

**Г.А. КРЕТОВА
И.В. ОЛЕВСКАЯ
Г.Г. ОСИКА**

Спец- практикум на МИНИ- и МИКРОЭВМ

**ДОПУЩЕНО УПРАВЛЕНИЕМ
ВЫСШЕГО И СРЕДНЕГО
СПЕЦИАЛЬНОГО ОБРАЗОВАНИЯ
ГОСУДАРСТВЕННОГО
АГРОПРОМЫШЛЕННОГО КОМИТЕТА СССР
В КАЧЕСТВЕ УЧЕБНОГО ПОСОБИЯ
ДЛЯ СТУДЕНТОВ ВЫСШИХ УЧЕБНЫХ ЗАВЕДЕНИЙ
ПО ЭКОНОМИЧЕСКИМ СПЕЦИАЛЬНОСТЯМ**

Scan Pirat



**МОСКВА
"ФИНАНСЫ И СТАТИСТИКА"
1987**

ББК 40.7
К79
УДК 681.3-181.4(076.5)

Рецензенты: кафедра машинной обработки экономической информации Кубанского сельскохозяйственного института (зав. кафедрой д-р экон. наук, проф. М. И. Семенов) и канд. экон. наук В. Ф. Сухоруков

К $\frac{2405000000-045}{010(01)-87}$ 112-87

ВВЕДЕНИЕ

Ускорение научно-технического прогресса, повышение эффективности управления народным хозяйством требуют широкого использования имеющихся и разработки новых средств вычислительной техники.

Ставится задача повышения уровня автоматизации народного хозяйства примерно в 2 раза, ускорения разработки и внедрения автоматизированных систем управления технологическими процессами, создания и освоения новых поколений ЭВМ всех классов — от суперЭВМ до персональных компьютеров. «Общий выпуск средств вычислительной техники увеличится за пятилетие в 2,3 раза»¹.

Эта задача обусловлена непрерывным ростом объемов производства, усложнением взаимосвязей между объектами управления, все возрастающим объемом собираемой, регистрируемой и обрабатываемой информации.

В настоящее время отечественной промышленностью освоены производство и выпуск ЭВМ различных классов — больших ЭВМ (ЕС ЭВМ), мини-ЭВМ (СМ ЭВМ) и микроЭВМ, способных обеспечить потребности служб управления производственной деятельностью предприятий. При этом наибольший удельный вес по объемам обрабатываемой информации ложится на мини- и микроЭВМ, устанавливаемые на низших уровнях управления.

Особенностями мини- и микроЭВМ являются расширенная номенклатура внешних устройств, возможность выбора варианта конфигурации ЭВМ самим пользователем в зависимости от решаемых задач, хорошее программное обеспечение, включающее операционные системы, средства программирования на алгоритмических языках высокого уровня в режиме диалога пользователя с ЭВМ.

¹ Рыжков Н. И. Об основных направлениях экономического и социального развития СССР на 1986—1990 годы и на период до 2000 года//Материалы XXVII съезда Коммунистической партии Советского Союза. — М.: Политиздат, 1986. — С. 237.

Организация использования мини- и микроЭВМ предусматривает непосредственную работу на них работников служб управления. Специалисты предприятий, у которых на рабочих местах будут установлены персональные ЭВМ, должны быть готовы к работе с такой вычислительной техникой. В этой связи в нашей стране проводится большая работа по массовому овладению так называемой компьютерной грамотностью, которая предполагает знание принципов работы ЭВМ, их технико-эксплуатационных характеристик и возможностей, а также умение рационально использовать ЭВМ в профессиональной деятельности.

Большая народнохозяйственная задача, стоящая перед агропромышленным комплексом страны (АПК), требует первоочередного вооружения работников АПК современными средствами вычислительной техники с максимальной эффективностью их применения.

В системе АПК накоплен положительный опыт использования мини- и микроЭВМ. Специфика сельскохозяйственного производства, территориальная разобщенность производственных участков требуют применения именно таких типов ЭВМ. Ведутся работы по расширению сферы использования автоматизированных рабочих мест (АРМ), основой которых являются микроЭВМ. Разрабатывается программное обеспечение для АРМ по решению задач учета, анализа, планирования и оперативного управления.

В докладе на совещании в ЦК КПСС по вопросам ускорения научно-технического прогресса тов. М. С. Горбачев подчеркивал: «Катализатор прогресса — микроэлектроника, вычислительная техника и приборостроение, вся индустрия информатики. Они требуют ускоренного развития. Конечно, многое зависит не только от наращивания выпуска ЭВМ, но и от умелого использования их в народном хозяйстве»¹.

В этой связи одной из важнейших задач является подготовка специалистов в области вычислительной техники для народного хозяйства и в первую очередь для сельского хозяйства. Цель методического пособия — обучение студентов экономических специальностей работе на программируемых ЭКВМ, мини- и микроЭВМ. При этом ставятся следующие задачи:

научить студентов работе на программируемых ЭКВМ;

¹ Правда. — 1985. — 12 июня.

познакомить студентов с мини- и микроЭВМ, их характеристиками, назначением, сферой применения, принципами построения;

дать представление об организации решения экономических задач с помощью ЭВМ;

познакомить с возможностями и основами работы с операционной системой ОСРВ СМ-4;

научить студентов составлению, редактированию, отладке, выполнению программ на алгоритмическом языке Бейсик при работе на мини- и микроЭВМ.

Учебное пособие написано в соответствии с требованиями курса «Спецпрактикум на мини- и микроЭВМ» для студентов высших сельскохозяйственных заведений по экономическим специальностям.

Последовательность изложения материала в учебном пособии согласуется с программой курса.

В главе 1 приводятся основные сведения об электронных клавишных вычислительных машинах, принципах их построения и использования. Подробно описываются способы обработки информации, основы составления, редактирования и выполнения программ на ЭКВМ «Искра-124», программируемом микрокалькуляторе «Электроника МК-54». Изучение данного материала и выполнение предлагаемых в пособии заданий будет способствовать выработке у студентов практических навыков по обработке экономической информации сельскохозяйственных предприятий.

В главе 2 приводятся общие сведения о техническом и программном обеспечении отдельных моделей мини- и микроЭВМ.

В главе 3 излагаются основы программирования на алгоритмическом языке Бейсик. Язык Бейсик предлагается к изучению как сравнительно простой язык программирования для начинающих, получивший достаточно широкое распространение при работе пользователей на мини- и микроЭВМ в диалоговом режиме. При изучении элементов языка программирования Бейсик и процессов реализации Бейсик-программ на мини- и микроЭВМ данное пособие может использоваться учениками 9—10 классов в соответствии с новой программой, а также любым пользователем при работе с интерпретатором языка на ЭВМ индивидуального пользования.

В главе 4 описывается работа в операционных системах мини- и микроЭВМ. Основное внимание уделено операционной системе ОСРВ и ее применению в про-

цессе работы пользователя на мини-ЭВМ СМ-4. Наряду с этим достаточно подробно показана последовательность работы с интерпретатором Бейсик на ЭВМ СМ-4, «Искра 226», ДВК-2.

Включение в соответствующие разделы описания клавиатур, пультов управления, последовательности работы с операционными системами на отдельных, получивших наибольшее распространение в сельском хозяйстве машинах («Искра 226», СМ-4), делает данное пособие полезным при изучении курсов «Технические средства управления и вычислительные машины», «Алгоритмизация и программирование» для студентов старших курсов, а также для слушателей факультета повышения квалификации.

Введение, параграфы 2.1, 2.4 написаны Г. Г. Осика, глава 1, параграфы 2.2, 2.3, 4.2 — Г. А. Кретовой, глава 3, параграфы 4.1, 4.3 — И. В. Олевской.

Авторы выражают искреннюю благодарность за большую работу по рецензированию пособия профессору кафедры машинной обработки информации Кубанского СХИ М. И. Семенову и доценту кафедры экономической кибернетики Одесского СХИ В. Ф. Сухорукову.

Глава 1

ЭЛЕКТРОННЫЕ КЛАВИШНЫЕ ВЫЧИСЛИТЕЛЬНЫЕ МАШИНЫ

1.1. ОБЩИЕ СВЕДЕНИЯ

Электронные клавишные вычислительные машины (ЭКВМ) относятся к классу клавишных вычислительных машин с вводом информации ручным способом. ЭКВМ широко применяются для решения планово-экономических, статистических, инженерно-технических, научных задач с большим объемом исходных данных и сравнительно малым объемом несложных вычислений. ЭКВМ используются как централизованно на вычислительных установках, так и децентрализованно при выполнении вычислений отдельными пользователями.

ЭКВМ — машины высокого быстродействия, весьма надежные, бесшумные, экономичные в эксплуатации, имеют сравнительно небольшие массу и габариты, обладают широкими функциональными возможностями. ЭКВМ оперируют с числами, представленными в форме с фиксированной и плавающей запятой.

По степени автоматизации вычислительного процесса ЭКВМ можно разделить на ЭКВМ без программного управления и программируемые ЭКВМ. К программируемым относятся «Искра-124», «Искра-125», «Электроника БЗ-34», «Электроника МК-54», «Электроника МК-56», «Электроника МК-61» и др. Программы могут вводиться несколькими способами: а) с клавиатуры ЭКВМ; б) со съемных блоков энергонезависимой памяти (ППЗУ), подключаемых через специальные разъемы «Электроника МК-52»; в) с магнитных карт через устройства записи-считывания — «Искра-124»; г) с магнитных лент накопителей (НМЛ) — «Искра-125».

По способу вывода информации различают ЭКВМ: с выводом на индикатор, с выводом на печать, с выводом на индикатор и печать. Индикаторные устройства выполняются на люминесцентных, газоразрядных, нео-

новых лампах, электронно-лучевых трубках. Некоторые модели ЭКВМ, например «Искра-1121», «Искра-108», имеют печатающее устройство барабанного типа с печатью узкого документа (шириной 90 мм, емкостью до 18 печатающих разрядов).

В нашей стране широкое распространение получили ЭКВМ серий «Искра», «Электроника».

ЭКВМ серии «Искра» строятся на транзисторах и интегральных схемах, имеют единую унифицированную номенклатуру моделей, единую структуру и принципы конструкции. Разработаны три базовые конструкции ЭКВМ «Искра»: ЭКВМ для простейших расчетов — «Искра-1103», ЭКВМ для деловых расчетов — «Искра-111» и ее модификации; ЭКВМ для научных расчетов — «Искра-122» и ее модификации.

Расширенными функциональными возможностями отличаются «Искра-124», «Искра-125». На «Искре-124», кроме четырех арифметических действий, можно выполнять обратное деление, возведение в степень, извлечение квадратного корня, перевод градусной меры в радианную, вычисление натуральных, десятичных логарифмов и антилогарифмов, прямых и обратных тригонометрических функций, вычисления со скобками, накопление в регистрах памяти, число которых равно 10. Время выполнения операций сложения и вычитания — около 0,02 с, умножения и деления — 0,1 с, извлечения квадратного корня — 0,3 с, вычисления функций — около 0,1 с. «Искра-125» имеет те же возможности, что и «Искра-124», и, кроме того, ряд функциональных преимуществ: предусмотрены прямое и косвенное обращение к 256 регистрам памяти, возможность записи данных на магнитную ленту, на магнитную карту. Для отображения информации на «Искре-125» используется электронно-лучевая трубка (элт), имеющая 4 строки по 17 символов.

Наряду с широкими функциональными возможностями в режиме «Счет» ЭКВМ «Искра-124», «Искра-125» обеспечивают обработку цифровой информации в режиме «Программирование», при этом выполняются программы, разработанные на символично-кодowych языках, но не на алгоритмических языках высокого уровня.

ЭКВМ серии «Электроника». Первые модели ЭКВМ «Электроника ДД», «Электроника ДЗ» создавались для массового использования при решении несложных задач, в которых числовая информация обрабатывается при помощи арифметических операций сложения, вы-

читания, умножения, деления, а также операций накопления в регистре памяти.

Применение в конструкциях моделей в качестве элементной базы больших интегральных схем (БИС) способствовало созданию микрокалькуляторов серии «Электроника».

Микрокалькуляторы — ЭКВМ индивидуального пользования, построенные на микропроцессорах (МП), с питанием от автономных источников и сети переменного тока. Микрокалькуляторы обрабатывают числа, представленные в экспоненциальной форме $X = M \times 10^E$, где M — нормализованная мантисса, $E = \pm 99$ — порядок числа. Широкие функциональные возможности обеспечиваются режимом совмещенных функций¹.

Характеристики основных моделей микрокалькуляторов серии «Электроника» приводятся в табл. 1.1. Отметим, что параметры, режимы и последовательность работы на других микрокалькуляторах можно изучить, используя руководства по эксплуатации соответствующих машин.

Как уже говорилось, микрокалькуляторы по степени автоматизации вычислительных процессов делятся на две группы: микрокалькуляторы без программного управления и программируемые.

Характеристики программируемых микрокалькуляторов целесообразно рассматривать по поколениям. К первому поколению относятся «Электроника БЗ-21, МК-46»; ко второму поколению — «Электроника БЗ-36» и ее аналоги МК-54, МК-56. Программируемые микрокалькуляторы первого поколения имеют ограниченные вычислительные возможности, программную память небольшой емкости, ограниченный набор команд прямой адресации.

Программируемые микрокалькуляторы второго поколения имеют гораздо большие вычислительные возможности, большее количество оперативных регистров и адресуемых регистров памяти, более развитые программные возможности (табл. 1.1). Наличие команд прямой и косвенной адресации, команд организации условных и безусловных переходов, подпрограмм и циклов позволяет составлять и выполнять на «Электронике БЗ-36» и ее аналогах сложные программы разветвляющихся, циклических структур.

¹ Функции, обозначения которых даны над клавишами, вводятся после нажатия префиксной клавиши F и соответствующих клавиш.

Таблица 1.1

Параметры	Микрокалькуляторы серии «Электроника»				
	без программно-го управления		программируемые		
	БЗ-19М	БЗ-21	МК-46	БЗ-34, МК-56, МК-54	МК-52
Число разрядов манти-сы (порядка)	8/2	7,8/2	7,8/2	8/2	8/2
Число регистров опера-ционного блока	3	2	2	4	4
Наличие регистра вос-становления результата предшествующей опера-ции	Есть	Нет	Нет	Есть	Есть
Число адресуемых ре-гистров памяти	1	6	6	14	15
Максимальное число ша-гов программы	—	60	66	98	105
Вычисление функций X: $1/X$, X^2 , \sqrt{X} , E^x , $\ln X$, $\sin X$, $\cos X$	Есть	Есть	Есть	Есть	Есть

Параметры	Микрокалькуляторы серии «Электроника»				
	без программно-го управления		программируемые		
	БЗ-19М	БЗ-21	МК-46	БЗ-34, МК-56, МК-54	МК-52
Вычисление функций X : $\lg X$, $\arcsin X$, $\arccos X$, $\arctg X$, 10^x , $\lg X$	Есть	Нет	Нет	Есть	Есть
Адресация	Прямая	Прямая	Прямая	Прямая и косвенная	Прямая и косвенная
Оформление	Переносное	Переносное	Настольное	Переносное /настольное /переносное	Переносное
Габариты, мм	$86 \times 167 \times 41$	$185 \times 100 \times 48$	$280 \times 240 \times 90$	$185 \times 100 \times 48$; $208 \times 205 \times 60$; $167 \times 78 \times 36$	$212 \times 78 \times 34$
Масса, кг	0,4	0,39	2,5	0,39; 1,3; 0,25	0,25

Отметим, однако, что недостатком этих моделей являются сравнительно небольшая емкость программной памяти и отсутствие устройств для хранения программ и данных при выключенном питании микрокалькулятора. С целью устранения этих недостатков была разработана модель программируемого микрокалькулятора «Электроника МК-52» с энергонезависимой памятью (ППЗУ), работающей в трех режимах: «Запись», «Стирание», «Считывание». Основным устройством ППЗУ является накопитель, содержащий 4096 запоминающих элементов. Каждая команда программы занимает два четырехразрядных слова, следовательно, в ППЗУ можно поместить 512 шагов программы.

Дальнейшее повышение эффективности использования ЭКВМ требует наряду с улучшением элементной базы совершенствования программного обеспечения ЭКВМ. Выпускаемые в настоящее время в нашей стране программируемые ЭКВМ имеют возможность выполнять программы, разработанные на символично-кодových языках и не реализуют программирование на алгоритмических языках высокого уровня. Ликвидация этого недостатка позволит создать персональные ЭВМ типа РС-1210, РС-1211, РС-1500 (Япония), ZX80, ZX81 (Англия), на которых программирование ведется на алгоритмическом языке Бейсик. Кроме того, для повышения интенсивности использования ЭКВМ разрабатываются прикладные программы [10, 25], позволяющие выполнять статистическую обработку данных, сглаживание экспериментальных зависимостей и т. д.

1.2. ОСНОВНЫЕ УСТРОЙСТВА И ПРИНЦИП РАБОТЫ

Основными устройствами ЭКВМ являются: устройство ввода (УВв); устройство управления (УУ); арифметико-логическое устройство (АЛУ); запоминающие устройства — оперативное (ОЗУ), постоянное (ПЗУ); устройство вывода (УВыв).

Устройство ввода состоит из цифровой клавиатуры и клавиатуры управления. Схемы клавиатур ЭКВМ «Искра-124» и микрокалькулятора «Электроника МК-54» показаны на рис. 1.3 и 1.4. Цифровая клавиатура имеет 10 цифровых клавиш (0, 1, ..., 9), клавишу отделения целой части от дробной [.] . Числа вводятся в десятичной системе счисления, начиная со старшего

разряда. Для преобразования каждой вводимой цифры в двоичный код служит кодификатор, соединенный с цифровой клавиатурой. Далее цифра в двоичном коде с кодификатора поступает в операционный регистр ОЗУ и в зависимости от команды, поступающей с клавиатуры управления, обрабатывается.

Клавиатура управления служит для управления вычислениями вручную, команды с клавиатуры поступают на устройство управления.

Устройство управления осуществляет взаимодействие, координацию и синхронизацию всех устройств ЭКВМ, т. е. управляет процессами ввода исходных данных, их арифметической обработкой, хранением и передачей информации внутри машины, выдачей результатов вычислений на индикатор или устройство печати.

Запоминающие устройства (ОЗУ, ПЗУ) используются для хранения данных, программ. ОЗУ включает операционные регистры для занесения чисел при выполнении арифметических операций, а также регистры памяти для хранения промежуточных результатов, итогов вычислений и т. д. ЭКВМ, применяемые для несложных расчетов, имеют один регистр памяти («Электроника ДЗ», «Электроника МК-42» и др.); применяемые для решения научных, инженерно-технических задач «Искра-124» — 10 регистров, «Электроника БЗ-34» и ее аналоги — по 14 адресуемых регистров памяти. В ПЗУ хранятся константы, команды (микропрограммы) для вычисления элементарных функций и т. д.

Арифметико-логическое устройство (АЛУ) обрабатывает поразрядно десятичные числа. При обработке дробных десятичных чисел используется блок обработки запятой, который анализирует положение запятой в числах и определяет положение запятой в результате.

Устройство вывода (УВыв) почти во всех ЭКВМ построено на цифровых индикаторах, лишь в некоторых моделях («Искра-1121», «Искра-108») применяются цифронпечатающие устройства.

Остановимся на некоторых особенностях построения микрокалькуляторов. Их основой (рис. 1.1) является специальное устройство для проведения логических и арифметических операций — микропроцессор (МП), к которому через устройство управления подключаются запоминающие устройства (ОЗУ и ПЗУ), клавишный пульт, устройство вывода — индикатор. Микрокальку-

ляторы имеют два операционных регистра (X, Y), которые часто объединяются еще с несколькими регистрами (Z, T и т. д.), образуя стек. Числа в стеке перемещаются вверх ($X \rightarrow Y$, $Y \rightarrow Z$, $Z \rightarrow T$ и т. д.) или вниз ($X \leftarrow Y$, $Y \leftarrow Z$, $Z \leftarrow T$ и т. д.), для разделения вводимых чисел и передвижения информации в стеке используется специальная клавиша [B↑].

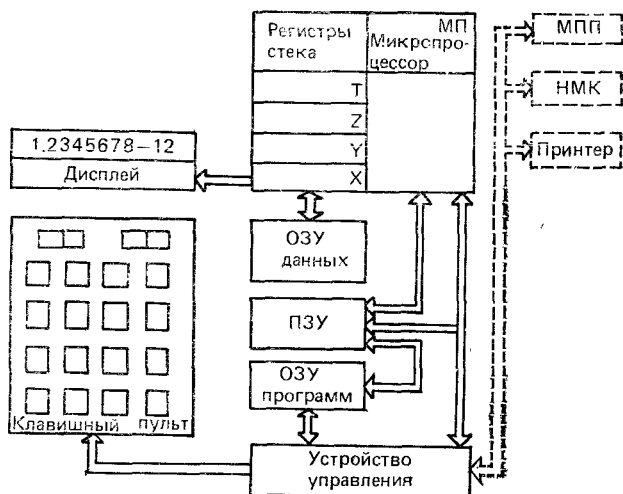


Рис. 1.1. Функциональная схема микрокалькулятора

Программируемые микрокалькуляторы наряду с ПЗУ для хранения микропрограмм и ОЗУ для хранения данных имеют программную память, емкость которой колеблется от 60 шагов у ЭКВМ «Электроника БЗ-21» до 105 шагов у ЭКВМ «Электроника МК-52».

К микропроцессору через специальные устройства могут подключаться дополнительные устройства: накопитель на магнитной карте (НМК), модуль с программами пользователя (МПП), печатающее устройство — принтер и др.

В ЭКВМ и микрокалькуляторах используется микропрограммный способ управления. При вводе внешней команды с клавиатуры управления выполняется микропрограмма поиска адресов, в которой код команды последовательно анализируется и устанавливается нужная программа.

Микропрограмма — последовательность управляющих слов, включающих в себя микрокоманды, условия, адреса.

Для реализации операций используются микрокоманды, являющиеся совокупностью микроопераций.

Микрооперация — это элементарное действие (сдвиг, установка, счет и т. п.). Микрокоманды и микрооперации формируются в устройствах управления с участием тактового генератора. Микрооперации и микрокоманды обеспечивают выполнение следующих функций: ввод числа, выделение целой части числа, выполнение арифметических операций, передачу и обмен, сдвиг, гашение, обращение к памяти, индикацию.

Режим работы ЭКВМ — *режим индикации*. При поступлении сигнала с клавиатуры управления устройство управления выполняет необходимую операцию и автоматически возвращает машину в режим индикации. Таким образом обеспечивается пошаговое выполнение всех операций. Использование принципа индикации и последовательной десятичной арифметики требует принудительного последовательного обращения к запоминающему устройству и жесткой синхронизации его работы со всеми другими устройствами, что обеспечивает устройство управления.

1.3. ПРИЕМЫ ВЫЧИСЛЕНИЙ НА ЭКВМ

1.3.1. ВЫЧИСЛЕНИЯ В РЕЖИМЕ «СЧЕТ»

Рассмотрим приемы вычислений на ЭКВМ «Электроника ДЗ», «Искра-124», микрокалькуляторе «Электроника МК-54». Схемы клавиатур этих машин приводятся соответственно на рис. 1.2, 1.4 и 1.3. Напомним, что «Электроника ДЗ» является непрограммируемой ЭКВМ, «Искра-124» может работать как в режиме «Счет», так и в режиме «Ввод программ» (ВП), на «Электронике МК-54» вычисления производятся в режимах «Автоматическая работа» и «Программирование».

Остановимся на некоторых особенностях построения и использования ЭКВМ.

1. Ввод и обработка чисел (целых и дробных) осуществляются в естественной форме или в форме с плавающей запятой. При вводе чисел отделение целой ча-

сти от дробной на ЭКВМ производится клавишей [.] на микрокалькуляторах — клавишей [.]

2. Вычисления на ЭКВМ производятся с заданной точностью, с автоматическим округлением результатов.

3. Сброс (гашение) чисел в регистре клавиатуры, операционных регистрах, вывод из блокировки при переполнении разрядной сетки индикатора и «некорректных операциях» осуществляется нажатием соответствующих клавиш (табл. 1.2).

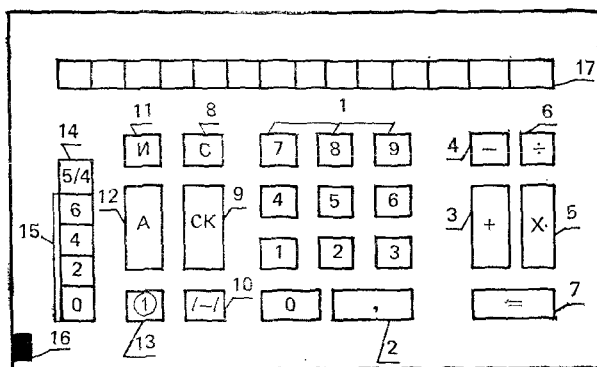


Рис. 1.2. Клавиатура ЭКВМ «Электроника ДЗ»

Клавиши: 1 — набора чисел; 2 — отделения целой части числа от дробной; 3 — сложения; 4 — вычитания; 5 — умножения; 6 — деления; 7 — результата; 8 — общего гашения; 9 — гашения набранного числа (текущего регистра); 10 — изменения знака набранного числа; 11 — вызова на индикатор одного из оперативных регистров попеременно; 12 — засылки и вызова из регистра памяти; 13 — накопления в регистре памяти; 14 — округления; 15 — установки фиксированного положения запятой; 16 — тумблер включения и выключения машины; 17 — индикатор (14-разрядный)

4. Регистры памяти ЭКВМ используются как накопители. Напомним, что «Электроника ДЗ» имеет 1 регистр памяти, «Искра-124» имеет 10 адресуемых регистров памяти: A0, A1, ..., A9, микрокалькулятор «Электроника МК-54» и его аналоги — 14 адресуемых регистров памяти: RG0, RG1, ..., RG9, RGa, RGb, RGc, RGd.

5. Многие модели ЭКВМ имеют двойную символику клавиш (см. рис. 1.3, 1.4). На микрокалькуляторах для выполнения операций, записанных над клавишами, предварительно нажимают клавишу F. Для выхода из режима совмещенных функций нажимают клавиши F. CF. Для выполнения операций, записанных в нижней части клавиш на ЭКВМ «Искра-124», предварительно нажимают клавишу A.

6. При вычислении значений прямых и обратных тригонометрических функций аргументы могут задаваться в радианах, градусах (для «Электроники МК-54», «Электроники МК-52» также в градах¹), что обеспечивается установкой переключателя 5 на ЭКВМ «Электроника МК-54» (см. рис. 1.3) или клавиши 27 на «Искре-124» (см. рис. 1.4) в нужное положение.

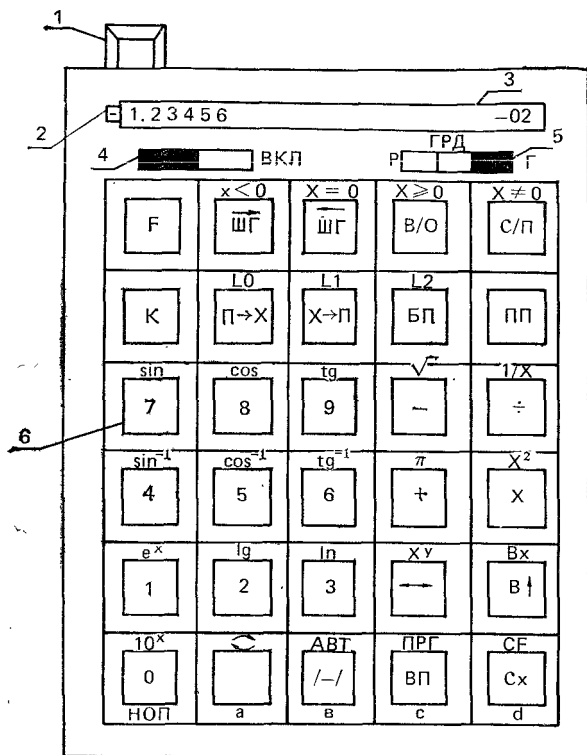


Рис. 1.3. Клавиатура микрокалькулятора «Электроника МК-54»:

1 — разъем для подключения блока питания; 2 — индикация знака числа; 3 — индикатор люминесцентный; 4 — переключатель питания; 5 — переключатель «радианы, градусы, грады»; 6 — клавиши набора чисел и управления работой

Последовательность работы оператора на ЭКВМ, единая для всех моделей ЭКВМ, и состоит из следующих этапов:

¹ Град — мера измерения углов, равная 1/100 части прямого угла.

1. Убедившись в исправности аппаратуры включения электропитания, включить машину в сеть.

2. Установить тумблер (кнопку) на корпусе машины в положение «Вкл» (включено).

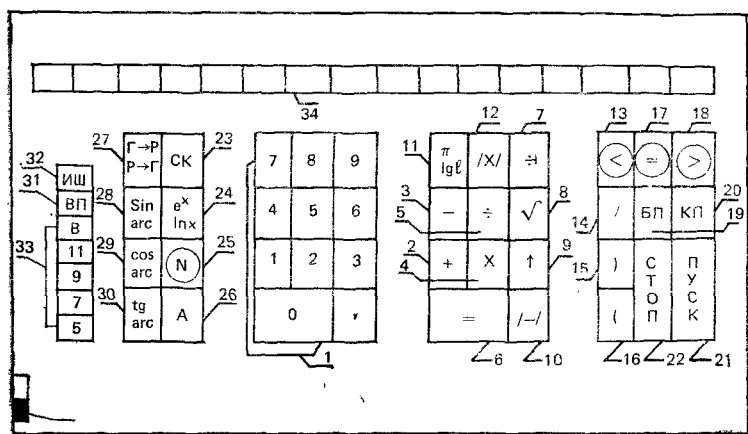


Рис. 1.4. Клавиатура ЭКВМ «Искра-124»

Клавиши: 1 — набора чисел; 2 — сложения; 3 — вычитания; 4 — умножения; 5 — деления; 6 — окончательного итога (равно); 7 — обратного деления; 8 — извлечения квадратного корня; 9 — возведения в степень; 10 — изменения знака; 11 — вызова констант π или $\lg e$; 12 — модуля числа; 13 — условного перехода «по меньше»; 14 — выделения целой части числа; 15 — закрывающая скобка; 16 — открывающая скобка; 17 — условного перехода «по равно»; 18 — условного перехода «по больше»; 19 — безусловного перехода; 20 — конца программы; 21 — запуска счета по программе; 22 — остановка счета по программе; 23 — сброса (гашения) регистра клавиатуры; 24 — вычисления значений e^x или $\ln x$; 25 — накопления в регистре памяти; 26 — обращения к регистру памяти; 27 — переводов из градусной меры в радианную и обратно; 28 — вычисления значений \sin и $\arcsin x$; 29 — вычисления значений $\cos x$ или $\arccos x$; 30 — вычисления значений $\lg x$ или $\operatorname{arctg} x$; 31 — переключатель режимов «ввод программы-счет»; 32 — индикация шага программы (с фиксацией); 33 — переключатель точности; 34 — индикатор (шестнадцатиразрядный); 35 — тумблер включения

3. Выполнить сброс (гашение), используя схемы табл. 1.2.

4. С учетом необходимой точности вычислений установить и зафиксировать положение запятой клавишей 15 на ЭКВМ «Электроника ДЗ» или клавишей 33 на ЭКВМ «Искра-124».

5. При необходимости нажать клавишу округления результата (клавиша 14 на ЭКВМ «Электроника ДЗ»). Округление на микрокалькуляторах и «Искре-124» осуществляется автоматически.

6. Выполнить необходимые вычисления, используя схемы, приведенные в табл. 1.3—1.5. Отметим, что схемы, приведенные для ЭКВМ «Электроника МК-54», мо-

Таблица 1.2

Операция	«Электроника ДЗ»	«Электроника МК-54»	«Искра-124»
Гашение регистра клавиатуры (набранного числа)	[СК]	[Сх]	[СК]
Гашение операционных регистров	[С]	[Сх] [Сх]	[=] [СК]
Гашение регистров памяти *	[С] [А]	[Сх] $[X \rightarrow \Pi] 0$, [Сх] $[X \rightarrow \Pi] 1$, ..., 9, [Сх] $[X \rightarrow \Pi] a, \dots$, [Сх] $[X \rightarrow \Pi] d$	0 [А] 0, 0 [А] 1, ..., 0 [А] 9
Возврат в исходное положение клавиши совмещенных функций [F]	—	[F] $\begin{bmatrix} CF \\ Сх \end{bmatrix}$	—
Вывод из блокировки	[С]	[Сх]	[СК][=][СК]

* Гашение всех регистров памяти можно произвести, выключив машину из сети. гут использоваться также и для «Электроника БЗ-34, МК-56». При этом необходимо учитывать имеющиеся различия в обозначениях клавиш:

«Электроника БЗ-34»	П	ИП	↑	\overleftrightarrow{XY}	arc sin	arc cos	arc tg	A	D
«Электроника МК-54»	X→Π	Π→X	B↑	↔	sin ⁻¹	cos ⁻¹	tg ⁻¹	a	d
«Электроника МК-56»									

7. Установить тумблер (кнопку) в положение «Выкл» (выключено).

8. Выключить машину из сети.

1.3.2. СОСТАВЛЕНИЕ И ВЫПОЛНЕНИЕ ПРОГРАММ НА ПРОГРАММИРУЕМЫХ ЭКВМ

Решение задач на программируемых микрокалькуляторах и ЭКВМ осуществляется по следующим этапам: постановка задачи, разработка алгоритма и программы, ввод программы в память ЭКВМ и редактирование, отладка программы, занесение исходных данных и выполнение программы.

Схемы выполнения простейших операций на ЭКВМ
«Электроника ДЗ», «Электроника МК-54», «Искра-124»

Операция	ЭКВМ «Электроника ДЗ»	Микрокалькулятор «Электроника МК-54»	ЭКВМ «Искра-124»
1. Сложение $173 + 194,83 + 32,67 = 400,5$	$[C][2]173[+]194,83[+]32,67[=]$	$[C_x]173[B\uparrow]194,83[+]32,67[+]$	$[CK]173[+]194,83[+]32,67[=]$
2. Вычитание $43,465 - 93,228 = -49,763$	$[C][4]43,465[-]93,228[=]$	$[C_x]43,465[B\uparrow]93,228[-]$	$[CK]743,465[-]93,228[=]$
3. Умножение с точностью до 0,01 $2,47 \times 0,7 = 1,73$	$[C]2[5/4]2,47[\times]0,7[=]$	$[C_x]2,47[B\uparrow]0,7[\times]$	$[7]2,47[\times]0,7[=]$
4. Деление с точностью до 0,01 $20 : 3 = 6,67$	$[C]2[5/4]20[:3][=]$	$[C_x]20[B\uparrow]3[:]$	$[7]20[:3][=]$
5. Возведение в степень $2,5^2 = 6,25$	$[C]2[5/4]2,5[\times][=]$	$[C_x]2,5[F][X^2]$	$[5]2,5[\uparrow]$
6. Арифметические операции с константой* $82 \times 1,2 = 98,40$ $18 \times 1,2 = 21,60$ $-35 \times 1,2 = -42,00$ и т. д.	$[C]2[82[\times]1,2[=]$ $18[=]$ $35[/-]/[=]$ и т. д.	$[C_x]82[B\uparrow]1,2[\times]$ $18[\times]$ $35[/-]/[=]$	$1,2[A]082[\times][=]$ $18[\times][=]$ $35[/-]/[=]$ и т. д.

* Аналогично выполняются операции сложения, вычитания, деления на константу.

Таблица 1.4

Схемы выполнения операций с использованием регистров памяти

Операция	«Электроника ДЗ»	«Электроника МК-54»	«Искра-124»
1. Гашение регистра памяти	[C] [A]	$[C_x][X \rightarrow \Pi][0]$ или $[C_x][X \rightarrow \Pi]$ 1, ..., 9 или [a], [в], [с], [d]	$[0][A][0]$ или $[0][A][1], \dots,$ $[0][A][9]$
2. Запись числа 25,3 (или любого другого) в память с одновременным гашением ранее записанного в память числа	25,3 [A]	25,3 $[X \rightarrow \Pi][0], \dots, [9]$ или 25,3 $[X \rightarrow \Pi][a], [в], [с], [d]$	25,3 $[A][0], \dots, [9]$
3. Нахождение, вычисление суммы произведений: $(1,5 \times 18,6) + (11,5 \times 30) = 372,90$	$[0][A]1,5[\times]18,6[=][1]$ $11,5[\times]30[=][1]$	$[C_x][X \rightarrow \Pi][0]1,5[B \uparrow]18,6[\times]$ $[X \rightarrow \Pi][0]11,5[B \uparrow]30[\times]$ $[X \rightarrow \Pi][0][\Pi \rightarrow X][0]$	$[0][A][0]1,5[\times]18,6[=]$ $[(N)][0]11,5[\times]30[=]$ $[(N)][0], \dots, [A][0]$
4. Вычисление разности произведений: $(15,4 \times 18,6) - (1,5 \times 3) = 281,94$	$[0][A]15,4[\times]18,6[=][1]$ $1,5[\times]3[=][1]$	$[C_x][X \rightarrow \Pi][0]11,4[B \uparrow]18,6[\times]$ $[X \rightarrow \Pi][0]1,5[B \uparrow]3[\times]$ $[/-][X \rightarrow \Pi][0], \dots, [\Pi \rightarrow X][0]$	$[0][A][0]15,4[\times]18,6[=]$ $[(N)][0]1,5[\times]3[=][/-]$ $[(N)][0], \dots, [A][0]$
5. Вывод на индикатор содержимого памяти	[A]	$[\Pi \rightarrow X][0]$ или [1], ..., [9], ..., $[\Pi \rightarrow X][a], [в], [с], [d]$	$[A][0]$ или $[A][1], \dots, [A][9]$

Схемы возведения в степень, извлечения корня квадратного, выполнения обратного деления, вычисления значений прямых и обратных элементарных функций

Примеры	Последовательность нажатия клавиш		Результат
	«Электроника МК-54»	«Искра-124»	
1. Вычислить 18^3	3[B↑] 18 [F] $\left[\begin{smallmatrix} xy \\ \leftrightarrow \end{smallmatrix} \right]$	18 [↑] [↑]	5832
2. Вычислить $\sqrt[3]{18,49}$	18,49 [F] $\left[\sqrt[3]{} \right]$	18,49 [V]	4,3
3. Вычислить $\frac{1}{21,5 \times 4}$	[C _x] 21,5 [B↑] 4 [x] [F] $\left[\begin{smallmatrix} 1/x \\ \vdots \end{smallmatrix} \right]$	21,5 [x] 4 [=] [÷] [=]	0,012
4. Вычислить $\sin 30^\circ$	30 [F] $\left[\sin \right]$	30 $\left[\begin{smallmatrix} r-p \\ p-r \end{smallmatrix} \right]$ $\left[\sin \right]$	0,5
5. Вычислить $\arcsin 0,5$	0,5 [F] $\left[\sin^{-1} \right]$	0,5 [A] $\left[\begin{smallmatrix} \sin \\ \text{arc} \end{smallmatrix} \right]$ [A] $\left[\begin{smallmatrix} r-p \\ p-r \end{smallmatrix} \right]$	30°
6. Вычислить $\lg 65^\circ$	65 [F] $\left[\lg \right]$	65 $\left[\begin{smallmatrix} r-p \\ p-r \end{smallmatrix} \right]$ $\left[\lg \right]$	2,145

Примеры	Последовательность нажатия клавиш		Результат
	«Электроника МК-54»	«Искра-124»	
7. Вычислить $\operatorname{arctg} 28,64$	$28,64 [F] \left[\frac{1}{6} \right]$	$28,64 [A] \left[\frac{\operatorname{tg}}{\operatorname{arc}} \right] [A] \left[\frac{r-p}{p-r} \right]$	88°
8. Вычислить $e^{2,833}$	$2,833 [F] \left[\frac{e^x}{1} \right]$	$2,833 \left[\frac{e^x}{\ln x} \right]$	17
9. Вычислить $\ln 73$	$73 [F] \left[\frac{\ln}{3} \right]$	$73 [A] \left[\frac{e^x}{\ln x} \right]$	4,2905

П р и м е ч а н и е. При вычислении тригонометрических функций на ЭКВМ «Электроника МК-54» переключатель 4 (см. рис. 1.3) следует установить в соответствующее положение.

1.3.2.1. Основные понятия программирования на ЭКВМ

Алгоритм — это точно сформулированное правило, указывающее последовательность действий, в результате выполнения которых обеспечивается переход от исходных данных к искомому результату. Такая цепочка действий называется *алгоритмическим процессом*, а каждое действие — его *шагом*.

Программа для программируемых ЭКВМ — это формализованная запись алгоритма в виде последовательности операций вычислительного процесса с указанием номера шага (адреса), символов операций и кодов их индексации. Разработанные программы могут оформляться в таблицы следующего вида:

Номер шага	Команда	Код команды
------------	---------	-------------

Оператор — совокупность символов, обеспечивающих выполнение одной из элементарных операций. Каждому оператору присвоен определенный код (см. табл. 1.6 и 1.7) и соответствует свой шаг в программе. Шаги последовательно нумеруются двузначными десятичными числами. Первому оператору в программе присваивается начальный адрес. В программную память ПП можно поместить одну или несколько программ в зависимости от емкости программной памяти конкретной ЭКВМ и количества шагов в программах.

Адреса определяют местонахождение кодов операторов в программной памяти. Непосредственное указание адреса называется *прямой* адресацией. Если адрес указывается содержимым одного из регистров памяти, то такая адресация будет *косвенной*.

Вычисления на ЭКВМ в режиме «Программирование» выполняются при помощи программного устройства, в состав которого входит программная память, постоянная память с библиотекой программ, программный счетчик (ПС), регистр начального адреса (РНА), регистр перехода (РП). В программную память можно записать несколько программ. Начальный адрес каждой программы задается с клавиатуры машины перед пуском программы, запоминается в РНА и передается в ПС. ПС управляет вводом и выводом команд через регистр команд клавиатуры ввода.

Программное устройство может работать в следую-

щих режимах: «Запись программы», «Счет по программе», «Просмотр программы».

В режиме записи команды последовательно вводятся в ПП, адрес команд фиксируется в ПС, каждый ввод команды сопровождается увеличением содержимого ПС на единицу. Для контроля вводимой программы на индикаторе «Искра-124» в двух старших разрядах происходит индикация кодов команд, представленных в табл. 1.6, в двух младших разрядах индикатора — номера выбираемого шага программы при включенном положении переключателя ИШ (индикация шага). На индикаторе микрокалькулятора «Электроника МК-54» отображаются коды трех последних команд из программной памяти и адрес, по которому записана команда. Например, на индикаторе отображено

13	64	61	25
----	----	----	----

Двузначные команды операций на индикаторе означают: код 25 — текущее состояние счетчика адреса, т. е. номер шага программы; коды 13, 64, 61 — три последо-

Таблица 1.6

Коды команд ЭКВМ «Искра-124»

Команда в режиме «ВП»	Код команды	Команда в режиме «ВП»	Код команды	Команда в режиме «ВП»	Код команды
0	4.	[÷]	3.5	[()]	0.7
1	4.1	[—]	3.4	[)]	1.L
2	4.2	[÷]	3.1	[СТОП]	0.1
3	4.3	[√]	2.2	[КП]	0.
4	4.4	[+]	3.	[r—p]	2.7
5	4.5	[×	3.9	[p—r]	
6	4.6	[↑]	2.1	[A]	2.
7	4.7	[=]	5.L	[(N)]	3.8
8	5.8	[π]	5.	[sin	2.3
9	5.9	[lg]		[arc	
[.]	1.8	[(=)]	0.3	[cos	2.4
[/—/]	5.1	[(<)]	0.4	[arc	
[/x/]	5.4	[(>)]	0.5	[tg	2.5
		[БП]	0.2	[arc	
				[e ^x	2.6
				[ln x	
				[/]	5.5

Таблица 1.7

**Коды команд и операций в режиме «ПРГ»
микрокалькулятора «Электроника МК-54»**

Команды и опера- ции	Клавиши	Код	Назначение клавиш и их сочетаний; выполняемые операции
1	2	3	4
0	[0]	00	Цифровые клавиши, осуществ- ляющие занесение цифр от 0
...	[.]	09	до 9 в регистр X
+	[+]	10	Сложение содержимого реги- стра X с содержимым реги- стра Y и передача результата в регистр X
—	[—]	11	Вычитание содержимого реги- стра X из содержимого реги- стра Y и передача результата в регистр X
×	[x]	12	Умножение содержимого ре- гистра X на содержимое реги- стра Y и передача результа- та в регистр X
÷	[÷]	13	Деление содержимого реги- стра X на содержимое регистра Y и передача результата в регистр X
↔	[↔]	14	Обмен содержимым между ре- гистрами X и Y
, CX	[.]	0—	Занесение десятичной запятой
/—/ ВП	[C _x]	0Г	Сброс содержимого регистра X
	[/—/]	0L	Смена знака числа и порядка
	[ВП]	0I	Подготовка для ввода поряд- ка числа
B↑	[B↑]	0E	Разделение вводимых чисел
10 ^x	[F] [0]	15	Вычисление степенной функ- ции 10 ^x
e ^x	[F] [1]	16	Вычисление показательной функции
lg x	[F] [2]	17	Вычисление десятичного логарифма
ln x	[F] [3]	18	Вычисление натурального логарифма
sin x	[F] [7]	1I	Вычисление функции синуса
cos x	[F] [8]	1Г	Вычисление функции косинуса
tg x	[F] [9]	1E	Вычисление функции тангенса
arcsin x	[F] [4]	19	Вычисление обратной функции синуса
arccos x	[F] [5]	1—	Вычисление обратной функции косинуса
arctg x	[F] [6]	1L	Вычисление обратной функции тангенса

Команды и операции	Клавиши	Код	Назначение клавиш и их сочетаний; выполняемые операции
1	2	3	4
$\sqrt{\quad}$	[F] [—]	21	Вычисление корня квадратного
X^2	[F] [×]	22	Возведение числа x в квадрат
$1/X$	[F] [÷]	23	Вычисление величины, обратной x
X^y	[F] [↔]	24	Возведение числа x в степень y
П0, П1, ...	[x→П] [0], ...	40, ...	Запись содержимого регистра X в регистры памяти RG0, RG1, ..., RG9
П9	[x→П] [9]	49	
ИП0	[П→x] [0]	60	Вызов содержимого регистров памяти RG0, ..., RG9 в регистр X
ИП9	[П→x] [9]	69	
ПРГ	[F] $\begin{bmatrix} \text{ПРГ} \\ \text{ВП} \end{bmatrix}$		Переход в режим «Программирование»
АВТ	[F] $\begin{bmatrix} \text{АВТ} \\ \text{—/—} \end{bmatrix}$		Переход в режим «Автоматическая работа»
СП	[C/П]	50	Команда прекращения программы в режиме «ПРГ» и фиксации содержимого регистра X на индикаторе; команда начала вычислений по программе в режиме «АВТ», а также прекращения вычислений в случае заикливания
БП	[БП]	51	Команда безусловного перехода
В/О	[В/О]	52	Команда выхода из подпрограммы в режиме «ПРГ»; команда перехода на нулевой адрес в режиме «АВТ»
ПП	[ПП]	53	Команда перехода на подпрограмму в режиме «ПРГ»; команда пошагового прохождения программы в режиме «АВТ»
Переходы по условию:			
$X < 0$	[F] $\begin{bmatrix} \rightarrow \\ \text{ШГ} \end{bmatrix}$	5[Команда прямого перехода по условию $x < 0$
$X = 0$	[F] $\begin{bmatrix} \leftarrow \\ \text{ШГ} \end{bmatrix}$	5E	Команда прямого перехода по условию $x = 0$
$X \geq 0$	[F] [В/О]	59	Команда прямого перехода по условию $x \geq 0$
$X \neq 0$	[F] [C/П]	57	Команда прямого перехода по условию $x \neq 0$

Команды и операции	Клавиши	Код	Назначение клавиш и их сочетаний; выполняемые операции
1	2	3	4
Косвенные переходы	[K]		Клавиша косвенных переходов и косвенных обращений к адресуемым регистрам
	[K][БП][0], ..., [9]	80, ... 89	Команды косвенного безусловного перехода по модифицированному адресу, хранящемуся в регистрах памяти RG0, RG1, ..., RG9
	[K][ПП][0], ..., [9]	—0, ... —9	Команды косвенного перехода к подпрограммам по модифицированному адресу, хранящемуся в регистрах памяти RG0, RG1, ..., RG9
			Команда косвенных переходов по модифицированному адресу, хранящемуся в адресуемом регистре, индекс которого входит в команду по условию:
	[K][ШГ][0], ..., [9]	E0, ... E9	$x=0$
	[K][С/П][0], ..., 9	70, ... 79	$x \neq 0$
	[K][В/О][0], ..., 9	90, ... 99	$x \geq 0$
	[K][ШГ][0], ..., 9	C0, ... C9	$x < 0$
	[K][X→П][0], ..., [9]	L0, ... L9	Команда косвенной записи содержимого регистра X в регистр по модифицированному коду, хранящемуся в адресуемом регистре, индекс которого входит в команду
	[K][П→X][0], ..., [9]	Г0, ... Г9	Команда косвенной индексации вызова в регистр X содержимого адресуемого регистра по модифицированному коду, хранящемуся в регистре, индекс которого входит в команду
	[K][0] НОП	54	Команда «Нет операции»

вательные команды, обеспечивающие вызов и деление содержимого регистра RG1 на RG4, команды расположены соответственно по адресам 24, 23, 22. Коды команд микрокалькулятора представлены в табл. 1.7.

В режиме «Просмотр программы» команды из программной памяти через регистр команд клавиатуры вводятся на индикатор для контроля.

В режиме «Счет по программе» команды последовательно выбираются в регистр команд клавиатуры и выполняются машиной. Окончание выполнения программы сопровождается передачей содержимого РНА в ПС, что позволяет повторно выполнять расчеты по той же программе. Сохранение выполненной программы и запись новой можно осуществлять в свободные (не занятые старой программой) адреса. При записи в занятые адреса записанная в них ранее программа автоматически стирается.

Регистр перехода (РгП) служит для переадресации ПС при выполнении команд условных или безусловных переходов. Адрес перехода указывается двумя цифрами после команды перехода.

Безусловный переход на ЭКВМ «Искра-124» формируется включением в программу команды безусловного перехода БП с указанием адреса, к которому нужно перейти по этой команде. Переходы по условиям «меньше», «больше», «равно» осуществляются включением в программу соответственно операций ($<$), ($>$), ($=$) с указанием адреса, к которому нужно перейти при выполнении условия перехода. В противном случае выполняется команда, следующая за адресом перехода.

Решение задач с повторениями отдельных частей программы, с изменением последовательности расчетов в зависимости от условий и промежуточных результатов и т. д. на программируемых калькуляторах выполняется при помощи команд переходов. Безусловные прямые и косвенные переходы используются для построения разветвляющихся вычислительных процессов. Безусловный переход на шаг программы с адресом пп вводится оператором БП. Безусловный косвенный переход $K[X \rightarrow П]$ обеспечивает переход по адресу, указанному содержимым адресуемого регистра памяти.

Условные прямые и косвенные переходы проводятся по данным анализа содержимого операционного регистра X на выполнение одного из условий: $x < 0$, $x = 0$, $x \geq 0$, $x \neq 0$.

Подпрограммы — повторяющиеся фрагменты программ. Подпрограммы начинаются с определенного адреса пп и кончаются оператором возврата из подпрограммы В/О. Обращение к подпрограмме из любого ме-

ста программы производится командой, содержащей операторы ПП и начальный адрес подпрограммы пп.

Остановка и пуск программы. Для остановки программы в нужном месте используются: на ЭКВМ «Искра-124» команда «КП» (конец программы), на микрокалькуляторе «Электроника МК-54» — «С/П». Нажатие клавиши С/П в режиме «Автоматическая работа» ведет к запуску программы. На «Искре-124» запуск программы обеспечивается клавишей ПУСК в режиме «Счет».

1.3.2.2. Разработка программ

При разработке программ необходимо учитывать технико-эксплуатационные возможности ЭКВМ: наличие и количество регистров памяти, емкость программной памяти.

При разработке программ обработки таблиц и ведомостей следует предусматривать:

- распределение регистров памяти для исходных, промежуточных и окончательных итогов;

- гашение регистров, в которых будут накапливаться итоги по графам таблиц;

- останов для набора исходных данных и ввод их в соответствующие регистры;

- запись операций выполнения расчетов по строкам;

- накопление в регистрах памяти для определения итогов по графам;

- проверку окончания ввода исходных данных;

- конец программы.

Приведем ряд программ для обработки экономической информации на программируемых ЭКВМ с кратким изложением их содержания.

Программа 1

Пусть есть сведения о размере посевных площадей и урожайности зерновых культур сельскохозяйственных предприятий района (области). Требуется составить программу для ЭКВМ «Искра-124», которая позволит в автоматическом режиме определить сумму площадей, валовой сбор по каждой культуре, валовой сбор в целом по зерновым культурам.

Отметим, что программа может использоваться также для определения валового производства молока по фермам и в целом по хозяйству; для определения стои-

мости различных видов продукции и выручки от реализации в целом по хозяйству и т. д.

Распределение регистров памяти: А1 — для исходных данных «Площадь», А2 — для исходных данных «Урожайность», А3 — для показателя «Всего посевных площадей», А4 — для показателя «Всего валовой сбор». Возможный вариант программы представлен в табл. 1.8.

Таблица 1.8

Номер машинного адреса команды	Команда	Код команды	Номер машинного адреса команды	Команда	Код команды
00	0	4.	12	А	2.
01	А	2.	13	2	4.2
02	3	4.3	14	х	3.9
03	0	4.	15	А	2.
04	А	2.	16	1	4.1
05	4	4.4	17	=	5.1
06	СТОП	0.1	18	(N)	3.8
07	А	2.	19	4	4.4
08	1	4.1	20	БП	0.2
09	(N)	3.8	21	0	4.
10	3	4.3	22	6	4.6
11	СТОП	0.1	23	КП	0.

Содержание шагов программы:

00—05 — гашение регистров А3 и А4;

06—08 — останов перед вводом и ввод исходных данных «Площадь» в А1;

09—10 — накопление суммы площадей в А3;

11—13 — останов перед вводом и ввод исходных данных «Урожайность» в А2;

12—17 — определение валового сбора по культурам и вывод результата на индикатор;

18—19 — накопление показателя «Всего валовой сбор» в А4;

20—22 — безусловный переход к 06 шагу, т. е. к вводу исходных данных по 2-й строке и т. д., до тех пор, пока не будет закончен ввод и расчет по всем строкам;

23 — конец программы.

Программа 2

Составить программу получения оборотной ведомости по учету материальных ценностей на ЭКВМ «Искра-124», используя данные табл. 1.15.

Распределение регистров памяти: А1 — «Итого» остатков на начало месяца, А2 — «Итого» приход, А3 — «Итого» расход, А4 — «Итого» остатков на конец месяца.

ца. Возможный вариант программы представлен в табл. 1.9.

Таблица 1.9

Номер машинного шага команды	Команда	Код команды	Номер машинного шага команды	Команда	Код команды	Номер машинного шага команды	Команда	Код команды
00	0	4.	14	5	4.5	28	5	4.5
01	A	2.	15	(N)	3.8	29	+	3.
02	1	4.1	16	1	4.1	30	A	2.
03	0	4.	17	СТОП	0.1	31	6	4.6
04	A	2.	18	A	2.	32	—	3.4
05	2	4.2	19	6	4.6	33	A	2.
06	0	4.	20	(N)	3.8	34	7	4.7
07	A	2.	21	2	4.2	35	=	5.L
08	3	4.3	22	СТОП	0.1	36	(N)	3.8
09	0	4.	23	A	2.	37	4	4.4
10	A	2.	24	7	4.7	38	БП	0.2
11	4	4.4	25	(N)	3.8	39	1	4.1
12	СТОП	0.1	26	3	4.3	40	2	4.2
13	A	2.	27	A	2.	41	КП	0.

Содержание шагов программы:

- 00—11 — обнуление регистров, в которых предусмотрен подсчет итогов по графам;
 12 — останов для ввода исходных данных по 1-й графе;
 13—16 — засылка чисел 1-й графы в A5 и накопление их в A1;
 17 — останов для ввода исходных данных по 2-й графе;
 18—21 — засылка чисел 2-й графы в A6 и накопление их в A2;
 22 — останов для ввода исходных данных по 3-й графе;
 23—26 — засылка чисел 3-й графы в A7 и накопление их в A3;
 27—35 — определение остатка на конец месяца по строкам;
 36—37 — накопление остатков на конец месяца в A4;
 38—40 — безусловный переход к 12-му шагу программы: к вводу исходных данных по следующим строкам и т. д.;
 41 — конец программы.

Программа 3

Составить программу вычисления заработной платы с вычетом подоходного налога на микрокалькуляторе «Электроника МК-54» по формуле

$$N_1 = N - [100 \cdot 8,2\% + (N - 100) \cdot 13\%] = N - (0,13N - 4,8),$$

где N — заработная плата, N_1 — заработная плата, причитающаяся к выплате.

В программе предусмотреть занесение N в RG2 и вывод на индикатор N_1 . Возможный вариант программы представлен в табл. 1.10.

Таблица 1.10

Адрес команды	Клавиша	Код операции	Адрес команды	Клавиша	Код операции
00	[B↑]	0E	09	[8]	08
01	[X→Π][2]	42	10	[—]	11
02	[0]	00	11	[Π→X][2]	62
03	[.]	0—	12	[↔]	14
04	[1]	01	13	[—]	11
05	[3]	03	14	[C/Π]	50
06	[X]	12	15	[BΠ]	51
07	[4]	04	16	[0][1]	01
08	[.]	0—			

Содержание шагов программы:

- 00 — разделение вводимых далее чисел (N и $0,13$);
- 01 — запись в RG2 содержимого регистра X. В регистр X с клавиатуры будем вносить при выполнении программы заработную плату N ;
- 02—06 — вычисление произведения $0,13 N$;
- 07—10 — вычитание из полученного произведения числа $4,8$;
- 11 — вызов в операционный регистр X числа N , содержащегося в RG2;
- 12 — обмен содержимым между регистрами X и Y;
- 13 — вычитание, т. е. определение разности между N , находящимся в регистре X, и выражением $(0,13 N - 4,8)$, находящимся в регистре Y;
- 14 — останов;
- 15—16 — безусловный переход к 01 шагу программы, т. е. к операции введения N в RG2.

Процесс вычисления заработной платы по такой программе сводится после окончания ввода программы к набору на клавиатуре величины заработной платы N и нажатию клавиши C/Π (пуск программы на исполнение).

Программа 4¹

Вычисляются среднее значение m_x и дисперсия D_x совокупности x_1, x_2, \dots, x_n на программируемых микрокалькуляторах (табл. 1.11).

$$m_x = \frac{\sum_{i=1}^n x_i}{n}; \quad D_x = \frac{1}{n} \sum_{i=1}^n \left(\frac{m_x}{n} - x_i^2 \right).$$

Распределение регистров памяти:

$$RG0 \rightarrow \sum_{i=1}^n x_i, m_x; \quad RG1 \rightarrow \sum_{i=1}^n x_i^2, D_x; \quad RG4 \rightarrow i, n.$$

Содержание программы:

00—03 — гашение регистров RG0, RG1, RG4;

04 — останов перед началом вычислений, необходимый для ввода x_i ;

Таблица 1.11

Ад-рес	Клавиша	Код	Ад-рес	Клавиша	Код	Ад-рес	Клавиша	Код
00	[C _x]	0Г	10	[↔]	14	20	[X→Π][0]	40
01	[X→Π][0]	40	11	[Π→X][1]	61	21	[C/Π]	50
02	[X→Π][1]	41	12	[+]	10	22	[Π→X][1]	61
03	[X→Π][4]	44	13	[X→Π][1]	41	23	[Π→X][4]	64
04	[C/Π]	50	14	[K]Π→X[4]	Г4	24	[÷]	13
05	[F][X ²]	22	15	[БΠ]	51	25	[Π→X][0]	60
06	[F][B _x]	0Е	16	[04]	04	26	[F][X ²]	22
07	[Π→X][0]	60	17	[Π→X][0]	60	27	[—]	11
08	[+]	10	18	[Π→X][4]	64	28	[X→Π][1]	41
09	[X→Π][0]	40	19	[÷]	13	29	[C/Π]	50

05 — вычисление квадрата вводимых чисел;

06 — восстановление x_i в регистре X;

07—09 — вычисление $\sum_{i=1}^n x_i$ и занесение в регистр RG0;

10 — обмен содержимым между регистрами X и Y;

11—13 — вычисление $\sum_{i=1}^n x_i^2$ и занесение в регистр RG1;

14 — косвенная индексация вызова в регистр X числа n , хранящегося в RG4;

15—16 — безусловный переход к 04 шагу, организующий цикл по вводу исходных данных;

¹ Программа взята из книги: Цветков А. Н., Епанечников В. А. Прикладные программы для микроЭВМ «Электроника БЗ-34» и «Электроника МК-56». — М.: «Финансы и статистика», 1984. — С. 102.

- 17—20 — вычисление m_x и занесение m_x в RG0;
- 18 — безусловный переход к 04 адресу;
- 21 — останов, обеспечивающий вызов на индикатор m_x при нажатии клавиши С/П в режиме «АВТ»;
- 22—28 — вычисление D_x и занесение в RG1;
- 29 — останов, обеспечивающий вызов на индикатор D_x при нажатии клавиши С/П в режиме «АВТ».

1.3.2.3. Ввод программ в память ЭКВМ и редактирование

Перед вводом программ машины следует подготовить к соответствующему режиму работы: ЭКВМ «Искра-124» — к режиму «ВП», а программируемые микрокалькуляторы — к режиму «ПРГ».

На ЭКВМ «Искра-124» это осуществляется в следующей последовательности:

установить переключатель ВП в режим «Счет», т. е. нажать клавишу В;

нажать клавишу БП и две клавиши, соответствующие начальному адресу программы;

установить переключатель ВП в нижнее положение, т. е. в режим «Ввод программы», при этом клавиша В автоматически займет верхнее положение;

нажать клавишу ИШ. На индикаторе высвечивается: 0 _ _ _ _ _ _ _ _ _ _ 00, что свидетельствует о готовности машины к вводу программы.

Ввод программ в программируемую память ЭКВМ осуществляется последовательным набором операционных клавиш в строгом соответствии с ранее составленной программой. Следует контролировать при вводе номер шага программы и коды команд. Редактирование программ, записанных в первой зоне «Искра-124», осуществляется вызовом на индикатор любого шага программы в следующей последовательности: в режиме «Счет» нажать клавишу БП и две цифры соответствующего шага программы, далее однократно нажать клавишу ПУСК в режиме «ВП», что обеспечит вызов нужного шага программы. При последующем нажатии клавиши ПУСК обеспечивается пошаговый просмотр программы. Просмотр программ, записанных во второй зоне, можно осуществить только последовательно, начиная с последнего шага в первой зоне, нажатием клавиши ПУСК в режиме «ВП».

Исправление ошибочно введенной команды осуществляется следующим образом:

задается адрес шага программы в режиме «Счет» (клавиша БП и две цифры номера шага программы, в котором есть ошибка);

вводится новая команда по данному шагу программы в режиме «ВП».

Подготовка и занесение программ на микрокалькуляторы «Электроника БЗ-34» и «Электроника МК-54» осуществляются двумя способами, каждый из которых предполагает определенную последовательность действий.

Первый способ — занесение программы с нулевого адреса:

а) очистить программный счетчик в режиме «Автоматическая работа», последовательно нажав клавиши F, АВТ, В/О;

б) перейти в режим «Программирование», нажав клавиши F, ПРГ, при этом на индикаторе появится адрес счетчика 00;

в) ввести программу, нажимая клавиши, записанные в программе, и контролируя на индикаторе коды вводимых команд и номера шагов программы.

Второй способ — занесение программ с произвольного адреса:

а) установить клавиши F, АВТ, БП и затем две цифровые клавиши, соответствующие номеру адреса, с которого начинается программа;

б) перейти в режим «ПРГ»; на индикаторе индицируется адрес, с которого вводится программа;

в) ввести программу нажатием соответствующих клавиш с одновременным контролем.

Исправление ошибок при вводе программы осуществляется следующим образом:

переходом на адрес, по которому записана ошибочная команда, с помощью клавиш F, АВТ, БП и двух цифровых клавиш, указывающих нужный адрес;

переходом в режим «Программирование» посредством нажатия клавиш F, ПРГ;

исправлением ошибки посредством нажатия клавиши требуемой операции или команды после появления в режиме «ПРГ» на индикаторе адреса ошибочной команды.

Перевод на нужный адрес можно осуществить нажатием клавиш ПРГ или ПРГ, если адрес ошибочной команды находится недалеко от текущего.

Для исключения команды из программы следует перейти на адрес этой команды, нажав клавиш К и

НОП. В программную память запишется команда «Нет операций».

1.3.2.4. Отладка программ

Для контроля правильности составления и ввода осуществляют отладку программ на контрольных примерах.

Для ЭКВМ «Искра-124» последовательность действий следующая:

отжать клавиши ИШ, ВП, нажав клавишу В;

установить начальный адрес программы: БПп, где пп — две клавиши (цифровые), указывающие начальный адрес программы;

нажать клавишу ПУСК;

осуществить последовательный набор исходных данных на клавиатуре и их ввод клавишей ПУСК.

Для микрокалькуляторов «Электроника БЗ-34» и «Электроника МК-54» выполняются следующие действия:

переход в режим «Автоматическая работа» посредством нажатия клавиши F и АВТ;

набор исходных данных контрольного примера и их ввод;

нажатие клавиши В/О, если программа начинается с адреса 00, или клавиши БП и двух цифр, указывающих начальный адрес программы;

нажатие клавиши пуска программы С/П.

На программируемых ЭКВМ можно осуществлять пошаговое выполнение программ:

а) на ЭКВМ «Искра-124» — клавишей ПУСК в режиме «Счет»;

б) на микрокалькуляторах «Электроника БЗ-34» и «Электроника МК-54» — клавишей ПП в режиме «АВТ».

При отладке программ возможны переполнения или заикливания, которые ликвидируются на «Искра-124» нажатием клавиши СК, на программируемых микрокалькуляторах клавишей С/П.

1.3.2.5. Занесение исходных данных и выполнение программ

Последовательность занесения исходных данных и выполнение программ показаны в табл. 1.12.

1.4. ЗАДАНИЯ

1. Подготовить имеющуюся в Вашем распоряжении машину к работе, используя методические указания п. 1.3.1.

Таблица 1.12

Последовательность действий	Выполняемые операции	
	«Искра-124»	«Электроника МК-54»
1. Подготовка машины к вводу программы	[СК] [В]	[F] [АВТ]
2. Установка начального адреса		
если начальный адрес 00	[БП] [00]	[В/О]
если начальный адрес nn	[БП] nn	[БП] nn
3. Включение режима «Программирование»	[ВП] [ИЩ]	[F] [ПРГ]
4. Пошаговый ввод программы	Согласно командам программы	
5. Подготовка к режиму «Счет» по программе	отжать клавишу [ИЩ]	[F] [АВТ]
6. Пуск программы	нажать [В]	
с начального адреса 00	[БП] [00] [ПУСК]	[В/О] [С/П]
с начального адреса nn	[БП] nn [ПУСК]	[БП] nn [С/П]
7. Занесение исходных данных совокупности X_1, \dots, X_n	X_1 [ПУСК] X_2 [ПУСК], \dots, X_n [ПУСК]	X_1 [С/П] X_2 [С/П], \dots, X_n [С/П]
X_1, X_2, \dots, X_n в регистры памяти	X_1 [А] [0], X_2 [А] [1], \dots, X_n [А] [9].	X_1 [X→П] [0] X_2 [X→П] [1], ..., X_n [X→П] [d].
8. Счет по программе	Автоматически	
9. Чтение результатов вычислений Y_n с индикатора или регистров памяти	Y_1 : [А] [0], Y_2 : [А] [1], \dots, Y_n : [А] [9].	Y_1 : [П→X] [0], Y_2 : [П→X] [1], ..., Y_n : [П→X] [d].

2. Для приобретения навыков работы «слепым» методом левой рукой произвести набор чисел с последующим гашением:

123, 456, 789, 987, 654, 321
456, 789, 123, 301, 509, 807

3. Для освоения техники выполнения арифметических операций выполнить следующие примеры:

1521+754+868+7117+235
1431,3—114,7—287,1+51,6+916,7
1073×0,25 73,5×21
с точностью до 0,01 (с использованием клавиши округления):
20 : 3 987,5 : 38 40,7 : 57

4. Для усвоения назначения клавиш, имеющих на изучаемой машине, выполнить примеры, используя следующие клавиши:

а) клавишу изменения знака /—/:

130,5+40+(—18,25)	10127+(—7913)+25
12,65×(—3,72)	1,36×10,5×(—0,25)
8 : (—3)	31,5 : (—105)
407×(—2,5) : (—80)	14,07×50 : (—208,3)

б) клавишу обратного деления [\div] без округления и с округлением частного:

$\frac{250}{12+3}$	$\frac{305}{7 \times 309}$	$\frac{625}{25 : 5}$	$\frac{790}{17-2}$
--------------------	----------------------------	----------------------	--------------------

в) клавишу вычисления процента от числа [%]:

25% от числа 1200	39,5% от числа 1,5
43% от числа 405	25,5% от числа 800
1,5% от числа 730	7,3% от числа 98,8

г) клавишу вычисления процентного отношения чисел [/ %]:

5 и 100	396 и 18
60 и 1085	1088 и 21,5
1,5 и 307	4253 и 4153

д) клавишу извлечения квадратного корня [$\sqrt{\quad}$] с точностью до 0,01 с округлением и без округления:

$\sqrt{125}$ $\sqrt{0,125}$ $\sqrt{5,35}$ $\sqrt{115}$ $\sqrt{405}$

е) клавишу возведения в степень [\uparrow]:

3^2 $5,5^3$ $10,2^4$ $-33,5^4$

ж) клавиши вычисления значений прямых и обратных тригонометрических функций:

sin	30°	arc sin	0,5
cos	45°	arc cos	0,888
tg	45°	arc tg	1
ctg	120°	arc ctg	20

з) клавиши вычисления экспоненты e^x и $\ln x$:

e^2	$e^{0,5}$	$e^{3,2}$
$\ln 7,502$	$\ln 2,7183$	$\ln 105$

5. По данным табл. 1.13 рассчитать:

а) количество поступившего зерна от каждого структурного подразделения (гр. 5=гр. 2+гр. 3+гр. 4);

б) количество принятого зерна каждым током и в целом по совхозу, т. е. итоги по гр. 2, 3, 4, 5. Правильность расчетов проверить балансовым методом: сумма итогов по строкам гр. 5 должна быть равной сумме итогов по гр. 2, 3, 4.

Таблица 1.13

Ведомость поступления зерна, ц

Код структурного подразделения совхоза	Номер тока			Всего
	1	2	3	
1	2	3	4	5
01	1285	2105	4001	
02	2184	3084	3962	
03	1964	4014	4840	
04	1825	2895	1998	
05	1938	3134	2194	

Итого

6. По данным табл. 1.14 определить по каждой ко-
рове:

а) годовые плановые и фактические надои молока;

б) процент выполнения годового задания по удою молока.

7. На основании исходных данных табл. 1.15 выполнить расчеты в ведомости движения продукции.

Правильность расчетов проверить балансовым методом.

Примечание. 1. Гр. 5=гр. 2+гр. 3—гр. 4.

Таблица 1.14

Код коровы	Надои по месяцам, кг												Всего за год	Выполнение плана, %
	1	2	3	4	5	6	7	8	9	10	11	12		
01														
План	290	320	400	350	300	280	200	110	—	250	320	280		
Факт	284	351	351	295	295	300	228	148	—	246	315	258		
02														
План	300	310	350	400	420	460	300	160	—	—	400	300		
Факт	390	373	397	370	351	395	293	140	—	90	385	361		
03														
План	—	450	540	600	600	550	500	500	450	350	260	—		
Факт	—	441	590	610	573	545	512	517	435	296	176	—		
04														
План	430	500	540	500	500	520	500	400	210	—	—	300		
Факт	428	472	518	530	486	510	492	387	260	10	—	205		

2. Получить итоги по гр. 2, 3, 4, 5.

3. Наличие продукции на конец года в целом по хозяйству проверить балансовым методом.

Таблица 1.15

Код продукции	Наличие на начало года	Приход	Расход	Наличие на конец года
1	2	3	4	5
2030	200,4	84,7	157,8	
2031	1305,7	107,3	484,4	
2032	807,4	208,7	698,5	
2033	2450,3	842,7	1875,3	

Итого по
хозяйству

8. По данным табл. 1.16 определить годовой фонд основной заработной платы с оплатой отпусков:

- а) на 1 работника (гр. 4=гр. 3×12);
- б) по группам работников (гр. 5=гр. 4×гр. 2);
- в) в целом по цеху растениеводства (сумма строк по гр. 5).

Данный итог определить, используя регистр памяти.

Таблица 1.16

Группа работников	Среднегодовая численность работников	Среднемесячный заработок	Годовой фонд основной заработной платы, с оплатой отпусков, руб.	
			на 1 работника	всего
1	2	3	4	5
Зав. отделением	1	105,65		
Бригадиры	3	98,60		
Учетчики	5	75,50		
Рабочие полеводства	28	65,85		
Прочие работники	13	64,50		

Итого х

9. По данным табл. 1.17 рассчитать показатели использования тракторов различных марок в расчете на 1 трактор (гр. 4=гр. 3 : гр. 2; гр. 6=гр. 5 : гр. 2).

Таблица 1.17

Марка трактора	Среднегодовое количество машин	Выработано машино-дней		Выполнено работ в переводе на эталоинные га	
		всего	в расчете на 1 трактор	всего	в расчете на 1 трактор
1	2	3	4	5	6
Т-75, Т-74, ДТ-75	22	2886		16984	
К-700	2	992		4986	
МТЗ всех видов	43	7520		30931	
Прочие марки всех других видов	11	1401		8829	
Всего в расчете на условный трактор	59	8888		31640	

10. По данным табл. 1.18 рассчитать с точностью до 0,1% структуру затрат на производство яровых зерновых и картофеля, используя рациональные приемы вычислений (деление на константу). Проценты накопить в регистрах памяти. При индикации содержимого регистров памяти на табло должно быть число 100,0.

Таблица 1.18

Статьи затрат	Яровые зерновые		Картофель	
	затраты, руб.	% к итогу	затраты, руб.	% к итогу
1. Заработная плата	20800		31893	
2. Семена и посадочный материал	47646		49367	
3. Горючее и смазочные материалы	8630		3378	
4. Удобрения	44830		16862	
5. Автотранспорт	10026		2910	
6. Накладные расходы	32640		22076	
7. Прочие затраты	67320		26750	
Всего		100,0		100,0

11. На основании данных табл. 1.19 определить:

а) производство молока по каждой ферме (гр. 4 = гр. 2 × гр. 3) и в целом по совхозу (сумма строк по гр. 4). Для определения итога по хозяйству использовать регистр памяти ЭКВМ;

б) удельный вес каждой фермы в общем поголовье коров и в общем производстве молока с точностью до 0,1%. Расчет произвести методом деления на постоянный делитель;

в) средний надой на одну корову по хозяйству (итог по гр. 4 : итог по гр. 2).

Таблица 1.19

Номер фермы	Поголовье коров, голов	Удой на 1 корову, ц	Валовое производство молока, ц	Удельный вес ферм в % к итогу	
				в общем поголовье коров	в общем производстве молока
1	2	3	4	5	6
1	1200	35,5			
2	440	32,0			
3	880	37,4			
4	120	29,5			
5	180	28,0			

Итого по хозяйству

12. На основании данных табл. 1.20 определить:

а) выполнение плана реализации продукции государству с точностью до 0,1 % (гр. 4=гр. 3:гр. 2×100);

б) себестоимость продукции по видам в тыс. руб. (гр. 6=гр. 3×гр. 5:1000) и в целом по хозяйству. Для определения суммарных затрат по хозяйству использовать регистр памяти ЭКВМ.

13. По данным табл. 1.21, используя рациональные приемы вычислений, определить:

а) урожайность овощных культур с точностью до 0,1 ц (гр. 4=гр. 3:гр. 2);

Таблица 1.20

Вид продукции	Продажа государству		Выполнение плана, %	Себестоимость продукции, руб. за 1 ц	Себестоимость всей продукции, тыс. руб.
	план	факт			
1	2	3	4	5	6
1. Зерновые и зернобобовые	3500	5160		7,35	
2. Картофель	12011	12011		6,50	
3. Овощи	550	120		9,80	
4. Молоко	35000	29862		16,50	
5. Мясо крупного рогатого скота	590	560		120,48	
6. Мясо свиней	6200	6750		88,50	
Итого по хозяйству	×	×	×	×	

б) объем продажи овощей государству с точностью до 0,1 ц, учитывая, что реализуется 85% валового сбора овощей (гр. 5=0,85×гр. 3);

в) стоимость реализуемой продукции государству в тыс. руб. (гр. 7=гр. 5×гр. 6:1000);

г) удельный вес каждой овощной культуры в стоимости реализованной продукции в % с точностью до 0,1;

д) среднюю урожайность овощных культур (итог по гр. 4:итог по гр. 2);

е) среднюю цену реализации овощей (итог по гр. 7:итог по гр. 5).

14. Установить имеющуюся в вашем распоряжении программируемую ЭКВМ в режим «Программирование». Нажимая различные клавиши, осуществить индикацию кодов команд и операций. Уяснить назначение клавиш, смысл команд и операций при работе в режиме «Программирование» на ЭКВМ.

15. Составить для имеющейся в вашем распоряжении программируемой ЭКВМ фрагмент программы гашения регистров памяти 1, 2, 3, начиная с нулевого адреса.

Таблица 1.21

Овощные культуры	Посевная площадь, га	Валовой сбор, ц	Урожайность, ц/га	Продано государству, ц	Цена реализации 1 ц, руб.	Стоимость реализованной продукции	
						тыс. руб.	% к итогу
1	2	3	4	5	6	7	8
1. Капуста	148	38319			6,18		
2. Огурцы	43,7	1064			22,30		
3. Помидоры	23,9	989			13,92		
4. Свекла столовая	24	3965			6,38		
5. Морковь	49,2	10668			8,28		
6. Лук	35,5	2064			26,80		

Итого или
в среднем

16. Составить и ввести, используя команды условного перехода, программу вычисления:

$$Y = \begin{cases} X^2, & \text{если } X < 0 \\ 2X + 1, & \text{если } X \geq 0 \end{cases}$$

Результаты вычислений оформить в табл. 1.22.

Таблица 1.22

X	-2	-5	0	5	0,5	-3,2	4	10,5	-3,2
Y									

17. Ввести программу 1 табл. 1.4, контролируя правильность ввода. Выполнить расчеты, используя данные табл. 1.19.

18. Ввести программу 2 табл. 1.5, контролируя правильность ввода. Выполнить расчеты, используя данные табл. 1.15.

Таблица 1.23

Показатели	I вариант			II вариант			III вариант			IV вариант			V вариант			VI вариант		
	I вариант			II вариант			III вариант			IV вариант			V вариант			VI вариант		
	X ₁	X ₂	X ₃	X ₁	X ₂	X ₃	X ₁	X ₂	X ₃	X ₁	X ₂	X ₃	X ₁	X ₂	X ₃	X ₁	X ₂	X ₃
1	23,6	70	1,0	37,6	99	3,0	24,5	72	1,2	34,2	68	4,2	27,0	70	3,2	28,2	91	4,2
2	31,9	79	4,1	18,5	57	1,1	31,1	86	3,0	24,4	46	2,7	19,9	55	2,2	25,0	80	4,1
3	35,2	91	3,8	29,1	76	2,9	15,3	44	1,0	29,1	86	2,4	36,5	73	5,2	12,6	67	3,9
4	36,4	94	3,9	38,5	90	3,0	31,0	89	2,9	23,2	82	1,2	22,5	45	2,9	22,8	67	3,9
5	23,6	72	1,2	18,8	45	0,8	36,0	95	4,3	40,3	99	3,3	33,3	81	4,0	27,8	82	4,9
6	34,0	81	3,9	20,6	39	1,5	18,7	51	1,0	31,1	86	3,0	21,6	43	3,1	14,8	46	2,3
7	38,2	96	4,1	29,6	68	2,1	30,0	63	3,1	19,5	57	1,3	24,6	58	3,2	20,5	55	3,9
8	17,3	45	0,8	36,8	83	3,2	24,6	54	1,7	28,5	70	2,5	40,1	91	4,8	20,4	64	3,2
9	23,8	60	0,7	15,8	35	1,2	42,2	93	4,0	18,0	68	1,4	28,0	64	4,3	15,4	49	2,2
10	19,7	50	1,3	33,4	71	3,0	18,0	47	1,7	32,4	71	2,6	22,5	49	3,6	29,1	85	5,0
m_x																		
D_x																		
σ_x																		
V_x																		
l_x																		

19. Произвести статистическую обработку данных табл. 1.23, где X_1 — урожайность зерновых в ц/га, X_2 — качество почв в баллах, X_3 — дозы внесения минеральных удобрений в центнерах действующего вещества.

а) Ввести программу определения:

$$m_x = \frac{1}{n} \sum_{i=1}^n x_i; D_x = \frac{\sum_{i=1}^n x_i^2 - n \cdot m_x^2}{n-1};$$

$$\sigma_x = \sqrt{D_x}; V_x = \frac{\sigma_x}{m_x} \cdot 100; l_x = \frac{\sigma_x}{\sqrt{n}}.$$

б) Определить значение m_x , D_x , σ_x , V_x , l_x по данным табл. 1.23. Результаты решений по программе поместить в табл. 1.23.

20. Составить и ввести программу определения

$$t_\alpha = \frac{m_1 - m_2}{\sqrt{l_1^2 + l_2^2}},$$

где m_1 и m_2 — среднее значение, l_1 и l_2 — ошибка средних. Определить t_α для произвольных пар показателей табл. 1.23. Полученные значения t_α сопоставить с табличным для $p=0,05$. Сделать выводы о достоверности (существенности) отклонений средних.

Глава 2

МИНИ- И МИКРОЭВМ

2.1. ОБЩИЕ СВЕДЕНИЯ

Мини- и микроЭВМ — это электронные цифровые вычислительные устройства малых габаритов, построенные на больших (БИС) и сверхбольших (СБИС) интегральных схемах.

До недавнего времени различие между мини- и микроЭВМ состояло в том, что основой микроЭВМ являлся микропроцессор, т. е. программно-управляемое функционально законченное устройство обработки цифровой информации, выполненное в виде одной или нескольких БИС. В мини-ЭВМ роль этого устройства выполняли несколько самостоятельных функциональных блоков, связанных между собой сложной электронной схемой с несколько повышенными параметрами.

Развитие микроэлектронной техники и появление более мощных по своим параметрам микропроцессорных систем приводят к стиранию существенных различий между мини- и микроЭВМ с точки зрения пользователя.

Мини- и микроЭВМ характеризуются следующими особенностями:

- большим быстродействием (от 10 тыс. до 1 млн. операций в секунду);

- простотой в эксплуатации;

- хорошо развитым набором внешних устройств;

- магистрально-модульным принципом построения, заключающимся в компоновке и расширении ЭВМ стандартными унифицированными, функционально и конструктивно законченными совместимыми блоками (модулями), что позволяет пользователю собирать необходимую ему конфигурацию ЭВМ в зависимости от решаемых задач;

- хорошо развитыми программными средствами, что обеспечивает возможность программирования на алго-

ритмических языках высокого уровня в режиме диалога пользователя с ЭВМ;

совместимостью на аппаратном и программном уровнях, что позволяет эффективно использовать мини- и микроЭВМ в вычислительных системах и сетях, информационно-поисковых системах и т. д.

Основными направлениями применения мини- и микроЭВМ являются:

управление технологическими процессами;

автоматизация научных экспериментов;

осуществление сопряжения больших ЭВМ с экспериментальными установками;

выполнение научно-технических, экономических расчетов;

использование в качестве терминалов в сложных вычислительных системах и сетях сбора и обработки информации;

использование в качестве АРМ (автоматизированных рабочих мест) для оперативного управления производством;

выполнение функций инструментальных машин для создания программного обеспечения;

использование в качестве обучающих систем, а также в качестве ЭВМ индивидуального пользования и т. д.

В нашей стране наибольшее распространение получили мини-ЭВМ системы малых ЭВМ (СМ ЭВМ) СМ-4, СМ1340, СМ1420, СМ1620; микроЭВМ СМ1800, СМ1810, «Электроника 60», «Электроника НЦ-80-01Д» и построенный на ее основе диалоговый вычислительный комплекс ДВК-2, «Электроника НЦ-80-20/3» (ДВК-3) и др.

СМ ЭВМ выпускаются с 1977 г. и являются результатом совместной работы социалистических стран СЭВ.

В сельском хозяйстве успешно применяются следующие модели мини- и микроЭВМ:

при обработке учетно-плановой информации предприятий — мини-ЭВМ СМ1600, построенная на базе мини-ЭВМ СМ-4 и вычислительного комплекса М-5000; электронная фактурно-бухгалтерская машина «Искра-555», которую по эксплуатационным возможностям можно отнести к мини-ЭВМ;

в системах управления производством — мини-ЭВМ СМ-4, микроЭВМ «Искра 226», выполняющие функции автоматизированных рабочих мест специалистов «АРМ-сельхоз».

Мини- и микроЭВМ по способу использования можно разделить на машины индивидуального и коллектив-

ного пользования. К машинам индивидуального пользования, в которых рабочее место предоставляется одному человеку (пользователю), относятся микроЭВМ «Искра 226», «Электроника-60», СМ1800, СМ1810, ДВК.

К машинам коллективного пользования, предоставляющим возможность одновременной работы с ЭВМ нескольким пользователям, относятся мини-ЭВМ СМ-4, СМ1340, СМ1620 и другие, оснащенные несколькими устройствами обмена информацией — дисплеями.

2.2. ОСНОВНЫЕ УСТРОЙСТВА И ИХ ХАРАКТЕРИСТИКА

Принципы работы мини- и микроЭВМ практически не отличаются друг от друга, а также от принципов работы универсальных больших ЭВМ, однако архитектура их имеет существенные особенности.

Функционально многие модели мини- и микроЭВМ построены из набора модулей и устройств, подключаемых к устройству обмена — интерфейсу.

На рис. 2.1 приводится структурная схема, поясняющая принципы построения и работы мини- и микроЭВМ.

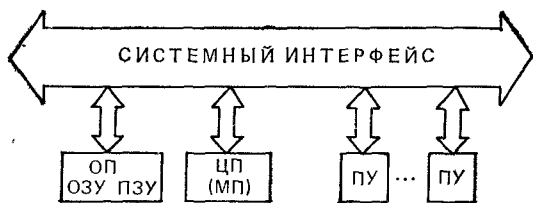


Рис. 2.1. Структурная схема мини- и микроЭВМ

К основным устройствам машин относятся:
центральный процессор (ЦП);
основная память (ОП);
оперативное запоминающее устройство (ОЗУ);
постоянное запоминающее устройство (ПЗУ);
периферийные устройства, включающие внешние запоминающие устройства (ВЗУ) и устройства ввода-вывода информации (УВВ).

Центральный процессор — электронное устройство, обеспечивающее преобразование информации по заданной программе и управление всеми остальными

функциональными устройствами ЭВМ. В него включаются арифметико-логическое устройство (АЛУ), предназначенное для автоматического и логического преобразования информации, и центральное устройство управления (ЦУ), обеспечивающее координацию работы всех устройств ЭВМ. Основными характеристиками ЦП являются быстродействие и длина машинного слова, т. е. максимальный размер обрабатываемых чисел. Длина машинного слова измеряется в двоичных разрядах (битах) и составляет для мини- и микроЭВМ 8—16 бит.

Роль центрального процессора в микроЭВМ выполняет микропроцессор (МП). МП — программно-управляемое устройство на одной или нескольких интегральных микросхемах, осуществляющее процесс обработки цифровой информации и управление этим процессом. В состав МП входят операционный блок, выполняющий арифметические и логические операции; блок микропрограммного управления, определяющий алгоритм функционирования операционного блока в соответствии с поступившей командой; интерфейсный блок, обеспечивающий сопряжение МП с системной магистралью.

Основная (оперативная) память — электронное устройство, предназначенное для кратковременного хранения (запоминания) и (или) выдачи данных. ОП взаимодействует с ЦП (МП) через системную магистраль.

В состав ОП входят оперативное запоминающее устройство (ОЗУ) и постоянное запоминающее устройство (ПЗУ).

ОЗУ обеспечивает возможность оперативного изменения информации, используется для записи, хранения и выдачи информации, в том числе во время выполнения программы. ОЗУ имеет длительность цикла обращения, соизмеримую с длительностью цикла выполнения ЦП (МП) основных операций.

ПЗУ — запоминающее устройство с неизменяемым содержимым памяти, выполняющее роль хранилища констант, стандартных программ (например, вычисления элементарных функций), микропрограмм управления внешними устройствами и т. д. Отметим, что в полупроводниковые ПЗУ информация записывается в процессе изготовления микросхем путем соответствующего соединения запоминающих элементов на поверхности кристалла.

В качестве запоминающих устройств микроЭВМ могут использоваться также программируемое постоянное

запоминающее устройство (ППЗУ), в которое информация заносится однократно потребителем и не может быть изменена; репрограммируемое постоянное запоминающее устройство (РПЗУ), в котором информация может изменяться при помощи специальных средств стирания и записи. Записывается информация пользователем при помощи специального программатора.

Важной характеристикой основной памяти ЭВМ является емкость, измеряемая в килобайтах (Кбайт).

Примечание. 1 Кбайт равен 1024 байтам, 1 байт равен 8 битам, бит — двоичная единица измерения информации.

Мини- и микроЭВМ имеют ОП небольшой емкости, что обусловлено целевым назначением этих машин. Это обеспечивает сравнительную простоту и невысокую стоимость мини- и микроЭВМ по сравнению с большими ЭВМ. Основные характеристики мини- и микроЭВМ содержатся в табл. 2.1.

Таблица 2.1

Марка ЭВМ	Быстродействие, тыс. оп./с	Длина машинного слова, бит	Объем ОП, Кбайт
СМ-4	800	16	32 ÷ 256
«Электроника 60»	400	16	64
«Электроника НЦ-80-01Д»	500	16	64
СМ1800	1000	8; 16	64
«Искра 226»	500	13 десятичных разрядов	64 — для ОЗУ; 8—24 — для ПЗУ

Примечание. Для микроЭВМ в объем ОП включен объем ОЗУ и ПЗУ.

Периферийные устройства, их состав и структура определяются в зависимости от ориентации мини- и микроЭВМ. К ПУ относятся устройства ввода-вывода информации (УВВ) и внешние запоминающие устройства (ВЗУ).

Устройства ввода предназначены для ввода информации и представлены в современных мини- и микроЭВМ пультовыми пишущими машинками (ППУ), видео-терминалами (дисплеями), перфоленточными устройствами считывания информации, устройствами ввода данных с перфокарт. Отдельные ЭВМ имеют возможность прямого ввода информации через специальные каналы связи.

В связи со значительным снижением стоимости машинного времени по сравнению с большими ЭВМ информацию в мини- и микроЭВМ целесообразно вводить не с заранее подготовленного носителя (перфоленды, перфокарты, магнитных носителей), а с клавиатуры дисплеев в диалоговом режиме.

В качестве устройств вывода информации выступают алфавитно-цифровые печатающие устройства на широкую бумажную ленту, ленточные перфораторы, графопостроители, графические и алфавитно-цифровые дисплеи, устройства выдачи управляющей информации по каналам связи.

Особое место среди периферийных устройств занимают внешние запоминающие устройства (ВЗУ). Они предназначены для длительного хранения больших объемов информации. ВЗУ делятся на две большие группы: устройства с прямым доступом, т. е. такие, в которых время доступа не зависит от расположения информации на магнитном носителе;

устройства с последовательным доступом, в которых эта зависимость значительна. К первым относятся различные виды накопителей на магнитных дисках (НМД), ко вторым — накопители на магнитных лентах (НМЛ).

НМД можно разделить на жесткие (постоянные) диски с фиксированными или подвижными головками записи-считывания, со сменным пакетом дисков, с комбинацией сменных и постоянных дисков, накопители типа винчестер) и гибкие магнитные диски. Наиболее распространенными для микроЭВМ являются накопители на гибких магнитных дисках (НГМД). Носителем информации в НГМД является гибкий диск (дискета), покрытый с одной или двух сторон магнитным слоем и помещенный в специальный конверт. Различают мини-дискеты с диаметром 200 мм и 130 мм и микродискеты с диаметром 89 мм. Габариты, масса, диаметр, количество дорожек и емкость дискет значительно колеблются. Так, емкость НГМД с диаметром 200 мм может составлять 1,6 Мбайта (1 Мбайт равен 1024 Кбайта), емкость микродискеты — 437,5 Кбайта. Малые габариты и масса, не слишком строгие требования к условиям хранения и транспортировке, большая емкость делают гибкие магнитные диски весьма перспективными ВЗУ микроЭВМ.

НМЛ — устройства с последовательным доступом к памяти, меньшими по сравнению с НМД скоростями обмена информацией и самой низкой стоимостью хранения информации. Различают НМЛ со стандартной маг-

нитной лентой и НМЛ со стандартными мини-кассетами.

Системный интерфейс осуществляет обмен командами, адресами, данными между ЦП, ОП, ПУ. Системный интерфейс представляет собой унифицированную проводниковую магистраль (для СМ ЭВМ она называется «Общая шина» и состоит из 56 линий связи).

Унификация устройств, их совместимость, магистрально-модульный принцип построения позволяют создавать вычислительные комплексы на базе мини- и микроЭВМ различной конфигурации в зависимости от круга решаемых задач и требований, предъявляемых пользователями.

Рассмотрим подробнее состав наиболее известных и широко применяемых мини-ЭВМ СМ-4 и микроЭВМ «Электроника 60» и СМ1800, микроЭВМ «Искра 226».

СМ-4 — базовый комплект включает в себя: центральный процессор (СМ-4П); системный канал «Общая шина»; оперативное ЗУ (СМ3101); устройство ввода-вывода перфоленточное (СМ6201) со скоростью считывания 300 строк/с и скоростью вывода 50 строк/с; устройство печатающее алфавитно-цифровое СМ6300 (скорость печати 180 зн./с, размер строки 132 символа); алфавитно-цифровой терминал СМ7204 (количество строк 24, размер строки 80, количество символов 96); накопитель на магнитной ленте СМ5002 (10—12 Мбайт); накопитель на малом магнитном диске СМ5400 (емкость накопителя — 13,8 Мбайта).

Дополнительно могут подключаться устройства на гибких магнитных дисках (СМ5603), накопители на кассетных магнитных лентах (РК-1), широкоформатный графический экранный пульт проектировщика (графический дисплей — ШГЭПП2), устройства ввода-вывода дискретных сигналов и др.

«Электроника 60» минимальной конфигурации включает: центральный процессор (ЦП), общий канал обмена информацией, электрическую пишущую машинку (ЭПМ типа «CONSUL-260») для низкоскоростного ввода и вывода данных и программ, перфоратор ленточный (ПЛ-150), устройство чтения лент (FS-1501). Дополнительно могут подключаться блоки оперативной (ОЗУ) и постоянной (ПЗУ) памяти, накопители на гибких магнитных дисках (НГМД), дисплеи, аналого-цифровые и цифроаналоговые преобразователи, устройства прямого доступа.

СМ1800 komponуется из модулей и устройств, объединенных в функциональные группы:

Первая группа — управления и обработки — предназначена для арифметической и логической обработки данных.

Вторая группа — ввода-вывода и связи с другими ЭВМ — предназначена для организации общения оператора с ЭВМ через перфоленточное устройство, видеотерминалы (ВТА), а также для связи с другими ЭВМ через видеотерминалы (ВТА-2000-13, ВТА-2000-30), устройства связи — для подключения с СМ-3, СМ-4.

Третья группа — группа связи с объектом — предназначена для ввода в СМ1800 стандартных сигналов с датчиков и вывода сигналов управления к объектам управляемой системы.

Четвертая группа — внешней памяти — предназначена для создания накопителей расширения информационного обеспечения систем управления на базе устройств внешней памяти, состоящих из накопителей на гибких магнитных дисках и модуля сопряжения с накопителями.

Пятая группа — вспомогательная — предназначена для создания особых режимов питания, контроля и наладки устройств и модулей ЭВМ. В качестве устройств оперативного обмена информацией человека с ЭВМ в вычислительных комплексах, построенных на базе СМ1800, используются видеотерминалы алфавитно-цифровые — ВТА 2000-30, ВТА 2000-31, ВТА 2000-32. Эти устройства используются также в качестве пультов программистов и операторов.

Диалоговые вычислительные комплексы ДВК-2 компактны по построению, по представленным ресурсам их можно отнести к персональным ЭВМ. В состав ДВК-2 входят микроЭВМ «Электроника НМС 1201.2», накопитель на гибких магнитных дисках (ГМД-7012), печатающее устройство, дисплей (15 ИЭ 00-013-01). Внешние устройства подключаются к микроЭВМ через каналы связи. На рис. 2.2 приводится схема пульта управления ДВК-2, включающая следующие зоны: 1 — клавиатура режимов работы; 2 — лампочная индикация режимов работы и состояния терминала; 3 — клавиатура управления курсором; 4 — цифровая клавиатура; 5 — алфавитно-цифровая клавиатура.

Характер взаимодействия дисплея с внешними устройствами определяется режимными клавишами: «ЛИН» — линия (автономный); «ДУП» — дуплекс (полудуплекс); «ПЧ» — печать (передача); «РЕД» — редактирование. В скобках указаны режимы, соответствующие

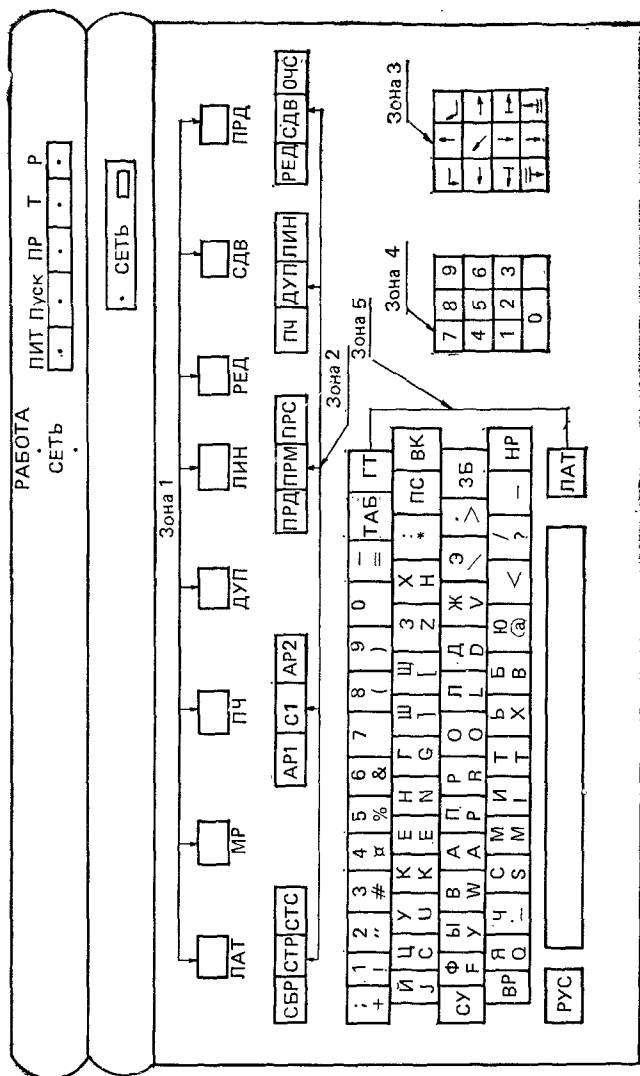


Рис. 2.2. Схема пульта управления ДВК-2

щие включенной клавише. Включение клавиши режимов индицируется соответствующими светодиодами.

Автономный режим применяется при отсутствии обмена информацией с внешними устройствами.

Для передачи данных и команд с дисплея во внешнее устройство необходимо установить режимы «Линия» и «Передача», при этом информацию можно передавать с клавиатуры дисплея (знаковый режим) или запоминающего устройства дисплея (блочный режим). Для включения блочного режима необходимо нажать клавиши ПРС (передача строки) и ПРД (передача страницы).

Для приема данных и команд от внешнего устройства устанавливаются режимы «Линия» и «Дуплекс» нажатием клавиш ЛИН и ДУП.

Выполнение команд в режиме «Редактирование» осуществляется в нижнем (включенном) положении клавиши РЕД и горячем светодиоде РЕД. Команды редактирования обеспечивают смещение маркера при помощи соответствующих клавиш: → (вправо), ← (влево), ↑ (вверх), ↓ (вниз), ↓[—] (в начало следующей строки), ↑ (сдвиг текста вверх), ↓ (сдвиг текста вниз), ←| (смыкание текста), |→ (размыкание текста), СБР (сброс), СТС (стирание строки), СТР (стирание страницы), ПРД (передача страницы), ПРС (передача строки), ПРМ (возобновление передачи) и т. д.

Вывод информации на печать при помощи термопечатающего устройства обеспечивается через разъем «линия» командами «Передача строки» (клавиша ПРС) и «Передача страницы» (клавиша ПРД).

Алфавитно-цифровая клавиатура содержит информационные и функциональные клавиши. Информационные клавиши обеспечивают ввод цифр, букв русского и латинского алфавитов, а также специальных символов. Информационные клавиши работают в двух регистрах: верхнем — клавиша ВР и нижнем — клавиша НР. В верхнем регистре вводятся цифры, буквы русского алфавита, отдельные специальные знаки (; — ., / :). В нижнем регистре вводятся буквы латинского алфавита, специальные символы (+ ! " # % & * < @ > ? \). Функциональные клавиши обеспечивают: РУС — русский регистр; ЛАТ — латинский регистр; ГТ — горизонтальную табуляцию — перемещение маркера вправо до следующей позиции табуляции или ряда 9, 17, 25, ..., 80; ПС — перевод строки; ВК — возврат каретки; ЗБ — стирание символа.

Расположенная между клавишами РУС и ЛАТ центральная нижняя клавиша без обозначения используется для пропуска одной позиции при вводе текста.

Для удобства ввода цифровой информации используется цифровая клавиатура (зона 4), включающая клавиши 0, 1, ..., 9 и клавишу десятичной запятой.

Порядок управления дисплеем следующий:

подключить дисплей к сети переменного тока, присоединив сетевой кабель к трехполюсной розетке, имеющей заземление;

установить сетевой тумблер в положение «ВКЛ», при этом должна загореться индикаторная лампочка «СЕТЬ»;

включить кнопки «СЕТЬ», «ПИТ», «ПР» на пульте управления дисплеем;

вставить гибкий магнитный диск в дисковод и включить тумблер на задней стенке дисковода;

установить режимы работы дисплея нажатием соответствующих клавиш:

а) клавиши СДВ и ОЧС обеспечат индикацию 25-й служебной строки с указанием режимов функционирования дисплея. Нажатие этих клавиш необязательно;

б) клавиши ДУП, ЛИН, РЕД обеспечат работу с внешними устройствами и выполнение команд редактирования;

по окончании работы выключить электропитание всех устройств.

«Искра 226» — микроЭВМ, широко используемая в сельском хозяйстве. Имеется несколько вариантов исполнения «Искры 226». Основой любого варианта исполнения являются: процессор интерпретирующий диалоговый; блок отображения символьно-графической информации (БОСГИ) с размером экрана 31 см по горизонтали, форматом 24 строки по 80 символов в каждой (физический адрес устройства — ФАУ=05); клавишное устройство (ФАУ=01); накопитель на гибких магнитных дисках (ФАУ=18); алфавитно-цифровое печатающее устройство (ФАУ=0С). В зависимости от круга решаемых задач подключаются к процессору также дополнительные периферийные устройства: накопитель на жестких магнитных дисках (кассетах) типа ИЗОТ-1370 (ФАУ=1С); графопостроитель типа Н 306 (ФАУ=14); накопитель на магнитной ленте (ФАУ=0В).

На рис. 2.3 показано клавишное устройство «Искра 226», предназначенное для формирования кодов символов, нанесенных на клавишах, и передачи их на вход

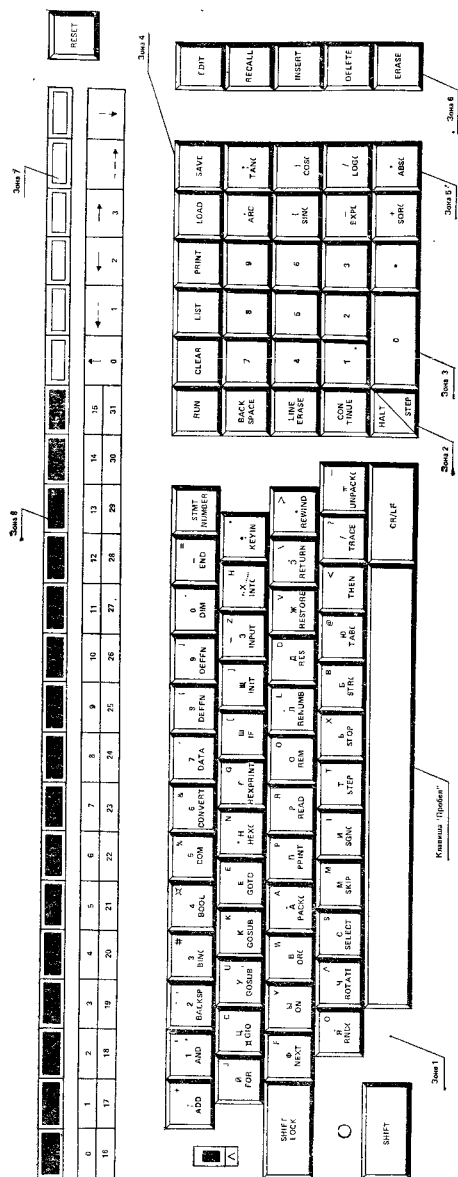


Рис. 2.3. Клавиатура микроЭВМ «Искра 226»

процессора. На клавишном устройстве можно выделить 8 функциональных зон.

1-я зона — алфавитно-цифровая клавиатура. В верхнем регистре вырабатываются прописные латинские буквы и специальные символы, в среднем — прописные русские буквы и специальные символы, в нижнем — операторы языка Бейсик. Для выработки операторов языка Бейсик необходимо, чтобы переключатель «ЛАТ/РУС» находился в положении «ЛАТ» и была нажата клавиша SHIFT или SHIFT LOCK.

2-я зона — клавиши управления счетом и редактированием при вводе: клавиша CONTINUE — продолжение выполнения программы, приостановленной оператором STOP; клавиша HALT/STEP — пошаговое (пооператорное) выполнение программы; клавиша BACK SPACE — удаление последнего символа и слова Бейсика; клавиша LINE ERASE — удаление текущей строки.

3-я зона — стандартная цифровая клавиатура, включающая цифры 0, 1, ..., 9 и клавишу точки для отделения целой части числа от дробной.

4-я зона — клавиши ввода наиболее часто используемых операторов Бейсик. RUN — запуск программы на выполнение, CLEAR — очистка ОП; LIST — вывод информации на экран; PRINT — печать; LOAD — загрузка программы в ОП с магнитного диска; SAVE — запись программы из ОП на магнитный диск.

5-я зона — клавиши арифметических операций (верхний регистр) и стандартных математических функций (нижний регистр). Для выработки стандартных функций нажимают соответствующую клавишу с одновременным нажатием клавиши SHIFT или SHIFT LOCK.

6-я зона — клавиши редактирования текста программы: EDIT — переход к режиму редактирования введенной строки; RECALL — вызов на экран редактируемой строки; INSERT — вставка пробела в текущую позицию строки со сдвигом вправо части строки, начиная с текущего символа; DELETE — удаление текущего символа со сдвигом влево части строки; ERASE — удаление правой части строки, начиная с текущего символа.

7-я зона — клавиши управления курсором, которые используются после нажатия клавиши EDIT, расположенной в 6-й зоне: ← — перемещение курсора на одну позицию влево; → — перемещение курсора на одну позицию вправо; ← — — перемещение курсора на 5 позиций влево; — — → — перемещение курсора на 5 позиций вправо.

8-я зона — клавиши, в двух регистрах реализующие 32 специальные функции пользователя.

На клавишном устройстве имеются также клавиши: клавиша-пробел; клавиша CR/LF — конец ввода строки (команды); клавиша RESET — прекращение выполнения программы.

Необходимо помнить, что ввод программы можно осуществлять символами или, что более рационально, операторами, используя соответствующие клавиши 2-й зоны. Использование при вводе программы русских букв, одинаковых по написанию с латинскими, приводит к ошибке. Следует также отличать клавишу с цифрой 0 (3-я зона) от буквы О (1-я зона).

МикроЭВМ «Искра 226» может эффективно использоваться при составлении, отладке и выполнении Бейсик-программ.

Ввод текста программы в ОП с клавиатуры производится при наличии на экране дисплея знака «:» и курсора. Операторы вводятся посимвольно или разовым нажатием соответствующей клавиши. Строки программы нумеруются, по окончании ввода каждой строки нажимают клавишу CR/LF.

Если пользователь собирается вводить программу с машины, на которой работали до него, он должен очистить ОП, нажав клавиши CLEAR и CR/LF; на экране появится сообщение о готовности ЭВМ

:—

Загрузка программы, записанной на гибком магнитном диске (DC), производится командой LOAD с указанием того, с какого накопителя (правого — F или левого — R) вводится программа. Имя программы заключается в кавычки.

Примеры:

а) загрузка с правого диска программы с именем APOL:

:—LOAD DCF «APOL» [CR/LF];

б) загрузка с левого диска программы с именем BEBI:

:—LOAD DCR «BEBI» [CR/LF].

Редактирование текста программы на ЭВМ «Искра 226» производится:

а) до команды ввода строки, т. е. до нажатия клавиши CR/LF, стиранием символов с возвратом курсора клавишей BACK SPACE или стиранием всей строки клавишей LINE ERASE;

б) исправление уже введенной строки производится повторным ее набором с тем же номером. Для удаления строки необходимо набрать ее номер и нажать клавишу CR/LF. Допускается вставка строк между строк ранее введенной программы.

Команда CLEAR позволяет стереть одну, несколько и все строки программы, например

CLEAR P 7 [CR/LF] — стереть 7-ю строку;

CLEAR P 7,20 [CR/LF] — строки с 7-й по 20-ю;

CLEAR P [CR/LF] — всю программу.

Редактирование можно осуществлять, используя специальный режим, с помощью клавиши EDIT и соответствующих клавиш 6, 7-й зон. По нажатии клавиши EDIT вместо «:» на экране появится *. Далее пользователь клавишами управления курсором (7-я зона) устанавливает курсор на символ, с которого начинается редактирование, и выполняет редактирование, используя клавиши 6-й зоны.

К командам редактирования относится команда вывода на экран текста программы LIST. Варианты использования программы следующие:

а) вывод на дисплей одной строки с указанным номером, например 20:

:__LIST 20 [CR/LF]

б) вывод текста программы между указанными строками, например с 20-й по 60-ю:

:__LIST 20,60 [CR/LF]

в) вывод текста с начала программы до указанной строки, например до 70-й:

:__LIST, 70 [CR/LF]

г) вывод на дисплей текста программы построчно, т. е. по 23 строки:

:__LIST S [CR/LF]

Вывести на экран дисплей следующую страницу можно, нажав клавишу CR/LF;

д) вывод текста программы на печатающее устройство

:__LIST/OC [CR/LF]

Запись программы на магнитный диск производится командой SAVE с указанием, на какой диск (правый F или левый R) записывается программа. Имя программы, не превышающее 8 символов, заключается в кавычки.

Пример записи программы BARK на правый диск:

:__SAVE DCF «BARK» [CR/LF]

При дальнейшем использовании этой программы необходимо загрузить ее в ОП командой LOAD:

:__LOAD DCF «BARK» [CR/LF]

Запуск программы, находящейся в ОП, производится командой RUN.

2.3. ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ

Программное обеспечение (ПО) мини- и микроЭВМ — совокупность методов, приемов и средств, позволяющих разрабатывать программы решения задач пользователя, эффективно управлять ресурсами ЭВМ (процессором, оперативной памятью, периферийными устройствами), а также обеспечивать проверку работоспособности оборудования.

ПО принято делить на системное программное обеспечение и прикладное программное обеспечение.

Системное программное обеспечение (СПО) включает исполнительные операционные системы, диалоговые системы, а также инструментальные операционные системы;

Прикладное программное обеспечение объединяет библиотеки стандартных программ, пакеты прикладных программ (ППП) для редактирования текстов, подготовки, статистической обработки данных, АРМ и т. д.

Кроме того, ПО включает тестовые средства для контроля работоспособности отдельных устройств ЭВМ.

Основу ПО мини- и микроЭВМ составляют *операционные системы*, главными функциями которых являются:

динамическое перераспределение ресурсов ЭВМ между несколькими программами пользователей (планирование работы процессора, распределение и защита памяти, управление вводом-выводом и т. д.);

управление операциями обмена между периферийными устройствами и оперативной памятью;

обмен информацией в диалоговом режиме между ЭВМ и пользователями и т. д.

Специфичность функций операционных систем мини- и микроЭВМ связана с магистрально-модульным принципом построения и ограниченной емкостью оперативной памяти данных машин. Отметим также, что наличие дисплеев и операционных систем, обеспечивающих

посредничество между пользователем и ЭВМ, способствуют тому, что любой пользователь, имеющий определенный минимум знаний по взаимодействию с вычислительной машиной и основам программирования, может самостоятельно разрабатывать и выполнять программы, а также обрабатывать информацию, используя ППП.

Операционные системы можно разделить на исполнительные и инструментальные. Исполнительные ОС обеспечивают функционирование устройств в различных режимах.

Инструментальные ОС обеспечивают выполнение технологических этапов процессирования задач: ввод исходного текста программ, трансляцию, редактирование связей, выполнение.

В состав инструментальных операционных систем мини- и микроЭВМ входят: интерпретатор Бейсика, программы-трансляторы с алгоритмических языков Фортран IV, ПЛ/М, Паскаль и др.

В главе 4 данного пособия показаны основные компоненты и последовательность работы с интерпретатором Бейсик на ЭВМ СМ-4, ДВК-2, а также состав и использование основной операционной системы ОСРВ мини-ЭВМ СМ-4.

Отметим некоторые особенности отдельных алгоритмических языков для мини- и микроЭВМ:

Фортран-80 разработан для программирования микроЭВМ, построенных на базе микропроцессора серии К580, при этом учтены требования программной совместимости с широкоиспользуемым языком Фортран IV и особенности микропроцессорной техники. Язык имеет средства, обеспечивающие ввод программы и данных не с перфокарт, а непосредственно с терминала, а также средства для работы программ с терминальными устройствами (дисплеем, телетайпом и т. д.). Язык удобен для программирования алгоритмов, содержащих значительный объем вычислительных операций;

Бейсик — популярный язык программирования для начинающих, широко используемый в силу своей простоты и диалогового характера работы в процессе разработки и отладки программ на мини- и микроЭВМ;

ПЛ/М — язык системного программирования для микроЭВМ и микропроцессорных систем, построенных на базе микропроцессора серии К580. Язык ориентирован на те типы данных и операций, которые близки к используемым на уровне машинных языков и допускают эффективную реализацию;

Паскаль — язык, используемый, аналогично Бейсику, в основном для обучения. Программы, разработанные на языке Паскаль, удобно читать, так как в них на процедуры и функции ссылаются при помощи имени, а не номера строки, а каждая переменная объявляется указанием ее типа. На основе языка Паскаль разработаны более поздние языки, такие, как Модуль-2, АДА и др.

Рассмотрим кратко программное обеспечение мини-ЭВМ СМ-4, микроЭВМ «Искра 226», ДВК-2.

Программное обеспечение СМ-4 включает:

операционные системы РАФОС, ДОСРВ, ОСРВ, ДИАМС, обучающую систему Бейсик, которые совместны с аппаратурой обеспечивают следующие режимы: интерактивный, пакетный, мультипрограммный, работу в реальном масштабе времени, в режиме разделения времени;

ППП, включающий программы обработки данных методами математической статистики, пакет программ методов оптимизации, имитационного моделирования, сетевого планирования и управления и т. д.

Перечислим основные функции некоторых операционных систем.

РАФОС (Разделения Функций Операционная Система) предназначена для сложных систем управления технологическим оборудованием и научными экспериментами, а также для разработки и отладки программ с применением алгоритмических языков высокого уровня в интерактивном и пакетном режимах.

ДОСРВ — дисковая операционная система, предназначенная для совместного выполнения нескольких задач в реальном масштабе времени в системах телекоммутиции, информационно-поисковых системах, а также для разработки и отладки программ с применением языков Бейсик, Фортран IV, Паскаль.

ОСРВ — универсальная по своим функциональным возможностям операционная система, обеспечивающая мультипрограммный режим и режим реального времени.

ДИАМС — система, ориентированная на управление базами данных и решение информационно-логических задач в системах оперативного управления, в системах управления научными и экономическими расчетами в многопользовательском режиме коллективного доступа к базам данных.

Обучающая система Бейсик — операционная система, применяемая в области образования для организации

учебного процесса в диалоговом многотерминальном режиме.

Тест-мониторная операционная система (ТМОС) для проведения технического обслуживания машины.

Программное обеспечение микроЭВМ «Искра 226» существенно отличается от ПО других микроЭВМ и имеет несколько ограниченные возможности, что частично компенсируется наличием специальных клавиш, объединенных в функциональные зоны «Редактирование текста программы», «Управление курсором», «Специальные функции пользователя» и т. д. На «Искре 226» возможны следующие режимы работы:

режим непосредственного счета (НС);

режим программирования на языке Бейсик;

работа с использованием ППП.

При работе в режиме непосредственного счета выполняемые операторы языка Бейсик не нумеруются и не запоминаются в памяти машины. Для получения результата на экране дисплея используется оператор PRINT. Строку непосредственного счета до нажатия клавиши окончания ввода CR/LF можно редактировать, используя клавиши BACK SPACE — стирание символа с продвижением курсора влево и LINE ERASE — стирание строки, а также с помощью клавиатуры управления курсором, предварительно нажав клавишу EDIT — переход в режим редактирования.

Работа в режиме непосредственного счета делится на два этапа.

Первый этап заключается в подготовке машины к работе и предполагает следующие действия:

включить процессор. На экране дисплея появится сообщение

ЗАГРУЗЧИК

:—

Одновременное изображение на экране двоеточия и курсора свидетельствует о том, что машина не производит никаких операций и готова к приему информации;

установить специальный (системный) диск в левое гнездо дискового устройства;

включить накопитель на гибких магнитных дисках (НГМД);

набрать на клавиатуре текст, контролируя его на экране дисплея:

LOAD R#1, где 1 — номер загружаемой версии языка Бейсик. Вместо 1 может быть любая шестнадцатеричная цифра.

Нажать клавишу CR/LF, что вызовет появление на экране сообщения

READY

:—

Второй этап заключается в выполнении вычислений. Используя оператор PRINT и соответствующие элементы и конструкции языка Бейсик, в режиме непосредственного счета произвести вычисления. Например, пусть требуется вычислить размер вклада в 1000 руб. по истечении 4 лет при 3 процентах годовых по формуле

$$B = A (1 + P/100)^T,$$

где A — начальный вклад;

P — число годовых процентов;

T — срок хранения вклада, лет.

Для выполнения примера после сообщения :— набираем

PRINT 1000 * (1 + 3/100) ^ 4

На экране появится ответ 1125,5.

При работе в режиме диалогового программирования на языке Бейсик машину следует подготовить к работе так же, как и в режиме непосредственного счета, далее построчно вводить с клавиатуры в память машины программу. Набор строки заканчивается нажатием клавиши CR/LF. После ввода последней строки программы дается команда RUN нажатием клавиш RUN и CR/LF.

Отметим, что в состав языка Бейсик на «Искре 226» включен ряд команд редактирования, повышающих эффективность работы пользователя при процессировании задач.

Режим непосредственного счета и программирования на языке Бейсик удобен при использовании «Искры 226» в качестве ЭВМ индивидуального пользования. Наряду с этим «Искра 226» рекомендуется для использования в сельском хозяйстве на различных уровнях управления, включая РАПО и сельскохозяйственные предприятия. С этой целью разработано прикладное программное обеспечение «Искра 226», включающее универсальную диалоговую систему проведения плановых и аналитических расчетов «Нива», диалоговую систему управления локальным банком данных «Колос», ППП «АРМ-сельхоз».

Системы «Нива», «Колос» обеспечивают:

ввод (загрузку) данных, их корректировку, считывание, запись, просмотр, распечатку;

обработку данных в диалоговом режиме при помощи арифметических и матричных операций. В качестве основной обрабатываемой единицы в системах выступает задача — таблица, которая состоит из головки (годы), левой части (наименование объектов) и матрицы с данными;

выполнение общесистемных функций (копирование, ведение справочника объектов и т. д.).

ППП «АРМ-сельхоз» включает ряд программ: программу графического представления данных для автоматической подготовки и графического отображения в виде круговых, прямоугольных диаграмм или графиков справочной или отчетной информации, а также для оперативного анализа числовой информации и результатов расчетов; программу «Редактор текста», осуществляющую различные операции (просмотр, редакцию, печать, стирание, сжатие и т. д.) с текстами и каталогами, и ряд других программ.

Использование прикладного программного обеспечения микроЭВМ «Искра 226» в сельском хозяйстве позволит значительно сократить сроки, повысить качество и эффективность обработки информации.

Программное обеспечение диалоговых вычислительных комплексов, ориентированное на различные области применения, включает операционную систему разделения функций (ОС ДВК) и систему контроля работоспособности устройств (ТМ ОС ДВК).

ОС ДВК построена на основе операционной системы РАФОС, что обеспечивает программную совместимость с ЭВМ «Электроника 60» и СМ ЭВМ. ОС ДВК предназначена для решения задач в реальном масштабе времени и разработки программ в режиме диалога.

Основными составляющими частями ОС ДВК являются: управляющая система, ядром которой является монитор; система программирования, редактирования и отладки программ; файловая система, системные библиотеки и программы.

Монитор выполняет функции копирования, стирания, переименования файлов, вывода каталога, запуска программы, задания даты, вызова прикладных, системных, сервисных программ и др.

Система программирования позволяет выполнять технологические этапы процессирования (ввод исходного текста программы, трансляцию, редактирование свя-

зей, выполнение) задач, запрограммированных на языках макроассемблера, Фортран IV, Бейсик. В ОС ДВК входят транслятор с макроассемблера — MACRO, компилятор Фортрана — FORTRAN, интерпретатор Бейсика — BASIC.

Каждая система программирования вызывается по команде монитора:

R «имя системы»

Основные компоненты и последовательность работы с интерпретатором Бейсик на ДВК-2 показаны в 4.2.2 данного пособия. Здесь же отметим диалоговый режим работы с интерпретатором, что значительно облегчает пользователю отладку Бейсик-программ. Обмен информацией осуществляется по схеме «оперативная память — периферийное устройство», и наоборот.

Файловая система обеспечивает создание, переименование, сравнение, копирование, обслуживание файлов на гибких магнитных дисках.

Системные библиотеки и программы обеспечивают связь между монитором и драйверами (программными средствами, управляющими работой периферийных устройств), системными и прикладными программами и т. д.

2.4. ЗАДАНИЯ

1. Перечислите области применения мини- и микроЭВМ.

2. Приведите основные характеристики, присущие мини- и микроЭВМ, микропроцессорам.

3. По каким направлениям и какие модели ЭВМ внедряются в сельское хозяйство?

4. Поясните принцип построения и мини- и микроЭВМ, приведите схему.

5. Перечислите и дайте краткую характеристику основных устройств мини- и микроЭВМ.

6. Охарактеризуйте техническое и программное обеспечение мини-ЭВМ СМ-4.

7. Охарактеризуйте техническое и программное обеспечение диалогового вычислительного комплекса ДВК-2.

8. Каково назначение информационных и функциональных клавиш на пульте управления дисплея ДВК-2?

9. Каков порядок управления дисплеем ДВК-2?

10. Дайте краткую характеристику микроЭВМ «Искра 226».

11. Каково назначение основных клавиш клавиатуры «Искра 226»?

12. Какова последовательность работы пользователя в режиме непосредственного счета на «Искре 226»?

13. Выполните расчеты по табл. 1.15, 1.17, 1.18 на «Искре 226» в режиме непосредственного счета.

14. Какие системы и прикладные программы разработаны на «Искре 226» для использования в сельском хозяйстве? Перечислите основные функции ППП «АРМ-сельхоз» для «Искры 226».

15. Каков состав и назначение программного обеспечения мини-ЭВМ СМ-4- Дайте краткую характеристику основных операционных систем ОСРВ, ДОС, РАФОС, ДИАМС.

16. Каков состав и назначение программного обеспечения ДВК-2? Дайте краткую характеристику ОС ДВК.

Глава 3

БЕЙСИК — ЯЗЫК ПРОГРАММИРОВАНИЯ ДЛЯ МИНИ- И МИКРОЭВМ

3.1 ОБЩИЕ СВЕДЕНИЯ

В состав программного обеспечения большинства современных мини- и микроЭВМ, а также персональных компьютеров входит язык высокого уровня Бейсик (BASIC).

Бейсик является одним из простых алгоритмических языков высокого уровня. Английское слово BASIC является аббревиатурой от Beginner's All-purpose Symbolic Instruction Code — универсальный код символических инструкций для начинающих. Существенное преимущество Бейсика — возможность работы в диалоговом режиме. В настоящее время Бейсик — один из самых распространенных языков программирования для мини- и микроЭВМ автоматизированных обучающих систем.

3.2. ЭЛЕМЕНТЫ И ОСНОВНЫЕ КОНСТРУКЦИИ ЯЗЫКА

В качестве основных символов языка применяются: прописные буквы латинского алфавита от A до Z; десятичные цифры от 0 до 9; знаки арифметических операций: + (сложение), — (вычитание), * (умножение), / (деление), ^ (возведение в степень)¹; знаки операций отношения: = (равно), > (больше), < (меньше); специальные знаки: запятая, точка, точка с запятой, вопросительный знак, круглые скобки (открывающая и закрывающая), " (кавычки), ' (апостроф), # (знак но-

¹ Знак возведения в степень в различных версиях интерпретатора может изображаться по-разному: ^, либо ↑, либо ˆ.

мера), %, © (знак денежной единицы)¹, ␣ (пробел), \ (обратная дробная черта), @ (коммерческое «а»).

Примечание. Буквы русского алфавита разрешается использовать только в комментариях и текстовых константах, выводимых на печать.

Основные элементы данных — константы и переменные. В Бейсике используются три типа *констант*: вещественные, целые и строковые.

Вещественная константа — это последовательность, состоящая из одной или нескольких десятичных цифр; возможен знак «плюс» или «минус».

Десятичная точка в явном виде может отсутствовать, в этом случае она предполагается помещенной за последней (правой) цифрой.

Например, константа 295 равносильна 295. либо 295.0; константа -6, равносильна — 6.0; .95 равносильно 0.95 и т. д. (Использование экспоненциального представления чисел здесь не рассматривается.)

Целые константы используются для представления чисел количественных характеристик реквизитов, существующих только в целом выражении (даты, количество студентов в группе и т. п.). Они представляют собой последовательность десятичных цифр, которые могут начинаться знаком «плюс» либо «минус» и заканчиваются обязательно знаком %, который служит признаком целого числа. Например, 1987%, 25% и т. д.

Строковые константы — это последовательность буквенно-цифровых и других символов, допустимых в Бейсике и ограниченных с обеих сторон кавычками либо апострофами. Например, «печать чисел», 'пункт шестой' и т. п.

Переменные — величины, изменяющие свое значение в процессе выполнения программы; допускаются типы: вещественный, целый, строковый.

Вещественные переменные обозначаются одной буквой либо одной буквой, за которой следует одна цифра. Например, A, B5, R8.

Целочисленное значение, присвоенное вещественной переменной, будет печататься как целое, но во внутреннем представлении ЭВМ оно рассматривается как число с десятичной точкой справа.

Целые переменные могут иметь имя, состоящее из одной буквы, либо буквы, за которой следует одна цифра, и должны заканчиваться обязательно знаком %.

¹ В тексте книги по технологическим причинам знак ⑈ заменен на знак ©.

Примеры целых переменных: A%, B8%; C1%. Следует подчеркнуть, что в случае, если целой переменной присваивается значение вещественной, дробная часть числа будет попросту при этом отбрасываться.

Именем строковой переменной может быть буква либо буква и цифра, но в конце обязателен знак @, который и служит признаком строковой переменной. Например, B1@; A@, S6@ и т. д.

Длина символьной строки, являющейся значением строковой переменной, допускается не более 255 символов.

Отметим, что вещественная целая и строковая переменные, имена которых начинаются с одной и той же буквы и цифры, с точки зрения Бейсик-интерпретатора представляют собой различные переменные. Например:

K6 — вещественная переменная;

K6% — целая переменная;

K6@ — строковая переменная.

Массивы представляют собой упорядоченную последовательность величин, обозначенных одним именем. Элементы массива — это составляющие его величины. Имя элемента массива указывается именем данного массива, за которым в круглых скобках записывается его индекс (или индексы), означающий положение элемента в массиве; индексы разделяются запятыми. Например, M2(1,4) может обозначать элемент массива стоящий в первой строке, в четвертом столбце.

Сам массив в операторе описания DIM указывается с тем же именем, а в скобках указываются максимальные значения индексов, например

10 DIM A(2,3), B(7,4)

Следует отметить, что в некоторых версиях Бейсика нумерация индексов начинается с нуля, т. е., например, двумерный массив, состоящий из 9 элементов (3×3), в операторе описания должен обозначаться DIM R4(2,2), а не DIM R4(3,3), так как первым в нем будет элемент R4(0,0), а не элемент R4(1,1).

Выражениями в Бейсике являются любые комбинации констант, переменных, функций, соединенных с помощью знаков арифметических операций, операций отношения и скобок. Например, арифметическое выражение

$(Y*3.4+B)/(C+D*X)$

Выражения отношения служат для сравнения величин и используют знаки операций отношения: = (рав-

но), $<$ (меньше), $>$ (больше), $< =$ (меньше или равно), $> =$ (больше или равно), $< >$ (не равно). В том числе могут сравниваться строковые значения (они сравниваются посимвольно, слева направо).

Бейсик содержит ряд системных функций, облегчающих труд программиста. Для вычисления функции нужно указать ее имя и в скобках список аргументов. Если функция помещена в каком-либо выражении, интерпретатор вычисляет ее значение и затем продолжает вычислять значение этого выражения, как если бы вместо функции было указано ее значение.

Из наиболее широко применяемых можно назвать: SIN, COS, SQR (корень квадратный из аргумента), EXP (экспоненциальная функция), LOG (вычисление натурального логарифма), LOG 10 — вычисление десятичного логарифма, INT определяет целую часть аргумента с округлением результата. Например:

SIN (M*10+R) ; LOG (X+Y)

Наряду с этим Бейсик предоставляет возможность пользователю создавать собственные функции и использовать их в программе.

Рассмотрим здесь основные операторы Бейсик-программы.

Оператор присваивания LET вызывает вычисление значения арифметического выражения, стоящего справа от знака равенства, и присваивает найденное значение переменной, стоящей слева. Оператор применяется и для присваивания строковых значений строковым переменным. В некоторых версиях Бейсика имя оператора LET является необязательным и может опускаться.

Формат оператора:

НСТР LET переменная = выражение

где НСТР — номер текущей строки программы (не нужен в диалоговом режиме);

LET — необязательное имя оператора;

переменная — переменная, значение для которой вычисляется (присваивается);

выражение — выражение, которое определяет новое значение переменной.

Например:

20 LET A = 42.5 или 20 A = 42.5
30 Y (I) = M1(5) * SIN(X) + B(I)

Оператор безусловного перехода GO TO вызывает нарушение естественного порядка выполнения операторов

ров программы и обеспечивает переход к строке, номер которой указан в нем. Затем выполнение программы будет продолжаться снова последовательно, начиная со строки, на которую был произведен переход.

Формат оператора:

НСТР GO TO НСТР1

где НСТР1 — номер строки, на которую осуществляется переход.

Отметим, что в Бейсик-программе каждая строка в обязательном порядке должна иметь номер, который и служит меткой. В одной строке допускается записывать два и более оператора; в этом случае операторы должны разделяться обратной дробью — наклонной чертой (\backslash).

Пример

```
10 A = 9 \ GO TO 30
20 A = A*2
30 B (1) = C*D+A
```

При такой записи строка программы 20 окажется невыполненной.

Оператор GO TO должен быть либо единственным, либо последним оператором в данной строке программы.

Операторы условного перехода IF THEN и IF GO TO передают управление другому оператору в зависимости от истинности (или ложности) выражения.

Формат оператора IF THEN:

НСТР IF вып.отн. THEN НСТР1

или

НСТР IF вып.отн. THEN ОПЕРАТОР

где вып. отн. — проверяемое условие, причем выражение отношения допускается как арифметическое, так и строковое;

НСТР1 — номер строки, куда осуществляется переход, если условие окажется выполненным;

ОПЕРАТОР — оператор, к которому совершается переход в случае, если условие окажется выполненным.

Если условие, проверяемое в операторе, окажется невыполненным, управление будет передано строке, следующей за строкой с оператором IF.

Пример

```
20 IF N = 0 THEN Y = A*2+B
30 IF X > A THEN 120
```

Следует отметить, что после THEN в операторе IF допускается запись следующего условного оператора, например

```
40 IF A1 > A2 THEN IF A2 < A3 GO TO 60
```

Операторы FOR и NEXT служат для организации циклически повторяемого участка в Бейсик-программе, причем перед каждым повторением будет автоматически проверяться условие выхода из цикла.

Формат оператора FOR:

```
НСТР FOR перем. = вып.1 TO вып.2[STEP вып.3]
```

где перем. — переменная цикла (индекс цикла, управляющая переменная);

вып. 1 — начальное ее значение (любое арифметическое выражение);

вып. 2 — конечное ее значение;

вып. 3 — приращение ее (шаг изменения).

В случае, если вып. 3=1, его в операторе FOR не указывают (он предусматривается по умолчанию).

Операторы FOR и NEXT используют только совместно: оператор FOR — определяет начало цикла (и проверку условия выхода из цикла, когда переменная цикла превысит свое заданное конечное значение), а оператор NEXT — конец циклического участка.

Формат оператора NEXT:

```
НСТР NEXT перем.
```

где перем. — переменная цикла, указанная в операторе.

Пример

```
20 FOR I = 1 TO 30 STEP 3  
30 X(I)=I*A \ Y(I)=B+X(I)*A  
40 NEXT I
```

Либо по умолчанию:

```
20 FOR I=1 TO 60  
30 FOR K=1 TO 40  
40 X(I,K)=Y(I)*B(K)+4  
50 NEXT K  
60 NEXT I
```

Из этого примера видно, что допускается вкладывать друг в друга операторы цикла; при этом внутренний цикл должен заканчиваться до окончания внешнего цикла.

Операторы STOP и END используются соответственно для приостанова и завершения выполнения Бейсик-программы. Оператор END должен быть последним в программе.

Формат его:

НСТР END

Для удобства организации общей точки выхода в Бейсик-программе можно рекомендовать присваивать этой строке всегда номер 9999; однако это не является обязательным.

Оператор STOP может помещаться в любой строке программы. Формат его:

НСТР STOP

Этот оператор вызывает приостанов выполнения данной программы, причем на экран дисплея пользователю будет выводиться сообщение с указанием номера строки останова, например

STOP AT LINE НСТР

Ответом READY сообщается пользователю о готовности машины работать в диалоговом режиме.

Оператор STOP приостанавливает выполнение программы, однако не закрывает файлы и не завершает работу программы (в противоположность оператору END).

Приостанов программы дает возможность пользователю распечатать (или вывести на экран дисплея) значения переменных, изменить их значения и т. п., т. е. оператор STOP является удобным средством отладки программ. Количество используемых операторов STOP в Бейсик-программе не ограничено.

Для того чтобы продолжить выполнение программы после приостанова, достаточно в диалоговом режиме указать номер той строки программы, с которой следует продолжить ее выполнение, т. е. в ответ на подсказку READY набрать на клавиатуре

GO TO НСТР1

где НСТР1 — требуемый номер строки продолжения работы, причем этот номер совсем не обязательно должен быть следующим за номером останова.

Оператор INPUT позволяет вводить и анализировать данные с клавиатуры дисплея в процессе выполнения программы.

Формат оператора INPUT:

НСТР INPUT перем.1 [, перем.2, перем.3,...]

где перем. 1, перем. 2, перем. 3 — элементы данных в программе, которым присваиваются вводимые значения.

Квадратные скобки указывают на необязательность многих переменных в списке оператора.

Встретив в программе оператор INPUT, машина выведет на экран знак вопроса и приостановит дальнейшее выполнение программы в ожидании ввода значений для указанных в списке переменных. Значение набирается на клавиатуре вслед за знаком вопроса и нажимается клавиша ВК. При этом анализируется соответствие типа переменных в программе вводимым значениям. Например, в программе записано:

```
10 INPUT D1%, D2○
```

На экране дисплея будет:

```
? 1987 %
```

```
? «год»
```

Количество вводимых данных должно точно соответствовать количеству переменных в операторе INPUT. При вводе недостаточного количества интерпретатор снова будет выводить знак вопроса, а избыток данных вызовет предупреждение об ошибке; лишние значения игнорируются.

Строковые переменные можно вводить и не заключенными в кавычки, но в этом случае следует поместить начальный и конечный пробелы (иначе будут игнорироваться первый и последний символы).

Оператор PRINT выводит данные на экран дисплея в процессе выполнения программы. Формат его:

```
НСТР PRINT список элементов
```

где список элементов — это перечень констант, переменных, строковых и арифметических выражений, разделяемых запятой либо точкой с запятой.

При отсутствии списка элементов оператор PRINT выводит строку пробелов (оставляет пустую строку). Если элементом списка является выражение, то интерпретатор вычисляет его значение и печатает результат, например

```
10 A5=8 \ A3=4
```

```
20 PRINT "значение A5 + A3 = "; A5 + A3
```

В результате выполнения этих операторов на экране дисплея будет выведено

```
Значение A5+A3=12
```

Следует отметить, что в операторе PRINT предусмотрено размещение печатаемых значений по зонам в каждой строке (одна зона печати охватывает 14 позиций). При этом значения, разделенные в списке точкой с запятой, печатаются непосредственно друг за другом,

а значения, разделенные запятыми, начинают печататься в данной строке только с первой позиции следующей зоны, оставляя множество пробелов.

После каждого оператора PRINT выполняется перевод строки и возврат каретки. Всего в строке предусмотрено 5 зон (т. е. 70 позиций). Если последняя зона в строке будет заполнена, будет продолжаться печать с начала первой зоны следующей строки.

Бейсик обеспечивает возможность обрабатывать файлы данных, находящиеся на магнитном диске.

Передача данных производится по внутренним каналам, номера которых в программе нужно связать с именем определенного файла, т. е. перед обращением к файлу его необходимо открыть, а при завершении работы с файлом — закрыть (и тем самым освободить данный канал). Для этих целей используются операторы OPEN и CLOSE.

Оператор OPEN открывает файлы как для ввода, так и для вывода, связывая номер канала с именем файла на диске. Формат его:

```
НСТР OPEN стр.выр.FOR INPUT AS FILE #выр.1
```

либо

```
НСТР OPEN стр.выр. FOR OUTPUT AS FILE #выр.1
```

где стр. выр. — строковая константа или строковое выражение, задающее спецификацию (имя) файла, заключенное в кавычки;

выр. 1 — номер присваиваемого канала ввода-вывода, связываемый с указанным файлом.

Оператор OPEN FOR INPUT открывает существующий уже на диске файл для считывания его в оперативную память ЭВМ.

Оператор OPEN FOR OUTPUT подготавливает создание нового файла путем записи информации из оперативной памяти на диск. Если при этом на диске уже был файл с таким же именем, то будет записана новая его версия (с номером версии на 1 больше).

Примеры:

```
10 OPEN "DANN" FOR OUTPUT AS FILE#1
```

```
20 OPEN "DATA4" FOR INPUT AS FILE#2
```

Оператор CLOSE закрывает файлы, как вводные, так и выводные, освобождая каналы, им соответствовавшие. Формат его:

```
НСТР CLOSE #выр.1 [выр.2, выр.3,...]
```

где выр. 1, выр. 2, выр. 3 — номера каналов ввода-вывода, подлежащие освобождению.

Если в программе указать оператор CLOSE без задания номеров каналов, то будут закрыты все открытые в программе файлы, например

```
90 CLOSE
```

По этой команде будут закрыты все файлы.

В простейшем случае в программе ведется работа с файлами с последовательной организацией. Следует иметь в виду, что такие файлы могут быть открыты либо для ввода, либо для вывода, но ни в коем случае не одновременно.

Если в операторе OPEN пропущены слова FOR INPUT либо FOR OUTPUT, то интерпретатор Бейсика откроет файл для считывания из него, если файл уже имеется на диске, и для формирования нового файла, если на диске его еще нет.

Оператор INPUT # производит считывание файла, открытого предварительно оператором OPEN.

Формат оператора:

```
НСТР INPUT #выр.,перем.1 [, перем.2, перем.3,...]
```

где выр. — номер канала ввода-вывода, связанного с файлом;

перем. 1, перем. 2 — элементы данных программы, которым присваиваются вводимые значения.

Например:

```
30 INPUT #1, N, H, P
```

Для обнаружения признака конца файла в Бейсике используется **оператор IF END #**. Обнаружив конец обрабатываемого файла, этот оператор передает управление той строке программы, номер которой в нем указан. Формат оператора:

```
НСТР IF END #выр. THEN НСТР1
```

или

```
НСТР IF END #выр. THEN ОПЕРАТОР
```

где выр. — номер канала ввода-вывода, связанного с обрабатываемым файлом;

ОПЕРАТОР — тот оператор, который надлежит выполнить при достижении конца файла.

Допустимым также является формат

```
НСТР IF END #выр. GO TO НСТР1
```

Признак конца файла с последовательным доступом обнаруживается, когда в файле данных больше нет.

Для помещения данных в определенный файл применяется оператор PRINT#. Его формат:

НСТР PRINT #выр, список

где выр. — номер канала вывода, связанного с файлом; список — список элементов программы, представленных в виде арифметических или строковых выражений, разделенных запятой либо точкой с запятой.

Оператор PRINT# без списка аргументов поместит в файл строку пробелов. Отметим, что если значение номера канала выр.=0, то вывод информации будет производиться на дисплей.

Для помещения в программе различного рода пояснений (комментариев, ремарок) используется оператор REM. Его формат:

НСТР REM текст комментария

Комментарий нужен только в распечатке программы пользователя и не оказывает никакого влияния на выполнение программы.

Пример

10 REM Группа 1125 Леонтьев

20 REM Лабораторная работа № 2

Непосредственный (не программный) режим работы Бейсик-интерпретатора является диалоговым режимом: каждый оператор, набранный на клавиатуре, выполняется тотчас же после нажатия клавиши <BK>. В диалоговом режиме номер строки перед оператором отсутствует, что и служит признаком диалогового режима¹.

3.3. СОСТАВЛЕНИЕ ПРОГРАММ ДЛЯ РЕШЕНИЯ ЭКОНОМИЧЕСКИХ ЗАДАЧ

Рассмотрим примеры составления программ для решения экономических задач.

Пример 1. Пусть даны размеры посевных площадей зерновых культур в гектарах (A)¹, урожайность в центнерах на один гектар — плановая (B), фактическая (C). Требуется вычислить валовой сбор в центнерах по каждой культуре — плановый (D) и фактический (E), а также определить урожайность в среднем по зерновым культурам — плановую (Z2) и фактическую (Z3).

¹ Работа с интерпретатором Бейсик в диалоговом режиме рассматривается в 4.2.

Значения А, В и С в первом, упрощенном варианте программы будем вводить с пульта N раз каждое, если N — количество строк исходного документа.

Программу на Бейсике можно записать следующим образом:

```

10 REM      Вычисление урожайности в среднем
20 S1=0 \ S2=0 \ S3=0      — очистка ячеек для накопления
                               сумм нарастающим итогом
30 PRINT    "Введите количество строк документа N"
40 INPUT N      — ввод числа значений А,В,С
50 FOR I=1 TO N      — организация цикла повторений
60 PRINT «Введите А,В,С» } — ввод очередной тройки значений
70 INPUT А,В,С
80 D =A*B \ E =A*C      — вычисление D и E по формулам
90 PRINT "A=";A, "B=";B, — вывод значений А, В, С, D, E на
    "C=";C, "D=";D, "E=";E дисплей
100 S1=S1+A \ S2=S2+      — накопление суммарных значений
    +D \ S3=S3+E          посевной площади и валового
                               сбора (по плану и фактически)
110 NEXT I      — проверка окончания цикла и пе-
                               реход к следующей строке исход-
                               ного документа
120 Z1 = S2/S1      — вычисление средней урожайности
                               по плану
130 Z2 =S3/S1      — вычисление средней урожайности
                               по факту
140 PRINT    "суммарная посевная пло- — вывод на экран дис-
    щадь = "; S1; "га"      плей соответствующих
                               показателей
150 PRINT    "суммарный валовой сбор
    по плану = "; S2; "ц"
160 PRINT    "суммарный валовой по
    факту = "; S3; "ц"
170 PRINT    "урожайность в среднем по
    плану = "; Z1; "ц/га"
180 PRINT    "урожайность в среднем
    по факту = "; Z2; "ц/га"
190 END      — конец программы

```

Рассмотрим теперь вариант, когда значения А, В и С заносятся заранее на магнитный диск и оформляются как файл.

```

10 REM Работа с файлом (А,В,С) на диске
20 S1 =0 \ S2 =0 \ S3 = 0
30 OPEN «ABC.DAT» AS      — открытие вводного файла ABC на
    FILE #1               диске
40 OPEN "LP" AS FILE #4 — назначение канала устройству пе-
                               чати
50 INPUT #1, А,В,С      — считывание из файла первой трой-
                               ки значений А,В,С

```

60 IF END #1 GO TO 180	— проверка окончания цикла, если все записи файла, соответствующие строкам исходного документа, исчерпаны
70 D = A*B \ E=A*C	— вычисление D и E по формулам
80 PRINT #4, "A=";A, "B=";B, "C=";C, "D=";D, "E="; E	— печать значений
90 S1=S1+A \ S2 = = S2 +D \ S3 =S3+E	— накопление сумм
100 GO TO 50	— переход к следующему повторению циклического участка
110 Z1 = S2/S1	— вычисление средней урожайности по плану
120 Z2 = S3/S1	— вычисление средней урожайности по факту
130 PRINT #4,	"суммарная посевная площадь = "S1 ; "га"
140 PRINT #4,	"суммарный валовой сбор по плану = ";S2; "ц"
150 PRINT #4,	"суммарный валовой сбор по факту = ";S3; "ц"
160 PRINT #4,	"урожайность в среднем по плану = ";Z1; "ц/га"
170 PRINT #4,	"урожайность в среднем по факту = ";Z2; "ц/га"
180 END	— конец программы

В этом примере вывод всех значений производится на печатающее устройство («LP:») вместе с пояснительным текстом.

Пример 2. Даны урожайности зерновых X_i (см. табл. 1.23).

Требуется определить статистические показатели:

а) среднее значение урожайности по культуре

$$\bar{X} = \frac{1}{N} \sum_{i=1}^N X_i$$

(в программе обозначим его Z);

б) дисперсию

$$D = \frac{1}{N-1} \sum_{i=1}^N (X_i - \bar{X})^2;$$

в) среднеквадратическое отклонение

$$\sigma = \sqrt{D}$$

- (в программе обозначим его S);
г) коэффициент вариации

$$V = \frac{\sigma}{\bar{X}} \cdot 100;$$

- д) стандартную ошибку

$$L = \frac{\sigma}{N}.$$

Программу можно записать в следующем виде:

```

10 REM Расчет статистических показателей
20 PRINT "Введите количество лет N " — вводится количество
                                         задаваемых значений
                                         урожайности
30 INPUT N
40 Z1 = 0 \ Z2 = 0 — обнуление ячеек для
                                         накапливаемых сумм
                                         в формулах для  $\bar{X}$ 
                                         и D
50 FOR I = 1 TO N — начало цикла вычислений
60 PRINT "Введите значение X"
70 INPUT X — ввод значения  $X_i$ 
80 Z1 = Z1 + X — подсчет Z1 нарастающим итогом
90 NEXT I — окончание цикла по I (ввод  $X_i$ )
100 Z = Z1/N — вычисление  $Z = \bar{X}$ 
110 FOR K = 1 TO N — начало цикла для вычисления D
120 Y = (X - Z) ^ 2 — вычисление  $(X_i - \bar{X})^2$ 
130 Z2 = Z2 + Y — накопление суммы для D
140 NEXT K — конец цикла для вычисления D
150 D = Z2 / (N - 1) — окончательно значение D
160 S = SQR (D) — вычисление  $S = \sigma$ 
170 V = S / Z * 100 — значение коэффициента вариации V
180 L = S / SQR (N) — значение стандартной ошибки L
190 OPEN "LP:" AS FILE #3 — назначение номера канала устройству печати
200 PRINT #3, "средняя урожайность по культуре Z= "; Z;"ц/га"
210 PRINT #3, "дисперсия D= "; D
220 PRINT #3, "среднеквадратическое отклонение S="; S
230 PRINT #3, "коэффициент вариации V = "; V
240 PRINT #3, "стандартная ошибка L = "; L
250 END

```

В этой программе вывод информации также происходит на бумагу, так как назначается канал 3 устройству печати «LP:».

3.4. ЗАДАНИЯ

1. Перечислите символы, допустимые в Бейсик-программе.

2. Перечислите способы записи целых, вещественных, строковых констант.

3. Какие переменные допустимы в Бейсике?

4. Укажите назначение массивов в Бейсик-программе, способы их описания.

5. Объявите в программе: а) целый массив из 15 строк и 8 столбцов; б) массив из 25 вещественных чисел.

6. Приведите примеры арифметических выражений.

7. Запишите в виде арифметического выражения правую часть алгебраической формулы

$$Y = A \cos (X + B) - \ln (B \sqrt{X^2 + A}).$$

8. Приведите примеры выражений отношений, применяемых в Бейсике.

9. Укажите роль оператора присваивания в Бейсик-программе, приведите примеры.

10. Запишите оператор перехода к строке 260.

11. Укажите возможные в Бейсике формы оператора условного перехода IF.

12. Запишите оператор, который при $X \leq 0$ и $B > 25$ передает управление строке 320.

13. Укажите операторы, применяемые для организации циклических участков в программе.

14. Укажите назначение и правила применения операторов FOR и NEXT.

15. Укажите способ приостановки выполнения Бейсик-программы и ее продолжения с указанной строки (приведите примеры).

16. Приведите примеры и осуществите ввод данных с клавиатуры дисплея.

17. Приведите примеры использования оператора вывода текста и значений переменных на экран дисплея.

18. Укажите примеры применения запятой и точки с запятой при размещении выводимых данных в строке по зонам.

19. Объясните смысл оператора OPEN при работе с файлами на магнитном диске. Приведите примеры.

20. Укажите назначение оператора CLOSE в Бейсик-программе и приведите примеры.

21. Напишите программу определения количества поступившего зерна от каждого структурного подразделения и количества принятого зерна каждым током и в целом по совхозу, используя данные табл. 1.13.

22. Используя ведомости движения продукции (табл. 1.15), напишите программу для определения наличия продукции на конец года по каждому шифру и итоговые значения по каждому хозяйству.

23. Используя табл. 1.18, напишите программу расчета структуры затрат на производство яровых зерновых и картофеля.

24. Напишите программу (табл. 1.21) для определения урожайности овощных культур, объема продажи овощей государству и стоимости реализованной продукции.

Глава 4

РАБОТА В ОПЕРАЦИОННЫХ СИСТЕМАХ

МИНИ- И МИКРОЭВМ

4.1. ОПЕРАЦИОННАЯ СИСТЕМА ОСРВ И ЕЕ ПРИМЕНЕНИЕ

4.1.1. ОБЩИЕ СВЕДЕНИЯ О СИСТЕМЕ ОСРВ

Операционную среду ЭВМ, обеспечивающую создание, корректировку и выполнение различных вычислительных задач, принято называть *операционной системой*.

Операционная система включает в себя комплекс программ, являющихся средствами редактирования и выполнения программ, созданных пользователями (программистами), а также средствами компиляции (трансляции) с языков высокого уровня (Фортран, Кобол, Паскаль и др.).

Основной частью операционной системы является управляющая программа, которая координирует выполнение всех программ, находящихся в оперативной памяти ЭВМ, и обеспечивает при этом максимальную эффективность использования всех ресурсов системы.

Операционная система ОСРВ обеспечивает мультипрограммный режим работы, т. е. в системе одновременно могут выполняться несколько различных программ. Такой режим возможен в результате того, что выполнение программы сводится не только к обращению к процессору ЭВМ: работающая программа, вызывая какое-либо действие (например, ввод и вывод информации из ЭВМ) и ожидая его завершения, на период этого ожидания освобождает процессор. Если какой-либо программе пользователя требуется доступ к процессору, управляющая программа ОСРВ обеспечит ей этот доступ. В то же время параллельно может производиться ввод-вывод информации по другой программе.

Система ОСРВ ориентирована на работу с магнитными дисками. Она использует их как для хранения управляющей программы и прочих системных программ, так и для хранения пользовательских программ и данных для расчетов. Благодаря такому использованию диска возможно создание общей файловой системы ОСРВ.

На каждом рабочем диске, где работает данный пользователь, должна быть выделена некоторая область, специальным образом инициализированная — директорий пользователя. Организация директория осуществляется системной программой UFD. Любая информация, хранящаяся в нем, будь то программа или данные, представляет собой файл, т. е. поименованный участок на магнитном диске.

Основной модуль, который может выполняться на мини-ЭВМ, — это задача, оформленная определенным образом. В операционной системе ОСРВ понятие «задача» равносильно понятию «программа». Любая задача в ОСРВ окончательно оформляется системной программой — построителем задач (ТКВ).

Для того чтобы выполнить задачу в системе ОСРВ, необходимо следующее:

1. С помощью программы EDI ввести исходный текст программы пользователя и записать его на диске.

Если исходная программа уже подготовлена на перфоленте, то в ОСРВ ее нужно ввести с помощью системной программы FLX. Если же исходная программа подготовлена на магнитной ленте, ее необходимо скопировать на диск с помощью системной программы EMT.

2. Программа пользователя должна быть оттранслирована соответствующим компилятором, например транслятором Фортрана, и тем самым переработана в объектный модуль.

Если обнаруживаются ошибки в исходном тексте, то на этом этапе они должны быть устранены с помощью программы EDI либо TED.

3. Объектные модули должны быть переработаны в образ задачи с помощью системной программы ТКВ — построителя задач.

4. Запуск задачи, т. е. ее загрузка и выполнение, осуществляется посредством системной программы RUN.

4.1.2. ФАЙЛЫ, ИХ СПЕЦИФИКАТОРЫ И ЗАЩИТА

Как уже говорилось, файлом называется поименованный участок на магнитном носителе (магнитном диске или магнитной ленте).

Командная строка в ОСРВ состоит из двух спецификаторов файлов, между которыми знак «=». Структура этой строки

OUTFILE = INFILE

где OUTFILE — спецификатор получаемого (выводного) файла;

INFILE — спецификатор вводного файла.

В общем виде спецификатор файла записывается так:

DEV: [UIC] FILE-NAME.TYPE; VER/SWITCH

где

DEV — имя устройства, на котором находится информация;

UIC — код пользователя, которому принадлежит директорий;

FILE-NAME — имя данного файла;

.TYPE — расширение имени, т. е. тип (например, программа, написанная на Фортране, имеет тип .FTN);

VER — версия (номер в восьмеричной системе может быть от 1 до заранее определенного номера, при каждой корректировке номер версии увеличивается на 1);

SWITCH — ключ (возможны несколько ключей).

Устройства, применяемые в ОСРВ, имеют свои наименования. Для упрощения этих наименований каждому устройству дается единственное имя, используемое во всех спецификациях.

Имя устройства состоит из двух буквенных символов, идентифицирующих тип, и необязательного одноили двузначного восьмеричного номера, за которым необходимо помещать двоеточие.

Двоеточие здесь является признаком того, что это имя устройства, а не какого-либо другого объекта системы.

В системе ОСРВ приняты следующие имена устройств:

DKnp: — диск кассетный емкостью 2,4 Мбайта ($2,4 \times 10^6$ байт);

MTnp: — накопитель на магнитной ленте;

TTnp: — алфавитно-цифровые видеотерминалы (дисплеи);

TInp: — системный терминал (дисплей пользователя).

Все пространство тома с файловой структурой (магнитного диска) в ОСРВ, как уже говорилось, разбирается на участки, или директории. Мнемоническое имя

директория, или код пользователя, принятое в ОСРВ, имеет следующую структуру:

[группа, член группы]

Здесь группа представляет собой восьмеричное число от 1 до 377, член группы — также восьмеричное число от 1 до 377.

Наличие квадратных скобок является обязательным.

Код пользователя назначается системным программистом или другим ответственным лицом вычислительного центра.

Доступ всех пользователей в процессе их работы с ОСРВ к файлам организуется и автоматически контролируется системой на основании кодов пользователей.

4.1.3. РАБОТА С ТЕРМИНАЛОМ

Основным рабочим местом пользователя системы является видеотерминал (дисплей). Терминалом принято называть устройство ввода-вывода, в котором для ввода информации используется клавиатура, для вывода — печатающее устройство или электронно-лучевая трубка (экран). В качестве терминалов могут использоваться алфавитно-цифровой дисплей либо пишущая машинка, подключенная к ЭВМ, и другие устройства.

Зарегистрировавшись в системе (в начале каждого сеанса работы) с любого терминала, пользователь получает доступ к системным ресурсам. Управление всеми внешними устройствами, поддерживаемыми системой, а также запуск задач, оперативный контроль за их выполнением, а в случае необходимости и вмешательство в их работу осуществляются с дисплея пользователя.

Если требуется, с дисплея может производиться ввод различных данных, запрашиваемых запущенной с него задачей.

Последовательно нажимая клавиши на клавиатуре терминала, пользователь может вводить в ЭВМ командные строки, которые принимаются и затем обрабатываются программой связи с оператором или другой задачей, запросившей ввод с терминала (например, экранный редактор или программа пользователя).

Вводимая командная строка размещается в специально отведенной области памяти — в буфере командной строки. Размер этого буфера не превышает 127 двухбайтовых слов; при этом вводимые символы высвечи-

ваются на экране дисплея, что обеспечивает визуальный контроль ввода.

Ввод каждой командной строки должен заканчиваться нажатием клавиши ВК. Этот символ воспринимается системой как признак окончания ввода строки, и она приступает к выполнению введенной команды. Автоматически при этом происходят перевод строки и установка курсора терминала в первую позицию новой строки.

Курсор представляет собой специальный знак подчеркивания (в некоторых дисплеях он имеет «мигающее» свечение). Курсор указывает на экране очередное знакоместо для отображения следующего символа.

В процессе работы с клавиатурой система дает возможность пользователю производить корректировку (изменение, удаление) командной строки. До того, как была нажата клавиша ВК, можно с помощью клавиши ЗБ удалить ошибочно набранный символ и взамен него набрать новый символ, а также можно полностью исключить набираемую командную строку — нажав одновременно клавишу УС и клавишу U, и набрать вместо нее новую командную строку.

Перед началом работы в операционной системе пользователю необходимо ознакомиться с клавиатурой того терминала, с которого он намерен работать, а также уточнить, какие символы из всего набора на клавиатуре в данной системе не используются, как производится переход с одного регистра клавиатуры на другой.

Относительно ввода информации с терминала имеются специальные соглашения. В системе заложены три типа отзыва, которые указывают, что данный терминал ожидает ввода командной строки:

> — отзыв по умолчанию;

MCR> — отзыв программы связи с оператором;

TSK> — отзыв задачи (TSK — имя данной задачи).

Отзыв по умолчанию означает, что терминал готов принять незапланированный системой вызов; при этом незапланированный вызов всегда в неявной форме обращен к программе связи с оператором (т. е. к монитору MCR). Само понятие «незапланированный вызов» означает, что в данный момент нет какой-либо задачи, присоединившей данный терминал и ожидающей ввода с него. Получая незапланированный вызов, программа связи с оператором анализирует его и передает управляющей программе системы либо соответствующей системной обслуживающей программе на выполнение.

Вызываемая с помощью программы связи с оператором задача выводит на экран свой отзыв-подсказку

TSK>

где TSK — имя вызываемой задачи, состоящее из трех литер.

Если данный терминал (дисплей) выполняет какую-либо задачу пользователя, то вся вводимая с него информация передается непосредственно этой задаче, а не программе связи с оператором MCR. Если при этом возникает необходимость обратиться к MCR, нужно набрать команду активизации программы связи с оператором путем одновременного нажатия клавиш УС и С. В ответ на это на экран выводится отзыв-подсказка

MCR>

и далее вся вводимая в ответ на подсказку информация будет принята программой связи с оператором.

Если пользователь по какой-либо причине передумал выполнять вызванную им задачу TSK, то в ответ на подсказку

TSK>

он должен ввести команду — управляющий символ двумя клавишами УС и Z, в ответ на это будет выведен на экран отзыв по умолчанию (>).

Следует иметь в виду, что большинство системных обслуживающих задач допускает ввод командной строки длиной не более 80 символов.

Командная строка, набираемая на клавиатуре, имеет следующую структуру:

имя команды параметр / ключевое слово < ограничитель строки >

Имя команды состоит из трех литер;

параметр определяет объект функции программы, который чаще всего представляет устройство. Параметр от имени команды (или параметр от параметра, если их несколько) должен отделяться пробелом. Например, команда начала выполнения задания ADDTWO

>RUN ADDTWO <BK>

/ключевое слово — модифицирует или действительную функцию команды, или параметр команды;

ограничитель строки представляет собой символ возврата каретки <BK> и означает завершение строки и отсылку ее в ЭВМ.

Как явная подсказка MCR (MCR>), так и косвенная его подсказка (>) указывают, что MCR готов к приему ввода от терминала пользователя.

Каждый терминал (дисплей) имеет набор управляющих символов.

Ввод управляющего символа осуществляется путем нажатия клавиши соответствующей буквы одновременно с нажатием (пальцем другой руки) клавиши <УС>.

На некоторые управляющие символы система отвечает выводом на экран направленной вверх стрелки (Λ), за которой следует набранная буква (например, ΛU). Другие управляющие символы в зависимости от функции, которую они выполняют, не отображаются на экране терминала.

Ниже приводятся наиболее часто применяемые управляющие символы с описанием работы каждого из них.

Символ	Описание
УС/С	Передает управление системной программе — монитору MCR. В подавляющем большинстве случаев система отвечает на управляющий символ путем подсказки > или реже MCR >. Эта подсказка даже без явной идентификации MCR означает, что операционная система готова принять ввод от данного терминала. Иногда во время работы за пультом нужно определить, чья подсказка светится на экране.
УС/С и УС/Q (STOP, QURRENT)	Управляющий символ УС/С останавливает движущееся отображение вывода на экране терминала. Информация на экране остается неподвижной до тех пор, пока пользователь не наберет на клавиатуре УС/Q; последний символ возвращает ей первоначальное движение.
УС/U	Остановки и запуски отображения на экране дисплея можно повторять неограниченно во время работы, в соответствии с ее условиями. Управляющий символ стирает всю текущую строку с экрана. Это позволяет осуществить заново полный набор строки, когда отдельные коррективы уже не удовлетворяют пользователя. Необходимо подчеркнуть, что для уничтожения данной строки символ УС/U необходимо набрать до возврата каретки. Если же клавиша возврата каретки уже была нажата, то стирание строки может быть достигнуто только использованием команд редактирования системной программы-редактора.
УС/Z	Этот управляющий символ обеспечивает выход из большинства программ и возвращает автоматически управление монитору MCR.

Примечание. Перечисленные управляющие символы в различных вариантах операционной системы могут не совпадать. Это обстоятельство следует уточнить перед началом работы за пультом дисплея.

4.1.4. РАБОТА В ОПЕРАЦИОННОЙ СИСТЕМЕ ОСРВ С МНОГОПОЛЬЗОВАТЕЛЬСКОЙ ЗАЩИТОЙ

Многопользовательская защита позволяет эффективно управлять доступом индивидуальных пользователей к системным ресурсам и обеспечивает надежную защиту файлов пользователя.

Каждый пользователь перед началом сеанса работы должен зарегистрировать свой вход в систему, иначе ему не будет предоставлен доступ к системным ресурсам. Регистрация должна производиться с того терминала, на котором пользователь собирается работать.

Для регистрации входа в систему применяется команда HEL программы связи с оператором. Параметрами этой команды являются имя или код идентификации пользователя и пароль.

Ни один пользователь не может успешно зарегистрироваться в системе, не указав правильно имя и пароль, так как на все обращения такого пользователя будет выдаваться сообщение

MCR — NOT LOGGED IN

или

MCR — ILLEGAL FUNCTION

Информация о всех тех пользователях, которым разрешен доступ к системе, хранится в специальном системном файле счетов. Каждый раз, когда пользователь пытается зарегистрировать свой вход в систему, ОСРВ проверяет указанные программе HEL параметры по файлу счетов для того, чтобы определить, разрешен ли данному пользователю доступ к системе.

Занесение имени и пароля пользователя в файл счетов производится один раз при первоначальном выходе его на машину. В дальнейшем в начале каждого сеанса работы пользователь ограничивается только указанием программе имени и пароля.

Системная программа, создающая файл счетов, доступна только привилегированному пользователю; обычно информацию о пользователе заносит в файл счетов или системный программист, или другое ответственное лицо на вычислительном центре.

По окончании сеанса работы пользователь должен зарегистрировать свой выход из системы, так как система не может гарантировать пользователю, не зарегистрировавшему свой выход из системы, сохранность его файлов. Регистрация выхода осуществляется командой BUE программы связи с оператором MCR.

Рассмотрим два основных варианта регистрации выхода в систему OCPB.

В а р и а н т I

В ответ на подсказку по умолчанию программы связи с оператором MCR «>» пользователь набирает команду

>HEL <BK>

Эта команда вызывает появление на экране дисплея запроса параметров команды

ACCOUNT OR NAME:

Затем пользователь должен ввести с клавиатуры свой код идентификации UIC или свое имя.

Например, если UIC пользователя — [12,10], а его имя IVANOV, то возможны следующие варианты ответа:

[12,10] <BK>

или

IVANOV <BK>

Оба ответа равносильны.

В случае, если UIC (либо указанное имя) найдены системой в файле счетов, система запрашивает пароль

PASSWORD

Примечание. Следует отметить, что для сохранения секрета от окружающих символы пароля, вводимые пользователем с клавиатуры, на экране не высвечиваются. Ввод пароля должен также заканчиваться нажатием клавиши BK.

Если обнаруживается, что пользователю разрешен доступ к системе (пароль соответствует его имени и UIC), на экран дисплея выводится текст

OCPB MULTI — USER SYSTEM

а затем следует приветствие, соответствующее времени дня

GOOD MORNING

или

GOOD AFTERNOON

или

GOOD EVENING

и далее на экране появляется текст из файла LB:

[1, 2] LOGIN.TXT

Файл LOGIN.TXT сформирован системным программистом и может содержать различную справочную информацию для пользователя.

Следующим выполняется файл LOGIN.CMD из директории пользователя на устройстве пользователя (в том случае, если он был подготовлен заблаговременно).

В а р и а н т II

Имя или код идентификации пользователя может быть указан в одной командной строке с командой HEL, причем сама команда HEL должна быть отделена от имени пользователя (или его UIC) пробелом.

Например:

>HEL [12, 10] <BK>

или

HEL IVANOV <BK>

Тогда в ответ система сразу же запрашивает пароль
PASSWORD:

Далее все происходит, как в первом варианте.

Если параметры пользователя, указанные в команде HEL, не соответствуют параметрам, хранящимся в файле счетов, то на терминал выводится сообщение

HEL —INVALID ACCOUNT

В ответ на последнюю подсказку MCR пользователь может попытаться вновь зарегистрировать свой вход в систему. Число попыток при этом не ограничено.

Команда BYE, регистрирующая выход пользователя из системы, имеет короткий формат

>BYE <BK>

По этой команде аварийно могут быть завершены все задачи, запущенные пользователем на выполнение, демонтируются личные тома запоминающих устройств и отменяются все назначения, сделанные пользователем.

На экран дисплея выводятся сообщения о всех действиях команды BYE и при нормальной регистрации выхода из системы прощание, соответствующее времени суток, например

HAVE A GOOD EVENING

4.1.5. ПОДГОТОВКА УСТРОЙСТВ ПОЛЬЗОВАТЕЛЯ К РАБОТЕ

В процессе работы с ЭВМ пользователю приходится обращаться к терминалу (дисплею), диску для создания, редактирования и хранения файлов, к алфавитно-цифровому печатающему устройству (АЦПУ) и др.

Любое из устройств, поддерживаемых системой, или же одновременно несколько устройств могут быть назначены как личные устройства пользователя.

Для того чтобы назначить устройство личным, применяется команда ALL (сокращение от ALLOCATE) программой связи с оператором MCR.

Привилегированный пользователь (системный программист), может назначить некоторые устройства общими (например, АЦПУ). К общему устройству имеют доступ все пользователи, но общее устройство уже не может быть назначено личным. Для этого применяется команда с привилегированного терминала

```
>SET / PUB = устройство :<БК>
```

Устройства, не назначенные ни личными, ни общими, считаются в системе неприсоединенными. К таким устройствам имеют доступ также все пользователи системы.

Все устройства, поддерживающие файловую структуру ОСРВ — диски и магнитные ленты, перед началом работы соответствующим образом подготавливаются.

Магнитные диски иницируются, на них создаются директории для хранения файлов. Магнитные ленты только иницируются. И те и другие должны монтироваться: монтирование диска делает его известным системе как устройство с файловой структурой.

Для подготовки дисков и магнитных лент к работе применяются следующие команды программы связи с оператором MCR:

```
ALL, DEA, INI, MOU, DMO, UFD.
```

Командой ALL пользователь назначает данное устройство личным, запрещая тем самым доступ к нему непривилегированных пользователей. Когда пользователь регистрирует свой выход из системы, это назначение будет автоматически отменено.

Например, для АЦПУ команда будет иметь вид

```
>ALL LP: <БК>
```

Если в процессе работы пользователя необходимо отменить сделанные ранее командой ALL назначения,

следует использовать команду `MCR:DEA` (сокращение от `DEALL`).

Например, отменить назначение устройства АЦПУ личным можно таким образом:

```
>DEA LP: <BK>
```

После этой команды устройство АЦПУ станет неприсоединенным.

Командой `MOU` пользователь монтирует диски и магнитные ленты, делая их известными системе.

Формат команды:

```
>MOU устройство: метка тома <BK>
```

Например, смонтировать магнитную ленту `SPRAV` можно с помощью следующей командной строки:

```
>MOU MT1 : SPRAV <BK>
```

При необходимости демонтировать том следует применить команду `DMO` (сокращение от `DISMOUNT`).

При регистрации пользователем выхода из системы с помощью команды `BYE` все смонтированные им личные устройства автоматически демонтируются системой.

Командой `UFD` (`USER FILE DIRECTORY`) программы связи с оператором `MCR` пользователь может создать директорию пользователя на диске.

Формат этой команды:

```
>UFD устройство: [g ,m]<BK>
```

При этом диск, на котором создается директорию, должен быть уже смонтирован.

Параметр `[g, m]`, означающий номер группы `g` и номер члена в группе `m` создаваемой зоны, должен совпадать с `UIC` пользователя, так как при регистрации пользователем входа в систему командой `HEL` ему будет сразу же автоматически предоставлен доступ к файлам в этом директории.

4.1.6. ОСНОВНЫЕ СВЕДЕНИЯ О ЯЗЫКЕ ПРОГРАММИРОВАНИЯ ФОРТРАН

В настоящее время Фортран является одним из наиболее распространенных языков для решения научно-технических задач. Его транслятор обеспечивает очень высокое качество объектных программ.

Программа на Фортране представляет собой последовательность операторов: каждая строка программы есть оператор. В качестве алфавита языка используются все буквы латинского алфавита, арабские цифры,

символы: точка, запятая, апостроф, открывающая и закрывающая круглые скобки, пробел, знак равенства, плюс, минус, звездочка (знак умножения), две звездочки (знак возведения в степень), знак деления (/), а также аналогичные математические знаки $>$, $<$, \neq , \geq , \leq .

Основные элементы, используемые для построения программ, — это константы и переменные.

Константы — не изменяют своего значения в процессе работы программы. Числовые константы могут быть целыми и вещественными. Вещественные чаще всего записываются с десятичной точкой. Например: 12 — целая константа, +7.52 — вещественная. К логическим константам относятся те, которые могут иметь одно из двух значений: истинное (.TRUE.) и ложное (.FALSE.). Текстовые константы состоят из последовательностей любых символов, допустимых в Фортране, и применяются для подготовки текста к печати на бумаге. В текстовых константах используются буквы русского алфавита. Чаще всего текстовые константы записываются с помощью апострофов (кавычек). Например:

```
' начало программы'  
' вторая ветвь '
```

Переменные — величины, обозначаемые символическим именем, могут принимать различные значения в процессе работы программы. Имя — это последовательность не более 8 символов, состоящая из латинских букв и цифр и начинающаяся обязательно с буквы. Например: X, Y, T48, ALFA.

Типы переменных соответствуют типам констант. Переменные хранятся (и соответственно изменяются) в определенных ячейках памяти ЭВМ.

Тип целый (INTEGER) и вещественный (REAL) можно назначить явно в программе либо воспользоваться правилом: все переменные, имена которых начинаются с букв I, J, K, L, M, N, — целые, все остальные — вещественные (с десятичной точкой).

Упорядоченная последовательность величин, обозначенных одним символическим именем, представляет собой массив. Величины, его образующие, — это элементы данного массива. Каждый элемент массива именуется соответственно символическому имени массива и одному или нескольким индексам, указывающим положение элемента в массиве. Одномерные массивы имеют элементы с одним индексом; двумерные (их можно рас-

сма́тривать как таблицы) — элементы с двумя индексами (индекс — номер столбца и индекс — номер строки, они разделяются запятой).

Например, элемент $A(1,3)$ — это элемент массива A , находящийся в первом столбце в третьей строке.

В общем случае значение индексов может записываться с помощью переменных и даже арифметических выражений. Количество индексов и максимальное значение каждого индекса описывается невыполняемым информационным оператором `DIMENSION`. Например, таблица чисел `RET`, состоящая из 7 строк и 5 столбцов, описывается

`DIMENSION RET (5, 7)`

Информация о количестве различных массивов в программе и о числе элементов в них должна быть сообщена компилятору раньше, чем в программе появится обращение к элементам какого-либо массива. Принято располагать оператор `DIMENSION` в самом начале программы.

Термин «выражение» используется в том же смысле, что и в математике; выражение состоит из констант и переменных, соединенных знаками операций и скобками.

Знаки арифметических операций: $+$ (плюс), $-$ (минус), $*$ (умножение), $/$ (деление), $**$ (возведение в степень). Например, арифметические выражения:

$X + Y - B$
 $ALFA * M$
 $D + (K * X - Y ** 2) * X$ и т. д.

Выражения отношения состоят из двух арифметических выражений, соединенных одной из операций отношения:

.LT. — меньше ($<$)
.LE. — не больше (\leq)
.EQ. — равно ($=$)
.NE. — не равно (\neq)
.GT. — больше ($>$)
.GE. — не меньше (\geq)

Например, выражение отношения

$Y \leq A + B$

запишется в виде

`Y.LE.(A + B)`

или

$X^2 + C > AM - P$

имеет вид

(X ** 2 + C).GT.(AM—P)

и т. д.

Логические выражения могут принимать только одно из двух значений: истинное (.TRUE.) либо ложное (.FALSE.). Операндами логических выражений могут быть выражения отношения, например

$A > B$ или $C \leq Y$

В программе это выражение примет вид

(A.GT.B). OR. (C.LE.Y)

Логические выражения в Фортран-программах используются для построения разветвляющихся программ.

Для удобства вычислений часто встречающихся математических функций в компиляторе Фортрана имеется набор специальных системных функций, в том числе:

SQRT(x) — вычислить квадратный корень вещественного аргумента;

SIN(x) — вычислить синус угла (в радианах);

COS(x) — вычислить косинус угла (в радианах);

EXP(x) — вычислить экспоненту вещественного аргумента;

ALOG(x) — вычислить натуральный логарифм и др.

Элементами действия в программе являются операторы. В Фортране различаются выполняемые и невыполняемые операторы. К выполняемым относятся операторы присваивания, управления, ввода-вывода. К невыполняемым операторам относятся: операторы описания, операторы, определяющие форматы ввода-вывода, и некоторые другие.

Операторы описания служат для описания свойств переменных, массивов, функций. Оператор формата FORMAT применяется с выполняемыми операторами ввода-вывода для обеспечения необходимых преобразований вводимой и выводимой информации.

При составлении программы на Фортране вначале необходимо записывать операторы описания, затем выполняемые операторы и в конце — операторы STOP и END, завершающие каждую программу.

Арифметические выражения и системные функции служат для записи арифметического оператора присваивания. Он предназначен для вычисления значения арифметического выражения, стоящего справа от знака равенства, и присваивания этого значения переменной, стоящей слева. Арифметический оператор присваивания

коренным образом отличается от обычного алгебраического равенства. Например, в алгебре лишено смысла уравнение вида

$$Y = Y + 5.26$$

а в Фортран-программе оно означает, что следует к числу, в данный момент находящемуся в ячейке памяти, где хранится переменная Y , прибавить число 5.26 и результат сложения снова записать в ту же самую ячейку Y в качестве нового значения Y , которое будет использоваться в дальнейших расчетах по программе. Отсюда следует, что в операторе присваивания отражена динамика изменения величин в соответствии с программой.

Выполнение Фортран-программы начинается с первого выполняемого оператора и далее все операторы выполняются последовательно.

Применение операторов управления, нарушающих эту последовательность, связано с наличием метки, вызывающей конкретную точку программы. Метка в Фортране обязательно должна быть числом; она может содержать от 1 до 5 десятичных цифр и помещается перед помеченным ею оператором.

Метки могут размещаться в программе произвольно, но все они должны быть различны. Один оператор может иметь только одну метку, которая таким образом служит для ссылки на этот оператор. Например:

$$150 \quad Y = X + A * B$$

Для передачи управления любому помеченному меткой выполняемому оператору программы можно использовать оператор перехода GO TO. Простейший вид его — безусловный оператор — вызывает переход к оператору с указанной меткой. Например, для перехода к вычислению Y по вышеприведенной формуле следует записать:

GO TO 150

Следует отметить, что компилятор Фортрана не требует строгого соблюдения пробелов внутри операторов (пробелы нужны для наглядности при составлении и чтении, проверке программы, т. е. для человека-программиста). Обязательными являются лишь пробелы, отделяющие метку от помеченного ею оператора.

Для передачи управления тому или иному оператору программы в зависимости от результатов проверки

выполнения некоторого условия предназначен условный оператор IF (если).

Арифметический оператор IF имеет вид

IF (A) M1, M2, M3

где A — арифметическое выражение;

M1, M2, M3 — метки выполняемых операторов программы.

Вначале вычисляется и анализируется значение арифметического выражения A. Если это значение отрицательно, то управление передается оператору с меткой M1; если оно равно нулю — оператору с меткой M2; если же оно положительно — оператору с меткой M3. Например:

```
IF (A + B * X) 150, 160, 170
150 Y = X + (A + B) ** 2
160 Y = SIN (X) + B ** 2
170 Y = X - (A + B) * 3
```

Если в программе следует различать только два значения арифметического выражения, например отрицательное и неотрицательное, то номера меток достаточно повторить (например, 150, 160, 160); но в любом случае метка в операторе IF должна записываться трижды.

При выполнении логического оператора IF вычисляется и анализируется значение логического выражения, стоящего в скобках. Если это значение истинно, то управление передается оператору, стоящему после скобок.

Если же значение логического выражения ложно, оператор, указанный в IF, игнорируется, и будет выполняться оператор, следующий за оператором IF. Например:

```
IF (Y.GT.A) GO TO 120
110 C = A * Y - B ** 2
120 C = (A - B / Y) ** 2
```

В этом примере, если условие $Y > A$ окажется ложным, переход к метке 120 игнорируется и выполняется оператор присваивания с меткой 110.

Исполняемым считается в Фортране фиктивный оператор CONTINUE, который не вызывает никакого действия в программе и служит только для помещения метки. Используется этот оператор в качестве конечного (последнего) оператора в циклически повторяемом участке программы, т. е. совместно с оператором DO.

Оператор DO используется для повторного выполнения участка программы (цикла) требуемое количество раз. Форма записи оператора в общем виде:

DO M I = K1,K2,K3

где M — метка последнего оператора цикла (оператора CONTINUE);

I — целая переменная (переменная цикла);

K1 — начальное значение переменной цикла;

K2 — конечное ее значение;

K3 — приращение ее (шаг изменения).

В случае, если приращение K3=1, его в операторе не указывают (оно подразумевается по умолчанию)

DO M I = K1,K2

Оператор DO выполняется следующим образом: в начале переменной I присваивается значение K1. После каждого повторения циклического участка значение I увеличивается на величину K3 (либо на 1). Если новое значение не превосходит конечного значения K2, то циклический участок программы повторяется снова; в противном случае начинает выполняться оператор, следующий за последним оператором цикла (за оператором CONTINUE).

Например, требуется подсчитать нарастающим итогом сумму 150 чисел WET. Опишем их с помощью DIMENSION WET (150). Сотрем содержимое ячейки SUM и будем накапливать в ней эту сумму с помощью оператора присваивания SUM=0.0.

Организуем циклический участок с использованием оператора DO; подсчитаем сумму SUM, начиная с первого числа (I=1) и кончая 150 (K2=150). В качестве последнего оператора цикла используем оператор CONTINUE. Соответствующий фрагмент программы будет в этом случае иметь вид:

```
DIMENSION WET (150)
SUM = 0.0
DO 10 I = 1, 150
  SUM = SUM + WET (I)
10 CONTINUE
```

Завершает выполнение Фортран-программы оператор STOP.

Последним оператором любой Фортран-программы должен быть оператор END, который используется с единственной целью: сообщить компилятору об окончании транслируемого текста Фортран-программы.

Данные в мини-ЭВМ вводятся с перфолент, магнитных лент, с клавиатуры дисплея и заносятся они на магнитный диск.

Операторы ввода-вывода в условиях работы с мини-ЭВМ предназначены для обмена информацией между магнитным диском и оперативной памятью ЭВМ.

Данные на магнитных лентах и дисках организуются в виде файлов, которые komponуются из логически связанных между собой групп данных. Обращение к каждому файлу происходит по присвоенному этому файлу номеру в данной программе.

Для преобразования данных при их вводе-выводе применяется оператор **FORMAT**.

Простейшими операторами ввода и вывода являются операторы **READ** (читать) и **WRITE** (писать). Данные, участвующие в передаче, указываются в списке ввода-вывода этих операторов. При вводе считываемые из файла данные преобразуются в элементы этих списков (элементы находятся в оперативной памяти ЭВМ), а при выводе данные записываются в файлы из элементов ввода-вывода. Последовательность передачи данных определяется порядком следования элементов в списке ввода-вывода.

Форма оператора чтения

READ (M1,N1) список данных

где M1 — номер файла (номер канала чтения);

N1 — метка оператора **FORMAT**, предназначенного для преобразования данных при чтении.

Форма оператора записи

WRITE (M2,N2) список данных

где M2 — номер файла, в который записываются данные;

N2 — метка оператора **FORMAT**, по которому производится преобразование данных при записи.

Например:

READ (2, 10) **AN**

WRITE (5,20) **Y7**

Оператор **FORMAT** состоит из слова **FORMAT** и записанного за ним перечня (списка) спецификаторов формата, заключенного в круглые скобки. Для ввода-вывода чисел в простейшем случае достаточно двух форм представления. Форма представления **In** означает, что считываемое (или записываемое) число является целым числом, содержащим *n* символов (цифры и знаки + и

—). Аналогично этому вещественное число с десятичной фиксированной точкой имеет форму представления $F_n.m$, где n — общее количество символов в числе (включая десятичную точку и знаки $+$ и $-$), а m — количество цифр после десятичной точки.

Например, целому числу $+127$ будет соответствовать спецификатор $I4$, числу 23 — спецификатор $I2$. Вещественному числу $7,25$ соответствует спецификатор $F 5.2$, а числу $3,5$ — спецификатор $F 3.1$.

При необходимости передавать текстовую информацию применяется спецификатор AW , где A — означает передачу символов, а W — указывает их количество в передаваемом слове.

В простейшем случае — при выводе на печать заголовков документов, кратких сообщений и т. п. — достаточно поместить в операторе соответствующую последовательность символов, заключенную в апострофы (кавычки). Например, для печати заголовка «Значения корней уравнения» достаточно записать операторы:

```
WRITE (5,30)
```

```
30 FORMAT ('значения корней уравнения')
```

В данном случае список переменных после оператора $WRITE$ вообще отсутствует, так как никаких переменных из оперативной памяти не выводится; печатается лишь сам текст, заложенный в операторе формата.

Для печати пробелов между выводимыми значениями переменных применяется символ X с требуемым повторителем. Например, чтобы получить поле шириной в три позиции, затем значения двузначного целого числа ($I2$), затем двух пробелов и значения вещественного числа, содержащего один знак до и два знака после точки ($F4.2$), можно указать формат

```
40 FORMAT (3X,I2,2X,F4.2)
```

Для переноса печати на новую строку в операторе $FORMAT$ используется символ $/$ (косая черта). Например, в результате выполнения операторов

```
WRITE (5, 10)
```

и

```
10 FORMAT (12X , ведомость ' /12X, 'начисления')
```

будет получен текст, расположенный в две строки:

```
ВЕДОМОСТЬ  
НАЧИСЛЕНИЯ
```

Оператор $FORMAT$ может быть записан в любом месте программы: это никак не повлияет на ее выполнение.

Следует отметить, что для помещения в Фортран-программе вспомогательной информации (пояснений к программе) используют оператор комментария (COMMENT), который записывается одной буквой С в первой позиции строки программы. Помещаемый при этом текст комментария в данной строке компилятором никак не анализируется, а просто выводится на печать при распечатке всей программы во время ее трансляции, например

С Лабораторная работа № 1
С группа 112 Иванов

Особенности размещения Фортран-программы на бланке состоят в следующем.

В первой позиции (колонке) помещается символ С, если данная строка программы является комментарием. С первой же позиции рекомендуется записывать метки операторов (для них отводятся с 1-й по 5-ю позиции). Позицию 6 используют только для помещения символа переноса (символ *) и только в том случае, если данная строка программы является продолжением предшествующего оператора, не уместившегося на предыдущей строке. В противном случае в колонке 6 должен быть пропуск (пробел).

Содержание самого оператора в каждой строке программы размещается с 7-й позиции по 72-ю включительно.

Текст, записанный на бланке с 73-й по 80-ю позицию, может использоваться только как индикация (нумерация) строк программы, так как этот текст компилятором не воспринимается, хотя и будет отпечатан на бумаге (аналогично комментариям) в ходе трансляции программы.

Следует отметить, что такое разделение бланка на отдельные поля имеет значение лишь для компилятора Фортрана и совершенно не нужно для записи значения числовых данных, которые в свою очередь должны записываться на бланки (и вводиться) лишь в строгом соответствии с записями в надлежащих операторах FORMAT. Это соответствие нуждается в тщательной проверке.

Рассмотрим конкретный пример составления программы на Фортране для ЭВМ типа СМ.

Пусть даны два массива по 15 значений: массив А — цены за продукцию (в рублях за единицу продукции) и массив В — количество продукции:

A : 12,75	32,81	10,24	21,32
B : 7,52	8,34	5,75	4,32

Требуется вычислить стоимость продукции каждого вида (SUM) и общую стоимость всей продукции (AITOG).

Решение: предварительно с помощью текстового редактора EDI записать на магнитном диске файлы соответствующих значений A.DAT и B.DAT, которые в Фортран-программе будут читаться с диска в оперативную память, соответственно в массивы A и B.

Оператор DIMENSION объявляет размерности этих массивов. Операторы CALL ASSIGN предназначены для присвоения массивам A и B соответствующих номеров, т. е. 1 — для A и 2 — для B. Операторы READ обеспечивают чтение информации из файла A.DAT в массив A в оперативной памяти и из файла B.DAT в массив B соответственно.

В операторе FORMAT содержится указание на то, что с форматами F5.2 и F4.2 вводятся по 15 данных.

Обнуление значения AITOG до начала циклического участка программы производится для правильного подсчета нарастающим итогом значения общей стоимости всей продукции.

Оператор цикла DO предназначен для циклического повторения участка программы, ограниченного оператором CONTINUE (с меткой 30).

В цикле оператор присваивания вычисляет стоимость SUM каждого из 15 видов продукции (путем умножения цены на количество).

Порядковый номер продукции и вычисленное значение суммы с выводом пояснительного текста 'сумма =' печатается с помощью операторов WRITE и FORMAT. Следует отметить, что канал 6 в операторе печати был назначен в начале программы для устройства печати ('LP:') соответствующим оператором CALL ASSIGN.

Значение AITOG подсчитывается нарастающим итогом в цикле оператором присваивания.

По выходе из цикла (после метки 30) печатается накопленное в цикле значение AITOG в соответствующем формате и с пояснительным текстом ('итого =', 'рублей').

Операторы STOP и END завершают программу:

```

С Лабораторная работа № 4
С группа 112 Иванов
  DIMENSION A(15), B(15)
  CALL ASSIGN(1, 'A.DAT')

```

```

CALL ASSIGN(2, ' B.DAT')
CALL ASSIGN(6, ' LP: ')
READ (1,10)A
10 FORMAT (15 F5.2)
READ (2,20)B
20 FORMAT (15 F4.2)
AITOG = 0.0
DO 30 K = 1, 15
SUM = A(K) * B(K)
WRITE (6,40) K, SUM
40 FORMAT (1X, 12', сумма = ', F6.2)
AITOG= AITOG + SUM
30 CONTINUE
WRITE (6,50) AITOG
50 FORMAT (1X 33' ИТОГО = ', F8.2, ' рублей ')
STOP
END

```

4.1.7. ОСНОВНЫЕ СИСТЕМНЫЕ ПРОГРАММЫ ОСРВ

4.1.7.1. Программа связи с оператором MCR

Программа MCR осуществляет связь системы ОСРВ с человеком-оператором.

Все команды MCR условно делятся на 4 группы: команды инициализации, команды информационные, команды управления и команды, отражающие организацию системы и ее состояние.

Команды инициализации реализуют ввод даты и времени; установку задач в системе; инициализацию дисков; распределение памяти разделов; установку статуса терминала.

Команда INI предназначена для структурирования тома диска в соответствии с требованиями файловой системы. При этом разрушаются все существовавшие на томе файлы; записывается метка тома.

Команда INS применяется для установки задач в системе, например

```
>INS DK1 :[1, 100]SKAN.TSK <BK>
```

где DK1: — диск, на котором находится директори́й;

[1,100] — директори́й пользователя;

SKAN — имя файла (имя задачи);

TSK — тип файла.

Команда MOU, как уже указывалось (4.1.2), монтирует магнитные диски, которые становятся доступными системе.

Команда SET устанавливает характеристики терминалов и устанавливает UIC; устанавливает или уничтожает разделы памяти и т. п.

Например, установка директория пользователя:

```
>SET / UIC = [g, m] <BK>
```

Команда TIM устанавливает текущее время дня:

```
TIM [hh : mm : ss] [mm/dd/yy] <BK>
```

Последовательность двузначных (цифровых) символов при этом:

```
[часы: минуты : секунды] [месяц/день/год] <BK>
```

Команда UFD создает файл оглавлений файлов пользователя на данном диске, например

```
>UFD DEV :[метка тома] [g, m] [/ключевые слова] <BK>
```

Команды информационные обеспечивают оператора данными о системе: время, дата, имена установленных в системе задач; распределение разделов памяти; имена устройств и т. п. Информационные команды не оказывают влияния на ход выполнения в системе задач.

По команде DEV (сокращение от device) на терминал выводятся имена всех устройств, известных системе, либо имена всех устройств данного типа.

При этом имена устройств печатаются в одной колонке, а вторая колонка содержит (если это необходимо) дополнительную информацию о каждом устройстве.

Формат команды:

```
>DEV <BK>
```

или >DEV устройство: <BK>

Первый формат обеспечивает распечатку всех устройств, известных системе, например

```
>DEV <BK>
DK0: PUBLIC MOUNTED
DK1: MOUNTED
DK2:
DK3: MOUNTED
LP0: PUBLIC LOADED
LP1: OFFLINE
PP0: LOADED
PR0: LOADED
TT0: [1,34] — LOGGED ON
TT1:
```

Здесь MOUNTED — означает, что данное устройство смонтировано;

PUBLIC — означает, что устройство установлено как общее;

OFFLINE — указывает на то, что устройство не под-

ключено к ЭВМ;

[UIC] — LOGGED ON — означает, что с данного терминала зарегистрировался пользователь;

LOADED — означает, что устройство подключено к работе.

Команда LUN применяется, если пользователю нужно узнать, какие физические устройства назначены для данной задачи.

Формат команды:

>LUN имя-задачи <БК>

Например, для программы PROG1

>LUN PROG1 <БК>

По этой команде список устройств, назначенных задаче PROG1, отображается на экране дисплея.

За каждой задачей, построенной программой ТКВ, закреплены физические устройства, необходимые для ее выполнения. При этом информация об этих устройствах записывается и хранится в заголовке данной задачи.

Заголовок, содержащий справочную информацию, является неотъемлемой частью задачи и загружается в оперативную память вместе с ней. Используется заголовок процессором системы, который выполняет соответствующие закрепления.

Следует подчеркнуть, что команду LUN можно подавать только в том случае, если задача в системе установлена.

Для выяснения того, какие задачи установлены в системе, применяется команда программы связи с оператором TAS. Формат команды:

>TAS <БК>

По этой команде на терминал для каждой задачи, установленной в системе, выводится не только имя, но и более обширная информация: имя задачи; идентификация каждой версии задачи; имя раздела памяти, соответствующего задаче; приоритет задачи; объем задачи, выраженный в байтах (восьмеричное число); идентификатор устройства, с которого установлена задача; адрес задачи на диске (номер блока на диске).

Команды управления позволяют запускать и выполнять задачи, возобновлять приостановленные и др.

Для инициации выполнения задачи применяется команда MCR RUN¹.

¹ Подробнее об этой команде см. в 4.1.8.

Формат ее:

>RUN имя-задачи <BK>

Если пользователю необходимо во время работы программы прервать выполнение задачи и освободить устройства, закрепленные за ней, применяется команда программы связи с оператором ABO. Формат ее:

>ABO имя-задачи <BK>

Если данная задача исключается, то освобождается также и занятый ею раздел памяти.

Например, чтобы прервать выполнение программы PROG2, достаточно набрать команду

>ABO PROG2 <BK>

Отметим при этом, что задачу, запущенную с другого терминала, можно прервать командой ABO только с привилегированного терминала.

4.1.7.2. Текстовый редактор EDI

Редактор текста позволяет создавать и модифицировать исходные программы (файлы).

С его помощью можно создать программу на Фортране, Коболе, языке ассемблера, файл данных и т. п.

EDI может читать строку или группу строк из входного файла в специальный буфер, где они могут подвергаться редактированию.

Когда строка (или блок) отредактирована, их можно записать в выходной файл, прочитать новый блок в буфер из вводного файла и т. д.

При этом выводной файл располагается на том же устройстве, что и вводной. Номер версии автоматически присваивается увеличением на единицу.

Запуск текстового редактора можно производить двумя способами:

>EDI <BK>

либо

>EDI спецификация файла <BK>

В первом случае на экране дисплея появляется отзыв-подсказка

EDI>

в ответ на которую пользователь должен ввести спецификацию требуемого файла. Спецификация файла должна иметь вид:

[устройство:] [код пользователя] имя файла.[тип ;версия]

Примечание. Здесь в квадратные скобки заключены элементы спецификации, которые могут подразумеваться по умолчанию.

Если специфицируется новый файл (т. е. указываемый файл еще не создан в данной директории на диске), предполагается, что пользователь хочет создать новый файл с указанным именем. При этом на экран дисплея выводится сообщение, напоминающее пользователю о том, что он создает новый файл, и разрешающее ввод

[CREATING NEW FILE]
INPUT

И редактор переходит в режим ввода текста.

Если указанный файл уже имеется в данной директории (его имя обнаруживается в оглавлении директории), то подразумевается, что пользователь намеревается его редактировать (или корректировать), и на экране дисплея высвечивается сообщение о том, какое количество информации прочитано с диска

[XXX LINES READ IN]
[PAGE1]
*

Здесь символы XXX означают количество строк редактируемого файла, перенесенных в буфер текстового редактора, а символ * означает, что EDI перешел в режим ввода команд (режим редактирования). Указатель строк редактора устанавливается при этом в начало текстового буфера.

Для перевода EDI из режима редактирования в режим ввода текста пользователю достаточно в ответ на символ * ввести команду

*I<BK>

Все строки текста, которые будут вводиться с клавиатуры, начиная с данного момента, автоматически добавляются к файлу как новый текст, следующий за текущей строкой редактируемого файла.

Для перевода редактора из режима ввода текста в режим редактирования достаточно нажать дважды символ BK; при этом на экране дисплея взамен курсора снова будет высвечиваться символ *.

Для завершения сеанса редактирования могут быть использованы следующие команды:

*EX [имя файла] <BK>

либо

*EX <BK>

По этой команде работа текстового редактора EDI завершается, управление передается программе связи с оператором. На устройстве пользователя создается новый файл, содержащий отредактированный текст с именем, заданным в спецификации; если же спецификация умалчивается, то созданный файл будет иметь такое же имя, как и исходный, и номер версии на 1 больше. При этом исходный (редактируемый или вводной) файл не изменяется и сохраняется по-прежнему на устройстве в директории пользователя;

*ED [имя файла] <BK>

или

*ED <BK>

Эта команда работает аналогично команде *EX, но по этой команде исходный (редактируемый) файл в директории пользователя уничтожается, сохраняется только результирующий (отредактированный текст);

*CL [имя файла] <BK>

или

*CL <BK>

Эта команда работает аналогично команде *EX; вводной файл сохраняется в директории пользователя; но управление не передается MCR, а остается в редакторе, поэтому пользователь может вызвать на редактирование новый (следующий) файл. При выполнении этой команды на экран дисплея выводится подсказка

EDI>

означающая, что редактор продолжает работу;

*CD [имя файла] <BK>

или

*CD <BK>

Работа этой команды аналогична работе команды *ED, но при этом управление сохраняется за редактором; на экране высвечивается подсказка

EDI>

означающая, что можно вызывать на редактирование следующий файл.

Если по какой-либо причине пользователь не хочет сохранить новый (выводной) файл, полученный в ре-

зультате редактирования, то он может применить команду

***KILL <BK>**

При выполнении этой команды исходный (вводной) файл сохраняется без изменений, а выводной файл уничтожается. На экран дисплея выводится подсказка

EDI>

В ответ на эту подсказку пользователь может вызвать для редактирования следующий файл либо командой <UC> <Z> передать управление монитору.

В случае системных сбоев работа редактора текстов EDI может завершиться аварийно. В этом случае в директории пользователя будет сохранен исходный файл и создан новый пустой файл с тем же именем и типом, но с номером версии на 1 больше, чем у исходного файла. Рекомендуется перед новым вызовом этого файла для редактирования распечатать каталог с помощью программы PIP и при необходимости удалить пустой файл.

ОСНОВНЫЕ КОМАНДЫ РЕДАКТОРА EDI:

1) ***I строка текста <BK>**

По этой команде вводится и помещается вслед за текущей строкой вводимая строка текста. Режим редактирования сохраняется.

2) ***I <BK>**

По этой команде редактор переключается в режим ввода.

3) ***AP подстрока <BK>**

Набранная подстрока добавляется (дописывается) к концу текущей строки и в откорректированном виде строка выводится на экран.

4) ***R строка <BK>**

По этой команде текущая строка заменяется на указанную строку.

5) ***DP[N] <BK>**

или

***DP <BK>**

По этой команде удаляются текущая строка и [N — 1] последующих строк, если N — положительное, и [N] предыдущих строк, если N — отрицательное. По умолчанию

принимается $N=1$ (т. е. удаляется одна лишь текущая строка). На экран выводится строка, первая после всех удаленных строк.

6) *NP[N] <BK>

Эта команда перемещает указатель строк на N строк вниз, если N — положительное, либо вверх (к началу файла), если N — отрицательное. Умолчания для значения N здесь применять нельзя. Новая текущая строка высвечивается на экране.

7) <UC> / [

Указатель строк по этой команде устанавливается на предыдущую строку. Новая текущая строка высвечивается на экране.

8) *P [N] <BK>

Распечатываются на экране текущая строка и $N - 1$ последующих строк.

9) *B <BK>

Указатель устанавливается в начало буфера.

10) *BO <BK>

Указатель устанавливается в конец буфера.

11) *C/подстрока 1/подстрока 2 <BK>

В текущей строке подстрока 1 заменяется на подстроку 2. Вместо символа / можно использовать любой служебный знак (для Фортран-программ рекомендуется #).

Если окажется, что подстрока 1 в команде отсутствует, то подстрока 2 вставляется в начало текущей строки с первой позиции, командная строка в этом случае имеет вид

*C // подстрока 2 <BK>

Если указанная подстрока 1 не будет найдена, то на экран дисплея будет выдано сообщение

[NO MATCH]

*

При этом текущая строка не изменяется.

12) *LC /подстрока 1/ подстрока 2 <BK>

Заменяется подстрока 1 во всех случаях, когда она встречается в текущей строке, на подстроку 2.

13) *PA /подстрока 1/ подстрока 2 <BK>

Команда предписывает EDI просмотреть все строки данного файла и заменить подстроку 1 на подстроку 2.

14) *L подстрока <BK>

С помощью этой команды обследуется буфер, начиная с текущей строки, в поисках строки, содержащей указанную подстроку. Подстрока может находиться в любой части строки. Найденная строка будет выводиться на экран.

15) *F подстрока <BK>

Примечание. В качестве подстроки для Фортран-программ в этой команде может указываться метка оператора.

Эта команда работает аналогично предыдущей команде, но искомая строка должна начинаться с указанной подстроки и именно с первой позиции в ней.

Рассмотрим пример действий редактирования с помощью EDI с подробным разбором команд.

Пусть редактируемая Фортран-программа содержит следующий фрагмент:

```
С ПРОГРАММА KENT
  DIMENSION CH (15)
  CALL ASSIGN (2, 'CH.DAT')
  READ (2, 10) CH
10 FORMAT (15 F4. 1)
  K1 = 0
  K2 = 0
  DO 20 I = 1, 15
```

и т. д.

Пусть в данном примере во второй строке после CH(15) необходимо добавить запятую и M(15); после третьей строки пропущен оператор назначения канала 5 устройству печати ("LP:"), а в операторе формат должно быть указано F3.1 (а не F4.1).

Последовательность действий пользователя:

>EDI <BK>	— запуск редактора;
EDI>KENT.FTN <BK>	— в ответ на отзыв редактора указывается программа пользователя, подлежащая редактированию;
.0065 LINES. READ IN]	— программа KENT обнаружена в директории пользователя; EDI устанавливается в режим редактирования, сообщает, что прочитано из файла 65 строк и на экран выводится первая страница;
[PAGE 1]	
*	— первая строка редактируемого текста станет текущей и будет видна на экране;
*<BK>	

С Программа KENT

* <BK>	— вывод на экран следующей строки;
* DIMENSION CH (15)	
* A,M(15) <BK>	— команда дописывает строку;
* DIMENSION CH(15), M(15)	— вывод на экран исправленной строки;
* <BK>	— текущей (и видимой) станет третья строка текста;
* CALL ASSIGN (2,'CH.DAT')	
* I CALL ASSIGN (6, 'LP:') <BK>	— вставляем пропущенную строку;
* CALL ASSIGN (6, 'LP: ')	— вывод пропущенной строки на экран;
* F 10 <BK>	— на экран выводится строка с меткой 10;
* 10 FORMAT (15 F4. 1)	— выведенная на экран строка текста программы стала текущей строкой;
* C # 4.1 # 3.1 <BK>	— замена 4.1 на 3.1;
* 10 FORMAT (15 F3.1)	— на экран выводится строка после замены;
* ED <BK>	и т. д. — выход из EDI по окончании редактирования;
[EXIT]	— вывод на экран подсказки
>	— отзыва MCR

4.1.7.3. Экранный редактор текста TED

В новых версиях системы ОСРВ помимо текстового редактора EDI может также применяться экранный редактор TED. Он работает не с файлом, где хранится исходный текст, а с его копией, содержащейся в текстовом буфере редактора.

Для чтения файла в буфер применяется команда

TED>R имя файла [N1 N2]

где R — означает чтение; имя — имя считываемого файла.

Если заданы необязательные численные значения N1 и N2, то в буфер будет прочитана часть файла со строки N1 до строки N2, если они не указаны, то будет переписан в буфер весь файл полностью.

Запуск экранного редактора производится командой

>TED <BK>

Редактор TED имеет два режима работы: командный и экранный. В командном режиме исполняются команды, которые требуют задания параметров. В экранном режиме пользователь видит перед собой на экране часть

текстового буфера и с помощью курсора (маркера) и клавиатуры может выполнять редактирование.

В командном режиме возможны следующие операции:

а) получение вспомогательной информации (HELP):

TED>HE<BK>

На экран дисплея выводится краткий справочник по работе с редактором;

б) завершение работы с TED осуществляется командой

TED> <УС> / <Z> <BK>

где <УС> — управляющий символ;

в) чтение текста файла и вставка его в буфер вслед за строкой Р либо за последней строкой в буфере, если Р не указано:

TED>R [Р] имя файла [N1 N2] <BK>

г) очистка текстового буфера (CLEAR):

TED> CL <BK>

д) приостановка работы редактора и освобождение терминала, чтобы дать возможность вызвать другую программу с этого же терминала, сохранив редактируемый текст в буфере:

TED> SS <BK>

е) запись (WRITE) содержимого текстового буфера от строки Р1 до строки Р2 в указанный файл вслед за строкой N этого файла:

TED>W [Р1 Р2] имя файла [N] <BK>

Если N не указано, то запись будет производиться с начала файла. (Записать весь текст в виде файла: TED>W<BK>.)

ж) вывод (PRINT) строк с Р1 по Р2 на экран дисплея:

TED> PR P1 P2<BK>

з) удаление (DELETE) строк с Р1 до Р2 включительно:

TED>D P1 P2<BK>

Удаление одной строки Р1:

TED>D P1 <BK>

и) замена группы символов string1 на string2 в строках с N1 по N2:

TED>SU/string 1 / string 2 [N1 N2]<BK>

к) сдвиг строк с P1 по P2 на N позиций вправо, если N — больше нуля, или влево, если N — меньше нуля:

TED>SH P1 P2 N <BK>

л) переход из командного режима в экранный режим; причем первой строкой экрана будет строка P:

TED>[P] <BK>

Если P не указано, то первой строкой экрана будет строка, на которой находился маркер перед переходом в экранный режим.

Кроме указанных имеется также ряд команд для более сложной работы, в том числе для восстановления текста, утраченного в случае аварийного завершения работы (за счет специального файла протокола).

Экранный режим — это режим редактирования текста. Для перехода в экранный режим достаточно нажать клавишу ВК (возврат каретки). Для обратного перехода в командный режим используется управляющий символ УС/С.

Управляющие символы, как обычно, задаются одновременно нажатием клавиши УС и соответствующей клавиши данного символа. Интерпретируются управляющие символы как команды.

К основным командам экранного режима относятся команды, с помощью которых можно выполнить следующие действия:

переместить курсор (маркер) на одну позицию вправо — УС/Х, на одну позицию влево — УС/Н;

переместить курсор на одну строку вверх — УС/У, — вниз — УС/З;

переместить курсор в первую строку экрана — УС/Л;

переместить курсор в первую позицию строки — ВК;

удалить текущую строку — УС/Λ;

удалить символ — УС/;

вставить пустую строку вслед за текущей — ПС;

вставить пробел в данной позиции — УС/ \;

перейти в режим вставки символов — УС/С.

4.1.7.4. Программа работы с файлами РІР

Программа РІР передает данные с одного стандартного устройства на другое, а также выполняет некоторые функции управления: копирует файлы, уничтожает их, переименовывает, выдает оглавления директориев, устанавливает по умолчанию устройства и UIC.

Командная строка имеет в общем случае следующий вид:

спецификатор выводного файла=спецификатор вводного файла/ключи.

Допускаются списки спецификаторов во вводной и выводной части командной строки.

В процессе создания, редактирования, компиляции и построения задачи LABOR1.FTN операционная система ОСРВ создает на диске в директории пользователя различные файлы.

Все эти файлы остаются на диске, занимая место до тех пор, пока не будут стерты специальной командой. ОСРВ не стирает автоматически старые варианты файлов при возникновении новых. Необходимо уметь ориентироваться среди этого множества и вовремя уничтожать информацию, ставшую ненужной с помощью РІР.

Вызов РІР можно производить одной командной строкой. В этом случае выполнится действие, указанное в командной строке, и тотчас же команда вернет управление монитору (выведется подсказка MCR), например

```
>PIR /LI <BK>
```

Возможен другой вариант вызова. На клавиатуре дисплея следует набрать:

```
>PIR      <BK>  
PIR>/LI <BK>
```

В этом варианте РІР после выполнения команды снова выйдет на связь, и чтобы вернуть управление MCR, следует набрать управляющий символ УС/З.

Перечень основных ключей РІР и описание их функций приводятся в табл. 4.1.

Если нужно вывести файл LAB2.FTN на экран дисплея, следует набрать на клавиатуре команду

```
>PIR TI: = LAB2.FTN <BK>
```

Если требуется получить распечатку этого файла на бумаге (твердая копия) с помощью печатающего устройства LP:, достаточно набрать на клавиатуре команду

```
>PIR LP: = LAB2.FTN <BK>
```

По ключу /LI осуществляется отображение каталога файлов, например

```
>PIR /LI <BK>  
DIRECTORy DK1: [210, 16]  
10-OCT-86      12 : 40
```

LAB2.OBJ;2	2		28— SEP—86
LAB2.TSK;1	19	C	20— SEP—86
LAB2.MAP;1	4	C	25— SEP—86
LAB2.FTN;2	1		30— SEP—86

TOTAL OF 26/49. BLOCKS IN 4 FILES

Этот вариант команды предусматривает вывод следующей информации:

физический носитель, на котором хранятся файлы и UIC пользователя, которому принадлежит директорий. По умолчанию носителем будет стандартный системный диск (SY:);

дата и время создания программой PIP перечня;

имя, тип и номер версии каждого файла;

количество блоков, использованных каждым файлом (блок составляет 256 двухбайтовых слов);

код файла (C — файл смежный, отсутствие кода — несмежный; L — заблокированный, т. е. содержащий испорченную информацию);

дата создания каждого файла в директории;

суммарный размер всех перечисленных файлов, выраженный в блоках, и количество блоков, выделенных для всех файлов.

Таблица 4.1

Ключ	Значение	Описание действия ключа
/LI	Перечислить	Выводит информацию о запрошенных файлах, содержащихся в директории пользователя
/PW	Скопировать с MJ на диск	Копирует файлы с магнитной ленты на диск
/BR	Кратко перечислить	Выводит сокращенный вариант каталога, вырабатываемого ключом /LI
/DE	Стереть	Уничтожает файлы, указанные в командной строке
/NV	Новая версия	Переписывает файл с новым номером версии
/PU	Очистить	Очищает указанные файлы (стирает все версии, кроме последней)
/PR	Изменить защиту	Изменяет код защиты файла указанным образом
/SP	Вывести на печать	Печатает на LP: указанные файлы
/RE	Переименовать	Переименовывает входной файл
/SD	Селективное стирание	Уничтожает каждый из указанных файлов только после соответствующего запроса и получения подтверждения

В программе PIP предусмотрены ключи для управления объемом информации, представленной в распечатках. Ключ /BR обеспечивает сокращенную форму с выводом на терминал пользователя, например

```
>PIP /BR <BK>  
  DIRECTORY DK1: [210, 16]  
  LAB2.OBJ;2  
  LAB2.TSK;1  
  LAB2.MAP;1  
  LAB2.FTN;2
```

Ключ /FU обеспечивает вывод каталога в более полной форме, в частности код защиты, который определяет, в какой мере данный файл может быть использован другими пользователями системы, например

```
[303, 12] [RWED, RWED, RWED, R]
```

Программа PIP позволяет получить информацию об одном файле или специфической группе файлов, имеющих в директории. Например, если нужно проверить, сколько версий файла LAB1.TSK существует в данной директории, следует набрать команду

```
>PIP LAB1.TSK; * /LI <BK>
```

Эта команда запрашивает перечень всех версий файла LAB1.TSK. Звездочка в поле номера версии указывает PIP на необходимость поиска всех версий файла.

Символ * в одном или нескольких полях спецификации файла ставится для обозначения всех спецификаций. Она также называется спецификацией свободного выбора.

Ниже приводится ряд случаев использования символа * для программы PIP:

*. *; * — обозначает все версии всех файлов всех типов;

*. DAT; * — обозначает все версии всех файлов типа DAT;

TEST. *; * — обозначает все версии и все типы файлов с именем TEST;

*. * — обозначает самые последние (по умолчанию) версии всех файлов всех типов;

*. DAT — обозначает самые последние версии всех файлов типа DAT;

TEST. * — самые последние версии всех типов файлов с именем TEST;

TEST.DAT — самая последняя версия файла TEST.DAT;

[*, *] — обозначает группу всех кодов идентификации

пользователя, включающую комбинацию кодов от 1 до 377;

[n1, *] — все номера членов группы в группе n1;

[*, n2] — все групповые номера для членов группы с номером n2.

Стирание файлов. Уточнив какие из файлов, находящихся в директории пользователя, более не понадобятся, их можно уничтожить.

Например, чтобы сохранить версии с наивысшим номером для каждого файла, требуется стереть следующие файлы:

LAB1.OBJ;1

LAB1.TSK;1

LAB1.TSK;2

Чтобы уничтожить эти файлы с помощью ключа /DE программы PIP, нужно набрать команду

```
>PIP LAB1.OBJ;1, LAB1.TSK;1, LAB1.TSK;2 /DE <BK>
```

Следует отметить, что ключ /DE требует обязательно либо явного задания номера версии, либо свободного выбора * в поле версии. Однако использование * нецелесообразно, когда директорий содержит файлы с этим же именем, но с другими типами и версиями, которые нельзя уничтожить. В этом случае необходимо явно специфицировать каждый файл, который подлежит уничтожению.

Очистка файлов. Если нужно уничтожить все версии файла, кроме самой последней, то ключ очистки /PU является более эффективным по сравнению с ключом /DE. Поэтому вместо предыдущей команды лучше набрать следующую:

```
>PIP LAB1. */PU <BK>
```

Очистка не затрагивает файлы, имеющие единственную версию. Заметим, что спецификация файла для ключа очистки не содержит поля версии.

Чтобы убедиться в том, какие файлы сохранились в директории после действия ключа /PU, можно воспользоваться следующей командой:

```
>PIP /LI <BK>
```

DIRECTORY DK1: [301 , 12]

28—MAR—86 09:15

LAB1.OBJ;2 2 25—MAR—86

LAB1.MAR;1 4 25—MAR—86

LAB1.STB;1 3 25—MAR—86

LAB1.FTN;2 1 24—OCT—86

LAB.TSK;3 29 C 25—OCT—86

TOTAL OF 39./49 BLOCKS IN 5 FILES

Селективное стирание файлов. Ключ селективного стирания PIP обеспечивает перед каждым стиранием запрос на его разрешение (Y — да, N — нет).

Можно специфицировать активное стирание следующим образом:

```
>PIP LAB1.*; */SD <BK>
```

В ответ программа PIP выдает детальный запрос:

```
DELETE FILE DK1: [301,12] LAB1.FTN;1 ?Y  
DELETE FILE DK1: [301,12] LAB1.OBJ;1 / ?N
```

В качестве реакции на ответы пользователя PIP стирает файл LAB1.FTN;1, но сохраняет файл LAB1.OBJ и т. д.

Копирование файлов. Копирование файлов является стандартной функцией PIP, поэтому при правильном наборе командной строки без ключей программа PIP осуществляет операцию копирования. Например, следующая команда копирует файл LAB1.MAP на системном устройстве SY: в директорию того же пользователя на диске DK2: (код идентификации пользователя выводится по умолчанию):

```
>PIP DK2: = LAB1.MAP <BK>
```

Здесь входной файл — тот, который следует скопировать, выходной файл — новая его копия.

Если опущены директории пользователя, имя файла, тип его или номер версии в спецификации выходного файла, программа PIP по умолчанию ассоциирует директорий со стандартным директориумом пользователя, а имя, тип и номер версии файла — с эквивалентными полями в спецификации входного файла.

Для обеспечения возможности копирования файла в директорий, размещенный на другом носителе, необходимо, чтобы к моменту копирования этот директориум уже существовал. В системе с защитой для многих пользователей директориум пользователя на системном устройстве SY: является единственным, который создается системой автоматически. Если пользователь заблаговременно не провел создание директориума на выходном носителе DK2:, то программа PIP выдает следующее сообщение:

```
PIP — — CANNOT FIND DIRECTORY FILE
```

и попытка копирования будет прекращена.

Переименование файлов. Существующие файлы переименовываются применением ключа /RE программы PIP, например

>PIP LAB2.*;* = LAB1.*;* /RE <BK>

Эта команда предписывает программе PIP заменить имена файлов LAB1 всех типов и всех версий на LAB2.

Отметим, что при использовании ключа /RE необходимо четко специфицировать цифрой или символом * в полях версии как входного, так и выходного файла. Символ * в спецификации выходного файла указывает, что типы и версии переименованных файлов не меняются.

Если требуется переписать файл на другой носитель с новым именем, необходимо использовать команду COPY (выполняется по умолчанию), введя новое имя в спецификацию выходного файла в командной строке, например

>PIP DK3: LAB2. *; * = LAB1.*;* <BK>

Эта команда предписывает программе PIP копировать файлы с именем LAB1 всех типов и всех версий, хранящиеся на системном устройстве SY:, на диск DK3, где они будут иметь имя LAB2.

Ключ изменения кода защиты файла /PR. Операционная система ОСРВ различает соответственно доступу к файлу четыре категории пользователей: системный, владелец, групповой, общий.

Системный — привилегированный пользователь, у которого номер группы в [UIC] меньше 10, он имеет доступ к системным программам.

Владелец — пользователь, в директории (и на устройстве) которого создан данный файл с [UIC], равным [UIC] директория.

Групповой — пользователь, имеющий в [UIC] номер группы, совпадающий с номером группы владельца.

Общий — все остальные пользователи.

Для каждой категории пользователей может быть установлен с помощью PIP доступ для чтения файла (R), записи (W), расширения (E) и удаления файла (D).

Категория пользователя в командной строке идентифицируются с помощью ключей: /SY — системный, /OW — владелец, /GR — групповой, /WO — общий.

Например, пусть нужно устанавливать файлу RAB.TXT следующие виды доступа:

системному пользователю — все виды доступа;

владельцу — все виды доступа;

групповому — только чтение и запись;

общему — только чтение.

В этом случае командная строка будет иметь вид:

>PIP <BK>

PIP >RAB.TXT/PR/SY:RWED/OW:RWED/GR:RW/WO:R <BK>

Конкатенация файлов. Один файл можно дополнять другим (или несколькими другими) с помощью ключа /AP. В частности, это бывает удобно использовать при создании библиотечных файлов.

По ключу /AP программа PIP открывает выводной файл, указанный в левой части командной строки и к его концу добавляет указанные вводные файлы (в правой части командной строки) в том порядке, в каком они перечислены в команде.

Формат команды:

>PIP COMFILE/AP = FILE1,FILE2,..., FILEN <BK>

В спецификациях файлов в этой командной строке допускаются умолчания в отношении устройства и UIC. Имя, тип и номер версии необходимо указывать.

Например, пусть к выводному файлу КОММ.TXT;2 необходимо присоединить файлы А.TXT;1 В.TXT;3. Командная строка в этом примере будет выглядеть следующим образом:

>PIP КОММ.TXT; 2/AP = А.TXT;1, В.TXT;3 <BK>

Следует подчеркнуть, что в действительности PIP в этом случае осуществляет конкатенацию с копиями, так что в результате выполнения команды вводные файлы А.TXT;1 В.TXT;3 остаются в директории неизменными.

Копирование файлов с магнитной ленты на диск. Если пользователю необходимо скопировать на диск несколько файлов с накопителя на магнитной ленте при выполнении одной командной строки PIP, применяется ключ /RW.

При использовании этого ключа после считывания очередного файла из списка в командной строке магнитная лента перематывается к началу и производится поиск очередного файла.

Если ключ /RW не указан, то после считывания очередного файла лента просматривается до метки конца, затем перематывается к началу, после чего производится поиск очередного файла из списка.

Формат командной строки:

>PIP [устр-во:] [UIC] [выводной файл]=MT[N] : FILE1,..., FILEK/RW

Примечание. Записи, заключенные в квадратные скобки, являются в строке необязательными.

Пример 1. Требуется скопировать с магнитной ленты MT1: в директорию пользователя все файлы типа .DAT с последними версиями; командная строка в этом случае может быть записана

```
>PIP =MT1: *.DAT/RW <BK>
```

Здесь в левой части строки по умолчанию подразумеваются диск и директорию отдавшего команду пользователя.

Пример 2. Требуется последовательно распечатать на АЦПУ все текстовые файлы, хранящиеся на магнитной ленте на устройстве MT0. Командная строка в этом случае

```
>PIP LP: = MT0:*TXT/RW
```

4.1.7.5. Трансляция исходного файла

После создания исходного модуля, например Фортран-программы LABOR1.FTN, следующим этапом работы является компиляция (трансляция) с целью получения объектного модуля, записанного в двоичном коде, понятном для ЭВМ.

Процесс трансляции начинается с вызова (загрузки в оперативную память) соответствующего компилятора, например Фортран IV. Вызов транслятора и задание на трансляцию запишутся следующей программной строкой:

```
>FOR LABOR1.OBJ = LABOR1.FTN <BK>
```

Здесь входным является файл LABOR1.FTN, выходным — объектный модуль LABOR1.OBJ.

Соглашение по умолчанию обеспечивает восприятие типа исходного файла, как .FTN, а выходного — как .OBJ, например

```
>FOR LABOR1 = LABOR1 <BK>
```

Если необходимо получить листинг трансляции (распечатку на бумаге) на устройство LP:, следует набрать

```
>FOR LABOR1, LP: = LABOR1 <BK>
```

Наряду с этим в тех случаях, когда необходимо транслировать одну за другой несколько Фортран-программ, можно осуществить вызов Фортран-транслятора другим способом:

```
>FOR <BK>  
FOR>LABOR1, LP:=LABOR1 <BK>
```

В этом случае транслятор каждый раз автоматиче-

ски будет выходить на связь по окончании предыдущей трансляции, даже в случае аварийного ее завершения.

Создание объектного модуля и отображение листинга на экране терминала без вывода его на печать реализуются следующей командой:

```
>FOR LABOR1, TI: =LABOR1 <BK>
```

По окончании трансляции вновь выводится подсказка транслятора (транслятор себя идентифицирует).

Для того чтобы закончить работу транслятора и удалить его из памяти, передав управление монитору, достаточно набрать управляющий символ `YC/Z`:

```
FOR LABOR2, LP: = LABOR2 <BK>  
FOR>YC/Z
```

4.1.7.6. Компоновка объектных программ построителем задач ТКВ

Для компоновки объектных программ используется системная обслуживающая программа — построитель задач ТКВ.

Построитель задач создает готовую к выполнению программную единицу, называемую задачей, из одной или более объектных программ. Он связывает между собой объектные программы, разрешает ссылки к отдельным модулям из системной библиотеки или других библиотек объектных модулей и в результате создает файл образа задачи, который размещается на магнитном диске.

Для пуска построенной задачи пользователь применяет команды программы связи с оператором MCR.

Формат полной командной строки построителя задач включает в себя три выводных файла и любое количество вводных файлов.

Вводными файлами могут быть следующие:
образ задачи (тип — .TSK), содержащий образ готовой к выполнению задачи;

файл распределения памяти (тип — .MAP), содержащий информацию о размерах и расположении в памяти составных частей построенной задачи;

файл определения символов (тип — .STB), содержащий информацию о символических именах данной задачи.

Список вводных файлов включает в себя один или более объектных модулей, собираемых в файл образа задачи.

Если пользователь опускает в командной строке один или несколько из выводных файлов, он должен сохра-

нить на месте имени этого файла запятую для того, чтобы система могла определить наличие пустого поля.

Ниже приведены варианты возможных запросов к построителю задач ТКВ на получение выводных файлов (поле каждого файла определяется системой по номеру позиции его имени в командной строке, в левой части которой размещаются имена выводных файлов, а в правой — вводных).

- | | |
|--------------------------|--|
| >TKB PROG,PROG,PROG=PROG | — запрашиваются все три выводных файла; |
| >TKB ,PROG = PROG | — запрашивается только файл определения символов; |
| >TKB ,PROG, PROG = PROG | — запрашивается файл распределения памяти и файл определения символов; |
| >TKB ,PROG=PROG | — запрашивается только файл распределения памяти; |
| >TKB PROG, ,PROG=PROG | — запрашивается файл образа задачи и файл определения символов; |
| >TKB PROG, PROG = PROG | — запрашивается файл образа задачи и файл распределения памяти; |
| >TKB PROG = PROG | — запрашивается только файл образа задачи |

Из приведенных вариантов написания командной строки ТКВ следует, что если опускается последний или два последних идентификатора, то запятая на их месте не сохраняется. В практической работе пользователей наиболее широкое применение находит последний вариант.

Следует отметить, что длинный список вводных файлов может привести к тому, что длина строки команды программы ТКВ превысит максимально допустимую длину строки терминала (80 символов).

В этом случае программа ТКВ должна вызываться следующим образом:

>TKB <BK>
TKB>

При этом командная строка может вводиться частями, причем ввод каждого участка строки завершается нажатием клавиши ВК, после чего снова на экране дисплея выводится подсказка

TKB>

Полностью ввод командной строки завершается сим-

волами прекращения ввода //, которые набираются на клавиатуре непосредственно после отзыва-подсказки ТКВ>.

Например, командная строка

```
>ТКВ PROG, PROG, SYM=PROG, STAT, F77LIB.OLB/LB  
<БК>
```

может быть задана следующим образом:

```
>ТКВ <БК>  
ТКВ> PROG, PROG, SYM=PROG <БК>  
ТКВ> STAT <БК>  
ТКВ > F77LIB.OLB/LB <БК>  
ТКВ> // <БК>
```

Для указания характеристик строящейся задачи могут использоваться необязательные параметры.

Если пользователь вводит символ «/» вслед за спецификацией вводных файлов, построитель задач будет запрашивать информацию о необязательных параметрах

ENTER OPTIONS:

и ожидать ее ввода. Например:

```
>ТКВ <БК>  
ТКВ> PROG, PROG = PROG <БК>  
ТКВ> / <БК>  
ENTER OPTIONS:  
ТКВ> ACTFIL=10 <БК>  
ТКВ> UNITS=16 <БК>  
ТКВ> // <БК>
```

В этой последовательности команд пользователь вводит необязательные параметры ACTFIL и UNITS, а затем символы // для завершения ввода необязательных параметров, инициации построения задачи и возврата управления программе связи с оператором.

В одной строке может быть более одного параметра. Для их разделения в этом случае используется восклицательный знак (!).

Например:

```
>ТКВ UNITS = 14 ! ACTFIL = 6 ! MAXBUF=1200 <БК>  
ТКВ > // <БК>
```

Из необязательных параметров распределения памяти отметим два параметра, предназначенных для программ, написанных на языках высокого уровня. По этим параметрам построитель задач может менять размер выделяемой памяти.

Число активных файлов (параметр ACTFIL). Этот параметр объявляет число файлов, которое задача может иметь открытыми одновременно. Для каждого активного файла выделяется примерно 512 байт.

Если число активных файлов, используемых задачами, меньше предполагаемого по умолчанию (значение по умолчанию равно четырем), можно воспользоваться необязательным параметром `ACTFIL` для экономии памяти. Если же число активных файлов больше предполагаемого по умолчанию, то нужно обязательно использовать параметр `ACTFIL` для выделения дополнительной памяти; в противном случае построенная задача не может быть запущена.

Формат параметра:

`ACTFIL` = максимальное число файлов

Здесь максимальное число файлов — десятичное число файлов, которые могут быть одновременно открыты в данной задаче.

Размер буфера записей (параметр `MAXBUF`). Необязательный параметр `MAXBUF` объявляет требуемый максимальный размер буфера записей для любого файла, используемого задачами. Данный параметр должен быть использован для расширения буфера, если требуется обработать файл, в котором размер максимальной записи превышает предполагаемую по умолчанию длину буфера (132 байта).

Формат параметра:

`MAXBUF` = максимальная длина записи

Здесь максимальная длина записи — целое десятичное число, больше 132.

Например, при подсчете максимальной длины буфера для обработки файлов в Фортран-программе следует учитывать, что 1 слово файла вещественных чисел занимает объем 4 байта; 1 слово файла целых чисел — 2 байта; символ составляет 2 байта. Если файл `M(400)` содержит целые числа, то для него объем буфера будет (400×2) 800 байт; для файла `A(300)`, содержащего вещественные числа (300×4) , — 1200 байт; для файла `KP(150)` символической информации (150×2) — 300 байт.

Из трех активных файлов следует выбрать максимальное значение для буфера (`MAXBUF=1200` байт).

Необязательный параметр использования логических устройств. Необязательный параметр `UNITS` объявляет число устройств ввода-вывода, используемых данной задачей. При этом число устройств и значение, указанное в параметре `UNITS`, должны быть согласованы: попытка назначить устройству номер, который больше общего числа объявленных устройств, воспринимается как ошибка.

UNITS = максимальный номер устройства

Здесь максимальный номер устройства — десятичное целое число в пределах от 0 до 250 в соответствии с условиями конкретной программы пользователя.

По умолчанию система выделяет 6.

Пример. Пусть в Фортран-программе содержатся следующие строки:

```
DIMENSION SK (150), TABI(150),SHP(150),  
COD(150),BH(150),TR(150),TK(150)  
CALL ASSIGN (3, 'LP:')  
CALL ASSIGN (7, 'SK.DAT')  
CALL ASSIGN (8, 'TABI.DAT')  
CALL ASSIGN (9, 'SHP.DAT')  
CALL ASSIGN (10, 'COD.DAT')  
CALL ASSIGN (12, 'BH.DAT')  
CALL ASSIGN (14, 'TR.DAT')  
CALL ASSIGN (11, 'TK.DAT')
```

.

В этом случае необходимо запросить у ТКВ максимальный номер

UNITS=14

Следует заметить, что обращение в программе может производиться только к 7 устройствам.

Сжатое написание такой командной строки для ТКВ имеет вид

>ТКВ ACTFIL = 7 ! UNITS = 14! <БК>

Порядок задания параметров UNITS, ACTFIL, MAX-BUF в командной строке значения не имеет.

4.1.8. ВЫПОЛНЕНИЕ ЗАДАЧИ

В результате выполнения описанных ранее процедур создается программная единица, называемая задачей пользователя, выполнение которой осуществляется под управлением системы ОСРВ.

Далее осуществляется компоновка задачи, в процессе которой создается файл образа задачи и происходит его размещение на системном диске пользователя.

Для установки и запуска задачи нужно набрать команду программы связи с оператором RUN, использующую идентификатор задачи как параметр.

Например, чтобы запустить задачу PROG1.TSK, следует набрать командную строку

RUN PROG1 <БК>

Тип файла .TSK берется по умолчанию (другим при запуске программы он быть не может).

Команда RUN указывает системе на необходимость выполнения следующих действий:

определение местонахождения файла PROG1.TSK на системном диске пользователя;

загрузка копии образа задачи в память ЭВМ;

запуск задачи на выполнение с выдачей сообщений об аварийных ситуациях;

очистка памяти после завершения ее работы.

Следует подчеркнуть, что файл образа задачи при этом остается на диске пользователя до тех пор, пока пользователь не удалит его с диска специально.

4.2. РАБОТА С ИНТЕРПРЕТАТОРОМ БЕЙСИК

4.2.1. РАБОТА НА МИНИ-ЭВМ СМ-4

Для работы с Бейсиком пользователь должен вначале как обычно зарегистрироваться в системе и затем в ответ на подсказку MCR (>) набрать на клавиатуре директиву загрузки Бейсика. Загрузившись в оперативную память, интерпретатор выводит на экран свой отзыв (READY), который будет наблюдаться впоследствии после каждой выполненной команды:

>BAS — вызов интерпретатора;
READY — отзыв (готов к работе).

Для выяснения в начале сеанса работы, какие именно программы имеются в данный момент в директории пользователя, можно воспользоваться командой CAT, которая применяется без параметров. По этой команде на экран дисплея выводится список всех программ, хранящихся в директории пользователя, с указанием номеров их версий.

При вводе программы применяется команда NEW, которая очищает содержимое памяти и присваивает имя вводимой программе, например

```
READY  
NEW PROG4
```

Программа набирается по одной команде (нажатием клавиши ВК), причем уже во время ввода она может подвергаться корректировке.

Для исключения одной строки программы можно набрать номер этой строки и нажать ВК.

Для того чтобы вставить строку, достаточно набрать ее номер, весь текст этой строки и нажать клавишу ВК. (С этой целью и рекомендуется нумеровать строки с шагом 10.)

Для того чтобы заменить строку, достаточно набрать номер этой строки, весь новый текст и нажать клавишу ВК.

Для сохранения введенной программы применяется команда SAVE, она записывает программу, находящуюся в оперативной памяти, в файл на диске, присваивая этому файлу указанное имя, например

```
READY  
SAVE PROG4
```

Если имя не указано, то будет взято имя программы в памяти.

Для того чтобы получить распечатку программы на бумаге, нужно указать имя устройства печати (LP:):

```
READY  
SAVE LP:
```

По команде REPLACE содержимое файла с указанным именем заменяется программой, находящейся в данный момент в памяти:

```
READY  
REPLACE PROG3
```

Команда OLD загружает программу, хранящуюся в файле на диске, в оперативную память, например

```
READY  
OLD VESNA
```

В этом случае загружается программа, хранящаяся в файле VESNA.BAS.

Для работы с текущей программой, находящейся в оперативной памяти ЭВМ, наиболее часто применяются следующие команды интерпретатора Бейсик.

Команда LIST обеспечивает вывод на экран дисплея программы. Для вывода всей программы записывается только слово LIST. Если нужно просмотреть программу с номера строки HCTP1 до номера строки HCTP2, следует набрать на клавиатуре (в ответ на отзыв READY)

```
READY  
LIST HCTP1—HCTP2
```

Если нужна всего одна строка с номером HCTP, то следует набрать

```
READY  
LIST HCTP
```

Для запуска программы, составленной пользователем и находящейся в памяти, используется команда RUN, например

```
READY  
RUN PROG1
```

Команда переименования находящейся в памяти программы — **RENAME**.

При редактировании программы в оперативной памяти чаще всего применяются команды **DEL** и **SUB**.

Команда **DEL** исключает строки программы: **DEL** — все строки программы; **DEL HCTP1** — строку с указанным номером; **DEL HCTP1 — HCTP2** — часть программы между указанными номерами **HCTP1** и **HCTP2**.

Для редактирования строки программы в памяти (т. е. замены в ней одной группы символов другой группой) применяется команда интерпретатора **SUB**. Формат ее:

```
SUB HCTP @ стр. 1 @ стр. 2
```

где **HCTP** — номер редактируемой строки программы;

@ — символ-ограничитель;

стр. 1 — последовательность заменяемых символов;

стр. 2 — последовательность заменяющих их символов.

Например, в редактируемой строке ошибочно записано:

```
30 PRINT "Введите значение X"
```

Необходимо заменить символ **X** символом **Y**

```
READY  
SUB 30 @ X @ Y
```

Отредактированная строка примет вид

```
30 PRINT "Введите значение Y"
```

Для прерывания выполнения программы набирается команда **STOP**. Чтобы затем продолжить выполнение программы, достаточно команды продолжения с указанной строки **GO TO HCTP** (**HCTP** — номер строки продолжения).

Если необходимо переименовать программу, хранящуюся на диске, применяют оператор **NAME TO**. Формат его:

```
NAME стр.1 TO стр.2
```

где стр. 1 — строковая константа, задающая спецификацию (наименование) файла, подлежащую изменению;

стр. 2 — строковая константа, определяющая новое наименование файла.

Например:

NAME «PROG1.BAS» TO «PROG2.BAS»

Оператор KILL удаляет указанный файл. Формат его:

KILL стр. 1

где стр. 1 — имя удаляемого файла. Например:

```
READY
KILL VESNA2
```

В заключение рассмотрим пример подготовки, записи на диск и выполнения Бейсик-программы.

Пример. Пусть требуется составить Бейсик-программу для выполнения расчетов по формуле

$$S = e^{-ax} \cdot \lg \sqrt{x+1} - a \cdot e^y$$

при условии, что $a=0,4$; шаг $\Delta x=0,4$; $1 \leq x \leq 3$; шаг $\Delta y=0,4$; $-1 \leq y \leq +1$.

Зарегистрировавшись в системе (см. 4.1.4), следует на отзыв-подсказку монитора вызвать интерпретатор Бейсика (BAS), сообщить ему директивой, что создается новая программа PROG1, и ввести ее по строкам. (нажимая после каждой строки клавишу ВК).

```
>BAS
READY
NEW PROG1
READY
10 REM гр. 1125 Сергеев
20 REM лабораторная работа № 2
30 OPEN "LP:" FOR OUTPUT AS FILE#4
40 PRINT "ВВЕДИТЕ ЧИСЛО А" \ INPUT A
50 PRINT "ВВЕДИТЕ НАЧАЛЬНОЕ ЗНАЧЕНИЕ Y1 и КОН-
    НЕЧНОЕ ЗНАЧЕНИЕ Y2, ШАГ Y3"
60 INPUT Y1, Y2, Y3
70 PRINT "ВВЕДИТЕ НАЧ. X1, КОН. X2, ШАГ X3"
80 INPUT X1, X2, X3
90 FOR Y=Y1 TO Y2 STEP Y3
100 FOR X=X1 TO X2 STEP X3
110 S=EXP(-A*X)*LOG10((X+1)^(.5))-A*EXP(Y)
120 PRINT # 4, "Функция S=" ; S, "Аргумент X="; X, "Ар-
    гумент Y="; Y
130 NEXT X
140 NEXT Y
150 CLOSE#4
160 END
SAVE PROG1 — пользователь записывает свою програм-
    му на диск
READY
RUN PROG1 — запуск программы
```

После запуска на выполнение интерпретатор будет выводить на экран сообщение программы о запрашива-

емых значениях, соответственно строкам 40, 50, 70, и знак вопроса — для каждого из вводимых значений A, Y1, Y2, Y3, X1, X2, X3; после каждого из них необходимо набрать на клавиатуре соответствующие значения, разделяя их запятыми, а в конце каждого ввода по оператору INPUT следует нажимать клавишу ВК.

В результате циклических расчетов по программе (здесь цикл по X вложен в цикл по Y) будут отпечатаны на бумаге следующие строки:

Функция S = — . 0462586 аргумент X=1 аргумент Y=1

Функция S = — . 0385617 аргумент X=1.4 аргумент Y= — 1

Функция S = — . 988999 аргумент X=3 аргумент Y= + 1

В конце сеанса для выхода из Бейсик-интерпретатора достаточно в ответ на отклик READY набрать директиву BYE.

После этого Бейсик закончит свою работу и передаст управление MCR (на экране появится подсказка >).

4.2.2. РАБОТА НА МИКРОЭВМ ДВК-2

В состав программного обеспечения диалоговых вычислительных комплексов (ДВК) входит интерпретатор Бейсика (BASIC). Интерпретатор предназначен для проверки, анализа и выполнения операторов языка Бейсик. Работают с интерпретатором в диалоговом режиме под управлением ОС ДВК. Обмен информацией осуществляется по схеме: оперативная память (ОП) — периферийные устройства (ПУ), и наоборот. В качестве ПУ могут выступать: ТТ: — системный терминал LP: — устройство печати; ДХ: — накопитель на гибких магнитных дисках и некоторые другие устройства.

Последовательность этапов работы пользователя с интерпретатором Бейсика следующая:

- подготовка машины к работе в ОС ДВК;

- запуск интерпретатора;

- запись в оперативной памяти имени создаваемой программы;

- ввод программы в оперативную память;

- редактирование;

- запуск программы на исполнение;

- запись программы на ГМД и ее сохранение;

- завершение работы с интерпретатором.

Подготовка машины к работе в ОС ДВК состоит из включения машины и дисплея, установки на дисковод ГМД с ОС ДВК и интерпретатором; их загрузки; установки, если требуется, рабочего ГМД.

Запуск интерпретатора можно осуществлять несколькими способами:

а) с указанием вариантов включения средств расширения языка Бейсик, например BASIC/OS DVK V.02. При запуске такой командой последует вопрос системы о вариантах включения средств расширения языка. Оператор выбирает один из вариантов: А — включать все средства, I — включать отдельные средства, N — не включать средства расширения языка;

б) с указанием размера памяти, выделенной пользователю:

F RUN BASIC/N: 3000

где 3000 — размер памяти;

в) с указанием имени устройства, например ГМД, с которого запускается интерпретатор:

.RUN DX1: BASIC

Возможен также запуск интерпретатора командным файлом, что используется программистами при условии более полного ознакомления с работой ОС ДВК.

Запись имени создаваемой программы производится командой NEW. Команда NEW очищает содержимое памяти и присваивает имя программе, которая будет загружаться в ОП. Ответ системы на команду NEW следующий:

NEW FILE NAME — —

Далее пользователь в этой строке должен указать имя программы, под которым его программа будет известна системе. Имя программы представляет собой одно-шестиразрядный алфавитно-цифровой идентификатор. Имя программы должно быть оригинальным во избежание повторений, первым символом должна быть буква.

Если пользователь не указывает имя программы, и после NEW FILE NAME — — нажимает клавишу ВК, создаваемой программе в ОП присваивается имя NO-NAME.

О завершении работы команды NEW свидетельствует сообщение системы READY.

Ввод исходного текста программы осуществляется построчно, последовательным нажатием соответствующих алфавитно-цифровых клавиш и клавиши завершения строки ВК.

Редактирование. Бейсик располагает большими возможностями по редактированию текстов программ

в диалоговом режиме. Отметим, что с помощью команд Бейсика можно осуществлять редактирование в любом месте программы:

а) сразу после ввода ошибочного символа, т. е. в процессе ввода;

б) после ввода ошибочной строки;

в) по окончании ввода всей программы;

г) в ходе выполнения программы при обнаружении ошибки (после сообщения READY). Исправление ошибок при вводе текста программы (до нажатия клавиши ВК) можно осуществлять клавишей ЗБ, которая стирает последние (один или несколько) символов в строке.

Пример. В строке

30 Y=STN(X+

пользователь заметил, что неверно набрано имя функции SIN. Нажав клавишу ЗБ 5 раз, он отменяет (стирает) последовательно знаки +, X, (, N, T — курсор остановится в позиции за символом S. Пользователь набирает символы IN (X+, после чего строка будет иметь вид

30 Y=SIN(X+

и далее окончание строки.

Если строка введена (нажата клавиша ВК) и пользователь, обнаружив в ней ошибку, решил стереть эту строку, он набирает номер этой строки и клавишу ВК. Если возникла необходимость изменения уже записанной в память строки, достаточно набрать номер исправляемой строки и полностью новое ее содержание. При этом новая строка, заменившая старую, будет иметь тот же номер.

Редактирование программы осуществляется также с помощью команд DEL, SUB, STR, RENAME, LIST и др. Команда DEL исключает (стирает) одну или более строк текущей программы. Возможны следующие способы использования команды:

а) если команда вводится без указания номеров строк, то уничтожаются все строки текущей программы;

б) стирается одна строка, номер которой указывается через пробел после команды DEL. Например, по команде DEL 25 будет уничтожена 25-я строка;

в) исключается часть программы — от начала программы до указанной строки, например по команде DEL 30 будут исключены строки от начала программы до 30-й строки;

г) исключается часть программы между указанными строками, например по команде DEL 30—55 стираются

все строки, номера которых находятся в пределах от 30 до 55-й;

д) исключается часть программы от указанной строки до конца, например по команде DEL 55-будут уничтожены строки с 55-й до последней.

Команда SUB редактирует строку текущей программы.

Пример.

```
50 PRINT «НЕВЕРНЫЙ СЕМВОЛ»;
```

Внесенное исправление:

```
SUB 50 * E * U*03
```

где 50 — номер строки для редактирования;

* — символ-ограничитель (любой символ, допустимый в языке Бейсик, за исключением пробела, табуляции и цифр);

E — заменяемый символ;

U — вновь вводимый символ;

03 — число, указывающее, что заменяемый символ «E» исправляется в редактируемой строке при написании в 3-й раз.

Команда SCR очищает содержимое памяти и присваивает имя программе, которая будет загружаться.

Команда RENAME изменяет имя программы, находящейся в памяти, пример: по команде RENAME PRAK имя программы, находящейся в памяти, изменится на «PRAK».

Команда LIST позволяет осуществить просмотр программы, находящейся в памяти, путем распечатки на экран дисплея. Способы использования команды LIST такие же, как и команды DEL.

Запуск программы на исполнение осуществляется командой RUN.

Запись программы на ГМД. Многократное использование программ или работа пользователя с несколькими программами требует их сохранения путем записи на ГМД, что возможно с помощью команды SAVE. Для записи на ГМД программы с текущим именем достаточно команды SAVE. Если при этом пользователь намерен вывести текст программы на печать (получить листинг текущей программы), необходимо ввести команду

```
SAVE LP:
```

где LP: — имя устройства печати.

При дальнейшем использовании программы, записанной на ГМД, необходимо вначале загрузить ее в ОП командой OLD с указанием имени программы, затем дать команду RUN. Перед командой RUN можно вывести текст программы на экран дисплея командой LIST, чтобы убедиться, что в ОП загружена нужная пользователю программа.

Для удаления программы с определенным именем, например PROG1, используется команда UNSAVE PROG1.

Завершение работы с интерпретатором можно выполнить несколькими способами:

а) командой BYE, на что система ответит сообщением B>, далее следует набрать команду UNLOAD FG;

б) командой CTRL/C, что обеспечит немедленное прерывание команды или программы. На экране появится строка STOP;

в) функцией ABORT, которая обеспечивает принудительное завершение выполнения программы. Функция ABORT должна входить в текст программы. Формат функции:

НСТР LET A = ABORT(*),

где LET — оператор присвоения;

A — переменная, содержащая неопределенное значение после выполнения оператора присвоения;

* — аргумент функции прерывания, который может быть либо 0, если программа остается в ОП, либо 1, если программа стирается из ОП, оставляя заголовок NO-NAME.

4.3. ЗАДАНИЯ

Первое

1. Дать краткую характеристику операционной системы ОСРВ.

2. Опишите последовательность обработки программ в ОСРВ.

3. Как производится управление выполнением задач в ОСРВ?

4. Какова последовательность действий при выполнении программ в системе?

Второе

1. Поясните понятие файла. Как располагается информация на магнитном диске?

2. Как записывается командная строка в ОСРВ?

3. Какова полная запись спецификатора файла?
4. Какие устройства поддерживаются системой ОСРВ?

5. Что представляет собой директорий пользователя на диске?

6. Каковы виды доступа к файлам?

7. Как осуществляется защита файлов?

Третье

1. Как используется терминал для связи человека-оператора с операционной системой?

2. Каковы специальные клавиши терминала и их назначение?

3. Как используются управляющие символы при работе с терминала?

4. Как набирается командная строка на клавиатуре терминала?

Четвертое

1. В чем состоят функции регистрации в системе?

2. Запишите последовательность выполнения регистрации с терминала.

3. Какие сообщения будут выводиться на экран дисплея при попытке пользователя не указать пароль и имя (или код идентификации пользователя)?

4. Ваш $UIC = [11, 14]$, а пароль РАБОТА, запишите последовательность набираемых вами и выводимых системой на экран строк при регистрации в системе.

Пятое

1. Каковы команды MCR, применяемые для подготовки устройств пользователя к работе? Формат и назначение каждой команды.

2. Пользователь создает том с меткой LABOR на диске ДК2. Запишите последовательность необходимых для этого команд с клавиатуры дисплея.

Шестое

1. Объявите в Фортран-программе целые массивы из 35 элементов; из 150 вещественных чисел.

2. Объявите массив, состоящий из 10 строк и 5 столбцов.

3. Запишите в виде арифметического выражения правую часть алгебраической формулы

$$y = Ax + \sqrt{x^2 - B}.$$

4. Запишите в виде логических выражений алгебраические неравенства:

$$Ax^2 > B \cdot C + y; \quad z^3 < (A + X)^2 - C.$$

5. Запишите оператор перехода к метке 270.
6. Как организуются разветвления в программе с помощью оператора IF?
7. Приведите примеры выполнения логического оператора IF.
8. Запишите оператор цикла для повторения участка программы до метки 70, если начальное значение переменной цикла равно 10, конечное — 120, шаг изменения — 1.
9. Запишите оператор чтения из файла номер 3 с форматом чисел F5.2 в массив A, состоящий из 10 элементов.
10. Запишите оператор вывода на печать из указанного в предыдущем вопросе массива A.
11. Составьте Фортран-программу для чтения чисел из файла X.DAT в массив X, состоящий из 15 элементов. Формат чисел — F4.1.
12. Для предыдущего задания предусмотрите в программе печать (в цикле) каждого элемента массива X.
13. Усложните предыдущую программу вычислением суммы всех элементов массива X нарастающим итогом и печати значения этой суммы.

Седьмое

1. Каковы команды управления MCR? Поясните функции каждой из них.

Восьмое

1. Пользователю необходимо записать на диск файл Фортран-программу PROG1. Запишите соответствующую командную строку.

2. Как перевести EDI из режима ввода в режим редактирования? Какая подсказка EDI указывает на то, что режим редактирования установлен?

3. В каком режиме работы EDI воспринимает команды?

4. В тексте программы необходимо заменить число 5 на число 3 в строке

30 FORMAT('СУММА СОСТАВЛЯЕТ', I 5)

Как быстрее всего обнаружить исправляемую строку?

- Запишите команды EDI, которые обеспечат исправление ошибки.

5. Запишите команды EDI, с помощью которых можно вернуться к самому началу редактируемой программы и перед первой ее строкой вставить текст

С Лабораторная работа № 1

6. В процессе редактирования необходимо вставить строку текста (и вернуться в режим редактирования).

С ГРУППА 713 студент ПЕТРОВ

Запишите соответствующие команды EDI.

7. Требуется уничтожить в редактируемом тексте текущую строку и взамен нее вставить новую. Каковы соответствующие команды EDI?

8. Требуется в строке программы с меткой 25 заменить закрывающую скобку на открывающую. Запишите соответствующие команды.

9. Требуется выйти из режима редактирования и уничтожить при этом предыдущую версию редактируемого файла.

10. Требуется вставить в самое начало строки редактируемого текста пропущенные символы C и пять пробелов.

Девятое

1. Каковы основные функции экранного редактора текста TED?

2. Где хранится символьная информация, которую может обрабатывать TED?

3. В каком режиме TED осуществляет чтение текста из файла на магнитном диске в буфер?

4. Запишите команду, которая обеспечит чтение в буфер Фортран-программы с именем PROG4.FTN.

5. Как поместить в буфер часть предыдущей программы с 10-й по 25-ю строку?

6. Какова последовательность команд для выполнения следующих действий: запуска экранного редактора; чтения в буфер файла POS1.TXT с 20-й по 45-ю строку; вывода на экран дисплея этих строк; удаления 27-й строки; записи результирующего текста в файл на магнитном диске; завершение работы с TED.

7. Запишите последовательность команд, с помощью которых можно прочесть в буфер программу LABOR2.FTN, вывести ее на экран, перейти в экранный режим, в 5-й строке программы удалить символ «;», в 7-й строке удалить символ «!», удалить 8-ю строку, вернуться в командный режим, записать на диск результирующий текст.

8. Запишите последовательность команд, которая требуется для решения следующей задачи: запустить экранный редактор, прочесть строки с 17-й по 23-ю программы BACK2.FTN, вывести их на экран, вставить между 3-й и 4-й строками текст

25 FORMAT (2X, F4.3, 3X, I5)

записать исправленный текст на магнитный диск, очистить буфер и для контроля снова прочитать и вывести на экран исправленный участок текста.

Десятое

1. Каковы основные функции программы работы с файлами?

2. Назовите основные ключи программы PIP, их назначения и действия.

3. Вывести на экран дисплея директорий пользователя. Запишите соответствующие команды PIP.

4. Требуется распечатать на бумаге текст Фортран-программы пользователя RAB4.

5. Запишите команду, запрашивающую перечень всех имеющихся версий файла LAB7.CBL.

6. Требуется уничтожить все версии всех типов файла с именем LABOR, кроме самой последней. Какой командой PIP это достигается наиболее эффективно?

7. Требуется скопировать файл PROG4.CBL из директории пользователя на системном устройстве в директорий того же пользователя на диске DK3. Какие умолчания допускаются в этой команде?

8. Требуется переименовать имена файла CONTR.CBL всех версий на CONT2.CBL в директории пользователя.

9. Необходимо переписать все файлы на диск DK3 в директорий пользователя.

10. Распечатайте на АЦПУ все Фортран-программы, хранящиеся на магнитном диске DK3.

Одиннадцатое

1. Вызовите Фортран-транслятор для трансляции программы RETI2 с условием распечатки листинга трансляции на АЦПУ.

2. Выведите файл листинга трансляции программы KROT4 на экран терминала без распечатки его на АЦПУ.

Двенадцатое

1. Перечислите возможные варианты запросов построителю задач ТКВ на получение выводных файлов для вводного файла LABOR1.

2. Каким образом указать построителю задачи, что будут вводиться необязательные параметры?

3. Как обеспечить число одновременно открытых файлов в данной задаче, превышающее принятое по умолчанию число 4?

4. Напишите командные строки для ТКВ, осуществляющие запрос двенадцати при наличии шести действующих в программе файлов и максимальной длине

записи 1000 байт.

5. В каком порядке следует указывать в командной строке параметры UNITS, ACTFILE, MAXBUF?

6. Запишите командные строки, необходимые для запуска программ PROG4.TSK и LAB2.TSK.

7. Сохраняется ли файл образа задачи на диске пользователя после запуска задачи?

Тринадцатое

1. Запишите команду, позволяющую изменить имя файла на диске при работе с Бейсиком в диалоговом режиме.

2. Требуется удалить файл VESNA.BAS, хранящийся на диске. Как это выполнить?

3. Выведите на экран дисплея программу, находящуюся на диске (ZIMA.BAS).

4. Запишите команду подготовки оперативной памяти для программы PUT1.

5. Как записать программу PUT1 на диск, когда она полностью подготовлена в памяти?

6. Запишите, как загрузить программу PUT1 в память с диска в начале следующего сеанса работы.

7. Запишите команду, которая удалит с 25-й по 28-ю строки Бейсик-программы.

8. Поясните, как вставить пропущенную строку в программу.

9. Как заменить текст строки 40?

10. Запишите команду, которая в строке 70 заменит подстроку $Y > A + B$ подстрокой $Y < A + B$.

11. Запишите команду вывода на экран списка всех программ в директории на диске.

12. Составьте и выполните Бейсик-программу для ввода с дисплея значений x , y , A , B , вычисления значения и вывода на экран функции $z = Ax^2 + By - AB$.

13. Составьте и выполните программу, которая производит в цикле перемножение 25 значений количества продукции $K(I)$ на соответствующее значение расценки $R(I)$ и печатает найденные 25 значений.

14. Выполните на машине Бейсик-программы, используя задания 22, 23, 24 п. 3.4.

ЛИТЕРАТУРА

1. Материалы XXVII съезда КПСС. — М.: Политиздат, 1986. — 350 с.
2. Горбачев М. С. Коренной вопрос экономической политики партии: Доклад на совещании ЦК КПСС по вопросам ускорения научно-технического прогресса 11 июня 1985 г. — М.: Политиздат, 1985.
3. Балашов Е. П., Григорьев В. Л., Петров Г. А. Микро- и миниЭВМ. — Л.: Энергоиздат, 1984. — 376 с.
4. Бирюков В. В., Рыбаков А. В., Шакула Ю. П. Введение в систему программирования ОСРВ/Под ред. В. Н. Лебедева. — М.: Финансы и статистика, 1986. — 191 с.
5. Брусенцов Н. П. Мини-компьютеры. — М.: Наука, 1985. — 206 с.
6. Горбунов В. Л. и др. Основы построения микроЭВМ: Учеб. пособие. — М.: Высшая школа, 1984. — 144 с.
7. Гурова Л. И. Основы программирования: Учеб. пособие. — М.: Финансы и статистика, 1983. — 224 с.
8. Дьяков В. П. Справочник по расчетам на микрокалькуляторах. — М.: Наука, 1985. — 224 с.
9. Кетков Ю. Л. Программирование на Бейсике. — М.: Статистика, 1978. — 158 с.
10. Королев Л. И. Операционные системы. — М.: Знание, 1984. — 63 с.
11. СМ ЭВМ: Комплексование и применение/Под ред. Н. Л. Прохорова. — М.: Финансы и статистика, 1986. — 304 с.
12. Маслоков О. А. Вычислительная техника и программирование. — М.: Высшая школа, 1985. — 176 с.
13. МикроЭВМ СМ-1800: Архитектура, программирование, применение/А. В. Гиглавый, Н. Д. Кабанов, Н. Л. Прохоров и др. — М.: Финансы и статистика, 1984. — 136 с.
14. Наумов Б. Н., Песелев К. В. Малые ЭВМ в сфере управления. — М.: Знание, 1979. — 64 с.
15. Операционная система СМ ЭВМ РАФОС/Л. И. Валикова, Г. В. Вигдорчик, А. Ю. Воробьев, А. А. Лукин. — М.: Финансы и статистика, 1984. — 207 с.

16. *Петров А. В.* и др. Вычислительная техника в инженерных и экономических расчетах: Учебник. — М.: Высшая школа, 1984. — 320 с.

17. *Праченко В. Д.* и др. Дисковая операционная система коллективного пользования/В. Д. Праченко, А. Г. Самборский, М. В. Чумаков. — М.: Финансы и статистика, 1985. — 206 с.

18. *Сергованцев В. Т., Бледных В. В.* Вычислительная техника в инженерных и экономических расчетах. — М.: Финансы и статистика, 1983. — 208 с.

19. *Уокерли Дж.* Архитектура и программирование микроЭВМ: В 2 кн. — М.: Мир, 1984. — Кн. 1 — 486 с. Кн. 2 — 359 с.

20. *Фортран ЕС ЭВМ/З. С. Брич, Д. В. Капилевич, С. Ю. Котик, В. И. Цагельский.* — М.: Финансы и статистика, 1985. — 192 с.

21. *Цветков А. Н., Епанечников В. А.* Прикладные программы для микроЭВМ «Электроника БЗ-34» и «Электроника МК-56». — М.: Финансы и статистика, 1984. — с. 102—103.

22. *Шаров А. А.* Статистическая обработка экономических данных с применением ЭКВМ «Искра-124». — М.: МГУ, 1984. — 92 с.

23. *Экхауз Р., Моррис Л.* Мини-ЭВМ. Организация и программирование. — М.: Финансы и статистика, 1983. — 359 с.

ОГЛАВЛЕНИЕ

Введение	3
Глава 1. ЭЛЕКТРОННЫЕ КЛАВИШНЫЕ ВЫЧИСЛИТЕЛЬ- НЫЕ МАШИНЫ	7
1.1. Общие сведения	7
1.2. Основные устройства и принципы работы	12
1.3. Приемы вычислений на ЭКВМ	15
1.3.1. Вычисления в режиме «Счет»	15
1.3.2. Составление и выполнение программ на про- граммируемых ЭКВМ	19
1.4. Задания	37
Глава 2. МИНИ- И МИКРОЭВМ	48
2.1. Общие сведения	48
2.2. Основные устройства и их характеристика	50
2.3. Программное обеспечение	63
2.4. Задания	69
Глава 3. БЕЙСИК — ЯЗЫК ПРОГРАММИРОВАНИЯ ДЛЯ МИНИ- И МИКРОЭВМ	71
3.1. Общие сведения	71
3.2. Элементы и основные конструкции языка	71
3.3. Составление программ для решения экономических задач	81
3.4. Задания	85
Глава 4. РАБОТА В ОПЕРАЦИОННЫХ СИСТЕМАХ МИ- НИ- И МИКРОЭВМ	87
4.1. Операционная система ОСРВ и ее применение	87
4.1.1. Общие сведения о системе ОСРВ	87
4.1.2. Файлы, их спецификаторы и защита	88
4.1.3. Работа с терминалом	90
4.1.4. Работа в операционной системе ОСРВ с мно- гопользовательской защитой	94
4.1.5. Подготовка устройств пользователя к работе	97
4.1.6. Основные сведения о языке программирова- ния Фортран	98
4.1.7. Основные системные программы ОСРВ	109
4.1.8. Выполнение задачи	133
4.2. Работа с интерпретатором Бейсик	134
4.2.1. Работа на мини-ЭВМ СМ-4	134
4.2.2. Работа на микроЭВМ ДВК-2	138
4.3. Задания	142
Литература	148

Кротова Г. А. и др.

К79 Спецпрактикум на мини- и микроЭВМ: Учеб. пособие/Г. А. Кротова, И. В. Олевская, Г. Г. Осика. — М.: Финансы и статистика, 1987. 150 с.: ил.

Приводятся конкретные экономические задачи для программируемых ЭКВМ «Искра-124», «Электроника МК-54», мини-ЭВМ СМ-4, микроЭВМ СМ1800, «Искра 226», широко используемых в сельском хозяйстве. Большое внимание уделяется методическим вопросам решения задач, включая постановку, формулы и схемы расчета. Описывается исходная и нормативно-справочная информация.

Для студентов вузов, обучающихся по специальностям «Экономика сельскохозяйственного производства» и «Экономическая кибернетика».

К 2405000000—045
010(01)—87 112—87

ББК 40.7

Учебное пособие

**Галина Андреевна Кретьева,
Инна Викторовна Олевская,
Георгий Георгиевич Осика**

СПЕЦПРАКТИКУМ НА МИНИ- И МИКРОЭВМ

Зав. редакцией *И. Г. Дмитриева*
Редактор *Л. В. Речицкая*
Мл. редактор *Т. А. Студеникина*
Худож. редактор *С. Л. Витте*
Техн. редакторы *Г. А. Полякова, И. В. Завгородняя*
Корректоры *Г. А. Башарина, Н. П. Сперанская*
Обложка художника *Л. Е. Безрученкова*

ИБ № 2042

Сдано в набор 01.10.86. Подписано в печать 09.03.87. А 04347.
Формат 84×108¹/₃₂. Бум. кк.-жури. Гарнитура «Литературная».
Печать высокая. Усл. п. л. 7,98. Усл. кр.-отт. 8,19. Уч.-изд. л. 7,98.
Тираж 15000 экз. Заказ 2912. Цена 25 коп.

Издательство «Финансы и статистика», 101000 Москва,
ул. Чернышевского, 7.

Великолукская городская типография управления издательств,
полиграфии и книжной торговли Псковского облисполкома,
182100, г. Великие Луки, ул. Полиграфистов, 78/12