT? отвечает Y);

учитель выходит из программы связи и копирует записанный файл на квазидиск — в SC это делается очень легко: устанавливается курсор на имя копируемого файла и нажимается клавиша «M»;

вновь устанавливается программа связи для дальнейшей записи и т. д.

На эти операции уходит около 40—45 с, что позволяет собрать файлы со всех машин за 5—6 мин. После завершения урока все файлы копируются с квазидиска на рабочий диск.

Вышеперечисленные действия можно делать и вне SC, но с меньшим комфортом.

Для неопытного пользователя все это выглядит достаточно длинно и сложно, но на самом деле после нескольких попыток придет навык и все окажется просто.

Калейдоскоп УКНЦ

М. Игнатов, А. Томшин и А. Нещерет (Свердловск) предлагают для вывода на печать графической информации, выведенной на экран восьмым цветом, использовать следующую программу.

```
1000 *ПОДПРОГРАММА ПЕЧАТИ СОДЕРЖИМОГО
      *ЭКРАНА В ГРАФИЧЕСКОМ РЕЖИМЕ
      *В ДВА ЦВЕТА
1010 *-----
1020 *Инициализация принтера
      *и установка интервала между
      *строками, равного восьми пикселям
1030 LPRINT CHR$(27);"@";CHR$(27);"A";
      CHR$(8)
1040 *В переменную B заносится число
      *графических строк
```

```
1050 B=264
1060 *Цикл по столбцам
1070 FOR X=79 TO 0 STEP -1
1080 *Включение графического режима
      *принтера
1090 LPRINT CHR$(27);"K";
      CHR$(B MOD 256);CHR$(1);
1100 *Цикл по строкам
1110 FOR Y=0 TO B
1120 *Занесение адреса читаемого байта
      *в переменную ADDR
1130 ADDR=Y*80+X+01000000
1140 *Занесение значения ADDR в регистр
      *адреса планов
1150 POKE 0176640,ADDR
1160 *Чтение байта с экрана по текущему
      *адресу в переменную A
1170 A=INP(0176642,255)
```



```

118@ *Печать вайта
119@ LPRINT CHR$(A);
120@ *Конец цикла по строкам
121@ NEXT Y
122@ *"Перевод каретки" принтера
123@ LPRINT
124@ *Конец цикла по столбцам
125@ NEXT X
126@ *Возврат в основную программу
127@ RETURN
128@ *КОНЕЦ ПОДПРОГРАММЫ ПЕЧАТИ-----

```

В. Семаков (г. Курск) предлагает способ распечатки программ учащихся. Для этого после загрузки операционной системы на РМП вводятся команды

```

.SET LP DEV=2      /так задается D100M в
                   качестве типа исполь-
                   зуемого принтера в
                   FMONSJ.SYS/
.R LP.SYS

```

Когда возникает нужда распечатать программу, программа связи с классом NETUK.

52

А. БАРАНОВ

Астраханский технический институт

Использование псевдографики в обучающих программах

Ряд разделов электроники можно успешно изучать с применением псевдографических возможностей ПЭВМ. Ниже приводится пример такой программы, реализованной на ПЭВМ «Роботрон-1715». Алфавитно-символьная псевдографика машин данного класса позволяет практически без отступления от ГОСТов воспроизводить топологию некоторых электронных схем, обеспечивая широкие возможности процесса исследования.

Суть метода заключается в том, что представленный на экране дисплея графический модуль дополняется до принципиальной схемы устройства, заданного начальными условиями. При правильной реализации схемы предлагается программа ее исследования. Процесс обучения протекает в диалоговом режиме.

В качестве примера рассмотрим исследование функциональных свойств триггеров, нашедших наибольшее практическое применение:

- RS-триггера асинхронного;
- RS-триггера синхронного;
- JK-триггера синхронного;
- T-триггера асинхронного;

SAV устанавливается в режим записи, после чего ученику на своей машине остается набрать команду

SAVE "LP:"

А если нужно распечатать не программу, а результаты ее работы, в ученическую программу следует добавить строку

OPEN "LP:" FOR OUTPUT

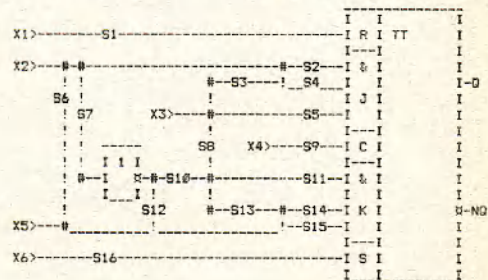
и заменить операторы PRINT на PRINT#. Заканчиваться выполнение программы должно строкой

CLOSE

Напомним для тех, кто подписался на наш журнал только в этом году, что если нужно не распечатать программу (или результаты ее работы), а вывести на монитор РМП, то вместо «LP:» ученик должен набирать «TT:». Управлять ходом этой выдачи преподаватель может с помощью клавиш УПР+Ч и УПР+О.

- TV-триггера асинхронного;
- TV-триггера синхронного;
- D-триггера синхронного;
- DV-триггера синхронного.

Исследование любого из них начинается со схематехники. С этой целью пользователь (обучаемый) выбирает из предлагаемого меню тип триггера, а затем дополняет представленный на экране дисплея графический модуль до нужной схемы. В модуль включены: универсальный JK-триггер (DD2) с логикой ЗИ на входе, инвертор (DD1), шестнадцать выключателей (S1 — S16) и шесть независимых входов (X1 — X6). Вы-



ходы триггера обозначены: прямой — Q, инверсный — NQ (аналог \bar{Q}). Задача составления функциональной схемы вызванного из меню конкретного триггера сводится к выделению из модуля единственно верной схемы путем «замыкания» соответствующих выключателей.

«Замыкание» выключателя сводится к нажатию соответствующей клавиши (клавиш) ПЭВМ. При наборе соответствующей выбранному триггеру схемы необходимо соблюдать следующие правила:

номера выключателей набираются в возрастающем порядке;

при многовариантных наборах схем внешних соединений выбирается та, для которой число замкнутых переключателей наименьшее;

если при оптимальном наборе схемы внешних соединений возникает необходимость выбора номера входа, то следует выбрать вход с наименьшим номером.

При соблюдении перечисленных правил и правильном нажатии клавиш изображение модуля трансформируется в схему вызванного триггера, причем входы триггера принимают вполне конкретные стандартные обозначения, соответствующие их функциональному назначению.

В табл. 1 приведены номера «замыкаемых» выключателей для генерации на экране изображения схемы соответствующего триггера и символьные обозначения входов.

Таблица 1

Название триггера	Номера замкнутых переключателей	Названия входов
RS-триггер асинхронный	1, 16	$X1=R, X6=S$
RS-триггер синхронный	1, 9, 16	$X1=R, X4=C, X6=S$
JK-триггер синхронный	2, 3, 4, 5, 9, 11, 13, 14, 15	$X2=J, X4=C, X5=K$
T-триггер асинхронный	2, 3, 4, 5, 6, 9, 11, 13, 14, 15	$X2=1, X4=T$
TV-триггер асинхронный	2, 3, 4, 5, 6, 9, 11, 13, 15, 14	$X2=V, X4=T$
TV-триггер синхронный	2, 4, 5, 6, 8, 9, 11, 14, 15	$X2=T, X3=V, X4=C$
D-триггер синхронный	2, 3, 4, 5, 7, 9, 10, 11, 13, 14, 15	$X2=D, X4=C$
DV-триггер синхронный	2, 4, 5, 7, 8, 9, 11, 12, 14, 15	$X2=D, X3=V, X4=C$

Одновременно с графическим модулем на экране появляется заготовка таблицы состояний с возможными сочетаниями входных сигналов (в двоично-цифровой форме). Перед обучаемым ставится задача заполнить таблицу состояний, для чего следует ввести значения логических состояний триггера из следующего набора: 0, 1, Q, NQ, x (произвольное логическое состояние). Значения размещаются в соответствующих строках таблицы состояний. Сочетания входных сигналов и соответствующие им значения логических состояний триггеров приведены в табл. 2. Каждая строка определяет состояние триггера по значению сигнала на его прямом выходе Q. К примеру, последняя строка таблицы расшифровывает логическое состояние синхронного DV-триггера ($Q=1$) при $C=1, V=1, D=1$.

Таблица 2

53

Входы						Выход		
C	V	R	S	J	K	T	D	Q
—	—	0	0	—	—	—	—	Q
—	—	0	1	—	—	—	—	1
—	—	1	0	—	—	—	—	0
—	—	1	1	—	—	—	—	x
—	—	—	—	0	0	—	—	Q
—	—	—	—	0	1	—	—	0
—	—	—	—	1	0	—	—	1
—	—	—	—	1	1	—	—	NQ
—	—	—	—	—	—	0	—	Q
—	—	—	—	—	—	1	—	NQ
—	—	—	—	—	—	—	0	0
—	—	—	—	—	—	1	1	1
0	x	x	x	x	x	x	x	Q
x	0	x	x	x	x	x	x	Q
1	1	0	0	—	—	—	—	Q
1	1	0	1	—	—	—	—	1
1	1	1	0	—	—	—	—	0
1	1	1	1	—	—	—	—	x
1	1	—	—	0	0	—	—	Q
1	1	—	—	0	1	—	—	0
1	1	—	—	1	0	—	—	1
1	1	—	—	1	1	—	—	NQ
1	1	—	—	—	—	0	—	Q
1	1	—	—	—	—	1	—	NQ
1	1	—	—	—	—	—	0	0
1	1	—	—	—	—	—	1	1

Естественно, что в процессе диалога обучаемый может допустить ошибки. Программа допускает совершение трех ошибок (составитель программы может изменить это число) как при заполнении таблицы состояний, так и на этапе выделения из графического модуля заданной схемы триггера. Весовые значения ошибок выбраны одинаковыми. Если их число превысит три, обучаемый возвращается к началу диалога.


```

10 DIM NAZ%(8),KT(8),KTI%(8,4),XV(6),
  YV(6),XS(16),YS(16),KV(8),NV(8,4),
  NZV%(8,4),KOS(8),NOS(8,16)
20 PRINT CHR$(12)
30 DEF FNAT%(X%,Y%)=CHR$(27)+
  CHR$(127+X%)+CHR$(127+Y%)
40 PRINT FNAT%(2,15)CHR$(133);" ***
  РЕАЛИЗАЦИЯ ЛОГИЧЕСКИХ ФУНКЦИЙ
  ТРИГГЕРА *** ";CHR$(134)
50 DATA RS-ТРИГГЕР АСИНХРОННЫЙ,RS-
  ТРИГГЕР СИНХРОННЫЙ,JK-ТРИГГЕР
  СИНХРОННЫЙ,T-ТРИГГЕР (СЧЕТНЫЙ)
  АСИНХРОННЫЙ,TV-ТРИГГЕР АСИНХРОННЫЙ,
  TV-ТРИГГЕР СИНХРОННЫЙ,D-ТРИГГЕР
  СИНХРОННЫЙ,DV-ТРИГГЕР СИНХРОННЫЙ
60 PRINT FNAT%(4,1)
70 FOR I=1 TO 8
80 READ NAZ%(I)
90 PRINT TAB(20)I;". ";NAZ%(I)
100 PRINT
110 NEXT I
120 PRINT FNAT%(23,20);:INPUT"ВЫБЕРИТЕ
  И ВВЕДИТЕ НОМЕР ---> ",N
130 PRINT CHR$(12)
140 PRINT FNAT%(2,20)CHR$(133);" --
  ЗАПОЛНИТЕ ТАБЛИЦУ ИСТИННОСТИ -- ";
  CHR$(134)
150 PRINT FNAT%(4,25)NAZ%(N)
160 DATA 2,R 0 0 1 1,S 0 1 0 1,
  Q 1 0 NQ,3,C 0 0 0 0 1 1
  1 1, R 0 0 1 1 0 0 1 1,
  S 0 1 0 1 0 1 0 1,0 Q 0 Q
  Q Q 1 0 NQ,3,C 0 0 0 0 0
  1 1 1 1,J 0 0 1 1 0 0 1 1,
  K 0 1 0 1 0 1 0 1 0 1
170 DATA Q 0 0 0 0 0 1 NQ,1,
  T 0 1,0 NQ,2,V 0 0 1 1,
  T 0 1 0 1,0 Q 0 Q NQ,3,
  C 0 0 0 0 0 1 1 1 1,
  V 0 0 0 1 1 0 0 1 1,
  T 0 1 0 1 0 1 0 1,
  Q 0 0 0 0 0 Q 0 Q NQ,2,
  C 0 0 1 1,D 0 0 1 0 1,
  Q 0 0 0 1,3
180 DATA C 0 0 0 0 1 1 1 1,
  V 0 0 1 1 0 0 1 1,
  D 0 1 0 1 0 1 0 1,
  Q 0 0 0 0 0 0 0 1
190 FOR I=1 TO 8
200 READ KT(I)
210 FOR J=1 TO KT(I)+1
220 READ KTI%(I,J)
230 NEXT J
240 NEXT I
250 PRINT FNAT%(10,1)
260 FOR I=1 TO KT(N)
270 PRINT TAB(25)KTI%(N,I)
280 PRINT
290 NEXT I
300 PRINT TAB(25);:INPUT"Q ",A$
310 IF A$<>KTI%(N,KT(N)+1) THEN 1040
320 PRINT CHR$(12)
330 PRINT FNAT%(1,10)CHR$(133);"
  ЛОГИЧЕСКИЙ МОДУЛЬ ";CHR$(134),
  " НЕОБХОДИМО СОЗДАТЬ "
340 PRINT FNAT%(2,50)NAZ%(N)
350 PRINT "
  -----"
360 PRINT "
  I I I"
370 PRINT "X1>-----S1-----
  -----I R I TT I"
380 PRINT "
  I---I I"
390 PRINT "X2>---#-#-----
  -----#-S2---I & I I"
400 PRINT "
  ! ! I"
  #--S3-----! S4---I I I-Q"
410 PRINT "
  S6 ! I J I I"
420 PRINT "
  ! S7 X3>---
  #------S5---I I I"
430 PRINT "
  ! ! I---I I"
440 PRINT "
  ! ! I---I I"
  S8 X4>---S9---I C I I"
450 PRINT "
  - ! ! I 1 I I"
  ! I---I I"
460 PRINT "
  ! #-I I I---S10-
  #------S11---I & I I"
470 PRINT "
  ! I---I ! I I I"
480 PRINT "
  ! S12
  #-S13---#-S14---I K I I---NO"
490 PRINT "X5>---#-----
  -----! S15---I I I"
500 PRINT "
  I---I I"
510 PRINT "X6>---S16-----
  -----I S I I"
520 PRINT "
  I---I-----I"
530 DATA 5,1,7,1,10,18,12,31,17,1,19,1
540 DATA 5,11,7,37,8,28,8,37,10,37,9,
  7,10,9,12,25,12,37,14,20,14,37,16,
  18,16,28,16,37,17,37,19,11
550 FOR I=1 TO 6
560 READ XV(I),YV(I)
570 NEXT I
580 FOR I=1 TO 16
590 READ XS(I),YS(I)
600 NEXT I
610 DATA 2,1,R,6,S,3,1,R,4,C,6,S,
  3,2,J,4,C,5,K,2,2,1,4,T,2,2,V,4,
  T,3,2,T,3,V,4,C,2,2,D,4,C,3,2,D,
  3,V,4,C
620 FOR I=1 TO 8
630 READ KV(I)
640 FOR J=1 TO KV(I)
650 READ NV(I,J),NZV%(I,J)
660 NEXT J
670 NEXT I
680 FOR I=1 TO KV(N)
690 X=XV(NV(N,I)):Y=YV(NV(N,I)):
  IF N=4 THEN X=X+1
700 PRINT FNAT%(X,Y)CHR$(133)+
  NZV%(N,I)+CHR$(134)
710 NEXT I
720 DATA 2,1,16,3,1,9,16,9,2,3,4,5,9,
  11,13,14,15,10,2,3,4,5,6,9,11,13,
  14,15,10,2,3,4,5,6,9,11,13,14,15,
  9,2,4,5,6,8,9,11,14,15,11,2,3,4,
  5,7,9,10,11,13,14,15,10,2,4,5,7,
  8,9,11,12,14,15
730 FOR I=1 TO 8
740 READ KOS(I)
750 FOR J=1 TO KOS(I)
760 READ NOS(I,J)
770 NEXT J
780 NEXT I
790 PRINT FNAT%(5,61)
  "ВВЕДИТЕ НЕОБХОДИМОЕ"
800 PRINT FNAT%(6,61)
  " КОЛИЧЕСТВО "
810 PRINT FNAT%(7,61)

```



```

      " ПЕРЕКЛЮЧАТЕЛЕЙ "
820 PRINT FNATX(8,61)
      " " ; INPUT"---> ",K
830 IF K<>KOS(N) THEN 1040
840 PRINT FNATX(10,61)"ВВЕДИТЕ НОМЕР"
850 PRINT FNATX(11,61)"ПЕРЕКЛЮЧАТЕЛЯ"
860 PRINT FNATX(15,61)
      "ВВОД НОМЕРОВ В ПО-"
870 PRINT FNATX(16,61)
      "РЯДКЕ ВОЗРАСТАНИЯ !"
880 FOR I=1 TO K
890 PRINT FNATX(13,61);:
      INPUT" ---> S",S
900 IF N=4 THEN XS(S)=XS(S)+1
910 IF S<>NDS(N,I) THEN 1040
920 IF S=4 THEN PRINT
      FNATX(XS(S),YS(S))"---":GOTO 950
930 IF S=6 OR S=7 OR S=8 OR S=12 THEN
      GOSUB 1020:GOTO 950
940 PRINT FNATX(XS(S),YS(S))"----"
950 PRINT
      FNATX(13,61)"
960 NEXT I

```

```

970 PRINT FNATX(23,25)" ВЫ
      СПРАВИЛИСЬ С ЗАДАНИЕМ !
      ПОРАБОТАЕМ ЕЩЕ ? <Е?>"
980 RESTORE
990 INPUT T
1000 GOTO 20
1010 END
1020 PRINT FNATX(XS(S),YS(S)-1)" " ;
      PRINT FNATX(XS(S),YS(S))"!"
1030 RETURN
1040 PRINT FNATX(23,35)CHRX(133);
      " НЕВЕРНО !";CHRX(134)
1050 RESTORE
1060 FOR I=1 TO 1000:NEXT I
1070 SS=SS+1
1080 IF SS>3 THEN PRINT CHRX(12);
      PRINT FNATX(10,30)"ВЫ ДОПУСТИЛИ
      УЖЕ 4 ОШИБКИ ! ДО СВИДАНИЯ !":END
1090 PRINT CHRX(12)
1100 PRINT FNATX(10,35)
      "НАЧНЕМ СНАЧАЛА !"
1110 FOR I=1 TO 1000:NEXT I
1120 GOTO 20

```

Д. МАЛЬЦЕВ, В. КИТАЙКИН

NETSY — инструментальное средство для КУВТ-86

В статье описывается автоматизированная обучающая система NETSY — инструментальное средство для подготовки и использования обучающих курсов по любому предмету для КУВТ-86. Рассмотрим основные принципы ее построения.

В терминах системы порция информации, одновременно отображаемая на экране, называется кадром. Содержимым кадра может быть как текстовая, так и графическая информация.

Совокупность кадров, объединенных в логическую последовательность, образует тему — самостоятельный законченный раздел обучения. Каждая тема имеет свое имя (до 80 символов), хранится в отдельном файле и может быть предложена для изучения на одной или нескольких периферийных ЭВМ. Все кадры темы имеют индивидуальные номера (от 1 до 999), которые и являются идентификаторами кадров при создании и редактировании темы. Эти номера используются также и для организации связей между кадрами при построении обучения.

В процессе обучения кадры темы сменяют друг друга на экране, как страницы при чтении книги, но, в отличие от последних, кадры отображаются не по порядку их записи в теме, а в зависимости от

ответов обучаемого. Здесь реализован так называемый принцип «книги с перепутанными страницами». Методист, который разрабатывает тему, должен предусмотреть варианты возможных ответов обучаемого (в терминах системы предусмотренные методом ответы называются эталонными) и для каждого из них указать, какой кадр нужно следующим отображать на экране при совпадении введенного ответа с данным эталоном, а также кадр, который должен отображаться в случае несовпадения ни с одним из предусмотренных эталонных ответов.

Для организации связей между кадрами темы, а также для определения формы ответа обучаемого в системе введено понятие «устройства ввода». Под «устройством ввода» здесь понимается формализованное указание системе о том, какие средства нужно предоставить обучаемому для ввода ответа и каким образом анализировать введенный ответ.

В NETSY имеется шесть типов «устройств ввода».

1. LAST — «устройство ввода» последнего кадра.

Используется в последнем кадре темы для информирования системы о том, что изучение данной темы закончено.

Методист при создании последнего кадра должен в качестве «устройства ввода» указать LAST. Это информирует систему о том, что из данного кадра нет ни одного перехода и что нужно ставить запрос на загрузку с центральной ЭВМ новой темы. После отображения содержимого такого кадра на экран ожидается нажатие клавиши ВВОД, после чего на экране появляется заставка обучающей системы, которая отображается всегда, пока в данную БК тема еще не загружена.

2. EMPTY — пустое «устройство ввода».

Используется в информационных кадрах, где обучаемому не задаются вопросы, а предлагается некоторый учебный материал, после изучения которого отображается содержимое следующего кадра (будем называть его кадром перехода). Номер кадра перехода методист должен указать при создании кадра в качестве параметра EMPTY. Во время обучения система в этом случае ожидает нажатия клавиши ВВОД, после чего осуществляется переход к тому кадру, который был указан методистом.

Рассмотренные «устройства ввода» фактически никакой информации от обучаемого не вводят и служат только для организации связей между кадрами. Следующие типы «устройств ввода» используются непосредственно для ввода ответа обучаемого и должны присутствовать в кадрах с вопросами.

3. CHOISE — выбор альтернативы.

Используется, если обучаемый должен выбрать ответ из множества предложенных ответов, заданных текстовыми строками.

При создании кадра с вопросом (предположим, что это кадр номер 56) методист в качестве «устройства ввода» должен указать CHOISE, а затем для каждого возможного ответа указать все его параметры: позиция печати строки-ответа на экране (ответы могут быть напечатаны в любом месте экрана, не обязательно друг за другом);

текст строки-ответа;

номер кадра перехода по данному ответу (для организации связей между кадрами);

признак правильности ответа (для сбора статистики ответов на центральной ЭВМ).

4. STRING — ввод текстовой строки.

Используется, когда ответ обучаемого должен представлять собой последовательность алфавитно-цифровых символов. В отличие от CHOISE, где обучаемый из набора готовых ответов должен выбрать один, по его мнению, правильный, в STRING ученик в качестве ответа должен сам набрать некоторую строку и завершить ее нажатием

клавиши ВВОД. В указанной позиции на экране появляется символьный курсор и обучаемый получает возможность набирать любые видимые алфавитно-цифровые символы от позиции курсора до конца текущей строки, при этом клавишей ЗБ можно уничтожить последний введенный символ. После нажатия клавиши ВВОД введенная строка сравнивается с заданными методистом эталонными ответами, и после совпадения с одним из них на экране отображается кадр с тем номером, которые соответствует данному эталону.

5. VALUATOR — ввод числа.

Используется для ввода числовых ответов. В системе предусмотрено два типа VALUATOR: для работы с целыми числами и числами с плавающей точкой. При наборе целых чисел можно пользоваться только цифровыми клавишами и клавишами для указания знака числа «+» или «-», а при наборе чисел с плавающей точкой еще и клавишей «.» для ввода десятичной точки и «Е» или «е» для указания начала набора порядка. При создании кадра методист должен в качестве эталонных ответов указать некоторые числовые диапазоны, а также соответствующие им кадры перехода. При попадании введенного числа-ответа в один из таких диапазонов на экране будет отображен соответствующий кадр.

6. LOCATOR — выбор позиции.

Используется, если в качестве ответа нужно указать некоторую точку на экране. Предположим обучаемому предлагается некоторое изображение (например, стилизованная географическая карта) и ставится задача указать на этом рисунке некоторый фрагмент (например, город или область). В указанном месте на экране появляется графический маркер, который с помощью стрелочных клавиш можно перемещать по экрану во всех направлениях и таким образом установить его в любую допустимую позицию. Клавиша ВВОД означает, что позиция выбрана. Методист при создании кадра должен в качестве эталонных ответов указать различные области экрана и соответствующие им кадры перехода. Если установленный обучаемым маркер попадет в одну из таких областей, указанный кадр отобразится на экране.

Система NETSY рассчитана на эксплуатацию в классах КУВТ-86. С помощью графического редактора, входящего в комплект поставки системы, преподаватель может самостоятельно «запрограммировать» любой учебный материал. Слово «запрограммировать» взято в кавычки, поскольку собственно программировать здесь не требуется.

Всю работу по записи информации и организации связей между кадрами берет на себя система. Преподаватель должен только изобразить картинку, указать тип и параметры (варианты возможных ответов и соответствующие им кадры перехода) «устройства ввода» и присвоить полученному изображению некоторый номер, по которому оно в дальнейшем будет распознаваться. При формировании изображения графический редактор предоставляет разнообразные возможности для работы как с алфавитно-цифровыми символами, так и с графическими примитивами — точками, ломаными, окружностями, а также возможности копирования кадров, изменения их номеров и т. д. Сформированные кадры сохраняются на накопителе, образуя своего рода базу данных.

Во время обучения программа, запускаемая на центральной ЭВМ, получает всю необходимую информацию из этой базы данных. Преподавателю предлагается меню существующих на данный момент тем. Выбрав одну из них, нужно указать номера периферийных ЭВМ, на которых будет изучаться данная тема, при этом на другие периферийные машины может быть назначена для изучения другая тема, и т. д. В общем случае на всех рабочих местах учеников могут изучаться различные темы.

Для начала работы системы преподавателю необходимо перейти в рабочий режим. Кадры начинают передаваться по сети в периферийные машины и отображаются там по заданному сценарию в зависимости от ответов учащегося, а на экране центральной ЭВМ преподаватель может наблюдать за ходом ведения занятий. Экран ДВК в этом режиме разделен на 12 частей (по числу рабочих мест учащихся), и в каждом из образовавшихся окошек указывается фамилия учащегося, а также количество верных и неверных его ответов к данному моменту времени. При желании эта информация может записываться на накопитель для дальнейшей ее обработки.

Что же дает применение системы?

1. Преподаватель получает возможность самостоятельно в интерактивном режиме создавать обучающие курсы. Создавать темы в графическом редакторе намного проще по сравнению с использованием для этого языков программирования. В комплект поставки системы входит руководство пользователя по системе, с помощью которого можно освоить работу с графическим редактором буквально за пару часов.

2. Применение системы для обучения позволяет в полную силу использовать все возможности класса КУВТ-86. До настоящего времени не существовало никакого про-

граммного продукта, который использовал бы КУВТ-86 именно как класс, а не как набор отдельных ЭВМ. Те программы, которые были разработаны для БК, имеют, во-первых, ограниченный объем вследствие малого количества доступной памяти, а во-вторых, их загрузка отнимает довольно много времени и требует некоторых действий со стороны учащихся, что часто вызывает определенные затруднения, особенно среди учеников младших классов. Использование возможностей сети в NETSY решает эти проблемы. Очеренные порции информации по мере необходимости подгружаются в периферийные машины без чьего-либо участия. Это происходит в то время, пока ученик обдумывает ответ или изучает отображенный на экране материал, и фактически незаметно ни для учащегося, ни для преподавателя. Система сама в нужное время считывает необходимую порцию материала с накопителя, размер которого теперь является единственным ограничением на длину темы. Следовательно, дополнительное время на загрузку программ практически не тратится и снимается ограничение на объем темы.

3. Появилась возможность передачи результатов работы учащегося на центральную ЭВМ с целью их сохранения и дальнейшего анализа. Система в этом смысле не только позволяет получить информацию о количестве верных и неверных ответов, но и предоставляет возможность анализировать сами ответы ученика на любой из предлагаемых ему вопросов. Кроме того, система позволяет проследить весь путь движения учащегося при изучении темы, что бывает важно при наличии в теме большого количества разветвлений в зависимости от ответов учащегося. При желании все эти данные могут быть записаны в отдельный файл и выведены на печать. Такой подход позволяет преподавателю не только более качественно оценить работу ученика, но и выявить недостатки в разработанной им методике. Если, например, все ученики на один и тот же вопрос отвечают одинаково неверно, значит, причину следует искать в недостаточно продуманной методике изложения учебного материала. Таким образом, применение NETSY, кроме всего прочего, позволяет преподавателям повышать свою квалификацию.

4. После загрузки системы во все периферийные машины управление ходом ведения занятия производится с центральной ЭВМ без участия учеников. Никаких действий по управлению системой со стороны учащихся не требуется, они должны выполнять только те задания, которые подготовил для них методист при создании темы.

5. Отсутствует возможность «сломать» систему со стороны учащегося. В любой момент времени ему разрешено нажимать только те клавиши, которые необходимы для ввода ответа. В противном случае подается звуковой сигнал. Никаких самостоятельных действий ученик делать не может. Ему позволено только то, что разрешил методист при составлении курса. Это устраняет страх некоторых учеников перед компьютером, позволяя сосредоточиться на изучении материала, вместо того чтобы думать, каким образом управлять ЭВМ. Для преподавателя это является гарантией того, что ученик не сможет проникнуть в систему незаконным путем, и поэтому он может не следить за ходом обучения. Таким образом можно организовывать самостоятельную работу учащихся без преподавателя, а только в присутствии лаборанта.

58

6. Одним из достоинств является исключительная простота использования системы преподавателем во время обучения. От него практически не требуется даже знания команд операционной системы. Нужно только запустить рабочую программу, а затем указать, на каких рабочих местах какая тема будет изучаться, а все остальное берет на себя система. При наличии обучающих курсов уроки с использованием NETSY может провести преподаватель, фактически впервые познакомившийся с системой.

При наличии хорошей методической и технической поддержки применение ЭВМ в учебном процессе сулит неоспоримые преимущества, против которых не станут воз-

ражать самые активные скептики. NETSY в этом плане не панацея от всех бед. Система не претендует на всеобщность: в данной версии, например, невозможно задавать движущиеся по экрану объекты. Тем не менее при широком ее внедрении появляется возможность наработки большого количества обучающих программ, которые имеют единую структуру и поэтому смогут легко распространяться между различными учебными заведениями и применяться в учебном процессе любым преподавателем.

Для проведения обучения с использованием NETSY стандартная конфигурация класса КУВТ-86 не требует никаких аппаратных доработок. Однако для работы графического редактора системы необходимо наличие на ДВК платы КГД, которая в составе класса не поставляется. Эту плату заказчики могут приобрести вместе с комплектом программ.

В настоящее время ведутся работы по адаптации описанной АОС к классам ЭВМ БК-0011 и УКНЦ, в которых все элементы системы будут рассчитаны только на стандартные аппаратные средства.

Адаптация системы на БК-0011 производится таким образом, чтобы обеспечить совместимость обеих версий системы на уровне данных. Это означает, что наработанные для БК-0010 обучающие курсы смогут без дополнительных затрат использоваться в классе БК-0011. Совместимость с УКНЦ не гарантирована.

Адрес для справок: Москва, К-498, МИЭТ, военная кафедра, Мальцеву Д. В.

В. КАНИВЕЦ

Активное средство защиты авторских прав

Для защиты от несанкционированного использования программных продуктов пользователями операционной системы MSX-DOS («Ямаха» MSX-2) предлагается самостоятельно репродуцирующаяся программа (вирус) JUST. Нормальная работа программы-носителя JUST зависит от зараженности и порядка расположения соседствующих на дисковом пространстве файлов, а также от числа ее запусков. Будучи «вырванной» из подобранного пользователем окружения или по истечении определенного числа запусков (зависящего опять-таки от окружения) программа-носитель в лучшем случае уничто-

жит себя, в худшем же будет приписывать тело вируса окружающим ее загрузочным модулям (COM-файлам). Программа с приписанным телом вируса (JUST-зараженная программа) либо разрушается, либо превращается в носителя вируса, способного нормально работать и заражать как другие программы, так и себя при копировании с места заражения. Как показывает практика использования JUST, возможны и непредсказуемые действия зараженных программ (полное уничтожение содержимого дискеты, перезагрузка либо зависание системы и т. п.). Непредсказуемые дей-

ствия будут иметь место и в случае отсутствия на дисковом пространстве зараженного или не защищенного от заражения COM-файла.

Работа JUST основана на следующих особенностях файловой организации операционной системы MSX-DOS:

файл на дискете занимает целое число кластеров, длина одного кластера равна 1024 байтам (2 сектора);

COM-файл загружается системой для выполнения с адреса 100H;

длина COM-файла обычно кратна 128 (80H) байтам.

Первое означает, что в пяти случаях из восьми последние 378 байтов последнего кластера каждого COM-файла свободны. В них и записывается вирусная часть, адрес вызова которой, в силу второй особенности, вычисляется по формуле

$$100H + \text{длина файла} + \text{смещение},$$

где смещение — расстояние в байтах между концом COM-файла и 280H-м байтом в последнем кластере. Из третьей особенности следует, что смещение можно брать равным одному из чисел 0, 80H, 100H, 180H, 200H, 280H, 300H, FF80H.

```

        .280
        ASEG
ABUF   EQU 80H
BEGIN  EQU 58A0H
ADRES  EQU 58A0H
ASEC   EQU 5900H
ABYTE  EQU 27H
HELP   EQU 0A5H
        ORG 100H
; 1-й шаг алгоритма
;
ST:    PUSH HL
        PUSH BC
        PUSH DE
        PUSH AF
        LD HL,HELP
        LD (HL),0E1H
        INC HL
        LD (HL),0E5H
        INC HL
        LD (HL),0C9H
        CALL HELP
        LD BC,11H
        SCF
        SBC HL,BC
        LD (BEGIN),HL
; 2-й шаг алгоритма
;
        LD DE,ASEC      ; УСТАНОВКА
        LD C,1AH        ; ОБЛАСТИ ОБМЕНА
        CALL 5          ; С ДИСКОВОДОМ.
        LD DE,7H        ; ЧТЕНИЕ
        LD HL,0200H     ; СЕКТОРОВ
        LD C,2FH        ; 7,В
        CALL 5H        ; (КАТАЛОГА).
        LD HL,ASEC+40H
LOOP:  LD DE,ABUF
        LD BC,32
        LDIR

```

```

LD A, (ABUF+8H)
CP 43H
JR NZ,LOOP
JR NZ,NEXT
LD A, (ABUF+0FH)
CP 15H
JR Z,NEXT
JR OP
NEXT:  ADD HL,BC
JR LOOP
OP:    SCF
        LD BC,16
        SBC HL,BC
        LD (HL),13H
        LD BC,14      ; РАЗМЕР
        ADD HL,BC     ; ФАЙЛА
        LD A, (ABUF+29) ; УВЕЛИЧИВАЕТСЯ
        ADD A,4
        LD (HL),A
        LD DE,7H      ; ЗАПИСЬ
        LD HL,0200H   ; СЕКТОРОВ
        LD C,30H      ; 7,В
        CALL 5H
3-й шаг алгоритма
        LD A, (ABUF+28) ; ОПРЕДЕЛЕНИЕ
        CP 00         ; СМЕШЕНИЯ
        JR NZ,C80
        LD A, (ABUF+29)
        BIT 0,A
        JR Z,C000
        BIT 1,A
        JR Z,C0100
        LD BC,80H
        JR L11
C000:  BIT 1,A
        JR Z,C0000
        LD BC,180H
        JR L11
C80:   LD A, (ABUF+29)
        BIT 0,A
        JR Z,C080
        BIT 1,A
        JR Z,C0180
        LD BC,00H
        JR L11
C080:  BIT 1,A
        JR Z,C0080
        LD BC,100H
        JR L11
C0080: LD BC,300H
        JR L11
C0180: LD BC,200H
        JR L11
C0000: LD BC,0FF80H
        JR L11
C0100: LD BC,280H
L11:   LD HL, (ABUF+28) ; ВЫЧИСЛЕНИЕ
        ; АДРЕСА
        ADD HL,BC      ; ВЫЗОВА
        LD (ADRES),HL ; ВИРУСНОЙ ЧАСТИ
; 4-й шаг алгоритма
;
        LD HL, (ABUF+26) ; ВЫЧИСЛЕНИЕ
        ADD HL,HL      ; НОМЕРА ПЕРВОГО
        LD BC,10      ; СЕКТОРА
        ADD HL,BC
        PUSH HL        ; ЧТЕНИЕ
        POP DE         ; ПЕРВОГО
        PUSH DE        ; СЕКТОРА
        LD HL,0200H
        LD C,2FH
        CALL 5

```



```

LD HL,ASEC      ; ПОИСК                               ; 6-й шаг алгоритма
LD A,@CDH      ; БАЙТА                               ;
LD BC,@FFFFH   ; СОДЕРЖАЩЕГО
CPIR           ; ЧИСЛО CD                               ; (=CALL)
DEC HL
LD DE,ABYTE
LD BC,3
LDIR
LD A,@C9H
LD (DE),A
DEC HL
DEC HL
LD A,(ADRES)
LD (HL),A
INC HL
LD A,(ADRES+1)
LD (HL),A
POP DE         ; ЗАПИСЬ
LD HL,@200H   ; ПЕРВОГО
LD C,30H      ; СЕКТОРА
CALL 5
; 5-й шаг алгоритма
;
PUSH IX       ; ВЫЧИСЛЕНИЕ
LD IX,@H      ; НОМЕРА
LD HL,@H      ; ПОСЛЕДНЕГО
LD BC,@0H     ; СЕКТОРА
LD D,@
CALC: LD A,(ABUF+29)
CP H
JR NZ,COUNT
LD A,(ABUF+28)
CP L
JR Z,OUTS
COUNT: INC D
LD A,D
CP B
JR NZ,INCS
INC IX
LD D,@
INCS: ADD HL,BC
JR CALC
OUTS:  PUSH IX
      POP BC
      POP IX
      LD HL,(ABUF+26)
      ADD HL,BC
      ADD HL,HL
      LD BC,10
      ADD HL,BC
      PUSH HL           ; ЧТЕНИЕ
      POP DE           ; ДВУХ ПОСЛЕДНИХ
      PUSH DE          ; СЕКТОРОВ
      LD HL,@200H
      LD C,2FH
      CALL 5
      LD HL,ASEC
      LD BC,640;
      ADD HL,BC
      PUSH HL
      POP DE
      LD HL,(BEGIN)
      LD BC,171H
      LDIR
      LD HL,ABYTE+1
      LD BC,6
      LDIR
      POP DE           ; ЗАПИСЬ
      LD HL,@200H     ; ДВУХ ПОСЛЕДНИХ
      LD C,30H        ; СЕКТОРОВ
      CALL 5
      LD HL,(BEGIN)
      LD BC,170H
      ADD HL,BC
      LD DE,(BEGIN)
      LD BC,5
      LDIR
      POP AF
      POP DE
      POP BC
      POP HL
      CALL @000
      RET
      END

```

Программа JUST реализует следующий алгоритм.

1. Определение начального адреса вирусной части зараженной программы, запись его в ячейку BEGIN.

2. Поиск в секторах каталога указателя на незараженный COM-файл, увеличение размера найденного файла на 1024 байта, запись в 16-й байт указателя числа IZH — отметки о зараженности.

3. Анализ «старой» длины найденного COM-файла и вычисление адреса перехода на вирусную часть (полученный адрес записывается в ячейки ADRES, ADRES+1).

4. Поиск в первом кластере найденного файла байта, содержащего число CDH; последнее трактуется как CALL, а содержащее следующих двух байтов (обозначаемое A_L , A_H) — как адрес перехода. В эти байты записывается адрес перехода на вирусную часть, а четыре шестнадцатеричных восьмизарядных числа CD, A_L , A_H , C9 запоминаются с адреса ABYTE.

5. Запись в последний кластер найденного файла тела вируса, т. е. содержимого 372 байтов оперативной памяти, начиная с адреса, содержащегося в ячейке BEGIN, плюс содержимое 4 байтов с адреса ABYTE.

6. Запись в ОЗУ с адреса, содержащегося в BEGIN, последовательности шестнадцатеричных чисел CD, A_L , A_H , C9. Таким образом, вирусная часть «срабатывает» не более одного раза за один запуск носителя.

Последний пункт необходим для избежания многократности заражения при нахождении оператора вызова вирусной части в области действия цикла.

Способы превращения произвольной программы в JUST-носитель, как и возможные модификации JUST, легко могут быть получены после анализа предложенного исходного текста и описания программы и поэтому здесь не рассмотрены. Заинтересованным читателям, не желающим тратить время на указанный анализ, автор готов сообщить свои рекомендации.

Беседы с «Сократом»

Коммерческие успехи первых проектов фирмы «МИГ» в области компьютерного обучения определили направление дальнейших поисков: сетевые обучающие системы. К этому времени (осень 1990 г.) завод «Экситон» перестал экспериментировать с содержимым ПЗУ и приступил к выпуску БК-0011М и классов на его основе — УКНЦ-01; таким образом определился и объект приложения сил наших программистов.

В ту пору класс практически не имел специализированного программного обеспечения, способного использовать все немалые возможности нового компьютера; основу предлагаемого заводом обеспечения составляли старые RT-шные программы для ДВК и программа-эмулятор БК-0010. Ситуация была привычно парадоксальной: единственным применением «старшего» БК стало прокручивание обширной библиотеки игр младшей модели.

Проект «Сократ» начался с написания драйвера перекодировки клавиатуры PK.SYS, позволившего работать с текстовым и кодовым редакторами RT-11, и драйвера квазидиска VD.SYS, после чего программирование на БК-0011 из аттракциона для хакеров превратилось в нормальную работу.

Ядром системы является сетевой диспетчер преподавательской ПЭВМ (см.: ИНФО. 1991. № 4. С. 92), обслуживающий файловые классы с РМУ, позволяющий машинам класса обмениваться текстовыми сообщениями, дающий возможность педагогу не только «подсматривать» за экранами РМУ, но и перехватить программу, с которой работает ученик, продолжить ее выполнение на РМП, возможно, что-то изменить в моделируемой среде и вернуть на РМУ, где ученик продолжит работу с того места, на котором остановился преподаватель. Таким образом закладывается основа для полноценного компьютерного диалога «ученик — преподаватель», сотрудничества и сотворчества, педагогическую ценность которых трудно переоценить.

«Сократ» — это оболочка, которую можно наполнить обучающими программами произвольного характера, как текстовыми, так и моделирующими. Преимущество вторых — эффективных ассемблерных программ, моделирующих изучаемый объект, — вполне очевидно. Такая программа должна предоставлять пользователю возможность изменять некоторые параметры моделируемой системы

и затем с достоверностью, достаточной для целей обучения, воспроизводить реакцию системы. В процессе общения с такой программой, весьма напоминающей игру, учащийся опытным путем, погружаясь в моделируемую среду, постигает закономерности ее функционирования, приобретает навыки управления или использования, проверяет корректность своих представлений о данном объекте.

В настоящий момент наполнение «Сократа» составляют две программы: «Бэби-САПР» и «PDP-11». Первая визуализирует временные диаграммы процессов, происходящих в несложных (до 16 логических элементов) цифровых схемах, которые предвзительно должны быть «собраны» на экране монитора «Бэби-САПР». При работе с программой (интерфейс ее, конечно, рассчитан на новичка и несложен) приобретаются не только углубленные знания принципов работы ЭВМ, но и навыки компоновки и трассировки печатных плат.

«PDP-11» — действующая модель процессора ЭВМ PDP-11, аналогом которой являются процессоры компьютеров ряда СМ, ДВК, «Электроника», БК. Программа демонстрирует работу процессора, знакомит с набором его команд и языком ассемблера, позволяет создавать и отлаживать на моделируемом процессоре программные модули на ассемблере или непосредственно в машинных кодах. Учащиеся узнают, как происходит обмен данными с ОЗУ, наблюдают сигналы системной магистрали, видят, как исполняется каждая из команд, что меняется в работе процессора при изменении способа адресации, легко обнаруживают ошибки в своей программе. «PDP-11» можно использовать как для поверхностного ознакомления с работой процессора, так и в качестве мощного тренажера при подготовке программистов, существенно сокращая время изучения ассемблера, избавляя от ошибок, характерных для начинающих.

Сетевой диспетчер позволяет сохранять на дискете трассировку платы для «Бэби-САПР» или программу на ассемблере для «PDP-11».

Эти программы, несомненно, найдут широкое применение в лабораторных практиках курсов «Элементарная база ЭВМ», «Узлы ЭВМ», «Микропроцессоры» в ПТУ, техниках и вузах. Хороший сервис и при-

лагаемые варианты заданий избавляют преподавателей перечисленных предметов от рутины, освобождая время для творческого труда.

Готовится перенос «Сократа» на УКНЦ-02.02 (МС 0511) завода «Квант», так что к моменту выхода статьи фирма «МИГ» предложит этот пакет программ пользователям обоих классов УКНЦ.

В заключение несколько слов о впечатлении, производимом УКНЦ-01. Несомненные достоинства БК-0011 — элегантная архитектура, простота переключения восьми страниц ОЗУ и четырех — ПЗУ, наличие двух страниц видео-ОЗУ и возможность их непосредственного включения в адресное пространство процессора, что существенно облегчает создание графического сопровожде-

ния. Приятной неожиданностью стала исключительно надежная работа локальной сети. Хотя реальная скорость передачи данных оказалась ниже паспортной — 16—20К бод,— она достаточна для реализации систем с активным обменом. Прочее «железо» также исправно работало по 8—10 часов в сутки. А весной появилась версия 3.2 ОС БК11 с устраненными недостатками и «нестучащим» драйвером дисковода.

В числе недостатков — традиционно грубоватый дизайн с висящими на одних лишь разъемах внешними модулями и соединителями, что лишает конструкцию прочности (весьма желательной при эксплуатации в учебных заведениях!).

Здравствуй, «Русич»!

Слух обо мне пройдет по всей Руси
великой...
А. С. Пушкин

Двенадцать столетий известны всему свету государство Русь и сыны земли русской — русичи. Испокоп веков славилась они трудолюбием и талантами, не было в мире такой премудрости или чуда техники, освоить и превзойти которые они не могли бы. А в конце XX века собрались русичи из славного Белгорода, с Белгородского радиоприборного завода, и решили создать акционерное общество «Прогресс», чтобы было на чем детям нашим осваивать зело мудрую науку — информатику. И сделали они КУВТ и нарекли его «Русич». Трудно живется сейчас на Руси, не хватает денег на самое необходимое, а заморская техника дорогая. Поэтому и решили на АО «Прогресс», что классы должны быть по цене доступны школам, учитель должен стать «хозяином» компьютера IBM PC/AT, а белгородские умельцы позаботятся о детях — сделают недорогой и надежный компьютер для рабочих мест учащихся. При таком решении вопроса учитель получает возможность пользоваться всем богатством программного обеспечения IBM-совместимых компьютеров, а школа — подготовиться как к информатизации учебного процесса на базе нового поколения учебной вычислительной техники, так и к информатизации управленческой деятельности (использование АРМов бухгалтера, директора, завуча, завхоза и т. д.).

В настоящее время семейство КУВТ «Русич» включает две модели компьютер-

ных классов: с 8-разрядными ПК «Русич» и с 16-разрядными IBM-совместимыми компьютерами на рабочих местах учащихся. В состав КУВТ «Русич» входит одно рабочее место преподавателя (IBM PC/AT) и 12 рабочих мест учащихся, объединенных в локальную вычислительную сеть. Программное и аппаратное обеспечение КУВТ «Русич» обеспечивает организацию управления и обмена информационным потоком в локальной сети, поочередную или одновременную загрузку программ с гибких или жестких дисков с головной машины в периферийные, пофайловый обмен между ученической и учительской ПЭВМ, совместное использование периферийных устройств IBM PC.

ПК «Русич» представляет одноплатную ПЭВМ в клавиатурном исполнении с низким энергопотреблением (всего 12 Вт), что определяет ее высокую надежность, а отсутствие вентилятора делает ее практически



бесшумной. ПК «Русич» сделан на основе 8-разрядного микропроцессора 1821BM85 (аналог Intel 8085) с быстродействием 500 тыс. оп/с, с 8-разрядной шиной данных. Оперативная память объемом 192 Кбайта позволяет создавать электронный диск объемом 110 Кбайт. Объем ПЗУ — 24 Кбайта. Локальная сеть типа «Звезда» обеспечивает передачу информации объемом до 64 Кбайт со скоростью 9,6 или 19,2 Кбит. Надежность и скорость позволяют снизить стоимость класса за счет отсутствия в базовой конфигурации на учебных машинах накопителей на гибких магнитных дисках. По желанию заказчика ПК «Русич» может снабжаться выносным НГМД 5,25" объемом 720 Кбайт. Машина обеспечивает цветную (8 цветов) и черно-белую графику с разрешающей способностью 764×256 (черно-белая) и 384×256 (черно-белая и цветная). В качестве дисплеев в стандартной поставке используются мониторы «Электроника МС 6105.05» и «Электроника МС 6106».

ПК «Русич» обеспечивает под управлением сетевого монитора создание и выполнение системных и прикладных программ. В качестве базовой операционной системы применяется самая распространенная среди 8-разрядных ПЭВМ система CP/M, что позволяет использовать все богатство системного программного обеспечения, накопленного более чем за 15 лет существования этой ОС. В состав программного обеспечения, поставляемого с компьютером, входят трансляторы с языков BASIC, FORTRAN, ASSEMBLER, PROLOG, LISP, PASCAL, FORTRAN, база данных dBASE II, электронные таблицы SuperCalc и Multiplan, текстовые и графические редакторы, игровые и обучающие программы по информатике, физике, химии, математике, иностранным языкам и другим предметам.

Вторая модель КУВТ «Русич» используется в качестве рабочего места ученика IBM-совместимый компьютер, представляющий одноплатную ПЭВМ с безопасным питанием 42 В, практически бесшумную, с естественной вентиляцией. Отличительными особенностями этой модели от других выпускаемых «так называемых IBM-совместимых компьютеров» является использование в качестве процессора Intel 80286, ставшего стандартом «де факто» для 16-разрядных ПЭВМ и позволяющего учащемуся работать с программным обеспечением не только для компьютеров класса IBM PC/XT, но и IBM AT, большая оперативная память — от 640 Кб до 1 Мб и стандартная графика с высоким разрешением. Компьютеры вы-



63

пускаются разных модификаций с различными видеоадаптерами (Hercules, EGA, VGA) и комплектуются импортными видеомониторами с хорошими санитарно-гигиеническими характеристиками. В IBM-совместимой модели используется локальная сеть типа общая шина со скоростью передачи 1 Мбит. По желанию заказчика на учебных машинах могут ставиться накопители на гибких магнитных дисках.

На базе 8 и 16-разрядной модели разработаны и выпускаются переносные компьютеры класса LAPTOP с питанием от обычной сети или автомобиля. В качестве видеомонитора используется газоразрядная панель с хорошим разрешением и гигиеническими характеристиками (полностью отсутствуют излучения и мерцание экрана).

Обычно школьные кабинеты вычислительной техники оборудуются демонстрационными цветными телевизорами, главными недостатками которых являются мелкое и некачественное изображение и сложность их подключения к компьютерной технике.

Впервые в нашей стране в АО «Прогресс» разработано табло коллективного пользования, представляющее газоразрядную панель размером 1544×772 мм, обеспечивающее либо монохромное (зеленое), либо трехцветное (красно-зелено-желтое) изображение. Информационная емкость табло — 512×256 световых элементов, размер светового элемента — 2×2 мм, время построения изображения на экране — 0,1 с. Табло позволяет выводить изображение с компьютеров IBM PC, удаленных на расстояние до 100 м.

С 1991/92 учебного года акционерное общество «Прогресс» будет предоставлять пользователям «под ключ» уникальный компь-



64 лекс компьютерного класса, который оборудуется рабочими местами со встроенными в мебель компьютерами и электронными табло, специально разработанными дизайнерами. А на очереди уже 32-разрядные ПЭВМ на базе разработанного в АО «Прогресс» RISC процессора.

Для семейства КУВТ «Русич» разработан учебно-методический комплекс, полностью поддерживающий школьный курс информатики. В его создании активно участвовали члены авторского коллектива школьного учебника «Основы информатики и вычислительной техники» (Просвещение, 1989) Ерохина Е. А., Федюшин Д. П., Щеголев А. Г. и автор языка «Пролог-Д» Григорьев С. Г. В состав комплекса входят: учебник, методическое пособие для учителей и электронный задачник по логическому программированию, интегрированный пакет «Компьютерная грамотность».

Многоальтернативный компьютерный учебник «100 уроков информатики на КУВТ «Русич» представляет конструктор уроков и содержит следующие основные разделы: введение в информатику и компьютерная грамотность, логика и логическое программирование, алгоритмизация и процедурное программирование, решение задач на ЭВМ, основы вычислительной техники, применение ЭВМ. Учебник строится как система поурочных блоков, позволяющих конструировать курсы для классов с различной специализа-

цией (гуманитарных, технических, физико-математических и общеобразовательных). Блоки включают теоретический раздел, лабораторный практикум и задачи. Наличие в КУВТ «Русич» хорошего принтера позволяет учителю для каждого ученика распечатать материалы к текущему занятию, дифференцировать их в зависимости от индивидуальных особенностей учащихся, а если содержание каких-то уроков его не устраивает, вносить в них свои изменения.

Методическое пособие по логическому программированию для учителей позволяет преподавателям освоить новое перспективное направление в информатике и программировании, использовать язык Пролог в качестве базового языка курса.

Электронный задачник по логическому программированию на языке Пролог-Д представляет собой лабораторный практикум по разделу курса «Логика и логическое программирование». Каждая лабораторная работа (урок) состоит из демонстрации решения задачи, ее модификации школьником и самостоятельного решения аналогичной задачи. Электронный задачник работает в среде интерпретатора языка «Пролог-Д», одна из версий которого специально разработана для КУВТ «Русич».

Интегрированный пакет «Компьютерная грамотность» поддерживает начальный этап обучения информатике и содержит клавиатурный тренажер, текстовый редактор, графический редактор, учебные электронные таблицы и учебную базу данных, объединенные единым унифицированным интерфейсом.

Основным принципом, положенным в основу успеха КУВТ «Русич», является принцип комплексности: поставка и установка класса, обеспечение пользователя учебно-методическим комплексом, базовым и прикладным программным обеспечением, гарантийное обслуживание в течение 12 месяцев, обучение преподавателей работе с КУВТ «Русич», авторские курсы по преподаванию информатики по учебнику «100 уроков информатики на КУВТ «Русич».

Так здравствуй «Русич», и счастливого тебе пути в компьютерные классы наших школ, ПТУ и вузов!

Д. БОГОЛЮБОВ

«РУСИЧ»

Стоимость 8-разрядной модели КУВТ «Русич» (включая пусконаладочные работы, программное и методическое обеспечение, гарантийное обслуживание и обучение):
в монохромном варианте — 67/72 тыс. рублей;
в цветном варианте — 100 тыс. рублей.

Покупателям предоставляется скидка с каждого КУВТ на:
1,5 тыс. рублей при закупке партии в 10 КУВТ «Русич»;
3,0 тыс. рублей при закупке партии в 20 КУВТ «Русич»;
5,0 тыс. рублей при закупке партии в 30 и более КУВТ «Русич».

Стоимость табло коллективного пользования:
монохромный вариант — 90 тыс. рублей;
цветной вариант — 120 тыс. рублей.

Стоимость КУВТ «Русич» с IBM-совместимыми компьютерами:
от 100 тыс. рублей и выше в зависимости от комплектации.

За дополнительной информацией по применению и приобретению КУВТ «Русич» обращайтесь по адресу:

СССР, 308600, г. Белгород, ул. Победы, 85, АО «Прогресс»

Телефоны: (8-072-22)-2-13-51.

Телекс: 156355 АНОД.

Телефакс: (8-072-22)-2-13-98.

Специализированные программные средства

Русификация системы проектирования печатных плат PCAD 4.5.

Выпуск перфолент для разнообразных сверлильных станков:

КД-46;

ОФ-100;

СМ-600;

SCMOLL и других.

Связь различных графических систем проектирования:

PCAD;

AUTOCAD;

DRAGON.

Прикладные базы данных в среде русифицированных СУБД dBASE и RBase.

Средства для использования в области экономики и финансов в среде русифицированной оболочки SuperCalc версии 4.

Программы тепловых и вибрационных расчетов печатных плат и блоков РЭА. В результате расчетов вы получаете температуру:

△ корпуса блока;

△ воздуха внутри блока;

△ нагретой зоны около печатной платы;

△ распределение температуры на поверхности печатной платы, а также:

△ первую и вторую собственные частоты колебаний печатной платы,

произвольно закрепленной и произвольно нагруженной электрорадио-элементами;

△ поля напряженно-деформированного состояния материала печатной платы.

Мы решим все ваши проблемы!

Адрес-контакт:

105187, Москва, Кирпичная ул., 39—41, ПТО «Магистр-2».

Тел.: (095) 366-15-57.

Нераскрытые возможности компьютера «Atari-XE/XL» откроются перед вами, если вы приобретете три новых разработки фирмы «КОМТЕКС»

КОМТЕКС Текст Тест (КТТ)

Пакет «КТТ» состоит из нескольких программ, позволяющих учителю в течение нескольких минут подготовить и записать на дискету самозагружающийся программный продукт (содержит операционную часть, учебные и контрольные тексты), проводящий опрос на знание учебного материала учащимися, оценивающий их ответы по заданному вами алгоритму и сохраняющий на дисковом носителе и выводящий на печать результаты работы. Оригинальное графическое решение с использованием новейших разработок, возможность применения по большинству предметов учебной программы, ненавязчивость и простота диалога ученика с компьютером — это отличительная черта обучающей программы, создаваемой при помощи «КТТ». Надеемся, что, купив этот пакет, вы получите удовольствие от работы с ним, а ваши ученики — от работы с изготовленными с его помощью программами.

Стоимость пакета 1400 рублей.

КОМТЕКС Текстовой Редактор (КТР)

Такие возможности, как:

- * длина строки 250 знаков;
- * автоматическая установка курсора на начало и конец строки, на начало и конец текста;
- * режимы вставки и замещения;
- * помощь;
- * заглавные и прописные буквы кириллицы и латиницы;
- * распечатка текста и сохранение его на любом из внешних носителей информации одним нажатием клавиши;
- * объем текста до 30 000 знаков

и многие другие будут сопровождать вас в «КТР». Мы надеемся, что вы сможете достойно оценить его возможности.

Стоимость программы 250 рублей.

Текст-информация

Для видеостудий и студий эфирного и кабельного вещания предлагаем программу «Текст-информация». Она предназначена для формирования, редактирования и вывода на телевизионный экран текстовой информации с вертикальным перемещением ее снизу вверх с возможностью фиксации и регулировки скорости. Текст может формироваться шрифтом как русского, так и латинского алфавитов. Максимальный объем разового вывода информации — 10 240 знаков. Возможен вариант поставки к этой программе компьютера «Atari-65XE».

Стоимость программы 1050 рублей.

Программы поставляются на дискетах, а «Текст-информация» и на кассете.

Программы рассылаются по почте наложенным платежом.

Заявки и гарантийные письма высылайте по адресу:

167031, Сыктывкар, а/я 1315, фирма «КОМТЕКС».

Тел. (8-821-22) 2-22-34, 2-06-64;

факс (8-821-22) 2-06-64.

Ю. ЗАЛЬЦМАН

Архитектура и ассемблер БК

Архитектура БК

Векторы прерывания и программирование

Большинство случаев использования сведений о векторах прерывания сводится к изменению векторов и созданию новых программ обработки прерываний. Например, таким путем можно запретить останов программы клавишей **СТОП** или ввести дополнительные команды процессора, дополнительные символы клавиатуры — да мало ли что еще!

Перейдем к конкретным примерам. Предположим, нам необходимо, чтобы клавишей **СТОП** программу нельзя было остановить; это бывает необходимо в случаях, когда останов программы может привести к нежелательным последствиям.

Итак, поскольку клавиша **СТОП** внеприоритетная и запретить прерывание по вектору 4 мы не можем, нам остается только изменить вектор 4 и написать новую программу обслуживания прерывания. Включим в программу, где-нибудь в начале, команду, которая будет работать при запуске программы:

```
MOV #V4,@#4
```

Мы переписали вектор 4. Теперь при нажатии на клавишу **СТОП** управление будет передаваться не по прежнему адресу, а по адресу метки 4. А какую программу разместить по этому адресу? Это зависит от наших целей. Если нам надо только, чтобы клавиша **СТОП** вообще не работала, достаточно единственной команды возврата из прерывания (**RTI**), ее выполнением и закончится обработка прерывания по вектору 4:

```
V4: RTI
```

Окончание. Начало см.: Информатика и образование. 1990. № 4—6; 1991. № 1—4.

Но не все так просто. Необходимо учитывать, что клавиша **СТОП** может быть нажата в совершенно произвольный момент, и прерывает программу она не совсем так, как, скажем, команда **EMT** или прерывание от клавиатуры. При прерывании программы по клавише **СТОП** процессор может не закончить выполнение очередной команды! При этом в стеке будет сохранено значение **PC**, соответствующее адресу следующей команды, и с нее-то и продолжится выполнение программы. В зависимости от конкретных, выполняемых в этот момент команд такой «финт» может привести к самым неприятным последствиям, вплоть до зависания системы. Как же быть?

Необходимо предусмотреть в программе специальный, так называемый *холодный*, вход, т. е. адрес, при передаче управления на который не произойдет нарушения работы программы (иногда такой вход называют горячим — по аналогии с горячим резервом аппаратуры; но программирование не электроники, и такой, не инициализирующий программу, вход правильнее называть холодным). На этот вход и надлежит передать управление по клавише **СТОП**, для чего его адрес указывается в программе обработки вектора прерывания. Пусть адрес холодного входа нашей программы — по метке **BEG**. Напишем:

```
V4: MOV #732,SP
     JMP BEG
```

Вторая команда ясна — это просто передача управления на холодный вход. А зачем первая, что мы пишем в регистр **SP** и для чего?

Вспомним, как выполняется прерывание: в стек заносится на сохранение **ССП** и **PC**. При этом, конечно, изменяется и указатель стека **SP** — его значение уменьшается на 4. Команда **RTI** в конце программы

Вспомним, как выполняется возврат из прерывания. Из стека восстанавливается сначала РС, а затем ССП. То значение PS, которое было занесено в стек при переходе к обработке прерывания, пишется обратно в регистр PS. А потом? Потом начинается выполнение следующей команды программы... Вот это нам и нужно! Что, если перед возвратом из прерывания «подменить» в стеке ССП на такое, в котором установлен Т-разряд? Процессор его «скушает», не подзревая подвоха, и выполнит очередную команду, после чего проверит Т-разряд и обнаружит, что там 1. Ему придется выполнить прерывание по вектору 14. А тут-то мы его и будем ждать со своими средствами, чтобы посмотреть то, что нам нужно...

Но сначала нужно организовать прерывание, чтобы при выходе из него «обмануть» ЦП. По какой команде это сделать? В принципе по любой, лишь бы это было именно командное прерывание. Существует специальная команда BPT, которая вызывает прерывание по тому же вектору 14 (но не отладочное, по Т-разряду, а командное, вроде EMT). Воспользуемся ею и не будем трогать другие векторы.

А что мы будем делать во время отладки? Что хотим посмотреть? Прежде всего, какая команда выполняется, а для этого желательно знать ее адрес и код (пока мы не будем пытаться дизассемблировать команды — переводить их на язык ассемблера, это чересчур сложно). Кроме того, допустим, что нас интересует, к примеру, содержимое регистра R1. А затем мы напомним какую-нибудь программку, которая будет работать именно с R1, и посмотрим, что получится.

Желательно также иметь хотя бы минимальный сервис — как-то маркировать выводимые данные, чтобы можно было понять, где адрес, где код команды, а где — содержимое регистра. А как переходить к следующей команде? Пусть для простоты отлаживаемая программа выполняется тогда, когда нажата любая клавиша, а если клавиша опущена, можно просмотреть данные, которые выдает отладчик.

Итак, «техническое задание» на программный продукт готово. Осталось написать программу.

```

; Запуск отладчика
MOV   #BPT, @#14  Записать пусковой
;                               вектор 14
BPT   Прерывание по
;                               вектору 14

; Демонстрационная программа
MOV   #1, R1      Записать в R1
;                               единицу
@:    ASL   R1      Сдвиг влево R1
      ADC   R1      Прибавить перенос
      BR   @        Программа зациклена

```

```

; Трассировка демонстрационной программы
TRS:  MOV   R0, -(SP) Сохранить в стеке
      MOV   R3, -(SP) регистры,
      MOV   R4, -(SP) используемые в
      MOV   R5, -(SP) "отладчике"
      MOV   'A=', R0  Сообщение "АДРЕС"
      EMT   16        выдать
      SWAB  R0        на
      EMT   16        экран
      MOV   10(SF), R3 Извлечь из стека
;                               FC в R3
      MOV   R3, R4    Печать адреса
      CALL  @#16322@  команды
      MOV   'K=', R0  Сообщение "КОД"
      EMT   16        выдать
      SWAB  R0        на
      EMT   16        экран
      MOV   @R3, R4   Печать кода
      CALL  @#16322@  команды
      MOV   'R=', R0  Сообщение "РЕГИСТР"
      EMT   16        выдать
      SWAB  R0        на
      EMT   16        экран
      MOV   R1, R4    Печать
      CALL  @#16322@  содержимого R1
      MOV   #12, R0   Перевод
      EMT   16        строки
@:    BIT   #10@, @#177716  Клавиша
;                               нажата?
;
;                               BNE   @        Нет — ждать, иначе
;                               продолжай
      MOV   (SP)+, R5  Восстановить
      MOV   (SP)+, R4  регистры
      MOV   (SP)+, R3  из
      MOV   (SP)+, R0  стека

```

```

; Запись Т-разряда, выход из прерывания
BPT:  BIS   #20, 2(SF) Занести Т-разряд
;                               в ССП в стеке
;
;                               MOV   #TRS, @#14  Записать рабочий
;                               вектор 14
;                               RTT           Выход из
;                               прерывания
;
;                               END

```

Разберемся с некоторыми деталями. Сначала мы пишем в качестве вектора 14 адрес метки BPT и даем прерывание BPT. Происходит переход к BPT, а там мы заносим в ССП в стеке (оно на 2 выше вершины, ведь мы в прерывании!) единицу в Т-разряд. Затем мы записываем новый вектор 14 (рабочий, т. е. тот, по которому будет передаваться управление при прерывании по Т-разряду), после чего следует команда возврата из прерывания, но не RTI, а RTT. Она отличается от RTI незначительно (ради экономии места не будем детализировать чем) и применяется для выхода из отладочного прерывания.

Теперь запускается наша «отлаживаемая» демонстрационная программа. Она зациклена и делает следующее: сначала в R1 пишется 1, затем R1 сдвигается влево. Когда 1 достигнет старшего разряда R1, возникает перенос, но бит переноса у нас каждый раз прибавляется к R1, единица снова попадет в младший разряд и опять начнет сдвигаться, и так до бесконечности.

После выполнения каждой команды происходит прерывание по Т-разряду (ведь наш отладчик включен!), и управление передается на метку TRS — «Трассировка».

Вот теперь в работе отладчик; трассировка уже выключена — ведь после прерывания переписалось новое ССП из адреса 16, второго слова вектора. Мы не знаем, что там было, но в данном случае это неважно, Т-разряд там равен нулю, будьте уверены! И отлично, иначе отладчик не смог бы работать, а прерывал бы сам себя после каждой команды.

В отладчике мы будем использовать часть РОН, а чтобы потом не возникло неприятностей, сохраним их пока в стеке (с данной демонстрационной программой это не нужно, но на будущее пригодится — вдруг там окажется нужная информация).

Сначала будем выдавать адрес команды. Даем сообщение «А=» — это наш сервис, а потом... потом вспомним, что у нас сейчас хранится в стеке. По порядку туда было записано: ССП, РС, R0, R3, R4, R5. РС нам и надо, именно там адрес — какой? Следующей команды. Что ж, для простоты давайте индцировать как раз следующую команду; это даже удобно — мы сначала посмотрим, что «там, за поворотом», а потом уже разрешим ей работать.

Итак, РС в стеке за 8 байтов от вершины (после него мы записали в стек еще 4 регистра). Применим косвенную индексацию: 10(SP). Мы «достали» РС. Он нам еще пригодится, поэтому мы не пожалели для него отдельного регистра R3. А как выдать его на экран? В МСД есть подпрограмма, которая выдает на экран содержимое регистра R4. Чтобы не писать нечто подобное заново, просто обратимся к ней, ее адрес 163220. Она не сохраняет регистры R0, R4 и R5, вот почему мы их сохранили в стеке. Итак, число (РС) сначала в R4, потом на экран. Готово!

Дальше — сервис: «К=», это код команды. Вот теперь нам пригодится R3, ведь там адрес, а по этому адресу — сама команда. Словом, косвенное обращение через регистр. В R4 его и на экран! Ну а теперь содержимое R1, которое мы хотим посмотреть, — туда же, но сначала опять сервис: «Р=». Выведена строка параметров: адрес, код команды, содержимое R1. Теперь переведем строку.

Что мы еще хотели? Если клавиша нажата — программа работает, если нет — ждет. Регистр 177716 позволяет решить задачу. Вот мы нажали любую клавишу, программа пошла дальше: восстановлены регистры, и все сначала — заносим Т-разряд (впрочем, вектор 14 мы могли бы не заносить снова) и командуем возврат из прерывания. Вперед, нас ждет следующая команда!

Не слишком сложно, не правда ли? А ведь эта программа имеет практически все основные черты настоящего отладчика. Конечно, есть масса деталей, которые мы не учли. Например, если команда отлаживаемой программы состоит из двух или трех слов, то будет выдано только ее первое слово. А если в отлаживаемой программе встретится ЕМТ 14, то переписутся все векторы, в том числе 14-й, и прощай, отладочный режим! Не все, что надо, мы видим на экране, нет точки останова, нет дизассемблера, наш отладчик неперемещаемый, и... Но чего же вы хотели? Едва освоив ассемблер, сделать настоящий отладчик? Главное, что требовалось, — понять, как он работает, и эта цель достигнута.

Следующий по порядку вектор прерывания — 20. Прерывание по этому вектору — командное, возникает по команде ИОТ и аналогично командным прерываниям ВРТ, ЕМТ и TRAP, только в отличие от двух последних не имеет аргументов. Может быть использовано для выполнения каких-нибудь действий. Допустим, нужно подать в определенный момент звуковой сигнал. В начале программы запишем MOV #ИОТ, @ #20 — это мы переписали вектор 20. Напишем программу обработки прерывания:

ИОТ:	MOV	#7, R0	Код звукового
;			сигнала
;	MOV	#100, R1	100 повторов (64д)
@:	EMT	16	Выдать звук
	SDB	R1, @	Цикл по R1
	RTI		Выход из прерывания

Теперь в любой момент можно вызвать звук одной командой ИОТ. Вообще же команда ИОТ предназначена для вызова процедур и обработки ошибок в операционных системах.

Вектор 24 для нас практически бесполезен, так как на БК-0010 он не имеет аппаратной поддержки. Но можно убедиться, что программа его обработки в ПЗУ имеется. Дайте в МСД команду 160714G, и получите ответ «СВОЙ ПИТАНИЯ».

Вектор 30 разобран в заводском руководстве подробно. Здесь же мы рассмотрим использование его изменений. Допустим, мы написали программу-отладчик и нам требуется защитить ее от действия команды ЕМТ 14 программ пользователя. Для этого требуется «перехватить» код команды ЕМТ 14. Проще всего это сделать, переписав вектор ЕМТ. Попробуем.

MOV	#EMT,	@#30	Записать
;			вектор 30
;		Дальнейший текст
;		программы


```

; Перехват и исполнение команды EMT 14
EMT: MOV R5, -(SP) Сохранить R5
      MOV 2(SP), R5 Извлечь адрес
;
;
; CMP #104014, -(R5) Предыдущая
; команда -
; EMT 14?
;
; BNE 0 Нет - выход
EMT 14 Иначе выполнить
EMT 14
;
; MOV #EMT, @#30 Восстановить
; вектор 30
;
; ..... Выполнить другие
; ..... необходимые
; ..... действия
0: MOV (SP)+, R5 Восстановить R5
; JMP @#100112 Передать управле-
; ние EMT-диспетчеру

```

Наверное, после разбора предыдущих примеров все в этой программе понятно. Скажем только, что это за «другие необходимые действия». Возможно, кроме вектора 30 мы изменяли и другие векторы прерывания; тогда после выполнения EMT 14 их тоже надо восстановить.

Что касается вектора 34, мы на нем останавливаться не будем. Если вас интересует построение TRAP-диспетчера, то в ПЗУ БК-0010 есть пример — EMT-диспетчер, расположенный по адресу 100112. Можете с ним ознакомиться и по аналогии написать такой же (или другой) TRAP-диспетчер.

Теперь кратко рассмотрим, что можно сделать, изменяя векторы прерываний от клавиатуры. Например, многим не нравился издаваемый БК-0010 при нажатии клавиш звуковой сигнал. Можно, изменив вектор 60 (и 274), перехватить прерывание от клавиатуры, выполнить некоторые действия и передать управление драйверу клавиатуры, но не в начало его, а после подпрограммы выдачи звукового сигнала; тогда звука при нажатии клавиш не будет. Такая программа для работы в Бейсике публиковалась в журнале (ИНФО. 1989. № 4. С. 83), желающие могут с ней ознакомиться и, если нужно, перевести на язык ассемблера. Можно также, переписав вектор 274, ввести по нижнему регистру, например, спецалфавит или просто картинку, выводимые при нажатии клавиш. Это не очень сложно, но программы получаются довольно объемные, а кроме того, приходится создавать в каком-либо графическом редакторе специальные символы или картинки, поэтому подробно останавливаться на этом не будем.

Можно также ввести по нижнему регистру, например, выдачу всех операторов Бейсика или Фокала, но это требует подробного изучения структур интерпретаторов этих языков, и на этом мы тоже не останавливаемся.

Во многих случаях достаточно подать идею, а дальше — дело за читателем; приводимые же во множестве примеров часто только сковывают фантазию, направляя мысль по проторенному пути.

Рассмотрим последний из оставшихся векторов прерывания — вектор 100. Это, как уже было сказано, прерывание по таймеру. Таймер должен быть *внешним* (подключаемое к БК устройство, выдающее импульсы строго определенной частоты, например 10 импульсов в секунду). Импульсы, подаваемые на вход таймера (контакт В1 порта ввода-вывода), должны иметь уровень ТТЛ-логики (0, или *низкий уровень*, — от 0 до 0,4 В; 1, или *высокий уровень*, — от 2,4 до 5 В). При прохождении на вход таймера *отрицательного фронта* импульса (т. е. при смене «единицы» «нулем») возникает прерывание по вектору 100 (если, конечно, приоритет ЦП разрешает прерывание). Как можно использовать этот вектор? Во-первых, по прямому назначению, т. е. как таймер. Во-вторых, это дополнительный канал связи БК с внешним миром. Если, допустим, у вас есть магнитофон, в котором имеется счетчик (или просто датчик автостопа), выдающий импульсы при вращении кассеты, можно подать эти импульсы (конечно, преобразовав их к нужному уровню!) на вход таймера и, считая их, получить программный счетчик ленты. Правда, тут есть одна неприятность — при записи на магнитофон или чтении прерывания должны быть запрещены, иначе произойдет искажение информации. Но можно сделать счетчик, работающий при перемотке, и таким образом автоматизировать хотя бы поиск файлов (разумеется, для этого магнитофон должен иметь полное дистанционное управление, и вообще это не так просто). А можно подключить ко входу таймера, например, *аналого-цифровой преобразователь* (АЦП) типа ПНЧ (преобразователь напряжения в частоту) и, пользуясь системным таймером для отсчета времени, измерять число поступивших импульсов в единицу времени, а затем вычислять измеренное напряжение (или иной параметр). Или можно сделать так, чтобы при поступлении на вход таймера сигнала ЭВМ считывала информацию с внешних устройств (например, микрокалькулятора или цифрового измерительного прибора) или, наоборот, выдавала информацию на порт ввода-вывода или на ТЛГ-канал. Приведем простейший пример обработки прерывания по вектору 100.

```

; Инициализация программы
      CLR R1 Очистить счетчик
      MOV #TIM, @#100 Записать вектор
;
;
; HALT Конец

```



```

;Счет импульсов с индикацией
TIM: INC R1 Счетчик+1
MOV R1,R4 Выдать текущие
CALL @#163220 показания на экран
RTI Возврат из
; прерывания
END

```

Разумеется, программа обработки прерывания может быть и другой. Максимальная частота подаваемых на вход таймера импульсов ограничена временем, затрачиваемым после каждого импульса на обработку прерывания, и может достигать при простейших программах обработки 20—30 кГц; надо только учитывать, что при этом скорость обработки ЭВМ текущей задачи резко снижается, ибо почти все время уходит на обработку прерываний.

Ассемблер БК

Трансляция и компоновка

72 В свое время мы уже кратко познакомились с тем, как подготовить программу к работе (оттранслировать и скомпоновать ее) и обещали еще раз вернуться к этой теме. Мы так и сделаем, напомнив, что наше описание базируется на ассемблер-системе МИКРО.10К и все приводимые далее адреса и директивы относятся прежде к ней, а для прочих версий могут отличаться от указанных, хотя принципы те же самые.

Итак, имеется текст программы на языке ассемблера. В его состав могут входить:

- операторы прямого присваивания;
- метки;
- операторы;
- операнды с различными способами адресации;
- псевдооператоры с наборами символов;
- комментарии.

Весь этот чрезвычайно разнородный материал (размещенный в памяти по адресам 17001—37000) ассемблер должен превратить в программу в машинных кодах, готовую к исполнению — *загрузочный модуль*, «собрать» программу. Отсюда и название системы: «ассемблер» — значит «собирает».

Обычно ассемблер выполняет трансляцию за два *прохода* (или просмотра) программы. Первый проход называется *трансляцией* и может состоять из ряда операций, но проще рассматривать его как одно целое.

Запуск транслятора осуществляется из монитора по директиве СО. В процессе трансляции производится перевод текста программы в машинные коды и присваивание меткам определенных адресов. По мере перевода текста в машинный код производится запись полученных кодов в память ЭВМ (начиная с адреса 13000), и встречаю-

щиеся по ходу трансляции метки располагаются уже не просто в определенных строках программы, а по определенным физическим адресам. Эти адреса (вместе с именами меток в коде RADIX-50) заносятся в *таблицу меток*, формируемую с адреса 40000 «вниз», т. е. в сторону меньших адресов. Если меток слишком много, таблица может стать очень длинной и затереть при трансляции конец текста программы раньше, чем он будет прочитан транслятором. Естественно, при этом возникает ошибка, и трансляция прекращается. Избежать чрезмерного роста таблицы можно путем широкого использования локальных меток, так как таблица локальных меток формируется отдельно и существует в памяти только до появления очередной обычной метки.

Если все-таки таблица меток затирает текст (его на всякий случай перед трансляцией нужно записывать на МЛ), то такую программу оттранслировать невозможно; ее необходимо либо разделить на части, либо перейти в режим РП (ассемблер, за исключением редактора, рассчитан на работу в этом режиме; есть также специальная версия МИКРО.10К-РП, у которой в режиме РП работает и редактор). В режиме РП можно транслировать также тексты программ, составленные из нескольких частей, сливая их друг с другом путем «догрузки» директивой LF. Естественно, что при этом оператор END должен стоять только в конце последнего блока, а общая длина текста может достигать 50000. Во многих случаях такое слияние текстов программ значительно удобнее компоновки через объектные модули, о которой речь пойдет дальше. Чтобы ассемблер нормально работал в режиме РП, его необходимо либо запустить в этом режиме с адреса 1000, либо, перейдя в режим РП в мониторе, дать команду RS. Текст программы в памяти при этом, конечно, будет уничтожен. В режиме РП таблица меток формируется с адреса 70000*.

В результате трансляции (первого прохода) образуется так называемый объектный модуль. Он представляет собой программу в машинных кодах, в которую занесено все необходимое, кроме некоторых операндов, под которые ячейки памяти только *зарезервированы*. Дело в том, что при первом проходе транслятор еще «не знает» реальных физических адресов меток, ведь таблица только формируется, поэтому и не может записать операнды, обращение к ко-

* Самые последние версии МИКРО.10К-РП допускают переход в режим РП в мониторе ассемблера (нажатием клавиш НР/СБР) и выход в редактор (или трансляцию) с сохранением текста.

торым требует знания этих адресов. Операнды, ячейки под которые зарезервированы (туда заносятся нули), транслятор отмечает в таблице меток (выдаваемой на экран по окончании трансляции) инверсией с указанием адресов зарезервированных ячеек и имен меток, входящих в состав операнда. Наличие в таблице после трансляции инверсных меток свидетельствует, что программа перемещаемая и нуждается в компоновке — это второй проход транслятора.

Объектный модуль, полученный при трансляции, может быть после нее записан на МЛ директивой SL, при этом вместе с ним записывается и таблица меток. Ассемблер устроен так, что могут быть загружены с МЛ и скомпонованы вместе несколько объектных модулей, что позволяет получить (в режиме РП, разумеется) загрузочный модуль длиной до 16К байт. Но пока рассмотрим случай, когда у нас один модуль.

Итак, объектный модуль — в памяти, таблица меток — на экране (если таблица меток занимает более 24 строк в обычном режиме или 4 строк в режиме РП, ее выдача на экран производится по частям, с остановками до нажатия любой клавиши). Раз в таблице есть инверсные метки, значит, программа нуждается в компоновке. При компоновке не только производится вычисление и запись в загрузочный модуль недостающих операндов, но и *задание адреса загрузки*, т. е. того адреса, по которому программа будет загружаться и работать в дальнейшем. Адрес этот может быть задан любым, ассемблер соответственно внесет поправки во все адреса меток программы, как бы переместив ее в памяти (но в действительности загрузочный модуль остается на месте, по адресу 13000), поэтому наш ассемблер называется *перемещающим*.

Директива LL задает компоновку по умолчанию — по последнему указанному ранее адресу (после запуска ассемблера это адрес 1000), а директива LS запрашивает новый адрес компоновки. После окончания компоновки инверсных меток в таблице (а она выдается на экран заново) не должно остаться, а если они все же есть, — значит, в программе имеются ошибки. Это либо неопределенные метки (т. е. обращение к метке есть, а самой ее нет), либо обращение к локальной метке «через обычную» (в скобках заметим, что такое обращение «вперед» допустимо, но лучше его избегать, это только запутывает программу).

После того как получен загрузочный модуль (в таблице меток нет инверсных имен), его можно записать на МЛ директивой SA. При этом учтите, что располагается он с

адреса 13000 независимо от заданного адреса компоновки, и с этим адресом будет записан на МЛ. Для загрузки такой программы при работе нужно всегда задавать адрес загрузки, если он отличается от 13000, а так как удобнее грузить программу «по умолчанию» (АДРЕС=0), то лучше сразу переслать ее директивами MSD по нужному адресу, а потом уже записать на МЛ, для чего нужно выйти из ассемблера, нажав клавишу СТОП. Длина загрузочного модуля выдается при трансляции.

Можно также сразу запустить загрузочный модуль на исполнение директивой RU, при этом управление просто передается по адресу 13000. Учтите только, что для этого компоновка должна быть произведена именно по этому адресу, а ассемблер в памяти вполне может быть испорчен («затерт») работающей программой. Такой прием (прямой запуск) часто используется для целей отладки, иногда даже специально изменяют для этого некоторые параметры программы, чтобы она могла работать по адресу 13000 без затирания ассемблер-системы. Действительно, это очень удобно: запустив программу по RU и посмотрев на результаты ее работы, можно вернуться в ассемблер-систему по «холодному» входу (адрес запуска ассемблера 1002), войти в редактор, внести в текст изменения, снова оттранслировать и скомпоновать, вновь запустить и т. д., пока работа программы нас не удовлетворит. Затем можно внести в текст программы окончательные изменения для работы по нужному адресу и скомпоновать ее, задав этот адрес. Такая методика отладки особенно удобна, если длина загрузочного модуля невелика (до 4000) и он не затирает при трансляции начало текста программы; это не сложнее отладки программ на Бейсике или Фокале.

А как быть, если программа длиннее и ее текст не помещается в памяти ни в обычном режиме, ни в РП? Тогда ее нужно разделить на несколько частей, по возможности таких, чтобы их можно было отладить отдельно. После проверки каждой части ее транслируют и записывают объектные модули на МЛ. Каждая часть при этом должна содержать в начале необходимые операторы прямого присваивания (относящиеся к меткам, используемым в данной части), а в конце — оператор END.

Когда на МЛ есть все необходимые объектные модули программы, дают директиву LA, указывают адрес компоновки и загружают первый модуль. В дальнейшем дают команды LI и загружают остальные модули. Каждый вновь загруженный модуль компонуется с уже имеющимися в памяти.

После загрузки всех модулей (непрерывно в том порядке, в котором они должны входить в программу!) загрузочный модуль готов, его можно записать на МЛ или запустить. Перед компоновкой модулей с МЛ нужно перезапустить ассемблер командой RS.

Есть при работе на ассемблере кое-какие тонкости, которые полезно знать.

Синтаксические ошибки в тексте программы выявляет обычно транслятор. При этом трансляция прекращается, на экран выдается строка с ошибкой и запрос С/Е?, т. е. «продолжать или перейти к месту ошибки?». Команда С вызывает продолжение трансляции безотносительно к тому, что строка с ошибкой оттранслирована не будет. Это иногда нужно, чтобы выявить остальные ошибки или для других целей. В конце такой трансляции таблица меток не выдается. Команда E вызывает переход в редактор, при этом курсор указывает место ошибки. Если это «ошибка 3» (длины перехода), то указывается не метка, при переходе к которой допущено превышение длины перехода в операторах BR или SOB, а сам оператор, т. е. место обращения.

В последних версиях МИКРО.10К введена «ошибка 15» — затирание текста программы при трансляции, но ошибка эта только констатирует факт, а не защищает от затирания текст, так как выдается после трансляции последнего оператора, когда часть текста уже затерта. Но если это был обычный оператор, затертой окажется небольшая часть текста, которую легко восстановить. Если же последним был, например, оператор .+10000, то будет утерян текст длиной 10000, несмотря на останов по ошибке. Обычная реакция на эту ошибку — команда С, при этом трансляция идет нормально до конца**.

Не совсем приятная ошибка — обращение к неопределенной локальной метке. При этом в таблице меток происходит путаница — появляются инверсные метки без имени, а только с адресом или метки с именами, которых нет в тексте. Но зная эти характерные признаки данной ошибки, найти ее несложно.

Хуже обстоит дело с логическими ошибками, когда программа написана формально правильно, но работает не так, как ожидается. В этом случае, если простой просмотр текста и проверка алгоритма ошибок не выявляют, следует разбить программу на функциональные блоки, проверяемые отдельно. Частая ошибка — неправильное ветвление программы. Она может быть вызвана либо

непониманием логики программы, либо неправильным оператором (например, применение оператора «со знаком», когда надо «без знака»), либо ошибками при сравнении операндов. На последнем случае стоит остановиться подробнее.

Когда ветвление задается после оператора сравнения, то оно может происходить неправильно по следующим причинам:

неправильный порядок записи операндов (нужно помнить, что сравнение производится вычитанием второго операнда из первого);

сравнение слов, когда надо — байтов, и наоборот;

искажение одного из сравниваемых операндов за счет распространения знака после выполнения оператора MOVБ;

использование оператора ветвления с иной, чем нужно, логикой работы, например: BLO вместо BLOS, BGT вместо BGE, BEC вместо BNE и т. п.

Возможны, конечно, и другие причины, но эти — самые частые. Не менее частая причина ошибок ветвления — передача управления «не туда», не просто не на ту метку, а именно не на то место программы — не там стоит сама метка, например: в цикле при этом все время повторно заносится исходное значение переменной, или все время обнуляются регистры, или повторно задается исходный адрес и т. д.

Уточнить, как именно происходит ветвление, где программа «зацикливается» и т. п., можно разными приемами. Всего нагляднее и проще — включать в текст программы по направлениям ветвления операторы, действие которых заметно — HALT, звуковые сигналы, выдачу на экран символов и прочее. Можно временно исключить из текста отдельные строки, «закрыв» строку знаком комментария — поставив в начале ее символ «;», — или вписав в начале строки оператор NOP; можно заменять операторы условного ветвления на безусловные и т. п.

Можно использовать отладчик, но сам автор не очень к этому склонен, а потому не настаивает на этой рекомендации; лучшим «отладчиком» является логика и опыт программиста. Все же иногда отладчик может помочь в трудном положении. При пошаговой работе программы можно наблюдать содержимое регистров и т. п. Особенно интересен ОТЛАДЧИК.К С. А. Кумандина — он позволяет, кроме регистров, задать индикацию еще до пяти произвольных ячеек памяти, имеет RADIX-дамп и другие сервисные функции.

Работе с отладчиками нередко мешает то, что командные прерывания EMT и TRAP они обрабатывают как одну команду; это

** В версии МИКРО. 10К-РП «ошибка 15» исключена.

происходит потому, что при прерывании заменяется ССП и сбрасывается Т-разряд. Чтобы посмотреть, что делается «внутри» прерывания, можно записать предварительное вторым словом вектора данного прерывания ССП с установленным Т-разрядом. Но вообще возможности отладчиков довольно ограничены, они более подходят для того, чтобы «копаться» в чужих программах, чем для отладки собственных.

Позиционно-независимое программирование

Обычно программа, написанная на языке ассемблера, предназначена для работы в определенной зоне адресов. Но есть программы, которые просто обязаны работать по любым адресам, — это отладчики, дизассемблеры, программы выдачи дампов и прочие «инструменты», которые нужны не сами по себе, а для работы с другими программами. При этом мы заранее не знаем, где будет расположена программа, которая нуждается в отладке или дизассемблировании, и поэтому вынуждены размещать инструментальную программу в свободном месте. Как добиться перемещаемости программ? Для этого при программировании на ассемблере следует придерживаться ряда принципов, носящих название принципов *позиционно-независимого программирования*:

все адреса перехода и обращений к тексту программы задавать только относительно, в виде меток;

все буферы данных внутри программы

выделять тоже по меткам, т. е. делать их перемещаемыми;

ко всем ячейкам ПЗУ и системной области обращаться только по абсолютным адресам или по меткам, адреса которых заданы оператором прямого присваивания;

не выделять в ОЗУ (кроме системной области) буферы и ячейки по абсолютным адресам, так как при размещении программы в этой зоне она может быть затерта собственным буфером;

если необходимо определение физического адреса обращения к ОЗУ (или самой программе), например при задании векторов прерываний, адресов текстовых сообщений и т. п., то вычислять эти адреса, пользуясь псевдооператором $\cdot @$;

широко использовать регистры общего назначения, стек и косвенную адресацию через регистры и метки — эта адресация не связана с физическими адресами программы.

Немного подумав, вы, возможно, сможете сами предложить еще ряд таких же принципов.

Признаком *перемещаемости* программы служит то, что она не нуждается в компоновке, т. е. таблица меток, выданная транслятором, не содержит инверсных имен. Перемещаемые программы занимают обычно заметно больше места в ОЗУ, чем непере-мещаемые, и несколько сложнее по структуре, поэтому стремление делать перемещаемыми все программы вряд ли оправданно.

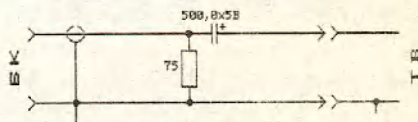
75

Улучшение сопряжения БК и ТВ

Владельцы БК обычно используют схемы подключения к телевизору, рекомендуемые в руководстве по эксплуатации компьютера или опубликованные в периодике (например, в «Науке и жизни»). Эти традиционные способы позволяют неплохо работать с телевизором как с монитором, но могут значительно ухудшать качество приема телепередач. Это вызвано тем, что низкочастотный тракт телевизора оснащается дополнительными, иногда довольно длинными, проводами, которые играют роль «низкочастотной антенны», принимая на себя и передавая в тракт телевизора низкочастотные помехи гармонического и импульсного характера.

Рекомендую подключать БК к ТВ по стандартной схеме, но с увеличением емкости

разделительного конденсатора до 500 мкФ. Соединение выполняется тонким коаксиальным кабелем или экранированным проводом, сигнал подается по центральной жиле кабеля.



Важно, чтобы конденсатор находился как можно ближе к монтажу телевизора; тогда он выполняет роль фильтра, препятствующего проникновению помех в низкочастотный тракт. Большая емкость разделительного конденсатора позволяет получить на

телеэкрane устойчивое, четкое изображение без характерных черточек и точек, образующих помехами от самого компьютера и от его блока питания. Полтора десятка малогабаритных телевизоров «Юность», «Электроника», «Шилялис», особенно чувствительных к такого рода помехам, доработанных

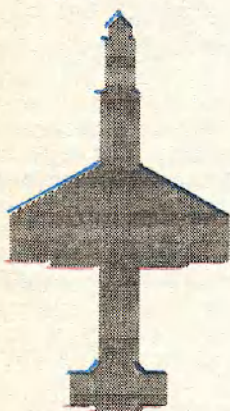
мною по этой схеме, радуют своих владельцев универсальностью: устойчиво работают в качестве монитора, прекрасно принимают телепередачи; отпадает нужда в переключателе «монитор—телевизор».

П. МИХАЙЛОВ

Упрощение организации движения изображений в Бейсике

Обычно движение изображений на экране программируют следующим образом: рисуют начальное изображение, стирают и рисуют его же, но с некоторым смещением.

— Дорисовать
— Стереть



Предлагаемый способ отличается тем, что дорисовывается лишь передний край рисунка (в направлении движения), а задний

```

10 FOR J%=60 TO 170%
15 LINE (90%, 221%-J%) - (107%, 215%-J%)
20 LINE - (108%, 203%-J%)
25 LINE - (110%, 200%-J%)
30 LINE - (112%, 203%-J%)
35 LINE - (113%, 215%-J%)
40 LINE - (130%, 221%-J%)
45 LINE (103%, 236%-J%) - (108%, 235%-J%)
50 LINE - (112%, 235%-J%)
55 LINE - (117%, 236%-J%)
60 LINE (90%, 225%-J%) - (106%, 226%-J%), 4
65 LINE (114%, 226%-J%) - (130%, 225%-J%), 4
70 LINE (103%, 240%-J%) - (110%, 241%-J%), 4
75 LINE - (117%, 240%-J%), 4
80 NEXT J%

```

стирается. Если же координаты в операторах графики сделать больше (меньше) максимальных (минимальных) координат экрана, то рисовать «статическое» изображение не нужно; создается эффект постепенного «выплывания» рисунка. В приводимой программе применены операторы LINE, имеющие меньшее быстродействие по сравнению с DRAW, но это позволяет реализовать выполнение программы как на БК-0010, так и на БК-0011 (в БК-0011 не реализован оператор DRAW!).

М. САДОВНИКОВ, О. ПЕТРУК

БК-0010.01 в роли осциллографа

В радиолюбительской практике при наладке цифровых устройств довольно часто возникает необходимость в одновременном наблюдении сигналов в нескольких точках схемы. Использование традиционных приборов в таких случаях малоэффективно, а двухлучевые осциллографы и логические анализаторы доступны лишь малому числу радиолюбителей. Между тем небольшая программа

превращает БК-0010.01 в 16-канальный цифровой низкочастотный запоминающий осциллограф.

Программа позволяет наблюдать на экране видеомонитора одновременно 16 каналов. Максимальная частота наблюдаемых сигналов 20 кГц. Ограничение определяется скоростью выполнения пары команд


```
1M: MOV (R4), (R1)+
   SDB (R5), 1M
```

Частоту наблюдаемых сигналов можно повысить примерно в два раза, если воспользоваться предложением по увеличению тактовой частоты микропроцессора (ИНФО. 1990. № 2).

Минимальная частота практически не ограничена. Задав период опроса, равный 100000, можно наблюдать сигналы с частотой порядка 0,005 Гц.

Значительно повысить верхний диапазон частоты исследуемых сигналов можно, обеспечив заполнение буфера памяти аппаратным образом, но он сложен. В большинстве случаев программного способа достаточно: практически всегда можно снизить тактовую частоту генератора отлаживаемого устройства и произвести отладку на пониженной частоте. Для цифровых схем такой способ позволяет определить большинство неисправностей.

Текст программы OSC1.TXT в системе МИКРО.11К

```
MOV #14,R0 ;ОЧИСТКА ЭКРАНА
EMT 16
MOV #232,R0 ;ОТКЛЮЧЕНИЕ
EMT 16 ;ИНДИКАЦИИ КУРСОРА
MOV #A2,R1 ;ВЫВОД СООБЩЕНИЯ
MOV #20000,R2 ;"ВВЕДИТЕ ПЕРИОД"
EMT 20
JSR PC,@#100472 ;ВВОД ЗНАЧЕНИЯ
MOV R5,@#230 ;ПЕРИОДА
N: MOV #177714,R4
   MOV #2000,R3
   MOV #2000,R1
   CMP #0,@#230
   BEQ 1M
   BR 2M
1M: MOV (R4), (R1)+ ;ЗАПОЛНЕНИЕ БУФЕРА
   SDB R3,1M
   BR BGN
2M: MOV @#230,R5
1S: SDB R5,1S
   MOV (R4), (R1)+
   SDB R3,2M
BGN: MOV #14,R0
   EMT 16
   MOV #0,R1 ;ВЫВОД НА ЭКРАН
   MOV #2,R2 ;ОЦИФРОВКИ
   EMT 24
   MOV #A1,R1
   MOV #20000,R2
   EMT 20
   MOV #1,R0
   MOV #20,R1
```

```
MOV #34,R2
MOV #2000,R3
7M: MOV #0,R4
   MOV (R3)+,R5 ;ЧТЕНИЕ БУФЕРА
   COM R5
5M: JSR PC,EMT
   INC R4
   CMP #20,R4
   BEQ 4M
   ADD #12,R2 ;УВЕЛИЧЕНИЕ БАЗЫ Y
   BR 5M
4M: MOV #34,R2
   INC R1 ;УВЕЛИЧЕНИЕ КООР-
   CMP #1000,R1 ;ДИНАТЫ X
   BNE 7M
   BR N
EMT: EMT 30 ;ПОДПРОГРАММА
   MOV R2,-(SP) ;ВЫВОДА БИТА
   ROR R5
   BCS 1E
   SUB #1,R2
   BR 2E
1E: SUB #6,R2
2E: EMT 30
   MOV (SP)+,R2
   RTS PC
```

```
A1: .A: 00^01^02^03^04^05^06^07^08^09^10^11^12^
      13^14^15
```

```
.E
A2: .A: ВВЕДИТЕ-ПЕРИОД:0...177777<ВВОД>^
.E
END
```

Входной порт БК-0010.01	Контрольные точки исследуемой схемы
BB00 B24	Канал 1
BB01 A24	Канал 2
BB02 B23	Канал 3
BB03 B17	Канал 4
BB04 B20	Канал 5
BB05 A20	Канал 6
BB06 B22	Канал 7
BB07 A23	Канал 8
BB08 B31	Канал 9
BB09 A31	Канал 10
BB10 B32	Канал 11
BB11 A32	Канал 12
BB12 B30	Канал 13
BB13 A29	Канал 14
BB14 B29	Канал 15
BB15 A30	Канал 16
Общий B19	К общей точке
Общий A18	исследуемой схемы

А. БАРСУКОВ

Поправка. В статье А. Барсукова «Подключение принтеров «Электроника МС-6312» и «Электроника МС-6313» к БК» («ИНФО». 1991. № 3. С. 55) в табл. 2 пропущена строка «DATA6 15 B6 ВД06».

Калейдоскоп БК

Уже хорошо известный читателям А. Ланев предлагает очередные программы. Первая позволяет распознавать состояние клавиатуры: 0 — клавиша нажата, 1 — все клавиши отжаты.

```
10 PRINT AT(10,10)MID$(BIN$(PEEK
      ($0177716)),10,1)
20 GOTO 10
```

Вторая превращает БК в нечто, напоминающее цветомузыкальную установку.

```
10 CLS
20 T=3
30 A$=MID$(BIN$(PEEK($0177716)),11,1)
40 IF A$="1" THEN GOTO 70 ELSE
      IF T>-60 THEN T=T-1
50 CIRCLE(122,122),T,4
60 GOTO 30
70 CIRCLE(122,122),T,INT(RND(1)*3)+1
80 IF T<10 THEN T=T+2 ELSE T=T-2
90 GOTO 30
```

Третья «приводит в норму» ключи после выхода в Бейсик из монитора.

```
DEF USR1=$0120606<ввод>
A%=USR1(A%)<ввод>
```

Четвертая отключает звуковой сигнал, выдающийся при нажатии на клавишу.

```
10 B0SUB 30
20 GOTO 10
30 K=0
40 IF PEEK(-80)=64 THEN RETURN
50 K=PEEK(-78)
60 ? CHR$(K)
70 POKE -80,64
80 RETURN
```

В. Ким (г. Самарканд) советует вместо отсутствующего в Бейсике БК оператора форматного вывода на экран PRINT USING использовать следующий прием:

```
10 K%=6%
20 A=12.34789
30 ? MID$(STR$(A),1%,K%)
```

В строке 10 задается число символов, выводимых на экран. Оно должно быть на 1 больше числа выводимых цифр, включая десятичную точку. В строке 20 задается произвольное число А. В строке 30 находится аналог PRINT USING, выводящий на экран в данном случае две цифры до и две цифры после десятичной точки: 12.34.

К. Толмазов (Москва) предлагает еще один способ отключения звукового сигнала,

сопровождающего нажатие клавиш БК. Его программа умещается в системной области.

```
10 DATA 12737,432,60,12737,470,274,
      207,4437,110340,10546,5737,
      224,1402,4737,110536,12737,
      17777,222,113700,177662,137,
      101150,4437,110340,10546,
      5737,224,1402,4737
15 DATA 110536,12737,17777,222,
      113700,177662,137,101374
20 FOR A%=$0414 TO $0524 ST 2%
30 READ C%
40 POKE A%,C%
50 NEXT A%
60 DEFUSR=$0414
70 R%=USR(R%)
```

Введя ее, следует дать команды RUN и NEW. Если же вы работаете с Фокалом, кодовую часть программы можно загрузить из режима ТС с адреса 414₈, заменив команду 207₈ на 0, затем запустить ее и далее запустить Фокал с адреса 120000₈.

С. А. Паничев (Мурманская обл.) предлагает усовершенствовать программу КОПИР-М В. Ярошенко. Приводимая таблица констант скоростей должна быть вставлена в подпрограммы, соответствующие адресам в таблице. Ее использование позволяет увеличить надежность хранения файлов.

Адрес	Данные
3400	10 10 50 50 10 10 20 20 10 10
3502	20 20 120 120 20 20 40 40 20 20
3604	40 40 240 240 40 40 100 100 40 40

П. Дейчко (Вильнюс) предлагает способ просмотра в системе МИКРО.10К текстовых файлов в формате EDASP. Для этого следует директивой LO загрузить нужный файл в ассемблер, выйти в редактор командой EN и нажать клавишу ←. Если удерживать ее нажатой, текст файла будет проплывать по экрану «бегущей строкой».

Оригинальную находку сделал А. Дарменко (Электрогорск). Он обнаружил в Фокале... лишний оператор. Оказывается, если вместо FOR поставить SET, работа Фокал-программы не меняется.

Книга-почтой для пользователей «Электроника БК-0010».

Адрес: 105187, Москва, Кирпичная ул., 39—41, ПТО «Магистр-2», ИТС

Н. ЛЕВШИН

НИИ педагогики УССР

Р. РИЖНЯК

Кировоградский пединститут

«Математический задачник» для V—VI классов

Проблема использования НИТ в школьном обучении, занимая достойное место в педагогической науке, страдает отсутствием практических исследований по самым разнообразным параметрам, относящимся к изучению разных предметов, и в частности к изучению математики в V—VI классах.

В этой статье мы хотим поделиться опытом построения педагогических программных средств (ППС) для обучения учащихся названных классов решению математических задач. Наш подход основывается на формировании у учеников обобщенных умений путем моделирования задачной ситуации на персональном компьютере (ПК), изучении основных особенностей структуры задачи и использовании в процессе решения задачи знаковых моделей различного уровня абстрактности: вербальной модели — текста задачи, наглядно-схематической — рисунка к ней, структурной — схематического изображения основных соотношений между характеризующими задачную ситуацию величинами — и решающей — алгоритма выполнения условий задачи. Эти положения явились исходными для создания и использования пакета прикладных программ «Математический задачник», состоящего из двух частей:

1. *Программные задачи.* Шесть программ представляют наиболее общие типы задач на движение, каждая из которых имеет особый характер взаимосвязей между величинами, определяющими задачную ситуацию. Зависимостями, характеризующими движение, можно описать и другие виды ситуаций изменения, представляя их в виде соотношения $M = t \cdot n$, где M — измеряемая величина, t — мера измерения, n — количество мер измерения в измеряемой величине [1].

2. *Нестандартные задачи.* Двенадцать

программ, представляющие следующие типы: задачи на простые арифметические соотношения чисел; на различные соотношения между множествами; на соотношения комбинаций элементов из числа данных; на бинарные соотношения элементов; на логические соотношения высказываний. Выделение этих типов было произведено после анализа опыта работы ведущих учителей-методистов школ Киева и Кировограда по использованию нестандартных задач в обучении математике.

Все программы пакета написаны на языке MSX-Бейсик для КУВТ «Ямаха» и допускают ускоренную рассылку по сети. Апробация пакета производилась на уроках и занятиях кружка по математике в СШ № 112 Киева.

Раскроем основные особенности построения пакета «Математический задачник». Во-первых, наибольший эффект от применения ПК в обучении решению задач по математике достигается за счет использования моделирующих возможностей компьютера. Но, на наш взгляд, при этом нужно учитывать следующие условия.

Целесообразнее моделировать не отдельные объекты с целью ознакомления с ними, а процессы во всей взаимосвязи их компонент. Например, одна из программ пакета предусматривает решение задачи о мостах Эйлера. На компьютере моделируется задачная ситуация во всей взаимосвязи ее компонентов: мосты, возможность и невозможность переходов. Органически входят в нее и такие блоки, как подсказка для правильного выбора начала и конца перехода. Таким образом, ученик воспринимает ситуацию в целом, «входит» в нее, что является необходимым условием успешного решения задачи.

Моделирующие возможности персональных

компьютеров должны использоваться на этапе материализации умственных действий.

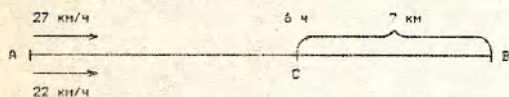
Отметим, что формирование сложного умения решать задачи мы понимаем как поэтапное формирование элементарных умственных действий [2], освоение которых обеспечивает правильное решение.

Задачная ситуация должна моделироваться в несколько этапов (в зависимости от степени абстрагирования).

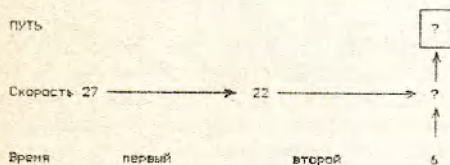
Для иллюстрации этого положения приведем пример программы, предусматривающей решение задачи на движение. Первый этап моделирования — изображение вербальной модели (текста задачи).

От одного причала в одном направлении отплыли два теплохода. Скорость первого теплохода — 27 км/ч, скорость второго — 22 км/ч. Каким будет расстояние между теплоходами через 6 ч?

Второй этап — построение наглядно-схематической модели.



Третий этап — построение графовой (структурной) модели задачи.



И последний этап — построение решающей модели.

$$(27 - 22) \times 6 = 30 \text{ (км)}$$

Как видим, с каждым последующим этапом степень абстрагирования модели повышается. Но при этом мы допускаем необходимость возвращения к рассматриваемым ранее моделям. Такое комбинирование на ПК моделей различной степени абстракции предоставляет широкие возможности для выработки у учащихся обобщенного способа решения математических задач.

Во-вторых, при обучении решению нестандартных задач с использованием ПК появляется совершенно новый важный блок — подсказка. Она выдается по требованию ученика после нескольких неудачных попыток ее решения. При такой методике повторное обращение к решаемой основной задаче становится более результативным. Например, рассмотрим программу поиска фальшивой (более легкой) из 243 одинаковых монет. Принцип решения задачи заключается

в последовательном разделении каждой «подозреваемой» кучки монет на 3 части. В качестве подсказок (почти во всех программах пакета они имеют 3-ступенчатую структуру) ученикам предлагается решить подобные задачи, но с меньшим количеством монет и взвешиваний. Соответственно — 27 монет и 3 взвешивания, 9 и 2, 3 и 1. Эксперимент показал, что большинство учащихся, испытывающих затруднения при решении основной задачи, уже после первой подсказки успешно справлялись с заданием. Мы использовали и другие виды подсказок: счетчик, при решении задач на вычисления результата производимых действий в некотором лабиринте, диалог-рекомендации, подсказывающие учащемуся путь к нахождению правильного алгоритма решения (как это было сделано при постановке на ПК задачи о трех рыцарях и их оруженосцах: в качестве первой подсказки использовалась задача об охотнике, волке, козе и капусте, неявно указывающая алгоритм решения основной задачи, вторая и третья подсказки имели вид рекомендаций, приближающихся к нахождению оптимального алгоритма решения задачи).

В-третьих, ученику следует обеспечить свободу выбора способа решения задачи, так как работа с задачей дает очень слабый обучающий эффект, если учащийся следует шагами намеченного алгоритма. Но возникает закономерный вопрос: какими средствами осуществлять коррекцию действия ученика? Их несколько: подсказка, возможность возврата к исходной точке решения, а также блоки защиты от нестандартных действий пользователя. Программа задачи, в которой требуется обойти шахматным конем всю доску 8×8 , устроена так, что не вмешивается в построение алгоритма решения, а лишь исправляет ошибки ученика.

В-четвертых, в упомянутых выше блоках защиты использованы «аварийные» сигналы. Отметим, что применение такого сигнала оптимально в том случае, если он, переключая мышление ученика на другую деятельность, сохраняет результат предшествующей. Это связано с особенностями «аварийного» сигнала, значимой характеристикой которого на первом этапе является его эффективность, а на втором — смысловая сторона раздражителя. В программе «Путешествие коня» это реализуется так: если ученик делает ход, не характерный для коня, повторно попадает на некоторую клетку шахматной доски или же ошибается в наборе координат, например набирает число, больше 8, то первый этап «аварийного» сигнала характеризуется звуком и миганием, а второй — выдачей соответствующего комментария:

- а) «Конь так не ходит!»;
- б) «Здесь конь уже был!»;
- в) «Проверь правильность набора координат!»

В-пятых, при определении числа сигналов, подаваемых ученику, следует предусмотреть их восприятие и объединение в образ. Экспериментально мы установили, что их максимальное число на одном кадре может быть не больше 4. При этом возникает вопрос: какова должна быть характеристика таких сигналов? Апробация программ показала их оптимальное сочетание: 3 зрительных и 1 слуховой сигналы. Причем слуховой сигнал дублирует только основной зрительный слуховыми, что способствует восприятию сигналов и объединению их в образ. Слуховые сигналы воспроизводились через наушники.

В программах пакета наиболее распространена следующая схема предъявления сигналов: инструкция о правиле вызова текста задачи и помощи (вверху), строка комментария (внизу), рисунок или схема решения (в центре) и в необходимых случаях звуковой сигнал.

Такой подход к использованию сигналов разных модальностей, с одной стороны, усиливает эффективность основного сигнала, а с другой — оказывает тонизирующее воздействие на ученика: чувствительность его основного анализатора повышается за счет раздражения других.

В-шестых, применение программ в учебном процессе привело к наиболее оптимальной, на наш взгляд, системе контроля за ходом решения задачи. Сразу оговоримся, что мы не предусматривали выставление

отметок ученикам непосредственно компьютером. Да и вообще, не рекомендовали учителям ставить оценки за работу с программой. Причина этого — сугубо обучающее назначение пакета «Математический задачник». В контролирующие функции программ входило следующее:

а) коррекция действия учеников в процессе решения задачи;

б) предоставление ученику информации о количестве его штрафных очков.

В зависимости от этого по завершении работы ПК «выдавал» дифференцированно-похвальный комментарий:

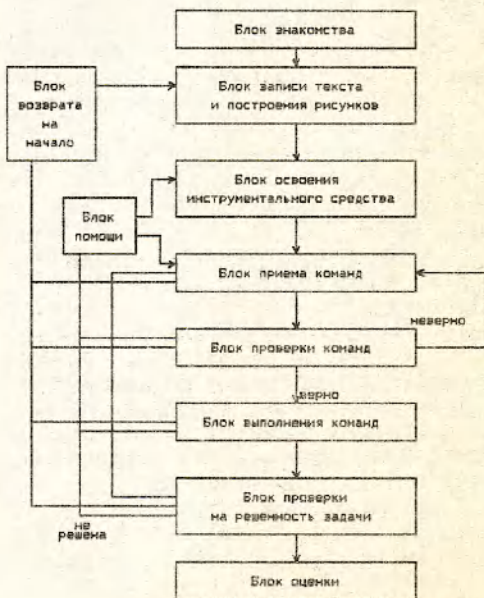
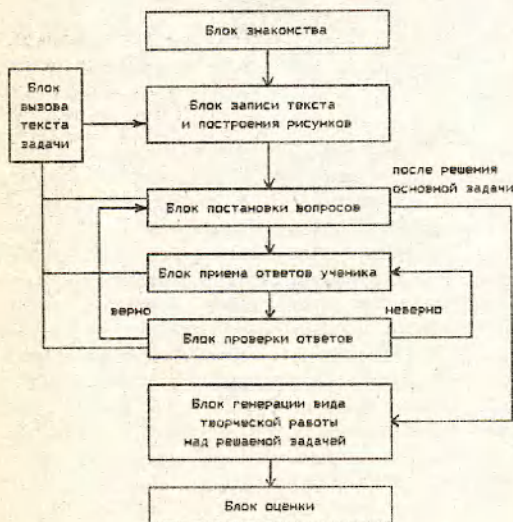
«Молодец! Спасибо тебе за отличную работу!»

«Хорошо! Хотя ты и допускал ошибки, но мне очень приятно было с тобой работать».

«Неплохо. У тебя было немало ошибок, но я знаю, что завтра ты будешь работать лучше».

При подсчете штрафных очков мы учитывали следующие показатели: синтаксические ошибки (A1), семантические ошибки (A2), количество вызовов текста задачи (A3), количество вызовов помощи (A4), рациональность решения, определяемую количеством ходов в решающем алгоритме (A5). Количество штрафных баллов определяется по формуле:

$$S = A1 + 2 \times A2 + A3 + 3 \times A4 + (A5 - X),$$



где X — оптимальное число шагов в алгоритме решения задачи.

И последнее. Нами была определена структура ППС для обучения решению программных и не стандартных математических задач.

Отмеченные выше особенности построения ППС для обучения решению математических задач в V—VI классах обеспечивает их эффективное использование в школе.

Адрес для справок: 252030, Киев, ул. Левшина, 10, НИИ педагогики УССР, Левшину Н. Н.

Н. АЛИПОВ, О. СЕРГБЕВА

2-й МОЛГМИ им. Н. И. Пирогова

Программы-репетиторы в обучении.

Несмотря на бурное развитие вычислительной техники и программного обеспечения, появление все более совершенных средств графических отображений, голосового ввода и т. п., эффективное использование компьютеров в обучении все еще остается проблемой как в СССР, так и за рубежом. Практика показывает, что самая совершенная обучающая программа очень часто уступает по педагогическому эффекту вдохновенному лектору, хорошо написанному учебнику или преподавателю, умеющему постоянно поддерживать живой контакт с аудиторией. Более того, объемные обучающие программы пригодны лишь в том случае, когда машинное время для учащегося не ограничено, т. е. когда он фактически имеет собственный персональный компьютер. Совершенно очевидно, что такая перспектива в СССР пока еще весьма отдаленна и, кроме того, она касается индивидуального, но не массового обучения.

Для эффективного применения компьютеров при массовом предметном обучении надо учитывать его специфику и отличия от индивидуального. Главное, на наш взгляд, состоит в том, что в массовом обучении основная роль в управлении обучением должна отводиться педагогу; компьютер же должен не заменять преподавателя, а освободить его от рутинной работы и помогать усовершенствовать учебный процесс.

Какие же функции преподавателя может взять на себя компьютер? Мы считаем (и в этом основное отличие применяемых нами программ от «классических» обучаю-

Литература

1. Гамезо М. В., Герасимова В. С. Знаковое моделирование в процессе решения учебных текстовых задач: Психологические проблемы переработки знаковой информации / Под ред. В. Ф. Венда, М. В. Гамезо и др. М.: Наука, 1977.
2. Исследование мышления в советской психологии / Под ред. П. Я. Гальперина. М.: Мысль, 1966.
3. Пономарев Я. А. Психология творчества и педагогика. М.: Педагогика, 1976.
4. Моныхов В. М. Что такое новая информационная технология? // Математика в школе. 1990. № 2. С. 47—52.
5. Ломов Б. Ф. Инженерная психология. Теория, методология, практическое применение. М.: Наука, 1977.

щих), что на компьютер не следует перекладывать функцию подачи информации по причинам, упомянутым в начале статьи. В связи с этим мы отказались от термина «обучающие программы», предпочитая ему «программы для обучения». В соответствии с теорией умственных действий [1], которая служит вполне оправдавшим себя рабочим инструментом, целью обучения служит умение оперировать полученной информацией, и, с другой стороны, это оперирование информацией при решении конкретных задач помогает более прочному и глубокому ее усвоению. Таким образом, решение обучаемым задач служит мощным методическим инструментарием. И именно здесь использование компьютеров весьма эффективно.

Общепринятая последовательность состоит в следующем. Учащемуся предлагается задача для самостоятельного решения. В случае если он не справляется, ему помогает преподаватель подсказками, наводящими вопросами и т. п. с целью максимально активизировать творческую мысль ученика.

Мы предположили, что помощь преподавателя учащемуся при решении задач — это достаточно рутинный и формализуемый процесс (в отличие от лекции, семинара и т. п.). Для этого необходимо было прежде всего определить, существует ли какая-либо универсальная схема решения, и если да — какие этапы она включает. Анализ работы ряда преподавателей позволил выстроить эту схему.

1. Анализируются общие закономерности

сти, на знание которых составлена данная задача.

2. Определяется алгоритм (способ) решения конкретной задачи.

3. Выбираются определенные формулы, коэффициенты, показатели и т. п., т. е. осуществляется конечное решение задачи.

По этой же схеме происходит и разбор задач, которые учащийся не смог решить самостоятельно.

1. Сначала преподаватель пытается определить, знает ли учащийся общие закономерности, которые отражены в задаче.

2. В случае если учащийся владеет этими закономерностями, преподаватель пытается выяснить, понимает ли он способ решения.

3. Наконец, если учащийся понимает и общие закономерности, и в общих чертах способ решения задачи, преподаватель выясняет, знает ли он конкретные математические выражения, коэффициенты и т. п., которые необходимо использовать при решении задачи.

Исходя из описанной схемы, мы разработали программы, объединенные условным названием «Репетиторы», которые как бы имитируют работу репетитора с учеником при решении задач. Поскольку программы использовались в преподавании физиологии в медицинском институте, на этапе определения алгоритма решения применялся способ решения в несколько действий: дело в том, что студенты-медики в силу особенностей медицины как науки мыслят в большей степени конкретно-образно и, кроме того, решение задач в физиологии — это не столько конечная цель, сколько способ прочнее усвоить различные закономерности, конкретные значения тех или иных показателей и т. п.

Программа-репетитор используется не сама по себе, а в комплексе с другими методическими средствами. На очередном занятии студентам задаются на дом задачи. На следующем занятии они работают с «Репетитором».

Прежде всего программа-репетитор предлагает учащемуся выбрать ту задачу, которую он решал дома. Далее он может выбрать один из трех вариантов работы:

1. Если он решил задачу дома, то ему предлагается ввести полученный ответ. В случае совпадения учащийся после соответствующей похвалы выходит из программы. В противном же случае он получает сообщение об ошибке и вновь возвращается в исходную точку.

2. Если учащийся не решил задачу дома, то он может выбрать вариант «Подсказка». При этом ему предлагается меню из трех вопросов:

а) Вы вообще не знаете, как решать подобные задачи?

б) Вы не знаете, как решить данную задачу?

в) Вы не знаете какую-либо конкретную формулу, показатель и т. п.?

В первом случае программа вместе с учащимся анализирует общие закономерности, на знание которых составлена задача, во втором — способ ее решения, в третьем — выясняет, какую конкретную формулу или показатель не знает студент. Ответы при этом, как уже говорилось, сообщаются не сразу, а после серии наводящих вопросов (программа подводит ученика к тому, что он сам находит верное решение).

3. Студент может выбрать работу со встроенной подпрограммой «Калькулятор», если ему необходимо произвести какие-либо расчеты.

После того как были составлены и отлажены в живом педагогическом процессе программы-репетиторы по некоторым темам физиологии человека, мы провели работу по оценке их эффективности. Предметом исследования был вопрос: помогают ли программы-репетиторы в решении задач?

Студентам были розданы на дом задачи на расчет легочной вентиляции, а на следующем занятии они работали с программой-репетитором. Студенты были предупреждены о том, что решение задач является добровольным и не влияет на оценку, в связи с чем давать в предложенной им анкете какие-либо ложные показания бессмысленно.

Самостоятельно дома пытались решить задачи 51 человек; из них справились с задачами 8, или около 15 % (следует отметить, что задачи были весьма трудными). После работы с «Репетиторами» с задачами справились 27, т. е. около 53 %.

Следует отметить, что среднее время работы с «Репетитором» составляло 15 мин и, с одной стороны, многие студенты не успели пройти программу до конца, а с другой, — напротив, часть студентов за это время успели разобрать несколько задач. Опыт показывает, что для того, чтобы практически каждый студент смог решить с репетитором хотя бы одну задачу, требуется около 30 мин.

Группа	Преподаватель	Студент			
	Дома	С компьютером			
Не решал	Решал, но не решил	Решил	Не решал	Решал, но не решил	Решил
Поставьте «+» в соответствующих графах					

Такие программы-репетиторы способны эффективно заменять преподавателя в одном из ключевых и вместе с тем рутинном этапе учебного процесса — в помощи учащимся в решении задач.

Отличие программ-репетиторов от обучающих программ состоит в том, что при работе с обучающими программами ученик сначала получает информацию, а затем следуют вопрос или задача. При этом в момент получения информации ученик не знает, на что следует обратить особое внимание; отсюда неизбежные возвращения к «информационному кадру» после получения задачи. Это приводит к большим затратам машинного времени. В случае «Репетиторов» ученик сначала получает задачу и самостоятельно и целенаправленно ищет необходимую информацию, лишь если он ее не находит, то запрашивает у компьютера. Таким образом, получение информации для него становится не самоцелью, а необходимым условием решения задачи. При таком подходе информация прочно закрепляется в памяти, так как, во-первых, она ищется активно и целенаправленно, а во-вторых, каждая закономерность запоминается в комплексе с другими зако-

номерностями, что делает запоминание более прочным [2]. Кроме того, ученик за компьютером получает информацию по данной теме не в полном объеме и отыскивает лишь ту, которая ему необходима; иногда достаточно лишь небольшой подсказки, чтобы задача была решена. Это резко сокращает использование машинного времени.

Описанные особенности делают программы-репетиторы эффективным средством педагогического процесса, обучение для студентов становится более увлекательным, а преподаватели освобождаются от рутинной работы, иными словами создается стимул и экономится время для истинно творческого труда.

Авторы признательны за помощь фирме «ОЛСИ», занимающейся разработкой систем для обучения и поддерживающей научно-методические работы в данном направлении.

Литература

1. Галызина Н. Ф. Управление процессом усвоения знаний. М.: Изд-во Московского университета, 1975.
2. Годфруа Ж. Основы психологии: Пер. с фр. М.: Мир, 1991.

84

И. ЗИНЧЕНКО

ЭВМ в начальной школе

Я хотела бы рассказать о своем шестилетнем опыте использования микроЭВМ на уроках в I—III классах начальной школы.

Мои первоклассники в течение года посещали базовое предприятие, где они работали на ЭВМ. Тогда и возникло желание использовать ЭВМ непосредственно на уроках в школе. Родители учеников по материалам журнала «Радио» сделали микроЭВМ «Радио РК-86» и подготовили в тесном контакте со мной, необходимое программное обеспечение, отвечающее следующим целям: повышению эффективности проведения урока,

организации режима индивидуально-го опроса,

усилению интереса учащихся к уроку, высвобождению учителя на определенном этапе урока.

В классе установлен один ПК «Радио РК-86» с двумя дисплеями: маленьким и большим — демонстрационным. Програм-

мы вводятся с кассетного магнитофона. Системное программное обеспечение хранится в съемных кассетах — блоках ППЗУ.

Второй компьютер, на котором я сама изменяю задания в стандартных заготовках контрольных работ, находится у меня дома.

Программное обеспечение написано на Бейсике (версия для ЭВМ РК-86) с использованием графического редактора для «Микроши», доработанного для данной ЭВМ. В его составе более 20 опросных программ по русскому языку (основные разделы), математике I—III классов, а также природоведению (раздел «Организм человека»). Кроме того, имеются заготовки демонстрационных программ для изучения алфавита и цифр в I классе и проведения контрольных работ.

Опросные программы по русскому языку работают в двух режимах: контролирующем и тренажерном. В первом режиме при неверном ответе учащегося ЭВМ выдает правильный ответ, а во втором — навядший

вопрос и вторую попытку для ответа.

ЭВМ используется на уроках при: объяснении нового материала (изучение алфавита и цифр), закреплении и повторении пройденного материала, проведении контрольных и самостоятельных работ, проверке домашней работы (природоведение).

На уроке в процессе устного счета, проведения словарной работы, закрепления и повторения, проверки домашнего задания я работаю с одной группой, а ученики из другой группы в это время по одному выходят к ЭВМ и выполняют предлагаемые задания. В конце ответа на экране появляется оценка и время, затраченное учеником на ответы, или число нерешенных примеров за заданный отрезок времени. Далее ученик включается в работу класса. По окончании этого этапа урока ученики называют оценки, которые им поставил компьютер, для занесения их в журнал.

Самостоятельная работа проводится в двух вариантах: первый аналогичен описанному выше, во втором — к ЭВМ выходит ученик быстрее всех справившийся с выполнением задания. Распределение контролирующих функций между учителем и ЭВМ повышает интенсивность проведения уроков и увеличивает число опрошенных. Нагрузка на учителя снижается.

Следует здесь же заметить, что ученики более охотно отвечают ЭВМ и если компьютер ставит им двойку, то горят желанием как можно скорее ее исправить. При проведении контрольных работ для всего класса (математические и словарные диктанты) средствами графического редактора на большой экран крупными символами выводятся задания. На этом этапе урока компьютер полностью заменяет учителя, позволяя ему в это время работать со слабыми учащимися или готовить доску к дальнейшей работе. А самое главное — учителю не нужно призывать учащихся к порядку и вниманию. Ученик знает, что если он отвлечется, то не успеет решить пример или записать слово, так как на экране через 10—15 с появится следующее задание. После выдачи контрольных заданий компьютер выводит на дисплей правильные ответы, и учащиеся производят самопроверку контрольных заданий. Учащиеся очень любят компьютерные диктанты. Эта работа проходит при полной тишине и огромном внимании. И если в начале года правильно справляются с этой работой 2-3 ученика, то к концу года большая половина класса выполняет эту работу на пятерки, а остальные — на четверки. Контрольные работы готовятся на домашнем компьютере перед текущим уроком.

Всем известны проблемы обучения, ориентированного на «среднего» ученика. Сильный ученик не получает должной нагрузки и у него снижается мотивация, а слабый — в силу обратной связи постоянно информируется о своих неудачах, и у него вырабатывается негативное отношение к школе, к учебе. Ученики, потерпевшие неудачу, проходят обучение менее успешно, теряют веру в себя, становятся более ранимыми в эмоциональном отношении. А если еще довести до сведения учеников их распределение по уровню успеваемости, то это не может не действовать как средство наказания. С целью избежания этого был введен дифференцированный опрос на компьютере. Сильные учащиеся получают более емкое сложное задание, и оценки выставляются им строже, что заставляет их не останавливаться на достигнутом и создает стимул к учебе. Слабым ученикам компьютер выдает меньшее число заданий и оценивает их соответственно.

Практически это осуществляется следующим образом. Все ученики звена получают разные варианты заданий, причем за каждым учеником резервируются два варианта — четный и нечетный.

Для слабых учеников программа под четными номерами варианта, для сильных — под нечетными. Ученик работает индивидуально с компьютером, нажимает номер своего варианта и отвечает на задаваемые вопросы. Таким образом, он не знает, меньше его задание или больше, чем у другого. Такое дифференцирование безболезненно для учащихся. По индивидуальным карточкам, в которые ученик заносит свое время и оценку, учитель видит его рост или спад. Если слабый ученик хорошо справляется со своей программой, его можно перевести на более сложную, и наоборот. В таких случаях учителю стоит только поменять вариант. При ответе у доски это было бы невозможно. В моем классе одиннадцать учеников работали по упрощенной программе. К концу первого полугодия с этой программой справились все. Из них пять учеников — на «хорошо», их я перевела на более сложную программу, т. е. изменила им вариант. Шесть учеников — на «удовлетворительно», они остались работать по тем же программам. И те и другие учащиеся продолжают совершенствовать свое мастерство, навыки и умения в меру своих возможностей.

Известно, что именно введение игровых моментов может вызвать дополнительный интерес учащихся к уроку. Поэтому с самых первых уроков в классе детей приветствует с большого дисплея робот Робик. Он показывает детям буквы, цифры, жонглирует и превращает их в любые слова или числа.

Введение игровых моментов в индивидуальный опрос на одной ЭВМ, требует боль-

шой осторожности, так как может вызвать ненужное оживление в классе или затянуть ответы учащихся, что при ограниченном времени урока нежелательно. Наиболее интересна в этом отношении программа, в которой ученику предлагается за 1 мин решить любое количество примеров, постепенно усложняющихся. Если ученик правильно решил пример — в корзину падает яблоко, если неправильно — нет. Ученик заносит в индивидуальную книжечку число яблок (правильно ре-

шенные примеры) и число, обозначающее неправильно решенные примеры. Причем между звездочками (звеньями) проводится соревнование (командное и личное), кто больше собрал яблок.

В заключение хочу подчеркнуть, что в нынешних условиях неудовлетворенного спроса на ЭВМ (особенно на компьютерные классы), опыт использования одной ЭВМ в I—III классах может оказаться полезным и в других школах.

А. ГЕЙДАР

Из опыта преподавания информатики в V классе

86 Преподавание информатики в V классе имеет свои сложности. При планировании материала приходится учитывать особенности подготовки пятиклассников, умеющих выполнять вычисления в пределах четырех арифметических действий и не владеющих такими важными для информатики понятиями, как десятичные дроби, стандартный вид числа, степень, корень и др. Они еще не знают, что такое алгоритм, не умеют пользоваться вычислительной техникой (микрокалькулятором). Отсутствие методических рекомендаций, учебного пособия еще больше усложняет дело.

Именно поэтому на первом году преподавания информатики в V классе особое внимание уделяется следующим вопросам:

введению самого общего понятия алгоритма, роли алгоритма при составлении программ, формального исполнения;

приобщению учащихся к работе на компьютерах не только в качестве пользователей, но и для ознакомления с основами программирования;

развитию логического мышления.

Всем этим требованиям удовлетворяет «Программа для робота Дежурика». Робот Дежурик может выполнять обязанности дежурного по классу по заранее составленной программе. Вначале даются правила записи команд:

команда записывается с новой строки прописными (заглавными) буквами;

команда заканчивается точкой с запятой — признак окончания;

не допускается введения знаков переноса при переходе на другую строку.

После этого вводится понятие «множество предписаний», т. е. множество команд, которые может исполнить робот. На доске

вывешивается красочно оформленный плакат:

Множество предписаний:

Закрой окно
Сотри с доски
Намочи тряпку
Открой окно
Сядь на место

Обращается особое внимание на соблюдение логической последовательности выполняемых команд. Ученикам предлагается пять программ для анализа. Они должны найти ошибки и определить, какие программы решают поставленную задачу, какие — нет, какие не могут быть выполнены. Перед этим вводится понятие начальных условий, т. е. с чего начинается работа программы: урок подходит к концу, вся доска исписана, тряпка совершенно сухая, единственное большое окно в классе закрыто.

Затем ученикам предлагаются программы:

Программа Ани:
Закрой окно;
Сотри с доски;
Намочи тряпку;
Открой окно;
Сядь на место;

Программа Наташи:
Отвори окно;
Намочи тряпку;
Подмети пол;
Вытри доску;
Закрой окно;
Вернись на место;

Программа Толи:
Открой окно;
Сотри с доски;
Намочи тряпку;
Сядь на место;

Программа Миши:
Открой окно. Намочи тряпку. Сотри с доски. Закрой окно. Сядь на место.

Программа Димы:
Открой окно;
Намочи тряпку;
Сотри с доски;
Закрой окно;
Сядь на место;

Робот не поймет и не сможет выполнить программу Миши и Наташи. Миша грубо нарушил правила записи программы, а Наташа использовала команды «Отвори окно», «Подмети пол», «Вытри доску» и «Вернись на место», которых нет в множестве предписаний.

В программе Ани нарушена логическая последовательность действий.

Толину программу Дежурик поймет, но ее выполнение не приведет к желаемому результату.

Программа Димы единственно правильная.

В начале года с учениками проводится несколько тренировочных занятий на компьютерах. Ребята знают, как включить питание, загрузить программу с диска, запустить или остановить ее, знают назначение клавишей редактирования, ввода, переключения реги-

стров. Практическое занятие закрепляет знания по данной теме. Ученики представляют себя сидящими за пультом управления роботом Дежурик. Набранные ими команды робот выполняет и выдает на экран сообщение об этом.

Программа составлена для КУВТ-86 (ДВК-2М и БК-0010Ш), но может быть легко адаптирована для любого другого типа ЭВМ.

Использование этой программы позволяет в виде игры обучать ребят программированию.

За справками обращаться по адресу: 714007, Киргизская ССР, г. Ош, м-р Тюлейкап, д. 4, кв. 2.

Л и т е р а т у р а

Звенигородский Г. А. Первые уроки программирования. М.: Наука, 1985.

Ю. СЕРЕДА

Разработчикам педагогических программных средств

Проблемы учителя информатики в средней школе известны всем: несовершенство, а зачастую и отсутствие вычислительной техники, хорошего программного обеспечения, учебников и т. д. У преподавателя техникума, кроме этого, есть и другие проблемы.

Одна из них — разное количество часов на различных специальностях (например, в нашем техникуме курс информатики на четырех специальностях — 54, 78 и 99 ч). В этом случае главное, на наш взгляд, — выработать единую методику преподавания, единую стратегию. В статье я хочу поделиться своим опытом.

Процесс преподавания информатики я условно разбил на 3 части.

1. Подготовительная. На нее для всех специальностей отводится 14 ч. Основные цели этих занятий — научить учащихся правильно записывать алгоритмы решения задач в виде блок-схем; знать правила записи каждого блока; уметь находить ошибки в готовой схеме. На этом этапе учащимся постоянно внушается мысль: программа на ЭВМ может быть правильной только в том случае, если правильно построена блок-схема решения задачи.

2. Основная. Эта часть содержит изучение языков высокого уровня (Фокал, Бейсик). На занятиях учащиеся учатся построению программ для тех задач, которые они освоили в первой части, т. е. они оттачиваются от уже готовых и проработанных

блок-схем. Основное внимание обращается на формат записи и принцип работы операторов данного языка. Заканчивается изучение примерами перевода программ с одного языка программирования на другой (с Фокала или Бейсика на Фортран или Паскаль). Учащийся должен понять: если у него не возникает проблем с построением блок-схем, если он хорошо знает хотя бы один язык программирования (пусть даже это язык программирования на ПМК), то при изучении другого языка проблем у него будет немного.

3. Дополнительная. Она включает в себя темы: «Решение математических, физических, экономических задач», «База данных», «Текстовый редактор», «Графический редактор», «Адресное пространство компьютера». Включение этих тем в курс информатики зависит только от объема часов (согласитесь, что если на предмет отводится 54 ч, то включить эти разделы в план нет никакой возможности).

На самостоятельных и контрольных работах по ч. 1 и 2 учащимся на выбор предлагаются задачи различной сложности, они знают, какую оценку получают за решение каждой из них. Преподаватель по итогам этих работ видит, кого он не сумел заинтересовать своим предметом, кто хочет иметь оценку выше удовлетворительной, но не дотягивает до хорошей, кого можно было бы привлечь к созданию программного обеспечения, в том числе и по другим предметам.

В нашем техникуме, кроме «ОИВТ» преподаются «Программирование», «Основы вычислительной, микропроцессорной техники», «Системы автоматизированного проектирования». Поэтому на занятиях мы не рассматриваем подробно такие темы, как «Работа процессора», «Физические принципы работы ЭВМ». Все внимание сосредоточено на решении следующих задач.

1. Научить учащихся свободному общению с компьютером, составлению несложных программ прикладного характера, игровых программ.

2. Убедить в невозможности развития науки и техники без дальнейшего развития и всестороннего применения средств микропроцессорной и вычислительной техники.

3. Привить интерес к дальнейшему, более глубокому изучению программирования. Для этого на занятиях по возможности рекламируется программируемый микрокалькулятор — объясняется соответствие операторов Фокала или Бейсика и команд ПМК, дается описание игровых программ. У ПМК, как мне кажется, есть два больших преимущества — низкая стоимость и наличие команд косвенной адресации.

Хотелось бы немного остановиться еще на одной проблеме (но не последней) — как эффективно провести лабораторную работу, если в техникуме один кабинет ВТ.

Описание лабораторной работы дается

учащимся накануне ее проведения на последних 10—15 мин теоретического занятия (одно описание на двоих). Они смотрят, как нужно оформлять данную работу, имеют возможность повторить пройденный материал, выписывают задание. Первые двое из учащихся, успевших до конца урока правильно выполнить задание (а такие ребята есть, хотя и не во всех группах), освобождаются от теоретической части лабораторной работы. Им даются или более интересная и сложная задача (не на оценку), или какой-либо материал из книги по программированию для просмотра. У этих учащихся появляется возможность сдать информатику раньше (преподаватель должен помочь им в этом), а у преподавателя — уделить остальным учащимся больше внимания, повысить их уровень знаний.

Группа (30 человек) делится на две подгруппы, и лабораторная работа проходит следующим образом: каждая подгруппа занимается 45 мин в кабинете ВТ, 45 мин — в другой аудитории. За время, проведенное в кабинете ВТ, учащемуся необходимо решить на компьютере поставленную задачу. Лабораторная работа засчитывается, если результаты задачи будут отображены на мониторе. 45 мин, проведенные в другой аудитории, учащиеся разбирают более сложные задачи или отвечают на теоретические вопросы по данной теме (письменно, на оценку).

Качество как функция размеров

В области компьютерных технологий часто встречаются ситуации, когда уменьшение изделия улучшает его качество. Чем меньше диаметр магнитного диска, тем выше его информационная емкость и скорость обмена данными; сверхминиатюрные транзисторы микросхем в десятки тысяч раз надежнее обычных.

Уменьшение размеров платы, на которую монтируются микросхемы, также дает выигрыш. Если кремниевые кристаллы «упакованы» в корпуса, то приходится использовать длинные (10—20 мм) проводники для создания

наружных выводов, что приводит к росту паразитных емкостей и индуктивностей, снижению скорости действия, увеличению помех — и это еще без учета длины соединений на печатной плате! А при монтаже бескорпусных микросхем их можно расставить практически вплотную друг к другу, в 2—3 раза уменьшая длину выводов. Укорачиваются и межкристальные соединения.

ЧТО?
МОЖЕТ?
ЭВМ

Можно и вообще отказаться от использования специальных проводников для выводов. Достаточно в тех точках кристалла, от которых должен отводиться сигнал, нарастить миниатюрные (сотые доли мм) металлические «шишечки», а на плате сделать в соответствующих местах контактные площадки. После этого остается только приклеить кристалл «верхом к плате» на нужное место. Такая технология сборки называется «метод перевернутого кристалла» и не только повышает возможности изготовления микросхем благодаря тому, что сигналы не нужно «разводить» на ограниченное число выводов по периферии корпуса.

И. ПЕРЕХОД, В. КАСАТКИН

Арифметика фибоначчией системы счисления

Математикам хорошо известен класс так называемых последовательностей Фибоначчи. Свойства чисел, образующих такие последовательности, изучаются давно. Сегодня найдено немало интересных и практически значимых применений этих чисел.

Приведем наиболее общее определение последовательности:

всякая последовательность Фибоначчи состоит из натуральных чисел и задается указанием двух первых ее членов (F_1 и F_2), а каждый член, начиная с третьего, равен сумме двух, непосредственно ему предшествующих ($F_k = F_{k-1} + F_{k-2}$).

Примеры.

$$1, 1, 2, 3, 5, 8, 13, 21, \dots \quad (1)$$

$$2, 5, 7, 12, 19, 31, \dots \quad (2)$$

$$3, 7, 10, 17, 27, 44, \dots \quad (3)$$

Традиционно основной последовательностью Фибоначчи называют (1), $F_1 = F_2 = 1$, а задание фибоначчией системы счисления определяется указанием:

цифрового множества $\{0, 1\}$;

базиса системы, в качестве которого берутся числа, образующие основную последовательность Фибоначчи

..., 34, 21, 13, 8, 5, 3, 2, 1.

Фибоначчией система позиционная, поэтому каждое число может быть записано в цифровой и многочленной форме:

$$101001 = 1 \times 13 + 0 \times 8 + 1 \times 5 + 0 \times 3 + 0 \times 2 + 1 \times 1 = 24_{10}$$

$$10101001 = 1 \times 34 + 0 \times 21 + 1 \times 13 + 0 \times 8 + 1 \times 5 + 0 \times 3 + 0 \times 2 + 1 \times 1 = 52_{10}$$

Всякое целое десятичное число всегда можно представить в виде равного ему числа, записанного в фибоначчией системе счисления:

$$37_{10} = 1 \times 34 + 1 \times 3 = 10000100$$

$$25_{10} = 1 \times 21 + 1 \times 3 + 1 \times 1 = 1000101$$

Целое число A_{10} представить в виде равного ему числа фибоначчией системы озна-

чает задать его в виде суммы чисел Фибоначчи, вводя в эту сумму возможно большие из чисел. Это, как известно, устраняет неоднозначность представления чисел в любой позиционной системе.

Для отыскания представлений чисел фибоначчией системы используется процесс вычитания чисел Фибоначчи сначала из данного числа, а затем из образующихся промежуточных разностей R_k .

Пример. Дано число 33_{10} .

Сначала из числа 33 вычитаем наиболее близкое к нему (снизу) число Фибоначчи $F_7 = 21_{10}$:

$$33 - 21 = 12 \quad (R_1 = 12, F_7 = 21)$$

$$12 - 8 = 4 \quad (R_2 = 4, F_5 = 8)$$

$$4 - 3 = 1 \quad (R_3 = 1, F_3 = 3)$$

$$1 - 1 = 0 \quad (R_4 = 0, F_1 = 1)$$

Процесс завершается при $R_4 = 0$; $33_{10} = 101010$.

Подчеркнем, что при записи чисел в фибоначчией системе счисления две цифры 1 не должны записываться рядом. Это следует из того, что два соседних числа последовательности Фибоначчи образуют в сумме следующее за ними число Фибоначчи.

Ниже приводятся программа замены десятичных чисел на равные им числа фибоначчией системы счисления и примеры ее работы.

```

10  ---- ДЕШИФРАТОР 10-->FIBO ----
20  INPUT "A="; A: N=64: PRINT "A=";
                                     A; "=";
30  DIM F(N+1): F(0)=1: F(1)=1
40  FOR I=1 TO N
50  F(I+1)=F(I)+F(I-1)
60  IF F(I+1)<=A THEN NEXT I
70  FOR K=I TO 1 STEP -1
80  IF F(K)<=A THEN PRINT 1; A=A-F(K)
                                     ELSE PRINT 0;
90  NEXT K: PRINT "(Ф)"
100 END
    
```


ЧТО? МОЖЕТ ЭВМ

Провода? Только электрические!

Вы спросите: а какие же еще? Развитее страны все плотнее опутывают провода «информационные» — телефонных, компьютерных, телевизионных сетей. Вот от компьютерных-то и пытаются избавиться (если не страны, то хотя бы учреждения — для начала) известная фирма из США Motorola.

Предлагаемая ею радиосеть для помещений работает в диапазоне 18 ГГц. Сигналы такой частоты способны проникать через мельчайшие трещины, так что большинство стен не экранируют их. В то же время они довольно быстро затухают с расстоянием, что позволяет радиосети в соседнем доме работать на той же частоте.

«Полупроводниковое обеспечение» этой сети исполнено на высшем уровне технологии и включает сверхвысокопроизводительные арсенид-галлиевые микросхемы для СВЧ-связи, цифровой процессор сигналов с пропускной способностью 15М бит/с, «разумную» шестисекторную антенну, позволяющую справиться с искажениями, появляющимися в результате отражения радиоволн от различных металлических предметов.

15 миллиметров светового луча

отводилось на каждый сигнал при рекордной — 20 000 000 000 битов в секунду! — передаче сигнала по волоконно-оптической линии длиной 114 км. Исследователи из фирмы Standard Electric Lorenz (ФРГ) сумели с помощью относительно дешевых технологий преодолеть «размывание» светового импульса в волоконно-оптических линиях.

Этой же фирмой совместно с учеными из Рурского университета установлен и другой рекорд: создана кремниевая микросхема,

работающая с такой же супер-скоростью — 20 Гбит/с. Она превзошла даже значительно более дорогие арсенид-галлиевые микросхемы. Как и предыдущий, рекорд был получен не благодаря использованию новейших и дорогостоящих технологических решений, а путем тщательной проработки общей конструкции и оптимизации работы каждого транзистора.

Когда есть ум, можно обойтись и без силы.

100 лошадиных сил и 2 РС ХТ

Возможно, именно так скоро будут выглядеть рекламные объявления о новых моделях автомобилей. «Вычислительная мощность» легковушек быстро растет. В средней фордовской модели стоимость электронного оборудования превысила 1000 долларов, а самый дешевый компьютер IBM PC XT сейчас стоит 400.

Новая вершина

Компанией Fujitsu America выпущен имеющий небольшие габариты винчестер емкостью 2 Гбайта, замечательный еще и чрезвычайно высоким быстродействием. Трудно сказать, однако, когда это прекрасное техническое достижение получит широкое распространение. Сегодняшним новинкам упорно сопротивляются новинки вчерашние, далеко не утратившие своей привлекательности. В частности, продажа компьютеров на микропроцессоре 80486 долго еще не превысит 0,1% от общего объема продаж, в то время как 40% эксплуатируемых ПК имеют процессор 80286.

Одна на двоих

Именно так пришлось бы распределять между компьютерами IBM PC XT, не самыми плохими по нашим стандартам, одну из новинок концерна Siemens — микросхему памяти емкостью 2 Мбайта. Она создана в рамках совместного проекта Mega двух гигантов — Siemens и Philips. Европейские «акулы империализма» объединяются, поскольку разработка подобных изделий стоит сотни миллионов долларов, а отказать от нее нельзя, чтобы не стать «сельскохозяйственным придатком» Японии и США.

Параллельно Siemens снимает с производства микросхемы емкостью 256 Кбит (что равно памяти БК-0010); достигло пика производство микросхем на 1 Мбит, цена которых снизилась до 5 долларов за штуку. Советским компьютерщикам-самодельщикам становится выгодно использовать импортные детали: 150 рублей за 128 Кбайт заметно ниже магазинных цен!

Но отставание Европы от Японии еще не преодолено. 256К-битные микросхемы Siemens стал производить на 3 года позже передовых японских компаний, 1М-битные — на год, 4М-битные — на полгода. Лишь в 1992 г., с началом серийного производства 16М-битных микросхем, удастся догнать соперников.

Go, Германия, go!

16 процессоров (каждый из которых содержит по два арифметических устройства) на одном кристалле площадью 10 см², работающих с тактовой частотой 125 МГц и способных выполнять 4 000 000 000 (миллиарда!) операций умножения в секунду, — таковы характеристики видеопроцессора Datawave немецкой компании Intermetall.

Трепещите, конкуренты!

ВСЕМ РУКОВОДИТЕЛЯМ УЧЕБНЫХ ЗАВЕДЕНИЙ

Брестское производственное объединение средств вычислительной техники предлагает приобрести комплекты учебных терминальных классов «Корвет» (12 рабочих мест, учительская ПЭВМ с цветным монитором, два дисковод, принтер Epson).

В составе класса дополнительно поставляются:

а) комплект учебной и методической литературы, включающий: набор из 20 учебников по информатике под редакцией В. А. Каймина (новая редакция);

методические пособия для учителя по методике В. А. Каймина; сборники задач по информатике;

б) программный пакет «Альтернатива-Корвет» производства НТП «Альтернатива» НИИ ИВТ АПН СССР (г. Барнаул). В составе пакета: пересылаемая по локальной сети операционная система СР/М, обеспечивающая быструю пересылку программ по сети; система программирования Турбо-Паскаль;

набор учебных программ (текстовой, графической, музыкальный редакторы, тренажеры, базы данных, электронные таблицы, пакет игровых программ).

Гарантируются:

● стандартность класса и совместимость с иными классами «Корвет»;

● высокая надежность поставляемых ПЭВМ;

● соответствие учебников и литературы последним методическим достижениям по курсу информатики;

● высокий технический уровень программного обеспечения, пересылаемость всех программ по локальной сети, качественное превосходство его над иными известными разработками («Корнет», «Парус» и т. д.).

Стоимость класса (с литературой и программным пакетом) — 35 157 рублей.

Заявки просим присылать по адресу:

224634, Брест, ул. Московская, 202, БПО СВТ, отдел сбыта (тел. 2-21-03, 2-62-94).

Вниманию владельцев и пользователей компьютеров БК-0010, БК-0010-01!

Саратовский завод «Эридан» предлагает вам

блок дополнительного оперативного запоминающего устройства (ДОЗУ 32К). Блок ДОЗУ 32К имеет габариты 105×175×24 мм, подключается к БК-0010, БК-0010-01 через разъем МПИ и обеспечивает увеличение емкости ОЗУ ЭВМ на 32К байт.

**Наш адрес: 410033, Саратов, ул. Панфилова, 1, з-д «Эридан».
Телефон для справок: 17-67-92.**

И. РЯЗАНОВА

г. Троицк, Московской обл.

АНГЛИЙСКИЙ ЯЗЫК И КОМПЬЮТЕР ДЛЯ ДОШКОЛЬНИКОВ

Обучая английскому языку 5—6-летних детей в кружке, я пришла к выводу, что занятия можно начинать гораздо раньше — с 3—4 лет. Тогда к 5—6 годам дети будут уже достаточно подготовлены к более продуктивной и серьезной работе. Именно ранний возраст характеризуется, по словам К. Чуковского, «изошренным чувством языка».

Занятия с детьми младшего возраста условно можно разделить на 2 этапа: 1-й — устный курс продолжительностью около 2 лет (для детей 3—4 лет); 2-й — смешанный курс (для детей 5—6 лет), когда наряду с традиционными устными формами вводятся чтение, элементы письма, работа на компьютере.

Мы занимались раз в неделю в компьютерном центре «Байтик» на машинах типа IBM PC.

Мне кажется, что начинать использовать компьютер полезно в начале второго этапа, когда дети выучили английский алфавит и непосредственно приступают к чтению.

Очевидно, что для детей этого возраста первое знакомство с компьютером (мы назвали его Compu) бессмысленно начинать с изучения команд DOS и принципов построения файловой системы.

Я просто говорю: «Чтобы поработать с программой, нам с вами нужно прийти в то место компьютера, где эта программа находится. Давайте скажем Compu: «Иди» — и укажем адрес. Для этого надо набрать на клавиатуре «CDAVC». «CD» звучит как «сиди», но на компьютерном языке это значит совсем другое — не сиди (на месте), а, наоборот, иди. Но наш Compu очень умный и не хочет «идти туда, не знаю куда», поэтому мы написали ему название улицы — «ABC». Но название улицы может быть лю-

бое, заранее он его не знает, поэтому, прежде чем пойти, он обязательно должен получить от нас сигнал-разрешение, а для этого нам следует нажать клавишу ENTER (выполний!). Видите, как Compu быстро привел нас на улицу «ABC» и остановился. Он не знает, в какой дом мы хотим войти. Давайте ему поможем, напомним номер — название дома и скажем: «Выполний!» Теперь наша программа заработала, значит, мы правильно объяснили Compu, что мы хотели сделать, а он правильно понял нас».

На этом практически заканчивается изучение с детьми операционной системы как таковой, но тем не менее предварительно еще необходимо ввести некоторые часто употребляемые английские термины (enter, escape, quit и др.), познакомить детей с общим устройством компьютера и клавиатурой, назначением некоторых клавиш.

Затем начинается непосредственно работа с конкретными программами.

Мы пользовались оригинальной программой «ALPHABET GAMES» (by Donald L. Pavia, May, 1986). В ней существуют 4 типа заданий: набрать букву (отдельно прописную, отдельно строчную); набрать букву, следующую за названной (отдельно прописную, отдельно строчную).

Машина изображением и музыкой реагирует на правильность выполнения. Эта программа хороша для повторения букв, их алфавитной последовательности и для развития навыков работы с клавиатурой (поиск нужной буквы на клавиатуре). Дети прекрасно с ней справляются. К ее недостаткам следует отнести не очень удачное графическое начертание букв: зная данную букву по книжному написанию, ребенок не всегда может узнать ее на экране.

Этого недостатка лишена программа АЛФАВИТ. Работая с ней, ребенок сам выбирает на экране курсором букву для изучения, а затем должен отметить эту же самую букву в предлагаемом компьютером предложении. Как только указаны все, звучит мажорная музыкальная фраза. Эта программа хороша тем, что, не требуя умения читать, подслупно подготавливает детей к этому и является переходной ступенью к следующим программам.

Программа GOOD DAY была написана специально для нас, на знакомом детям материале. Она заключается в том, что ребенок должен прочесть букву, потом слово с этой буквой (или с комбинацией букв), затем ответить на вопрос, как произносится данная буква (или комбинация) в данном слове.

В программе «живет» забавный клоун. Он реагирует на ответы ребенка. Если ответ правильный, машет шляпой, если нет — возмущенно топает ногами. Программа позволяет выявить, насколько дети знают определенные звуко-буквенные соответствия, а также символы для написания звуков в транскрипции. Программа, будучи контролирующей, является и обучающей, так как в случае неправильного ответа детям дается подсказка, которая предлагается до тех пор, пока они не усвоят звук и не ответят правильно.

Все набираемые учеником ответы хранятся в ОЗУ и могут быть вызваны учителем на экран или записаны в файл, что позволяет ему точно представить, насколько легко или, наоборот, трудно воспринимается ребенком материал.

Программа включает в себя 5 файлов. Файл ABC.BAT запускает драйвер русской клавиатуры и собственно программу, написанную на языке Бейсик. Необходимость драйвера русской клавиатуры объясняется тем, что не все символы английской транскрипции совпадают с буквами стандартной клавиатуры. Путем модификации драйвера EGAGA.EXE для недостающих символов транскрипции были использованы клавиши, не содержащие букв.

В процессе работы программа обращается к двум текстовым файлам, связанным между собой построчно. Один из них содержит предлагаемые детям вопросы, а другой ожидаемые ответы, с которыми сравнивается ответ, набранный ребенком. Положительным моментом является то, что по мере изучения нового материала он может легко вноситься в программу с помощью простейшего текстового редактора и предлагаться детям для повторения и контроля их знаний.

На основе вышеописанной была создана программа COLOURED CAT, в которой по-

являющийся кот различной окраски требует правильно написать свой цвет.

Еще одна программа WORDS, которой мы пользовались, была написана по нашему заказу на языке Паскаль. На экране возникают 10 английских слов с пропущенной буквой и русским переводом. Ученик имеет три попытки. Подогнав стрелками курсор в нужное место, он должен ввести правильный символ. При первом же верном ответе он получает оценку «пять», вторая правильная попытка оценивается четверкой, а последняя тройкой. Если и она неудачна, компьютер сам вставляет правильную букву, но при этом ученик вынужден довольствоваться только двойкой. В конце урока подводится итог: выдаются количество правильных ответов и сумма оценок (максимум за 10 слов 50 очков). Затем «страница пероворачивается» и учащемуся предлагаются новые 10 слов. Программа сопровождается музыкальными и цветовыми эффектами. При ее использовании преподавателю легко выявить знания ребенка, понять, на что следует обратить внимание, буквально тут же, вызвав в редактор файл нужного урока, ввести новые слова или пропустить другую букву. Дети 5—6 лет еще плохо и неохотно пишут, и навыки письма неизбежно отстают от навыков чтения. Здесь компьютер может оказать неоценимую помощь, так как нажимать клавиши куда приятнее, чем просто писать ручкой или карандашом. Таким образом, во время работы с этой программой усваивается и контролируется spelling, а с другой стороны, приобретаются навыки работы с клавиатурой, аналогичные навыкам работы с текстовым редактором. Для проведения быстрой контрольной работы существует специальная версия этой программы с контролем времени и блокировкой выхода.

Следует отметить, что написание программ дело очень хлопотное и дорогое и, скорее всего, наши программы, написанные, как правило старшекласниками, занимающимися в «Байтике» (С. Тихонов, С. Четвертков, А. Миронов и др.), не столь внешне эффектны, как те немногие «настоящие», которые довелось увидеть. Но я убеждена, что дошкольнику, который еще не способен быстро читать и только учится концентрировать внимание, как раз и не нужно иметь много различной информации на экране. Для него гораздо эффективнее система «вопрос — ответ», а предварительный комментарий к заданию должен даваться учителем устно.

Работа в компьютерном классе показала, что абсолютно все ребята охотно ведут диалог с машиной. У них отсутствует боязнь компьютера, которой страдают многие взрос-

лые. Даже у застенчивых и слабоуспевающих детей пропадает чувство неуверенности в себе, появляется раскованность, после упражнений на компьютере они с большим интересом работают на последующих (некомпьютерных) уроках. Уже одно то, что они работают с английской клавиатурой, с командами DOS на английском языке, способствует усвоению основных терминов как в специальном, так и в бытовом значении и повышает общую, компьютерную и языковую культуру ребенка.

Теперь мне хотелось бы остановиться на ряде трудностей, с которыми мы столкнулись в работе. Главной проблемой были и остаются программы, по которым можно заниматься с маленькими детьми. Нет достаточно полной информации, о многом мы просто не знаем, а из того, что есть, далеко не все нас устраивает. Так, например, оригинальная программа АЗБУКА (LEXICA, Москва, 1989) в режиме Find Word Games могла бы быть использована для дошкольников, однако содержит большое количество незнакомой им лексики и не позволяет самостоятельное ее введение или изъятие. То же самое относится и к программе READER RABBIT (The Learning Company, 1987). К тому же, «Байтик» пока располагает только тремя компьютерами типа APPLE, для которых она написана, что не позволяет вести занятия с группой. Этого недостатка лишена хорошая разработка («LEARN ENGLISH» Н. Лунина, г. Пушкино), но имеющаяся у нас версия написана для ПЭВМ «Ямаха» и в нашем компьютерном центре «Байтик» мо-

жет быть использована только в нецветном варианте, что значительно ее обедняет. Одним словом, необходима полная информация об обучающих программах по английскому языку, чтобы преподаватели имели возможность наладить контакты между собой и договориться о сотрудничестве на взаимовыгодных условиях (я имею в виду среди прочего и обмен программами).

В заключение хотелось бы сказать, что компьютеры, будучи сами по себе крайне привлекательными для детей, могут явиться мощным стимулом к изучению языка, особенно при некотором снижении к нему интереса детей. Однако использование одних лишь компьютеров вряд ли может привести к поставленной цели, ее можно достичь, только комплексно используя все средства и методы обучения.

А цель у нас прекрасная — как можно раньше научиться любить и знать английский язык, язык международного общения.

95

Литература

1. Амамджян Ш. Г. «Play and Learn English», М.: 1982.
2. Азурова Н. В., Гоздеуцкая С. И. Английский язык в детском саду. М. 1963.
3. Верещагина И. М., Дубровин М. И., Притыкина Т. А. Учебник английского языка для второго класса школ с преподаванием ряда предметов на английском языке. М. 1982.
4. Галеева Д. С. Некоторые вопросы использования микро-ЭВМ в преподавании иностранных языков в школах Великобритании и США // Иностранные языки в школе. 1990. № 2. С. 85—89.

«Парадокс» выходит вперед

Система управления базами данных Paradox становится в последние годы все более популярной, и не исключено, что вот-вот она станет лидером среди других систем управления базами данных (СУБД).

Фирма Computer Intelligence (США) занимается анализом покупок программного обеспечения крупнейшими компаниями. По ее данным, в сентябре 1990 г. количество продаваемых копий Paradox превысило суммарное количество продаваемых версий dBASE III+ и dBASE IV, которые для нас являются едва ли не синонимом СУБД. На долю Paradox пришлось 44 % вновь проданных СУБД, а на обе версии dBASE — 34 %.

ЧТО?
МОЖЕТ?
ЭВМ

Секс и компьютеры

Последняя новинка дисковой библиографии: в Италии на шести дисках CD-ROM издана первая часть (по 1930 г. включительно) каталога эротических фотографий. На диски записаны как сами снимки, так и комментарии к ним и статьи искусствоведов об эротике в изобразительном искусстве.

Мышка в норке

Продолжаются попытки приспособить манипулятор курсора к условиям эксплуатации портативных ПК. Последнее предложение — «мышка в норке», точнее, под одной из клавиш компьютера.

Под колпачком такой клавиши кроме обычных контактов размещается датчик усилия, реагирующий на боковое смещение колпачка. Отклонять клавишу можно вперед, назад, вправо, влево или в любом промежуточном направлении, — иными словами, это не джойстик, а именно мышь. Другое отличие от джойстика — переменная скорость движения курсора, зависящая от приложенной к клавише силы (от 3 до 100 г).

«Спрятала» мышку компания Home Row (США).

УЧЕБНЫЕ КЛАССЫ

на базе советских IBM-совместимых компьютеров
«ИСКРА-1031»

с пакетами учебных программ
для средних и высших учебных заведений

предлагают Курское производственное объединение «Счетмаш»
и Московский городской институт усовершенствования учителей

Учебные классы состоят из центральной учительской ЭВМ и от одной до двенадцати ученических ЭВМ, объединенных в локальную сеть типа «звезда».

В качестве учительской и ученических машин в классе используются персональные компьютеры «Искра-1031», программно совместимые с ЭВМ типа IBM PC.

Каждый из компьютеров имеет черно-белый монитор MC 6105, накопитель на гибких магнитных дисках (5,25') емкостью 360 или 720К байт, оперативную память объемом 640К байт и процессор типа i8086.

Учительский компьютер дополнительно имеет накопитель на жестком магнитном диске типа «Винчестер» емкостью 20М байт и печатающее устройство CM 6337 с шириной каретки 420 мм и скоростью печати 180 зн/сек.

Классы комплектуются ученическими компьютерами от 1 до 12 шт. по заказу; ученические ЭВМ могут быть доукомплектованы «Винчестером» или печатающим устройством. Стоимость учительской ЭВМ — 15 тыс. рублей, ученической — 6 тыс. рублей.

Директора школ и руководители учебных заведений могут использовать эти же компьютеры и для решения своих административно-управленческих задач: подготовки справок, отчетов, ведения архивов, бухгалтерских расчетов и т. п.

Техническое обслуживание учебных классов на базе ПЭВМ «Искра-1031» осуществляется во всех регионах Советского Союза. Гарантийное обслуживание в течение 1,5 лет. По истечении срока предоставляются льготы для приобретения нового усовершенствованного класса.

В составе учебных классов «Искра-1031» поставляется сетевой пакет «Net», позволяющий производить обмен информацией и программами между учительской и ученическими ЭВМ, и базовый пакет программ для обучения основам информатики и вычислительной техники в соответствии с учебником В. А. Каймина.

В состав базового пакета учебных программ входят тренажер для освоения клавиатуры, учебный текстовый и графический редакторы, учебные база данных и электронные таблицы, а также авторская система для подготовки контролирующих и обучающих программ.

Заявки на приобретение учебных классов и пакетов программ направлять по следующим адресам:

305901, Курск, ул. Республиканская, 6, ПО «Счетмаш». Генеральный директор Рыков Александр Михайлович. Справки по телефону 6-01-59.

125167, Москва, Авиационный пер., 6, МГИУУ. Заместитель директора Каймин Виталий Адольфович. Справки по телефону 151-59-31, 151-44-11.

«Альтернатива» расширяет деятельность

Среди разработчиков программного обеспечения для учебной ВТ большую известность имеет предприятие «Альтернатива» из Барнаула, учредителем которого является НИИ ИВТ АПН СССР (Новосибирск). Читателям нашего журнала оно известно (см. № 3 и 6 за 1989 г., № 3, 5, 6 за 1990 г.). Предлагаем вниманию наших читателей интервью с директором «Альтернативы» А. Н. Гриценко. С вопросами или предложениями можно обращаться по адресу: 656049, Барнаул, а/я 3475, «Альтернатива».

— Алексей Николаевич, как вы оцениваете современное состояние программного обеспечения для КУВТов?

— Состояние безрадостное. С завода комплекты приходят «голыми», с набором программ, который нельзя назвать даже минимальным. Обычно это Бейсик, малопригодная ОС и сеть для Бейсик-программ, чего до смешного мало даже для уроков информатики, не говоря уже о других предметах. На местах программные фонды формируются стихийно, путем взаимного обмена или случайного приобретения программ, как правило не лучших.

Мы стараемся по мере сил улучшить положение, поставляя своим заказчикам пакеты, созданные нашими разработчиками. При этом мы обычно рекомендуем стереть все пришедшее с КУВТом программы как устаревшие и при наличии нашего пакета не представляющие ценности.

— Возможности заводов в части производства программ равны нулю, а в части посреднической продажи минимальны. Но что мешает создавать хорошие программы многочисленным вузам, ИУУ, институтам АПН, тому же ИВТ АПН СССР?

— В свое время Минпрос СССР дал этой сфере лучшую тогда вычислительную технику — импортные «Ямахи» — и тем самым изолировал от рядовых школ. Вузы, институты АПН СССР варятся в собственном соку.

— А «Альтернатива» делает программы для «Ямах»?

— Нет. Мы всегда считали «Ямаху» одним из проявлений отрыва аппарата от народа и с самого начала отказались работать на пополнение ее программного обеспечения. Работая до недавнего времени при кабинете

информатики Алтайского краевого ИУУ, даже уговорили руководство ИУУ и ГУНО не брать «Ямахи» в ИУУ, а отдать в школу, чтобы не отрываться от учительских масс. К сожалению, этот поступок остался, насколько мне известно, единственным в своем роде.

— А не могут ли помочь школе в разработке программ разнообразные кооперативы, центры НТТМ, малые и совместные предприятия?

— Как сказать... СП, например, работают исключительно с IBM-совместимыми компьютерами. Огромное число кооперативов и центров НТТМ, создававшихся как центры передовой технической мысли, постепенно скатились к перепродаже такой же техники и программного обеспечения для нее, украденного за рубежом.

В то же время ведутся и разработки для школьной техники. В «ИНФО» постоянно публикуются рекламные объявления о продаже ППС. Но в большинстве случаев это объявления кооперативов, лишь недавно занявшихся подобной деятельностью. И второе объявление появляется редко. Причина проста: удержаться на рынке ППС, получить прибыль для новых капиталовложений нелегко. Нужно хорошо знать специфику школы, не гнаться за высокой прибылью, иметь хорошую репутацию. Поэтому большинство кооперативов и ИП быстро прекращают попытки создавать программы для школ, успех, однако, поднять рекламный шум, создающий впечатление наличия на рынке большого числа ППС. При ближайшем рассмотрении эти разработки оказываются весьма сырыми, а то и просто незавершенными.

— Но на рынке ППС присутствуют и весьма крупные организации, например Казанский научно-учебный центр.

— Качество программ, тиражируемых Казанским НУЦ и распространяемых через магазины наглядных пособий, ниже всякой критики. Они, как правило, методически слабы, технически недоработаны, а то и просто не отлажены. Причиной этого, видимо, являются условия приема программ на тиражирование. Для борьбы с такой ситуацией можно порекомендовать кабинетам информатики республиканских и областных ИУУ соз-

давать у себя экспертные комиссии для оценки новых поступлений в магазины наглядных пособий, чтобы в школы не попадала явная калтура. Неплохо бы и Гособразованию СССР разобратся, почему столь слабые разработки проходят-таки через его экспертов.

— А может быть, кабинеты информатики ИУУ сами могли бы обеспечить школы добротными ППС?

— Практика показала, что многие кабинеты успешно обеспечивают и широкое внедрение информатики в школы, и снабжение школ программным обеспечением, однако здесь очень многое зависит от работников, точнее даже от заведующего кабинетом. Он должен быть грамотным специалистом, инициативным, настойчивым. А если этого нет... В целом авторитет системы ИУУ невысок, и это, видимо, не случайность.

Сама жизнь требует предоставления кабинетам информатики ИУУ юридической и финансовой самостоятельности, преобразования их в малые предприятия, региональные центры информатики. Только это позволит решить вопросы финансирования, комплектования программных фондов, обеспечения качественным оборудованием, привлечения квалифицированных специалистов. Должна возникнуть целая индустрия школьной информатизации, чтобы решить имеющиеся проблемы.

Немало подобных центров уже создано. Мы готовы оказывать им техническую, методическую и организационную помощь.

— Что вы посоветуете нашим читателям в условиях складывающегося рынка программного обеспечения?

— Прежде всего сразу осознать, что приходящее с завода ПО подлежит немедленной замене.

При анализе рекламы, появляющейся на страницах периодических изданий, нужно учитывать несколько моментов. Прежде всего категорически не советую приобретать программы, написанные на Бейсике. Как правило, это непрофессиональные вещи, написанные наскоро. Кроме того, рекомендую обращать внимание на авторитет разработчика.

— И к каким же разработчикам можно относиться с доверием?

— На мой взгляд, к «Инфомиру», Центру НЦ ПСО (обе организации расположены в Москве).

«Инфомир» известен различными вариантами Е-практикума для «Корвета», «Ямахи», УКНЦ, текстовыми редакторами «Микромир». Их слабая сторона — склонность везде и всюду создавать маленький «мирок», отгородившись от совместимости с другими разработками.

Наиболее известные разработки НЦПСО — система LOGO, текстовый редактор WRITER, пересылщик KLNЕT для УКНЦ. Слабая сторона — высокие цены и малопривлекательные для заказчика условия поставок.

Ну а как на этом фоне смотрится «Альтернатива» — решайте сами.

— Что ж, расскажите, что сделано у вас.

— Мы с самого начала сделали ставку на создание не разрозненных программ и даже не систем программ, а комплексных пакетов, призванных решить сразу все насущные проблемы потребителей. Пакеты включают техническую и методическую документацию, позволяющую пользователям самостоятельно осваивать их. По желанию заказчиков пакеты комплектуются джойстиком, упрощающими работу с программами и сберегающими клавиатуру.

Начинали мы с интегрированного пакета по информатике для КУВТ-86, сделанного в кодах и одинаково подходящего для классов с Бейсиком и Фокалом. Состав пакета соответствует программному обеспечению профессиональной ПЭВМ — текстовые, графические, музыкальные редакторы, базы данных и т. д. Затем началась разработка обеспечения для «Корвета», УКНЦ (МС 0511), УКНЦ-01 (БК-0011).

— Расскажите об этом подробнее, пожалуйста. Для начала о «Корвете».

— Под названием «Корвет» скрываются разные ПЭВМ. Машины первых выпусков программно несовместимы с более новыми, так как у них изменен принцип обработки сигналов драйвером клавиатуры. Чтобы избавиться от этого и других недостатков, нам пришлось отказаться от штатной ОС МикроДОС и адаптировать для сети ОС СР/М. Наша ОС, будучи пересланной на РМУ, перехватывает обращения к дискуводу и адресуется их сети, так что РМУ получает такие же возможности работы, в том числе с дисками, как и РМП. Кроме новой ОС, для «Корвета» сделан аналог Norton Commander. Это делает КУВТ значительно более удобным для работы. Справедливости ради скажу, что нам известны и другие разработки аналогичного типа, но свою разработку мы ценим все же выше, потому что у нас на РМУ пересылается полная СР/М, под которой выполняются все рассчитанные на эту ОС программы.

Различие РМП и РМУ только в разных сетевых драйверах и том, что на РМП стоит сетевой монитор, обрабатывающий запросы. Для РМУ становится доступным Бейсик РМП, содержащий значительно меньше ошибок, чем ПЗУшный. Но программировать

мы на нем не рекомендуем. В наш пакет входит Турбо-Паскаль, соответствующий версии 4.0 Турбо-Паскаля для IBM PC. Именно этот язык должен использоваться на уроках и для «самостоятельного» программирования.

Имеется в нашем пакете и Пролог. Есть, естественно, текстовый редактор, база данных, графический и музыкальный редакторы. Такой набор — следствие нашей стратегии. Мы поставили цель для каждого вида техники создать программный пакет, включающий полный набор программного обеспечения современной ЭВМ: операционная система, обязательно пересылаемая по сети, база данных, текстовый, графический и музыкальный редакторы, клавиатурные тренажеры, языки высокого уровня (наиболее нужными сейчас мы считаем Пролог и Паскаль).

— А Си?

— Над Си мы работали пока только для КУВТ-86, он уже готов к поставке. Начали капитальные разработки для УКНЦ, но они пока застопорились. Си мы не отвергаем, но на все сразу сил не хватает.

— Пора, я думаю, перейти к УКНЦ.

— Для УКНЦ 02.02 (МС 0511) мы сделали полный комплект, включающий Турбо-Паскаль, транслирующий напрямую в коды, Пролог, аналог Norton Commander, ОС типа RT-11 (естественно, пересылаемую по сети) с повышенным быстродействием, обеспечивающую кэширование памяти при сетевых обменах.

КУВТ УКНЦ-01, т. е. классы на БК-0011, выпускались двух типов — на БК-0011 и БК-0011М, несовместимых между собой. БК-0011 является механическим наращиванием БК-0010. В нем увеличен объем ОЗУ, организованного странично, и переписан системный монитор, причем монитор БК-0011 (старого) содержит крупные недоработки, что затрудняет его использование. Первое, что мы хотим сделать для этих КУВТ, — адаптировать их к пакету «Альтернатива» для КУВТ-86, чтобы, пока суть да дело, можно было использовать все богатство программного обеспечения, наработанного для БК-0010. Для этого на центральной машине ставится пересылщик, «закидывающий» на РМУ эмуляторы компьютера БК-0010. Этим достигается полная совместимость с КУВТ-86, но возможности БК-0011 используются далеко не полностью. К сожалению, других возможностей быстро закрыть зияющие дыры в программном обеспечении УКНЦ-01 мы не видим.

— Все упомянутые вами программы предназначены в первую очередь для курса информатики и обучения программированию. А что делается для других предметов?

— Здесь есть определенное противоречие. Как правило, методисты, знающие, чему

и как учить, не являются квалифицированными программистами. Программисты же обычно не берутся за сугубо обучающие программы. Выход я вижу только один: программисты должны выполнять свою часть работы, создав специализированные авторские системы, позволяющие методистам разрабатывать высококачественные программные продукты. Такого рода системы известны на IBM PC, и мы считаем их совершенно необходимой частью программного обеспечения любого КУВТа. В качестве первого шага мы разработали систему для КУВТ-86, она называется «Автор». Система безязыковая в том смысле, что в ней не используется язык программирования. Входной информацией для нее является совокупность кадров — текстовых (создаваемых текстовым редактором), графических (создаваемых графическим редактором), кодовых (это могут быть программы на ассемблере или музыкальные вставки, созданные музыкальным редактором). Все кадры записываются на диск и компонируются специальной программой — линкером. В итоге получается кодовая программа, способная выполняться без всякой поддержки, вполне самостоятельно. Практика первых ее поставок и испытаний в школах показала, что учителя быстро ее осваивают.

— Самой существенной характеристикой подобных систем являются способы задания вопросов и получения и анализа ответов.

— У нас вопросы задаются в текстовом виде, а ответы принимаются в виде указания курсором определенного места экрана. Так можно выбрать и строку текста из меню, и точку на географической карте, нарисованной на экране. Расчет на весьма малоопытного пользователя, не владеющего клавиатурой.

Педагогическим коллективам на местах, имеющим серьезные методические идеи, мы готовы на льготных условиях предоставить наши программные средства и заключить договоры на создание ими по нашему заказу программных пакетов обучающих программ по различным предметам.

— Ну, и последнее: какой вы видите перспективу развития КУВТов?

— Я хорошо понимаю, что нам не избежать грядущего перехода на IBM-подобную ВТ. Известно, что основные заводы — производители школьной техники разрабатывают модели КУВТов, совместимые с IBM PC. Но переход должен произойти не прежде, чем цены на такую технику опустятся до уровня, сравнимого с сегодняшними ценами на терминальные классы. А существующая сейчас техника должна еще немало пора-

ботать и уйти достойно, оставив о себе хорошую память.

Готовиться к грядущим переменам уже пора, и мы вложили средства в разработку программ для IBM PC, в первую очередь авторской системы.

— Кстати, как вы относитесь к наделавшему немало шума советско-американскому проекту «Пилотные школы»?

— Неоднозначно. С одной стороны, он может (и должен!) «поторопить» становление в нашей стране индустрии школьной информатики и внедрение IBM-совместимой техники. С другой — чрезвычайно высокая цена установки классов, неудобство для потребителей многих условий установки, обучения и обслуживания, отсутствие локальной сети и, наконец, чрезвычайно убогий для такого проекта комплект программного обеспечения: ОС, текстовый и графический редакторы, авторская система — и все.

Было бы нормально, если бы IBM-овская техника потеснила отечественные КУВТы в открытом экономическом и методическом соревновании. Пока же создается впечатление, что движущая сила проекта — амбиции некоторых руководителей в центре и на местах. В то же время очень многие КУВТы используются далеко не в полную меру их возможностей, в первую очередь из-за недостаточного ответственного отношения их хозяев к подбору программного обеспечения. Улучшение использования КУВТов путем приобретения качественных программ в десятки и сотни раз дешевле приобретения новой техники, а эффект (по край ей мере на первых порах) — не меньший. Для эффективного использования больших возможностей PS/2 необходимы опыт и подготовка. Не следует пытаться перескакивать через естественные этапы эволюции. Другое дело, что пройти через них нужно быстро.

Вниманию всех учителей и руководителей учебных заведений, использующих терминальные классы на основе БК-0011 и БК-0011М

Научно-техническое предприятие «АЛЬТЕРНАТИВА» при НИИ ИВТ АПН СССР предлагает пакет учебных программ «АЛЬТЕРНАТИВА-БК-0011»

В состав пакета входят:

сетевой монитор, обеспечивающий быструю пересылку программ и их связь с дисководом;

текстовые редакторы, базы данных, графические, музыкальные редакторы, графический конструктор, электронные таблицы, пакет игровых программ;

подробная техническая и методическая документация на все программы.

Гарантируются:

высокое качество программ и пересылаемость их по сети связи класса. Все программы написаны в кодах или на языке ассемблера; обеспечение на программном уровне совместимости с кодовыми программами класса КУВТ-86 (на БК-0010);

работоспособность пакета на классах из БК-0011 и БК-0011М без всякой переделки машин;

низкая стоимость пакета (для учебного заведения порядка 1000 рублей перечислением).

**Заявки просим присылать по адресу:
656049, г. Барнаул, а/я 3475, НТП «АЛЬТЕРНАТИВА».**

Вербальный компонент и стратегия школьной информатики (реплика гуманитария)

Информационная культура легко расчленяется на центр и периферию использованием оппозиции «с помощью» — «без помощи ЭВМ». Периферия была и будет для центра поставщиком идеалов и идей; центр же стремится освоить периферию, что проявляется не только в компьютеризации, но и в распространении информационного взгляда на мир, уже сейчас заметном в гуманитарных областях.

Центр находится в ведении молодой науки информатики, получившей в школе постоянное, надо думать, представительство. Периферией же ведаёт прагматическая филология (риторика, герменевтика и их современные аналоги), развивавшаяся в течение веков, но представленная в школе разрозненными элементами, погребёнными в курсах языка и литературы, и факультативами.

Очевидно, что связь периферии и центра должна быть отображена на школьный курс в виде межпредметных связей между информатикой и прагматической филологией. Причем речь идет не о заурядной связи, которая соединяет информатику с любой другой наукой, но о пучке связей, которые, образно говоря, обеспечивают прямой доступ к информатике, так как проблемы передачи, обработки и накопления информации возникают в недрах филологии и все «искусственное» рождается по той или иной аналогии с «естественным».

Базовым элементом связи «центр-периферия» является отношение «искусственный знак — вербальный знак». Отсюда вытекает исключительная важность развития на уроках информатики не просто логического, но вербально-логического мышления, под которым здесь понимается способность вычленять в речи ее логическую основу. Не

стремиться сделать свою речь абсолютно «строгой» (задача нереальная по отношению к естественному языку, где лексическое значение не сводится к десигнату, аргументация — к логике и т. д.), а уметь постоянно производить как бы логическое вычитание из речи того, что «строгое», и анализировать то, что в остатке.

В этом смысле полезна та часть школьной математики, которая включает вербальный компонент (дефиниции, формулировки задач, теорем, доказательств). Парадокс состоит в том, что алгебраизация мышления «съедает сама себя», приводит к тому, что школьник больше манипулирует искусственными знаками по правилам синтаксиса, чем вдумывается в их семантику.

Очень слабо в отношении развития вербально-логического мышления работают школьные дисциплины филологического цикла. Пример тому — школьное сочинение, призванное, казалось бы, развивать вербальное мышление. Но школьное сочинение как жанр совершенно лишено прагматики (его реальная прагматика — понравиться учителю — не формулируется и не анализируется, другая же просто не постулируется), и совершенно непонятно, какую коммуникативную цель преследуют миллионы детей, рассуждая о литературных героях. Введение риторики, разумеется, изменило бы картину к лучшему.

Осознание стратегической роли вербального компонента в курсе школьной информатики сделает этот курс максимально универсальным и гибким и, возможно, послужит ключом к разрешению ряда спорных вопросов этого курса.

Т. ХАЗАГЕРОВ,
Ростов-на-Дону

ППП «Телец»

Пакет содержит полный анализ функций по программе школы и может быть использован на лабораторных и практических занятиях в технических учебных заведениях для исследования зависимостей, полученных как теоретическим, так и опытным путем.

В состав ППП «Телец» входят семь программ обработки и программа «меню», позволяющая пользователю выбирать в диалоговом режиме интервал изменения аргумента, число отсчетов функции, вид задания функции (формулой или таблично).

На экране справа появляется список программ обработки, оформленный в виде отдельных разноцветных рамок, и слева — график заданной функции. Перемещением курсора, изменяя режим рамки, вы можете выбрать одну из программ обработки:

102 *«Максимальное и минимальное значения»* — путем перебора всех значений функции по точкам отсчетов дает наибольшее и наименьшее значения функции в заданном интервале;

«Интеграл» — программа нахождения интеграла функции по методу Симпсона для заданного интервала измерения;

«Таблица» — выводит таблицу значений аргумента и функции (по десять значений) из заданного количества отсчетов;

«Нули» — с заданной точностью определяет значение аргумента, где функция обращается в нуль;

А. БЛОХ, С. ЯЩУК

Информатика в школе на основе программируемых микрокалькуляторов

Перейдя в массовой школе от счетов и логарифмической линейки к компьютеру, мы незаслуженно отбросили программируемый калькулятор (ПМК), так и не ощутив его преимуществ. По своим функциональным возможностям программируемый калькулятор сравним с персональной ЭВМ. Он широко используется при расчетах, не требующих больших объемов памяти. Небольшая память ПМК в определенном смысле является преимуществом при обучении, так как заставляет разрабатывать наиболее рациональные алгоритмы.

По многим причинам начинать изучение информатики в школе предпочтительнее на базе ПМК. Одна из них заключается в том,

«Аппроксимация» — осуществляет кубическую сплайн-аппроксимацию функции, заданной таблично или формулой. Программа запрашивает, что мы хотим получить — график или значение функции («g» или «y»). Если «y», то для любого значения аргумента из заданного интервала мы получим значение функции, т. е. результат аппроксимации. Если «g», то программа построит график заданной функции по всем точкам экрана (пикселям) с учетом разрешающей способности данного режима графического экрана; *«Экстремумы»* — определяет МАХ и MIN значения функции в заданном интервале аргумента методом квадратичной интерполяции — экстраполяции;

«Четность» — определяет четность или нечетность функции.

ППП «Телец» написан на Бейсике для КУВТ «Ямаха», адаптирован для КУВТ «Корвет» и по желанию может быть адаптирован для других видов КУВТ.

ППП «Телец» был использован на курсах для учителей математики, физики и информатики в институте усовершенствования учителей.

Пакет распространяется бесплатно.

По всем вопросам обращаться по адресу: Рязань-29, ул. Урицкого, 2а, РОИУУ, кабинет ЦЭВМ.

И. ПАНОВА

что в средней школе основным объектом вычисления служит формула. При программировании на ПМК ученику необходимо тщательно исследовать ее структуру, что совершенно не требуется при использовании алгоритмического языка: там формула попросту переписывается с учетом нескольких простейших правил.

Необходимый при использовании ПМК анализ структур формул обогащает курс информатики и значительно способствует развитию алгоритмического (конструктивного) мышления. Здесь уместно отметить, что формулы с более общей структурой являются базой, на которой строится математическая логика, теория формальных моделей.

После построения программ для формул осуществляется естественный переход к понятию алгоритма, к построению схем алгоритмов. Формируются правила построения программы по схеме алгоритма. В школах с углубленным изучением математики можно включить в программу курса информатики построение программ даже по решению дифференциальных уравнений первого порядка. В массовых школах подобную задачу на моделирование сводят к программированию формулы, пределом которой является дифференциальное уравнение.

Использование программируемых калькуляторов и компьютеров в школе — это не взаимоисключающие, а дополняющие друг друга возможности при изучении как информатики, так и других школьных предметов. Изучение начал информатики на базе ПМК хорошо подготавливает учащихся к последующему освоению одного из алгоритмических языков и использованию ПЭВМ. Разумеется, в тех школах, где компьютеров нет, весь курс информатики основывается только на ПМК. Конкретная разбивка часов и тематика зависят от типа школы, подготовки учеников, количества компьютеров.

Последнее обстоятельство еще долго будет играть существенную роль, учитывая сла-

бые финансовые возможности народного образования и высокую стоимость вычислительной техники. Ведь, например, класс «Корвет» из 12 ПЭВМ даже без учебного программного обеспечения стоит 33 тыс. рублей. На эти деньги школа может приобрести более 350 программируемых калькуляторов. А ведь нужно еще иметь средства для ремонта и обслуживания ПЭВМ, для приобретения учебных программ.

Один дисплейный класс, оборудованный 12 компьютерами, явно недостаточен для школы. С учетом его использования для курса информатики на долю других дисциплин остается очень мало времени — примерно на 8—10 уроков в год, что, по существу, означает безмашинный вариант преподавания. Все это является дополнительным аргументом, говорящим о целесообразности широкого применения ПМК в учебном процессе.

Актуальность и важность курса информатики в системе среднего образования не вызывает сомнений. Однако его введение не было ни хорошо продумано, ни материально обеспечено. Курс оказался в кризисном состоянии. Отсюда — наблюдающееся разочарование и, более того, тенденции к его ликвидации, чего, на наш взгляд, никак нельзя допустить.

103

УКНЦ в Малой академии наук

Секция информатики и программирования свердловской Малой академии наук занимается обучением школьников и разработкой программного обеспечения для школ и внешкольных учреждений. Вот уже два года пользуемся КУВТ УКНЦ. Накопили много материалов по технической и программной части: описания ОС и драйверов устройств, программы с ДВК и БК-0010 (адаптированные для УКНЦ) и программы, разработанные учащимися нашей секции.

Освоение УКНЦ начинали с освоения Бейсика, как большинство пользователей, но он не позволяет эффективно распоряжаться и без того малыми ресурсами ПЭВМ и к тому же прививает плохой стиль программирования.

Мы перешли на Паскаль. На этом этапе учащимися секции были созданы первые серьезные программы: графическая библиотека Паскаля, учебный графический редактор, редактор спрайтов, редактор шрифта знакогенератора, позволяющий создавать и

использовать свои собственные шрифты, кодировщик функциональной клавиатуры и т. д. Паскаль оказался мощным средством, но не без недостатка: при трансляции к программе добавляется существенный довесок, избавиться от которого можно лишь средствами ассемблера.

На сегодня транслятор макроассемблера является самым эффективным средством создания программ на УКНЦ, позволяющим максимально использовать все возможности машины. С его помощью учащиеся разработали MultiCommand (резидентная операционная оболочка типа Norton Commander), системные утилиты для обслуживания дискового хозяйства, динамические компьютерные игры и т. д.

Если кто-либо из читателей журнала заинтересуется нашей работой или пожелает с нами сотрудничать — наш адрес: 620014, Свердловск, Малышева, 6, Учебно-методический центр МАН. Телефон: (3432) 51-92-51.

«Агат»?

В последнее время в «ИНФО» появилось несколько публикаций, восхваляющих «Агат», и мне захотелось рассказать о своем опыте использования этой ПЭВМ.

Начну с того, что нашу школу не спросили, какие компьютеры нам нужны (это о нравах системы народного образования), и приобрели для нас семь машин «Агат-9». На эти деньги можно было купить полный КУВТ УКНЦ или «Корвет»!

Из семи исправными оказались три. Только три! И начались перипетии с ремонтом. Описывать их подробно — занимать полжурнала, поэтому упомяну только основные моменты: неверно указанный в паспорте машины адрес Новосибирского ремонтного завода ВТИ; отказ этого завода заняться нашими машинами раньше, чем через год; попытка гарантийного ремонта в Барнауле, не удавшаяся из-за отсутствия запчастей (не поставил Волжский завод ЭВТ); обращение в кооператив, ремонтирующий *все* компьюте-

ры кроме, как выяснилось, «Агатов»; согласие районного вычислительного центра помочь нам, если мы достанем схему «Агата-9» и ЗИП; неоднократные обращения по этому поводу на Волжский завод ЭВТ (в рекламном листке, пришедшем в роно, завод обещал выслать ЗИП), оставшиеся без ответа... Гарантийный срок тем временем кончился, четыре машины стоят, схемы нет, запчастей нет...

Помогите, кто может!!! Должен же где-то существовать этот таинственный документ — «Схема принципиальная электрическая ПЭВМ «Агат-9»!»

А. НИКИТИН

658564, Алтайский край, Мамонтовский р-н, с. Крестьянка, средняя школа

Р. С. Наглядевшись на наши страдания, краевой отдел народного образования отказался от получения «Агатов». Что ж, хоть другим облегчили жизнь...

104

Откликнитесь, биокибернетики!

Применение информатики в любой сфере народного хозяйства ведет к получению высоких результатов, достижению ранее недоступных целей. Относится это и к медицине, здравоохранению. Математические разработки, существующие в практике и позволяющие оценить в информационном плане патологический процесс в организме на основании количественных мер, достаточно точно дифференцируют заболевания, позволяют прогнозировать состояния, предлагают более оптимальное решение в сложной ситуации. Поэтому создание секции медицинской и биологической кибернетики является актуальной и необходимой задачей в наше время. И хотя наша секция еще молода, она дает о себе знать стабильными результатами. Наши школьники занимают призовые места в республиканских съездах Малых академий наук, студенческих научных конференциях.

А в мае этого года учащийся из нашей секции занял первое место по информатике во Всесоюзном сборе научных обществ учащихся, проходившем в Ставрополе. Но нам немного обидно, что на республиканском и всесоюзном уровне мы не можем найти такую же секцию. Поэтому мы просим откликнуться организацию, секцию или кружок, которые занимаются такими же вопросами. Мы готовы переписываться и поделиться своим опытом, методическими материалами, программными средствами.

Наш адрес: 286016, Винница, ул. 40-летия Победы, 4а, областная станция юннатов, секция медицинской и биологической кибернетики.

А. ХОШАБА

- Прошу помочь с педагогическими программами для «Ямахи-2»
261010, Володарск-Волынский, ул. Петровского, 17, кв. 5. Т. А. Николаенко.
- Группа пользователей ПК «Сура» (ПК-8000) хочет переписываться с владельцами этих компьютеров; купит или обменяется программами.
183070, Мурманск, а/я 1276.
Е. Н. МЫСКОВ.
- Хочу переписываться с владельцами «Коммодора-64». Имею большой выбор программ для обмена и безвозмездно.
143000, Красногорск-4, а/я 54.
В. МОСКОВЦЕВ.
- Хотел бы переписываться с пользователями «Электроники-85» (МС 0585). Для обмена предлагаю обучающие и контролирующие программы.
115551, Москва, Каширское шоссе, 102/2, кв. 331. С. ПАНЧЕНКО.
- Пользователям ПК «Сура» и «Веста» предлагаю программное обеспечение и информацию. Заинтересован в обмене.
603163, Нижний Новгород, ул. Бринского, 4, корп. 1, кв. 30. М. И. НАЗАРОВ.
- Приглашаю к обмену владельцев ПК-01 «Львов». Имею около 100 системных и игровых программ.
284019, Ивано-Франковск, пл. Освобождения, 7, кв. 44. С. И. ПУШКАР.
- Прошу откликнуться пользователей ПК «Криста», вторая модель. Предлагаю обмен ПО и т. п.
163045, Архангельск, просп. Ломоносова, 282, корп. 1, кв. 49. М. СИСИН.
- Предлагаю переписку и сотрудничество всем, занимающимся разработкой развитого ПО для БК-0010 (-01), коллекционированием и обменом программами (мой архив более 5000 программ), доработками БК (расширение памяти, контроллер для НГМД и др.).
196191, Ленинград, ул. Варшавская, 31, кв. 46. В. А. СОРОКИН.
- Хотел бы переписываться с владельцами ПК «Специалист». Имею широкий выбор игровых и системных программ.
346826, Ростовская обл., с. Куйбышево, ул. Восточная, 3, кв. 4.
А. А. ШВЕДЕНКО.
- Буду благодарен тем, кто поможет в подключении принтера МС 6312 к компьютеру ПК-01 «Львов».
603062, Нижний Новгород, ул. Горная, 32, кв. 97. Ю. П. КРИУЛЕВ.
- Ищу схему подключения «Корвета» к телевизору ЗУСЦТ-51-16.
422400, Татарстан, г. Буинск, ул. Интернационала, 139, СПТУ-56.
Р. Н. ДУНАЕВ.
- Хотел бы переписываться и обмениваться программами с пользователями БК-0010-01. Имею много игровых и прикладных программ, в том числе трансляторы Форты, Си, Паскаля.
420138, Казань, ул. Дубравная, 23, кв. 304. Р. АЙВАЗОВ.
- Всех обладателей КУВТ-86, а также индивидуальных пользователей БК-0010 просим откликнуться для обмена программами и текстовыми материалами.
357359, Ставропольский край, Предгорный р-н, с. Юца-1, ул. Овражная, 22. Д. Б. ГЛАДЬКО.

Поправка. В статью С. Комарова, М. Монина «Про Бейсик, быстродействие и кое-что еще» («ИНФО». 1991. № 3. С. 32) вкрались опечатки. В первой программе (с. 33) в строке 50 должно стоять $R=R+RND(1)$; в третьей (с. 34) в строке 50 запятая перед закрывающей скобкой — лишняя.

Всесоюзный конкурс «Юный программист» («Старшие для младших»)

Ассоциация учителей информатики и Главное управление народного образования Красноярского крайисполкома проводят конкурс «Юный программист» на лучшую программу, созданную школьниками для детей младшего возраста.

На конкурс принимаются клавиатурные тренажеры, демонстрационные, обучающие, контролирующие программы по предметам, исполнители, игровые программы и т. д., предназначенные для детей 5—13 лет.

При разработке программ следует ориентироваться на школьные персональные ЭВМ.

Для участия в конкурсе необходимо до 1 февраля 1992 г. прислать заявку и подробный сценарий, содержащий основ-

ные идеи и показовое описание программы. Сценарии с орфографическими ошибками не рассматриваются.

Авторы лучших сценариев будут приглашены в г. Красноярск в апреле 1992 г. для показа своих программ на заключительном туре конкурса.

Победителей ждут призы: радиоприемники, фотоаппараты, часы, библиотечки литературы по информатике, персональные приглашения в летние школы.

Лучшие сценарии будут опубликованы, лучшие программы будут занесены на Дискету Почета.

Материалы направлять по адресу: 660051, г. Красноярск, а/я 11533, Оргкомитет конкурса «Юный программист».

ЗАЯВКА

Ф. И. О. _____

Домашний адрес _____

Номер и адрес школы (СПТУ) _____

Класс _____

Тип и конфигурация машины (точное наименование модели компьютера, характеристика дисководов и т. д.) _____

Язык программирования _____

Название программы _____

Кому и для чего предназначена _____

Объем программы (К байт) _____

Среднее время работы программы (мин) _____

Ф. И. О. учителя информатики (руководителя кружка, преподавателя УПК), оказавшего помощь в работе над программой, обучении программированию _____

Всесоюзный семинар в Москве

В марте 1991 г. в Москве проходил VI Всесоюзный семинар «Разработка и применение программных средств ПЭВМ в учебном процессе», организованный ИПИ АН СССР. В семинаре приняли участие представители АН СССР, научно-исследовательских, учебных и промышленных организаций. К началу семинара был выпущен сборник докладов.

Результаты исследований и разработок в области учебной информатики, полученные с 1989 по 1990 г., были доложены на заседаниях пяти секций: «Тенденции информатизации образования» (рук.— С. А. Христочевский), «Базовые программные средства систем учебной информатики» и «Прикладные программные средства учебного назначения» (рук.— Р. С. Агамирова), «Инструментальные средства учебной информатики» (рук.— А. А. Федосеев), «Психолого-педагогические проблемы компьютеризации обучения и контроля знаний» (рук.— Г. Н. Александров).

В ходе семинара оценивались перспективные направления дальнейших работ. Специфика семинара заключается именно в том, что его участники, являясь представителями самых разнообразных областей деятельности, объединены использованием информационных технологий в образовании.

К VI семинару изменились условия его проведения. Во многих учебных заведениях появилась разнообразная вычислительная техника. Все большее распространение получают IBM PC-совместимые ПЭВМ.

Участники отметили большую и целенаправленную работу, проводимую:

в Северо-Осетинском государственном университете (рук.— профессор Г. Н. Александров) по общим и прикладным психолого-педагогическим проблемам компьютеризации и контроля знаний;

в Свердловском инженерно-педагогическом институте по исследованию психофизиологических и санитарно-гигиенических проблем использования ПЭВМ в учебном процессе;

в Институте проблем информатики АН СССР по разработке инструментальных средств для создания ППС;

в Курганском педагогическом институте по разработке архитектурных систем автоматизированного контроля знаний.

В то же время, несмотря на определенные успехи в исследовании теоретических основ обучения с использованием ПЭВМ, в решении психолого-педагогических проблем компьютеризации обучения и контроля знаний, серьезным недостатком является слабая разработанность общей психологиче-

ской теории обучения, которая была положена в основу технологии компьютерного обучения;

весьма невысокий уровень экспериментальных исследований в этой области, недостаточная координация усилий конструкторов, математиков, программистов, психологов, педагогов в решении задач оптимального конструирования ПЭВМ, разработки качественных обучающих программ, программ для контроля знаний;

большинство ППС разрабатывается в настоящее время для IBM PC, хотя ПЭВМ предыдущих поколений будут оставаться в школе еще 5—10 лет, в связи с этим необходимо продолжать для них разработки;

все большее значение приобретает инструментальный подход к созданию ППС, тем не менее инструментальных средств по-прежнему не хватает и затраты на разработку ППС все еще слишком велики.

Участники семинара рекомендовали: считать целесообразным и полезным периодическое проведение настоящего семинара, просить ИПИ АН СССР рассмотреть возможность преобразования всесоюзного семинара в международный;

рекомендовать Общесоюзному научно-методическому совету информатизации образования выпускать информационный бюллетень о проводимых в стране семинарах и конференциях по проблемам информатизации образования;

включить в программу следующего всесоюзного семинара сообщения о влиянии занятий, проводимых в дисплейных классах, на физиологическое состояние обучаемых; просить ИПИ АН СССР организовать координацию разработки инструментальных средств на основе объединяемых функциональных программных модулей;

призвать заинтересованные организации провести работу по систематизации ППС (выпуск каталогов, обзоров по типам ПЭВМ, разработка критериев оценки ППС и соответствующая аттестация их);

рекомендовать разработчикам поставлять вместе с ППС методическую литературу по их применению;

в качестве перспективных направлений для проведения исследований и разработок в области создания ППС считать:

средства компьютерного моделирования; экспертные системы для целей обучения; авторские системы на основе языков логического программирования и объектно-ориентированного подхода.

Омская конференция по проблемам информатизации образования

23—24 мая 1991 г. в Омске состоялась VIII республиканская научно-практическая конференция «Новые информационные технологии в учебном процессе и управлении», организованная Республиканским центром НИТО Минобразования РСФСР, действующим при Омском пединституте. В работе конференции приняли участие ученые и преподаватели педвузов и средних учебных заведений Российской Федерации, а также представители других регионов страны: Казахстана, Узбекистана, Литвы и др. Омск представляли преподаватели средних и высших учебных заведений, работники народного образования, ученые академических учреждений.

На конференции работало 5 секций: «Общие вопросы применения НИТ в учебном процессе» (подсекции: «Психолого-педагогические вопросы информатизации в сфере образования», «Санитарно-гигиенические вопросы применения ЭВМ в учебном процессе»), «Проектирование и разработка ППС», «Изучение и использование ЭВМ в средней школе» (подсекции по проблемам использования НИТ в преподавании математики, физики, филологических дисциплин, а также в обучении младших школьников), «НИТ в подготовке специалистов» (подсекции по проблемам подготовки учителей информатики, учителей других специальностей, специалистов технического профиля), «НИТ в научных исследованиях и управлении». Всего было заслушано более 150 докладов.

С основным докладом на пленарном заседании конференции «Информатика и технология: компоненты педагогического образования» выступил научный руководитель РЦ НИТО, председатель секции педкадров НМС информатизации образования ГКНО СССР, проректор по научной работе Омского пединститута, профессор М. П. Лапчик. Наряду со всесторонним анализом состояния и перспектив компьютеризации школы в докладе изложены требования к информационной и технологической грамотности современного учителя, а также вытекающие из этого направления совершенствования учебных планов педвузов. С докладом «Математика — проблемные области — информатика: стратегия взаимодействия» выступил директор Института информационных технологий и прикладной математики СО АН СССР В. А. Шапцев.

Большой интерес участников конференции вызвал доклад зав. лабораторией ВЦ СО АН СССР (г. Красноярск), доктора физ.-мат. наук А. Н. Горбаня о нейрокомпьютерах. Другие пленарные сообщения касались проблем информатизации Омской области и, в частности, школьного образования (зам. зав. Омского облУНО В. М. Благинин, председатель правления Омского отделения ВОИВТ профессор В. Г. Осипов, зав. кафедрой АСУ ОмПИ, доктор техн. наук, профессор В. И. Потапов).

Среди докладов, вызвавших наибольший интерес, можно выделить следующие: «Программно-методический комплекс по использованию ПЭВМ в учебном процессе» (Ю. С. Брановский, Ставропольский ПИ), «Гемодинамика и временная организация сердечного ритма подростков, обучающихся на ЭВМ» (В. А. Глотов и др., Омский ПИ), «К вопросу о языках и исполнителях в начальном обучении программированию» (В. А. Беркуцкий, Павлодарский ПИ).

Представители педвузов в очередной раз выразили недоумение по поводу политики центральных органов в вопросе реализации проекта «Пилотные школы», при которой педвузы (осуществляющие подготовку учителей!) не оснащаются современной техникой и технологией, направляемой сейчас напрямую в школы; крайне необходимы современные учебно-методические пособия и соответствующие пакеты ППС по всем направлениям информатизации школьного образования (курсы информатики начиная с младших классов; факультативы и специализации по вопросам, связанным с НИТ; школьные предметы или их отдельные разделы и т. д.).

В ходе работы конференции участники имели возможность познакомиться с фондом программных и методических средств РЦ НИТО (ФПМС в настоящее время отражен в трех выпусках каталога), передать в ФПМС имеющиеся разработки и приобрести заинтересовавшие их средства.

Участники конференции получили сборник тезисов докладов (Новые информационные технологии в учебном процессе и управлении: Тез. докл. VIII Республ. научно-практ. конф., 23—24 мая 1991 г., Омск/Редкол.: М. П. Лапчик (отв. ред.) и др. Омск, РЦ НИТО при Омском пединституте, 1991. 193 с.).

Следующая Омская конференция по проблемам информатизации образования состоится в мае 1992 г. Оргкомитет приглашает к участию в ней все заинтересован-

ные организации и лица. Адрес: 644099, Омск, наб. Тухачевского, 14, пединститут, РЦ НИТО.

О. Н. ЛУЧКО

Конференция в Нальчике

В июне 1991 г. в г. Нальчике состоялась Всесоюзная научно-практическая конференция «Использование новых информационных технологий в системе общего среднего образования». На ней присутствовали представители республиканских институтов усовершенствования учителей Азербайджана, Кабардино-Балкарии, Казахстана, России, Узбекистана, Украины. Ниже публикуется ее решение.

Решение

Конференция отмечает, что к настоящему времени:

сформировалось ядро школьного курса «Информатика» для X и XI классов средней школы;

создана система учебников, а также программное обеспечение к этим учебникам, ориентированное на все основные типы ЭВМ, используемые в школе;

подготовлены кадры учителей, способные проводить обучение по машинному варианту курса.

Фактически создана база для стабильного преподавания курса «Информатика».

В то же время эффективному преподаванию курса мешает ряд нерешенных на сегодняшний день проблем.

1. Ненадежность техники, поставляемой в школы.

2. Разнородность техники по основным характеристикам (объем оперативной памяти, быстродействие и т. д.) и несовместимость программного обеспечения.

3. Слабая методическая вооруженность учителя информатики, в особенности дидактическими материалами к урокам.

В связи с этим конференция рекомендует:

1. Создать стандарт по основным характеристикам персональных ЭВМ, их комплектности и надежности и запретить поставки техники в школы, не отвечающей этому стандарту.

2. Зафиксировать список основных типов техники, используемой в школе, и обеспечить программную совместимость внутри каждого типа.

3. Создать при Ассоциации учителей информатики Фонд дидактических материалов и поддержать этот Фонд в деле его распространения.

4. Включить в повестки дня ближайших конференций вопрос о правовых аспектах создания, тиражирования и использования педагогических программных средств.

Конференция отмечает, что курс информатики является *единственным целенаправленно формирующим алгоритмическое мышление*, т. е. того компонента мышления, который обеспечивает человеку эффективное планирование своей деятельности. Курс информатики является фундаментальным курсом, общезначимой составляющей которого является формирование модельного подхода к познанию окружающего мира, позволяющего адекватно прогнозировать результаты человеческой деятельности. Курс информатики обеспечивает внедрение информационных технологий в учебный процесс, повышая его эффективность.

Конференция считает, что преподавание курса информатики в перспективе должно быть распространено на всю среднюю школу. В качестве начального этапа рекомендуется ввести преподавание информатики с VIII класса, а после проведения соответствующей научно исследовательской работы — с V класса.

ЧТО? МОЖЕТ ЭВМ

Хождение за три моря

Глобальная информационная сеть — это множество различных программ, работающих на разнообразнейших компьютерах (от персоналок до супер-ЭВМ), обменивающихся самыми разными данными (тексовыми, графическими, реально-временными...) по сильно различающимся каналам связи (от телефонных до волоконно-оптических). Чтобы связать воедино это вавилонское столпотворение, Международная организация по стандартизации разработала семиуровневый протокол взаимодействия открытых систем.

В нем описываются параметры, которым должна соответствовать информация на каждом этапе передачи — от уровня 7 (представление данных прикладными программами) до уровня 1 (физическое представление в канале связи). Протокол позволяет производить обмен даже между БК-0010 из любого захолустья (был бы телефон) и пентагоновскими суперкомпьютерами (было бы желание).

Платить за это приходится, однако, дорого. «Спуск» по семи уровням от передающего компьютера и обратный «подъем» на принимающем конце требуют больших преобразований информации, добавления к данным объемных служебных блоков и, таким образом, значительных затрат времени центрального процессора на обслуживание обмена.

Чтобы освободить ЦП от лишней работы, приходится в компьютер встраивать еще один компьютер — интеллектуальную адаптерную плату сопряжения с сетью, процессор которой берет на себя всю «черновую» работу по подготовке информации к пересылке и «очистке» ее после приема. Логическое развитие этого приема привело корпорацию

IBM к разработке так называемой архитектуры блока управления подсистемами (Subsystem Control Block, SCB), позволяющей платам компьютеров взаимодействовать без участия центральных процессоров. Построенные в соответствии с SCB компьютеры, связанные информационной сетью, можно рассматривать как одну многопроцессорную машину, «размазанную» в пространстве. Эффективно и удобно — пользователям, а вот проектировщикам... Каждый компьютер нужно делать с оглядкой на остальные!

Ни жены, ни работы...

В таком грустном положении оказался австралиец Д. Бэрилек, обвиненный в занесении компьютерного вируса в вычислительную систему. Жена ушла, когда суд признал его виновным, а работы он не может найти, так как компьютерная индустрия тесно сплочена и не любит нарушителей. Судом второй инстанции бедолага полностью оправдан (первое рассмотрение дела велось тенденциозно), ему выплачена компенсация за причиненный ущерб, но вопрос с женой остается открытым.

Снова автомобиль и компьютер

На этот раз компьютер вполне обычный, разве что портативный, и автомобилем не управляет ни в каком аспекте. Просто для людей, не способных обойтись без ПЭВМ ни минуты, но ленящихся носить портативный ПК (обычно он весит 6—8 кг), компания LapStop организовала прокат дорожных компьютеров в конторах по прокату автомобилей. Если еще учесть, что во многих гостиницах США вместе с завтраком в номер можно заказать и ПЭВМ, возникает вопрос: а стоит ли покупать портативную модель?

Диплом об окончании ясель

В августе 1990 г. состоялся первый выпуск детей, прошедших обучение по программе Head Start, организованной фирмой Texas Instruments. Четырехлетние выпускники перешли в детский сад.

Эта программа уникальна по возрастной направленности, но не по организационной основе. Многие компьютерные фирмы вкладывают деньги в образование. Именно вкладывают, а не тратят. Вице-президент по НИОКР электронного «кита» Digital Equipment Corporation С. Фуллер говорит: «В конечном счете я думаю, мы получим обратно больше, чем вложим. Если то же самое сделают еще 50 корпораций, мы получим значительную долю лучше образованных ребят, выпущенных начальной школой».

Hewlett Packard (HP) — чемпион в этой деятельности: 65,3 млн. долларов для образования в 1989 г. IBM — 58,1 млн. Другие дают меньше денег, но... Как оценить деятельность DEC, оказывающей щедрую финансовую помощь своим инженерам в переходе на преподавательскую работу? Не дороже ли любых денег готовность поделиться людьми?

HP не только учредила должность менеджера по связям с начальными школами, но и организовала специальные школы, в которых сотрудники фирмы выступают в качестве индивидуальных педагогов-наставников.

В заключение — слова М. Райниса из Texas Instruments: «Это наш город. И в нем мы сталкиваемся с колоссальной проблемой проблемы людей, которые, не получая надлежащего образования, становятся париями общества. Качество жизни города снижается, так как мы не предоставляем того уровня образования, в котором нуждаемся. Я прожил здесь всю жизнь и не хочу, чтобы мой город разорился и погиб».

Коммерческо-промышленная
ассоциация «Электроника»



Предприятие
«Электроника-Маркет»

Программно-музыкальный комплекс «МЕНЕСТРЕЛЬ» для ПЭВМ «Электроника БК-0010(-01)

БК + «Менестрель» = музыка для вас

Такой знакомый и привычный БК! Казалось бы, что нового можно ожидать от него?

«МЕНЕСТРЕЛЬ» обогатит ваш компьютер новыми широчайшими возможностями.

«МЕНЕСТРЕЛЬ» — классный синтезатор + встроенный шестиголосный музыкальный редактор, не имеющие аналогов для советских компьютеров. И профессиональным музыкантам, и не умеющим играть ни на одном музыкальном инструменте «МЕНЕСТРЕЛЬ» предоставляет возможность записать и (!) с помощью входящего в комплекс синтезатора качественно воспроизвести любое музыкальное произведение, в том числе вашего собственного сочинения.

«МЕНЕСТРЕЛЬ» по вашему желанию разнообразит записанные музыкальные формы различными звуковыми эффектами: слитное или раздельное звучание (*legato, non legato*), постепенное замедление или ускорение темпа (*ritenuto, accelerando*), вибрато для отдельного голоса, плавный переход от текущей ноты к следующей (*glissando*); изменит темп воспроизведения; осуществит транспозицию (в диапазоне 13 октав).

«МЕНЕСТРЕЛЬ» превратит ваш компьютер в квалифицированного аккомпаниатора, надежного помощника в обучении и музыкальном творчестве.

Вы привыкли к стереозвучанию? «МЕНЕСТРЕЛЬ» сохранит этот уровень комфортности — вся записанная музыкальная информация может быть воспроизведена со стереоэффектом (два звуковых канала, по три голоса в каждом из них).

«МЕНЕСТРЕЛЬ» прост для освоения, снабжен подробными пояснениями (нажатие на клавишу СТОП в любой ситуации приводит к появлению на экране справочной информации, необходимой в данный момент), открыт для пополнения (дополнительное программное обеспечение). Начав с простейших мелодий, вы легко сможете перейти к сложным музыкальным произведениям.

«МЕНЕСТРЕЛЬ» умеет многое, а стоит фантастически мало.

«МЕНЕСТРЕЛЬ» — это именно то, что вам нужно.

Адрес: 117421, Москва, Ленинский проспект, д. 99.

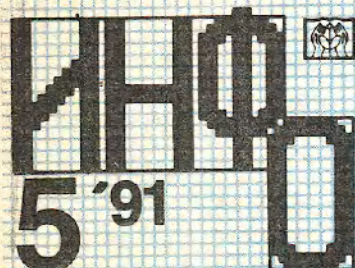
Телефон: 433-10-03

Телефакс: 4342419 Телекс: ЭКУРС 207108.

Свет

Цена 1 р. 20 к.

70423



**ИНФОРМАТИКА
И ОБРАЗОВАНИЕ**

ISSN 0234—0453. Информатика и образование, 1991, № 5, 1—128.

