№ стр	Формат	Обозначение	Наименование	Кол. экз	N экз	Место-нахождение
1						
2	А4	353872.30024-24	Резидентный се-	1	-	
3			тевой монитор			
4			RomBios			
5						
6	А4	353872.30014-229	Дисковая опера-	1	-	
7			ционная система			
8			EKDOS 2.29			
9						
10	А4	353872.30026-10	Сетевая опера-	1	-	
11			ционная система			
12			NETOS			
13						
14	А4	353872.30007-10	Пакет тестовых	1	-	
15			программ			
16			DIAGNOSTICS	1	-	
17						

353872.00001-00

Изм	Л	N докум	Подп	Дата
	Нов	ен 139582		2889

Разраб.	Мирошников		28.04.89
Пров.	Кириллов		28.04.89
Н.контр	Филимонов		29.06.89
Утв.	Плаксин		28.06.89

Интеллектуальный терминал для систем реального времени (ИТСРВ) Е5104 Программное обеспечение

Лист 3

20 01

Инв N подл.	Подп. и дата	Взам инв N	Инв N дубл.	Подп и дата
84181/1	4у 280889			

	Наименование	Кол. экз	N экз	Место-нахождение
02	Утилиты к ОС	1	-	
	EKDOS			
	UTY Ø.2			
-11	Интерпретатор	1	-	
	языка Бейсик			
	JBASIC			
5-20	Ассемблер	1	-	
	ASM			
-561	Транслятор	1	-	
	языка Паскаль			
	MTPLUS			
-55	Редактор связей	1	-	
	для языка Паскаль			
	LINKMT			

дп	Дата	353872.00001-01 20 01	Лист 2

и дата	Взам инв N	Инв N дубл	Подп и дата
889			

N стр	Формат	Обозначение	Наименование	Кол экз	N экз	Место-нахождение
1						
2	А4	353872.30037-52	Интерпретатор	1	-	
3			языка Бейсик			
4			В80 5.21			
5						
6	А4	353872.30043-00	Транслятор	1	-	
7			языка Бейсик			
8			BASCOM			
9						
10	А4	353872.30038-34	Транслятор	1	-	
11			языка Фортран			
12			F80			
13						
14	А4	353872.30039-34	Редактор связей	1	-	
15			L80			
16						
17	А4	353872.30036-14	Отладчик	1	-	
18			программ			
19			SID			
20						
21						
22						

Изм	Л	N докум	Подп	Дата	353872.00001-01 20 01	Лист 3

Инв N подл.	Подп. и дата	Взам инв N	Инв N дубл	Подп и дата
84181/1	03.08.89			

N стр	Формат	Обозначение	Наименование	Код экз	N экз	Место-нахождение
1						
2	А4	353872.30023-24	Система управле-	1	-	
3			ния базами данных			
4			DBASE			
5						
6	А4	353872.30004-15	Пакет для таб-	1	-	
7			личных расчетов			
8			MULTIPLAN			
9						
10	А4	353872.30005-61	Экранный тек-	1	-	
11			стовый редактор			
12			SED			
13	А4	353872.30001-25	Редактор графи-	1	-	
14			ки и текстов			
15			GTR			
16						
17	А4	353872.30018-13	Игра "Змейка"	1	-	
18			SNAKE			
19						
20						
21						
22						

Изм	Л	N докум	Подп	Дата	353872.00001-01 20 01	Лист 4

Инв N подл.	Подп. и дата	Взам инв N	Инв N дубл	Подп и дата
84181/1	08 08 89			

N стр	Формат	Обозначение	Наименование	Кол экз	N экз	Место-нахождение
1						
2	A4	353872.30017-11	Демонстрацион-	1	-	
3			ная программа			
4			RDEM			
5						
6	A4	353872.30040-10	Драйвер печатаю-	1	-	
7			щего устройства			
8			CM6329			
9						
10	A4	353872.30045-10	Драйвер печатаю-	1	-	
11			щего устройства			
12			FX800			
13						
14						
15						
16						
17						
18						
19						
20						
21						
22						

Изм	Л	N докум	Подп	Дата	353872.00001-01 20 01	Лист 5

Инв N подл.	Подп. и дата	Взам инв N	Инв N дубл	Подп и дата
54181/1	06.08.89			

Лист регистрации изменений

Изм	Номера листов (страниц)				Всего листов (страниц) в докум.	No докум.	Входящий No сопроводительного докум. и дата	Подп.	Дата
	Измененных	Замененных	Новых	Аннулированных					
			6а		6	84139582			9813

Изм	Л	N докум	Подп	Дата

353872.00001-01 20 01

Лист 6

Инв N подл.	Подп. и дата	Взам инв N	Инв N дубл	Подп и дата
84181/1	080889			

"ЭХТА"
СКБ вычислительной техники Института кибернетики АН ЭССР

ИНТЕЛЛЕКТУАЛЬНЫЙ ТЕРМИНАЛ Е5104

РЕЗИДЕНТНЫЙ СЕТЕВОЙ МОНИТОР
ROMBIOS

РУКОВОДСТВО ПРОГРАММИСТА

353872.3Ø024-24

28 стр.

Таллинн 1983

СОДЕРЖАНИЕ

1. ОБЩИЕ СВЕДЕНИЯ О ПРОГРАММЕ

ЕКТА Tallinn *ЭКТА* Таллин КОМПОНЕНТ

RomBios	353872.3Ø024-24	Версия: 2.4 , 3.4

РЕЗИДЕНТНЫЙ СЕТЕВОЙ МОНИТОР

ТРЕБУЕМОЕ ОЗУ ! Программа: 16К байт ! Данные 1К байт

НОСИТЕЛЬ: ПЗУ
Рег. No носителя:

НАЗНАЧЕНИЕ И МЕТОД:
Модификация резидентного монитора JMON, расширенная функциями передачи данных в условиях локальной сети. Выполняет инициализацию необходимых аппаратных средств и проверку ПЗУ. Обеспечивает символьный ввод-вывод и выполнение функций графики, управление звуковым сигналом, хронированием, прерываниями, переключением режимов адресации памяти, а также выводом информации на печатающее устройство. Имеет директивы для осмотра, изменения и проверки содержания памяти, некоторые средства отладки програми в ОЗУ, функции для обращения к внешней памяти и функции загрузки и запуска програми и операционной системы.

ТЕХН. ХАР-КИ:
16 директив, 51 функция.Символьный вывод 23 команд АР2. Время обслуживания системного прерывания таймера около 2,5 мс.

ОГРАНИЧЕНИЯ:
Прерывание таймера имеет низкий приоритет, все остальные прерывания имеют высокий и равный приоритет.

ПРОЦЕССОР / ЭВМ: КР58ØИК8Ø / Е51Ø4

ВНЕШНИЕ УСТР-ВА:
Видеомонитор, линия связи

Ø6.12.1987	ПРОГРАММИСТ:	Палуоя Р. Аменберг А. Иыги А.	Юрвисте Э. Кауквер А.

ИСХ. ТЕКСТ: ! ЯЗЫКИ: Ассемблер

2. ДИРЕКТИВЫ МОНИТОРА

При нажатии на клавишу сброса (RESET) или при передаче управления на адрес FFC4H на экране появляются текст "RomBios <номер версии>" и символ "*". Теперь пользователь имеет доступ к услугам монитора через посредство следующих директив (в описаниях знак "+" обозначает нажатие второй клавиши при нажатой первой клавише):

Распечатка памяти - D XXXX/YYYY

На экран строками выводится содержимое памяти в диапазоне адресов XXXX...YYYY, в виде

<адрес>:<байт 1><байт 2> ... <байт 8>

Вывод прекращается при нажатии CTRL+S и может быть продолжен путём нажатия на любую другую клавишу. Для возврата из этой функции следует нажать последовательно CTRL+S и CTRL+C.

Пересылка массива - M XXXX/YYYY ZZZZ

Содержимое ячеек памяти в диапазоне адресов XXXX...YYYY перемещается в область с начальным адресом ZZZZ.

Сравнение массивов - C XXXX/YYYY ZZZZ

Содержимое ячеек памяти в диапазоне адресов XXXX...YYYY сравнивается с областью, начальный адрес которой ZZZZ. Обнаруженные различия выводятся в виде

<адрес>:<байт x><байт z>

Вывод прекращается при нажатии CTRL+S и продолжается после нажатия на любую другую клавишу. Для возврата из функции следует нажать последовательно CTRL+S и CTRL+C.

Поиск байта - K XXXX/YYYY BB

Содержимое ячеек памяти в диапазоне адресов XXXX...YYYY сравнивается со значением BB. Обнаруженные различия выводятся в виде

<адрес>:<байт x><BB>

Вывод прекращается при нажатии CTRL+S и продолжается после нажатия на любую другую клавишу. Для возврата из функции следует нажать последовательно CTRL+S и CTRL+C.

Заполнение памяти - F XXXX/YYYY BB

В ячейки памяти с адресами XXXX...YYYY записывается значение BB. Допускается задать и YYYY=XXXX.

Изменение байта памяти - S XXXX BB-CC BB-CC....RETURN

Отображается значение BB находящегося на адресе XXXX байта с последующим занесением нового значения CC на этот адрес (шаг диалога "BB-" выполняет монитор). Если значение CC равно коду пробела, то значение BB сохраняется, а на экран выводится значение байта. Для возврата из функции нужно нажать RETURN.

Пуск программы - G...

Программа может быть запущена с адреса, заданного содержанием счётчика команд PC (например, с точки разрыва), или с адреса XXXX. Выполнение программы продолжается до конца
или до адреса YYYY.

G RETURN - с адреса, заданного в PC, до конца
G XXXX RETURN - с адреса XXXX до конца
G XXXX, YYYY - с адреса XXXX до адреса YYYY (точки разрыва)
G,YYYY - с адреса, заданного в PC, до адреса YYYY

Автономный режим/Режим он-лайн - E/CTRL+C

При вводе "E" имеет место переход в квитирующий автономный режим, когда на экране отображаются коды соответствующие любым нажатиям клавиш, в том числе последовательности типа ESCAPE <символ>. Для возврата в командный он-лайн режим следует нажать CTRL+C.

Выбор страницы внешней памяти - PXX

Выбирается страница XX(ØØ...FDH) внешней памяти и открывается окно внешней памяти на адресах 4ØØØH...BFFFH.

Открытие/закрытие окна внешней памяти - PS/PR

PS - открытие окна
PR - закрытие окна

Пересылка массива с внешней памяти в ОЗУ - R XXXX/YYYY ZZZZ

Содержимое ячеек в диапазоне адресов XXXX...YYYY внешней памяти перемещается в область основной памяти с начальным адресом ZZZZ.

Пересылка массива с ОЗУ во внешнюю память - W XXXX/YYYY ZZZZ

Содержимое ячеек в диапазоне XXXX...YYYY основной памяти перемещается в область внешней памяти с начальным адресом ZZZZ.

Запуск программ во внешней памяти - A

Программа, находящаяся в начале выбранной внешней страницы памяти загружается в основную память, начиная с адреса 100H и запускается.

Начало программы должно быть следующее:

```
JMP  103H
DW   XXXX
DW   YYYY, где
```

YYYY - длина программы в байтах и
XXXX = DCBAH - ключ для автоматического запуска программы после включения питания.

Трансляция - B

Запускается программа пользователя во внутреннем ПЗУ; начало программы должно соответствовать требованиям директивы A.

Вызов операционной системы - T

На экран выводится текст

<D>isk, <T>ape, <N>et? ("Диск/Лента/Сеть?")

После нажатия на клавишу "D", "T" или "N" управление передается соответственно загрузчику дисковой, ленточной или сетевой операционной системы. При отсутствие какого-либо из этих загрузчиков соответствующий элемент текста не выводится и нажатие на соответствующую клавишу игнорируется.

3. ПРИМИТИВЫ МОНИТОРА

К примитивам монитора имеется доступ только в режимах адресации 1 и 2. В виде исключения возможен доступ во всех режимах адресации к примитивам MEMMD, PMEMMD и MEMMDRST. При обращении к примитивам переключения режима адресации следует иметь в виду следующие ограничения к адресу стека:
- при выборе режима Ø адрес стека должен быть не менее 8ØØØH
- при выборе режима 2 адрес стека должен быть вне диапазона 4ØØØH...BFFFH.

При символьном или графическом выводе во время обработки прерывания таймера следует пользовать указатель стека > 88ØØH и сблокировать другие прерывания во время выполнения этих функций. Это надо учитывать и при использовании функций переключения способов адресации памяти.

В приводимой ниже спецификации символы A,B,C,D,E,H,L обозначают соответствующие регистры ЭВМ, а AC, CY, P, S, Z - флажки (двоичные индикаторы) процессора.
AC = 1: десятичный перенос (из младшего полубайта)
CY = 1: переполнение
P = 1: чётный результат
S = 1: отрицательный результат
Z = 1: нулевой результат
"Параметр" означает входной параметр примитива.
"Вход" означает входную точку примитива.
"H" в конце числа - признак шестнадцатеричной системы.

MONITOR - входная точка монитора

Вход: FFC4H

MONVER - код версии монитора X.Y

Вход: FF6EH
Выход: D = номер версии X+3ØH
E = Номер подверсии Y+3ØH

*
TTI - ввод символа без вывода на экран
*
Вход: FFD3H
Выход: A = код введённого символа

*
TTIO - ввод символа с выводом на экран

Вход: FFD6H
Выход: A = код введённого символа

KEYBRD - сканирование и чтение состояния клавиатуры

Вход: FF71H
Выход: A=Ø, если клавиши не были нажаты
A=FFH, если какая-либо из клавиш была нажата

Функция предназначена для использования в случаях, когда прерывания таймера запрещена (отсутствует системное сканирование клавиатуры). Функцию нельзя вызывать чаще чем через 60 мс; это необходимо для правильного хронирования повторных нажатий. Для чтения используются функции TTSAT, TTI или TTO.

*

TTSTAT - чтение состояния клавиатуры

*

Вход: FFC7H
Выход: если какая-либо из клавиш была нажата, то Z=Ø, A=код символа; если нет, то Z=1

CONSTAT - чтение состояния клавиатуры

Вход: FF98H
Выход: A=ØØ, если ни одна из клавиши не была нажата
A=FFH, если какая-либо из клавиши нажата

PICTON - разрешение 10 минутного вывода изображения на экран

Вход: FF68H
Выход: CY=1, если вывод изображения был запрещен

Эта функция выполняется при нажатии любой клавиши, а также при символьной или графическом выводе информации на экран.

После истечения 1Ø-минутного контрольного времени, установленного для вывода изображения на экран, можно нажатием на любую клавишу восстановить вывод (для пользовательского уровня это нажатие маскируется).

TTO - вывод символа на экран

*

Вход: FFD9H
Параметр: A = код отображаемого символа

*

CRLF - перевод строки на экране

*

Вход: FFCAH

*

TTCON - вывод текста на экран

*

Вход: FFCDH
Параметр: BC = начальный адрес текста; признаком конца текста является байт со значением ØØH или символ "$"

TTCLF - вывод текста на экран с переводом строки

Вход: FFDOH
Параметр: BC = начальный адрес текста; признаком конца текста является байт со значением ØØH или символ "$"

OUTH - вывод 16-ричного значения байта на экран
*
Вход: FFDCH
Параметр: A = значение отображаемого числа

OUTH2 - вывод 16-ричного значения слова (2 байта) на экран
*
Вход: FFDFH
Параметр: BC = значение отображаемого числа

INHEX - ввод 16-ричного числа длиной в 1 байт
*
Вход: FFE2H
Выход: при безошибочном вводе CY = Ø, A = введённое число;
в противном случае CY = 1

INHX2 - ввод 16-ричного числа длиной в 2 байта
*
Ввод: FFE5H
Выход: при безошибочном вводе CY = Ø, BC = введённое число;
в противном случае CY = 1

NIBBLE - проверка 16-ричной цифры
*
Вход: FFE8H
Параметр: A = проверяемый код
Выход: если в A содержится код 16-ричной цифры, то CY = Ø, в противном случае CY = 1

DMEM - отображение содержания памяти

Вход: FFF1H
Параметры: BC = начальный адрес отображаемой области
HL = конечный адрес отображаемой области
Останов: CTRL+S с последующим нажатием CTRL+C
Продолжение: любая клавиша
Возврат: CTRL+S

MOVE - пересылка массива

Вход: FFF4H
Параметры: BC = начальный адрес массива
HL = конечный адрес массива
DE = новый начальный адрес массива

COMP - сравнение массивов, с отображением различий

Вход: FFF7H
Параметры: BC = начальный адрес первого массива
HL = конечный адрес первого массива
DE = начальный адрес второго массива

FILL - заполнение области памяти постоянной

Вход: FFFAH
Параметры: BC = начальный адрес области памяти
HL = конечный адрес области памяти
A = записываемое значение

CNST - поиск постоянной в области памяти, с отображением несовпадений

Вход: FFFDH
Параметры: BC = начальный адрес области поиска
HL = конечный адрес области поиска
A = значение постоянной
Останов: CTRL+S
Продолжение: любая клавиша
Возврат: CTRL+S с последующим CTRL+C

PRINTCHAR- вывод символа на печатающее устройство

Вход: FFEEH
Параметры: A = код выводимого символа
Системные средства содержат интерфейс для печатающего устройства СМ 6329.02 производства ГДР. В случае применения печатающего устройства другого типа следует в ячейку, адрес которого содержится в ячейке FFEFH занести команду безусловного перехода во входную точку соответствующего драйвера. Вывод на печатающее устройство инизиализуется после нажатия клавиш CTRL+SHIFT+P.

HARDCOPY - вывод содержания экрана на печатающее устройство

Вход: FF77H
Для выполнения этой функции должен быть загружен драйвер печатающего устройства, а команда безусловного перехода в драйвер должен быть занесен в ячейку, адрес которого считывают с адреса FF78H.
Примитив HARDCOPY выполняется при нажатии клавиш ERASE, ESC, P или CTRL+SHIFT+F1.
Ввиду того, что примитив печати выполняется во время обработки прерывания таймера, счетчики времени сблокированы до конца выполнения этой функции.

BLEEP - управление звуковым сигналом

Вход: FF86H
Параметры: A = (продолжительность сигнала, мс) /20
DE = 2000000/ (частота сигнала, Гц) или
DE = 0000H - отключение сигнала

BLEEPVOL - выбор уровня звукового сигнала

Вход: FFB0H
Параметры: A=00 для выбора низкого уровня
A= ненулевое для выбора высокого уровня
Ввод с клавиатуры квитируется звуковым сигналом низкого уровня.

HGR - переход в графический режим

Вход: FF8CH

HCOLOR - выбор цвета в графическом режиме

Вход: FF8FH
Параметр: E = индекс цвета
0 : чёрный
ненулевое значение : белый

HPLOT - вывод точки на экран

Вход: FF92H
Параметры: DE = значение координаты X
B = значение координаты Y
При символном формате 40 x 24 или 53 x 24
X=0...319 и Y=0..239
При формате экрана 64 x 20 символов
X=0...383 и Y=0...199.

HPLOTTO - вывод отрезка на экран

Отрезок направляется из последней выбранной точки в точку с координатами (X,Y)
Вход: FF95H
Параметры: DE = значение координаты X
B = значение координаты Y
При символном формате 40 x 24 или 53 x 24
X=0...319 и Y=0...239
При формате экрана 64 x 20 символов
X=0...383 и Y=0...199.

INTCNT - программируемый таймер

Истечение заданного промежутка времени индицируется флажком. Флажок сбрасывается примитивом чтении его значения.
Вход: FF9EH
Параметр: DE = (выдержка, мс)/20 - задание выдержки
DE = 0 - чтение и сброс флажка
Выход: A = FFH - промежуток истек
A = 00H - промежуток не истек

TIMCNT - программируемый секундомер

Вход: FFA1H
Параметр: E = 00H - сброс секундомера
E = FFH - чтение измеренного времени
Выход: DE = (измеренное время, мс)/20

INTRSV - обработка прерывания
Вход: FF89H
Параметры: A = номер уровня прерывания
DE = адрес программы обработки прерывания
DE = 0000 для освобождения данного уровня прерывания
Выход: DE = 0000 - если данный уровень прерывания не был занят
DE = адрес, предыдущей программы обработки прерывания

(В этом случае пользовательская программа должна после окончания обслуживания прерывания передать управление по адресу полученному из DE.)

Для пользователя выделены внешние прерывания уровней 6 и 7, и прерывание таймера уровня 8 (с периодом 20 мс), в версии монитора 3.4 кроме названных выделены внешние прерывания 0, 1 и 4.

При обработке прерывания 8 все другие прерывания разрешены. Режим адресации всегда 1 (режим, установленный перед обработкой прерывания восстанавливается после завершения обработки прерывания). При выборе других режимов адресации (кроме 1) необходимо маскировать все другие прерывания с помощью операции DI во время работы в другом режиме.

В распоряжении пользователя имеются 8 уровней стека. Следует иметь в виду, что те функции монитора, которые сохраняют содержимое регистров, обычно используют больше 8-и уровней, поэтому при вызове их внутри программы обработки прерывания необходимо использовать собственный указатель стека (не ниже 8800H) и маскировать другие прерывания на время работы функций. Если эти функции вызываются и в главной программе, то согласование работы их должен обеспечивать пользователь.

К этому примитиву необходимо обратиться при загрузке пользовательской программы и при завершении выполнения программы.
При возникновении прерывания управление передается соответствующей программе. Системные средства обеспечивают сохранение содержимого регистров и стека. Обращение к этому примитиву из программы обработки прерывания запрещено.

MEMMD - выбор временного режима адресации памяти

Вход: FFB9H
Параметры: A=0...3 - номер режима

PMEMMD - выбор постоянного режима адресации памяти

Вход: FFBCH
Параметры: A=Ø...3 - номер режима
При включении питания ЭВМ выбирается 1-ый режим адресации.

MEMMDRST - восстановление постоянного режима адресации памяти

Вход: FFBFH
Поскольку системные программы для перемотки экрана, а также для графического и символьного вывода используют эту функцию, то при вызове их в режиме 2 следует использовать функцию PMEMMD.

EXTPG - выбор или считывание страницы внешней памяти

Вход: FEBDH
Параметры: A = номер страницы (Ø...FDH)
A = FEH когда внешняя память не используется
A = FFH чтение номера страницы
Выход: A = номер страницы (Ø...FDH) или FFH (если внешняя память не используется)

MEMCOPY - пересылка массива с изменением режима адресации

Вход: FF83H
Параметры: HL= длина массива в байтах (кратная 128-и байтами)
BC= начальный адрес массива
DE= новый начальный адрес массива
A = s + 16d, где
s = режим адресации области массива
d = режим адресации новой области массива

LDROMPRG - загрузка и пуск резидентной программы из ПЗУ

Вход: FF9BH
Размещенная в ПЗУ программа (напр. интерпретатор Бейсика) загружается на адрес 1ØØH и управление передается этой программе. Выполнение этого примитива во время обработки прерывания запрещено. Начало программы должно соответствовать требованиям директивы "A".

* - эти функции сохраняют содержимое всех регистров.

4. СИСТЕМНЫЕ ПОЛЯ МОНИТОРА

В дальнейшем (A) обозначает содержимое ячейки с адресом A и следущей (2 байт).

LATSYM - начало кодовой таблицы эстонского алфавита

Адрес: ((FF60H))
Кодовая таблица без столбцов управляющих кодов (таблица начинается с кода пробела). (См. также 5.1 и 6.)

RUSSYM - начало кодовой таблицы русского алфавита

Адрес: ((FF62H))
Кодовая таблица только для 64-х букв. (См. также 5.1 и 6.) Если адрес 0000H, то с данной версии эта таблица отсутствует.

EXTSYM - начало кодовой таблицы для кодов 80H...BFH

Адрес: ((FFC2H))
Кодовая таблица для 64 символов. Если адрес 0000H, то в данной версии этой таблицы нет и пользователь может загрузить свою таблицу. (См. также 5.1 и 6.) Коды 80H (128)...BFH (191) можно ввести с клавиатуры; для этого при нажатой клавише CTRL набрать на цифровых клавишах код символа в десятичной системе и отпустить CTRL.

SQIKEY - код нажатой клавиши

Адрес: (FFA5H)
Ячейка содержит код клавиши согласно табл. 3, если клавиша нажата или 00H, если клавиши не были нажаты.

SQKBSW - флажок состояния клавиатуры

```
Адрес: (FFA5H)-1
Код: XXXXXXX1
           !--- клавиша нажата (сбрасывается при чтении кода
                функциями монитора)
```

Пока флаток активен, другие нажатия клавишей не регистрируются.

SQHCODER - код строки клавиши

```
Адрес: (FFA5H)+2
Код: 11XXXXXX
     !!   ---
     !!    !-- код строки нажатой клавиши (1...6)
     !!
     !!------- SHIFT
     !
     !-------- CTRL
```

SQHCODEC - код столбца клавиши

```
Адрес: (FFA5H)+3
Код: XXXXXXXX
       ----
         !
         !--- код столбца нажатой клавиши (1...16)
```

CONSW - код режима клавиатуры

```
Адрес: (FFA5H)-4
Код: XXXXXX11
          !!--- CAPS LOCK
          !
          !---- RUS (если Ø, то LAT)
```

CONCW - управляющий код клавиатуры

```
Адрес: (FFA5H)-3
Код: 1Ø1X1X11
     !!! ! !!--- запрос столбца курсора
     !!! ! !
     !!! ! !---- запрос строки курсора
     !!! !
     !!! !------ флаток ЛОС (если Ø, то дисковая ОС)
     !!!
     !!!-------- флаток операции CTRL+P (устанавливается, если
     !!          установлены флаток 6-ой функции ЛОС и флаток ЛОС)
     !!
     !!--------- флажок 6-ой функции ЛОС (при других функциях 1)
     !
     !---------- сигнал квитирования нажатия клавиши
```

Флатки ЛОС и 6-ой функции ЛОС (если 1) разрешают прерывание символьного вывода на экран с помощью одновременного нажатия клавиш CTRL и S.

Последующее нажатие CTRL и S разрешает продолжение символьного вывода. Если кроме названных флатков установлен флаток операции CTRL+P, то при символьном выводе на экран символы дублируются на печатном устройстве.

SWSTATE - состояние микропереключателей на клавиатуре

```
Адрес: (FFA5H)-5
Код: XXXXXXXX
     !       !--крайний правый переключатель
     !
     !---------крайний левый переключатель
```

Состояние переключателей считывается после включения питания и после нажатия на кнопку "RESET". Если переключатели не установлены, то код = ØØ.

FUNKEY - код функциональной клавиши, нажатой вместе с SHIFT и CTRL

Адрес: (FFA5H)-6

Ячейка содержит код функциональной клавиши нажатой в месте с SHIFT и CTRL. (Не передается пользователю). При нажатии F1+SHIFT+CTRL управление передается во время прерывания таймера функции HARDCOPY (см. гл. 3).

Пользователь может эту ячейку использовать например для запуска резидентной программы. (Записывать нуль в ячейку должен пользователь.)

NOWATY - координата Y графического курсора

Адрес: (FFB7H)

NOWATX - координата X графического курсора

Адрес: (FFB7H)+1

LINELEN - количество символов на одной строке экрана (40, 53 или 64)

Адрес: (FF66H)

USEREND - адрес начала системных полей

Адрес: (FF64H)

Если пользователь желает, что после выхода из программы в операционную систему эта программ, например драйвер принтера, сохранилось в ОЗУ, то необходимо переслать программу в конец пользовательской области (адрес последнего ситемного байта находится в ячейках 0006, 0007). В ячейки 0006, 0007 и USEREND следует записать начала программы после пересылки. Пользовательская программа должна начинаться с команды перехода, адресная часть которой формируется из содержимого ячеек 0006 и 0007.

Например:
Содержание памяти перед и после загрузки программы

```
0005 : JMP BLOS        0005 : JMP PROG
  ....                   ....
                       PROG : JMP BLOS
                                программа
BLOS : JMP XXXX        BLOS : JMP XXXX


USEREND : BLOS         USEREND : PROG
```

5. РЕЖИМЫ ОТОБРАЖЕНИЯ ИНФОРМАЦИИ

Дисплей (телевизор) может работать в буквенно-цифровом или в графическом режиме. На экране можно отобразить 384 точек в горизонтальном направлении (по оси x) и 240 точек в вертикальном направлении (по оси y).

5.1. Буквенно-цифровой режим

В этом режиме экран разделён на 24 строки длиной до 40 позиций каждая. Формат матрицы отображения одного символа 8 x 10 точек. Диапазоны координат символов: в вертикальном направлении 0...23, в горизонтальном направлении 0...39. Начало координат, т.е. точка (0,0) - в левом верхнем углу экрана.

В версии монитора с расширенным символьным выводом возможны еще два формата экрана:

24 строки длиной 53 позиций, или 20 строк длиной 64 позиций. Формат матрицы отображения одного символа 6 x 10 точек. Для такой версии монитора возможно открытие "окна" на экране (см. табл.2). Коды регистра ESC влияют только внутри "окна", если оно открыто. Надо иметь в виду, что адресация курсора при таком режиме символьного вывода выполняется относительно правой верхней точки, т.е. точка (0,0) соответствует позиции (v1,v1) в абсолютных координатах (когда "окна" закрыто). Однако при чтении позиции курсора всегда возвращаются абсолютные координаты.

При "холодном" пуске экран гасится и в точке (0,0) появляется мигающийся курсор латинского алфавита (чередуются цвета фона и символа). При переходе на русский алфавит мигание прекращается, курсор чёрный на белом фоне.

Вывод на экран возможен через монитор или при помощи примитивов операционной системы (LOS).

Если пользователь желает изменить изображения символов, отображаемых на экране, то следует изменить адреса этих таблиц (см. гл.4-ая, функции LATSYM, RUSSYM).

Для каждого символа выделены 9 байт, причем каждый байт определяет одну строку изображения (младший байт соответствует верхней строке). Каждый символ выводится с пустой строки после 9-ой строки. Для прописных букв используется поле 7X5 (две последней строки пусты), а для строчных поле 5X5 (две верхней и две нижней строки пусты). Исключением являются строчные буквы g,j,p,q и b,d,f,h,k,l,t к полю которых соответственно добавляются две нижней или две верхней строки. Внутри байта обычно пусты наименьший разряд и два наистарших разрядов.

Служебные коды дисплея описаны в таблицах 1 и 2, а коды символов в таблице 3.

Служебные коды дисплея Таблица 1

Обозначение	Код	Результат обработки кода
BELL	07H	Звуковой сигнал
BS	08H	Перемещение курсора на один шаг налево, в пределах текущей строки
TAB	09H	Перемещение курсора на следующую позицию табуляции (шаг табуляции равен 8 позициям); при необходимости - сдвиг изображения вверх
LF	0AH	Перемещение курсора на одну позицию вниз; при необходимости - сдвиг изображения вверх
VT	0BH	То же
FF	0CH	То же
RETURN	0DH	Возврат курсора в начало строки
ESC	1BH	Переключение кодового регистра
Прочие в диапазоне 00...1FH		Не обрабатываются, реакция отсутствует

Коды регистра ESC Таблица 2

Символ *) после ESC	Код (...H)	Результат обработки кода
H	1B 48	Возврат курсора в позицию (Ø,Ø), т.е. в левый верхний угол экрана
L	1B 4C	То же, с гашением экрана
=<vv>	1B 3D rr vv	Перемещение курсора в позицию (r,v),
Y<rr>	1B 59 rr vv	где rr = номер строки (r) +20H, vv = номер столбца (v) +20H
J	1B 4A	Стирание от курсора до конца экрана
K	1B 4B	Стирание от курсора до конца строки
D	1B 44	Перемещение курсора на одну позицию влево
C	1B 43	Перемещение курсора на одну позицию вправо
B	1B 42	Перемещение курсора на одну позицию вниз
A	1B 41	Перемещение курсора на одну позицию вверх
R	1B 52	Запрос позиции курсора; за этим кодом должны следовать 2 примитива чтения клавиатуры, причём результатами чтения будут соответственно r и v
Ø	1B 3Ø	Запрет квитанции ввода звуковым сигналом
1	1B 31	Разрешение квитирующего звукового
2	1B 32	Запрещение перемотки экрана
3	1B 33	Разрешение перемотки экрана
4	1B 34	Гашение курсора и запрещение последующего вывода
5	1B 35	Разрешение вывода курсора
:	1B 3A	сигнала Переход в режим построчной перемотки экрана вверх
;	1B 3B	Переход в режим постраничной перемотки экрана вверх
'	1B 27	Начало обращённого изображения
(	1B 28	Конец обращённого изображения
N	1B4E	Закрытие "окна" , возврат курсора в позицию (Ø,Ø)
M<mm> **)	1B4D mm	mm=формат символьного вывода ØØ-4Ø x 24 Ø1-53 x 24 Ø2-64 x 2Ø, закрытие "окна"
*<rr1><vv1> <rr2><vv2>	1B 23 rr1vv1 rr2vv2	Открытие "окна" на экране с позиции (r1,v1), где rr1=номер строки (r1)+20H, vv1=номер столбца (v1)+20H до позиций (r2,v2), где rr2=номер строки (r2)+20H, vv2=номер столбца (v2)+20H, при формате 53 x 24 и 64 x 2Ø должны v1 и (v2+1) делиться на 4

*) Допускается ввод левой квадратной скобки после ESC, например: ESC [H . Код скобки не обрабатывается.
**) Только для версий монитора, в котором имеется расширенный символьный вывод.

5.2. Графический режим

В графическом режиме изображение можно составить из любых точек экрана. Диапазоны координат применяемых точек

- по оси x: 0...319,
- по оси y: 0...239.

или

- по оси x: 0...383
- по оси y: 0...199, если в буквенно-цифровой режиме формат 64 x 20 символов.

Начало координатной сетки, т.е. точка (0,0) находится в левом верхнем углу экрана.

Переход в графический режим возможен посредством примитива HGR монитора или операционной системы. Допускается чередование графического и буквенно-цифрового режимов. При каждом выполнении приказа в графическом режиме стирается курсор буквенно-цифрового режима. При переходе на отображение буквенно-цифровых символов курсор восстанавливается.

Под графическим курсором следует понимать последнюю выбранную точку, в случае отрезка - конечную точку отрезка. Графический курсор в виде отдельной метки на экране не отображается.

При выполнении примитива HGR графический курсор перемещается в точку (0,0), цвет - белый. Теперь возможно выполнение следующих графических функций:

HCOLOR n - выбор цвета по трём младшим битам индекса n (0: чёрный, 1...7: белый)

HPLOT x,y - перемещение графического курсора в точку (x,y); точка отображается выбранным цветом

HPLOT TO x,y - построение отрезка от графического курсора, т.е. от последней выбранной точки, до точки (x,y); отрезок отображается выбранным цветом, начальная точка отрезка не обновляется

6. КОДЫ КЛАВИАТУРЫ

Клавиатура содержит клавиши трёх функциональных типов:
1) буквенно-цифровые клавиши,
2) служебные клавиши,
3) клавиши переключения регистров и режимов.

Коды, выдаваемые буквенно-цифровыми и служебными клавишами, приведены в таблице 3.

Для ввода кодов 20H...FFH следует при нажатой клавиши CTRL набрать соответствующий код на цифровых клавишах в десятичной системе и отпустить CTRL.

Функции переключающих клавиш следующие:

CTRL	- при нажатии одновременно с какой-либо из буквенно-цифровых клавиш вырабатывается служебный код согласно табл. 3
CTRL+ESC	- переход на адрес ØØØØ, откуда управление передаётся монитору, ленточной операционной системе или резидентному интерпретатору Бейсика
SHIFT	- переключение кодового регистра: у клавиш с двумя символами активными являются верхние символы, у буквенных клавиш - прописные буквы
CTRL+SHIFT+ESC	- управление передаётся монитору при срабатывании прерывания таймера
SHIFT+ERASE	- стирание информации с экрана (если не имеет места обмен с магнитофоном)
CAPS LOCK	- переключение прописных/строчных букв; выбранные буквы активны до следующего переключения
RUS/LAT	- переключение латинского/русского алфавитов; выбранный алфавит активен до следующего переключения
ERASE ESC P или CTRL+SHIFT+F1	- вывод изображения с экрана на печатающее устройство

Буквенно-цифровые и служебные коды Таблица 3

	Ø	1	2	3	4	5	6	7
Ø			пробел	Ø	Õ(@)	P	Ō(`)	p
1	F5		!	1	A	Q	a	q
2			"	2	B	R	b	r
3		F1		3	C	S	c	s
4	F2		$	4	D	T	d	t
5	F3		%	5	E	U	e	u
6	F6		&	6	F	V	f	v
7	ЗВ	F7	'	7	G	W	g	w
8	ВШ/←	F4	(	8	H	X	h	x
9	ГТ		)	9	I	Y	i	y
A	ПС/↓	F8	*	:	J	Z	j	z
B	ВТ/↑	AP2	+	;	K	[	k	õ({)
C	ПФ/→		,	<	L	Ä(\)	l	ä(:)
D	ВК		-	=	M	]	m	ö(})
E			.	>	N	Ü(^)	n	ü(~)
F			/	?	O	_	o	ЗБ

Символ или обозначение в скобках соответствуют стандартной кодовой таблице КОИ-8 (ГОСТ 19768-74) и изображены на стандартной клавиатуре, однако на экран эти коды выводят соответствующую букву эстонского алфавита.

	C	D	E	F
Ø	ю	п	Ю	П
1	а	я	А	Я
2	б	р	Б	Р
3	ц	с	Ц	С
4	д	т	Д	Т
5	е	у	Е	У
6	ф	ж	Ф	Ж
7	г	в	Г	В
8	х	ь	Х	Ь
9	и	ы	И	Ы
A	й	з	Й	З
B	к	ш	К	Ш
C	л	э	Л	Э
D	м	щ	М	Щ
E	н	ч	Н	Ч
F	о	ь*)	О	пробел**)

*) Код DF вырабатывается комбинацией клавиш CTRL+<223>.
**) Код FF вырабатывается комбинацией клавиш RUS+DEL, на экран этим кодом выводится пробел.

7. ФУНКЦИИ ПЕРЕДАЧИ ДАННЫХ

SETPARM - установка параметров локальной сети
Вход: FF26H
Параметры: DE = адрес таблицы параметров в ПЗУ

Выполняются считывание таблицы параметров из ПЗУ, дополнение таблицы значениями, установленными на микропереключателях блока E51Ø1, формирование в ОЗУ таблицы параметров, необходимых для работы локальной сети.

Таблица параметров:

ØØ Адрес станции
Ø1 Максимальное число станций в сети
Ø2 Наличие спецпроцессора передачи данных
Ø3 Постоянная скорость передачи, для таймера
Ø4 Адрес списка для функции WRITE
Ø6 Адрес списка буферов, свободных для передачи
Ø8 Адрес списка передачи
ØA Адрес списка для функции READ
ØC Адрес списка буферов, свободных для приема
ØE Адрес списка приема
1Ø Адрес массива буферов
12 Число буферов для передачи
14 Об'ем буфера
16 Длина идентификатора сообщения
18 Максимальная длина сообщения
1A Максимальная длина блока
1B Число повторений блоков
1C Число повторений кадров данных
1D Постоянная контрольного таймера, в единицах 2Øмс
1E Постоянная счетчика обнаружения лишнего диспетчера
1F Число пользовательских контрольных таймеров
2Ø Адрес массива пользовательских таймеров
22 Код синхронизации

Микропереключатели для установки параметров сети:

7	6	5	4	3	2	1	Ø
P	M	M	A	A	A	A	A

P = 1: наличие спецпроцессора передачи данных
= Ø: отсутствие спецпроцессора

MM - максимальное число станций в сети:

MM = ØØ: 3 станции
MM = Ø1: 7 станций
MM = 1Ø: 15 станций
MM = 11: 31 станция

AAAAA - адрес станции

GETPARM - чтение адреса таблицы параметров локальной сети
Вход: FF29H
Выход: DE = адрес таблицы параметров в ОЗУ

JANETSTR - запуск локальной сети
Вход: FF2F

Устанавливаются начальные значения полей данных, необходимых для работы сети, устанавливается исходное состояние аппаратуры, запускается работа локальной сети.

ONGROUP - установка принадлежности станции к группам ЭВМ
Вход: FF2F
Параметры: E = код принадлежности к группам

X	6	5	4	3	2	1	X

- номер разряда соответствует номеру группы
- разряд = 1: принадлежность к данной группе

Выход: E = код принадлежность к группам

OFFGROUP - удаление станции из состава групп ЭВМ
Вход: FF32H
Параметры: E = код принадлежности к группам; для удаления из состава какой-либо группы соответствующий разряд должен иметь значение 0

Выход: E = код принадлежности к группам

MWRITE - передача сообщения
Вход: FF35H
Параметры: DE =адрес управляющего блока

Сообщение из ОЗУ пользователя переписывается в буфера для передачи.

Управляющий блок (входные значения):

ØØ Адрес (индивидуальный, групповой или глобальный) целевой станции
Ø1 Начальный адрес сообщения в ОЗУ
Ø3 Длина сообщения. Если длина = ØØØØ, то передается только идентификатор сообщения
Ø5 Адрес байта состояния
Ø6 Идентификатор сообщения; длина этого поля задается параметрами сети

Байт состояния:

7	6	5	4	3	2	1	Ø
F	E	S	T	R	X	X	X

F = 1: передача сообщения завершена
E = 1: синтаксическая ошибка
S = 1: отсутствие достаточно числа буферов передачи
T = 1: Ошибка при передаче
R = 1: переполнение приемных буферов

HREAD - прием идентификатора сообщения
Вход: FF38H
Параметры: DE = адрес управляющего блока

Выполняется считывание идентификатора принятого сообщения. Если сообщение состоит только из идентификатора, то освобождаются приемные буфера.

Управляющий блок:
ØØ Вход: индивидуальный адрес станции-источника или FFH; при FFH выполняется поиск сообщения, поступившего от любой станции
Выход: индивидуальный адрес станции-источника
Ø1 (Не используется)
Ø2 (Не используется)
Ø3 Выход: ØØØØH
Ø5 Вход (2 байт): адрес байта состояния
Выход (n байт): идентификатор сообщения; длина поля задается параметрами сети

Байт состояния:

7	6	5	4	3	2	1	Ø
F	E	M	X	X	X	X	X

F = 1: завершение приема идентификатора сообщения
E = 1: синтаксическая ошибка
M = 1: отсутствие сообщения

MREAD - прием сообщения
Вход: FF3BH
Параметры: DE = адрес управляющего блока

После перекачки сообщения в ОЗУ пользователя освобождаются приемные буфера.

Управляющий блок:

ØØ Вход: индивидуальный адрес станции-источника или FFH1, при FFH выполняется поиск сообщения, поступившего от любой станции
Выход: индивидуальный адрес станции-источника
Ø1 Вход: целевой адрес ОЗУ для размещения сообщения
Выход: адрес поля ОЗУ, следующего за размещенным сообщением
Ø3 Выход: длина принятого сообщения
Ø5 Вход: адрес байта состояния
Ø7 Выход (n байт): идентификатор сообщения; длина поля задается параметрами сети

Байт состояния:

7	6	5	4	3	2	1	Ø
F	E	M	X	X	X	X	X

F = 1: завершение приема сообщения
E = 1: синтаксическая ошибка
M = 1: отсутствие сообщения

MIGNORE - игнорирование поступившего сообщения
Вход: FF3EH
Параметры: DE = адрес управляющего блока

Освобождаются буфера принятого сообщения

Управляющий блок:
ØØ Вход: индивидуальный адрес станции-источника или FFH1; при FFH выполняется поиск сообщения, переданного от любой станции
Выход: индивидуальный адрес станции-источника
Ø1 (Не используется)
Ø3 Выход: ØØØØH
Ø5 Вход: адрес байта состояния
Ø7 Выход (n байт): идентификатор сообщения; длина поля задана параметрами сети

Баит состояния:

7	6	5	4	3	2	1	Ø
F	E	M	X	X	X	X	X

F - завершение игнорирования сообщения
E - синтаксическая ошибка
M - отсутствие сообщения

STMOUT - запуск таймера
Вход: FF41H
Параметры: A = номер таймера (нумерация начинается с нуля)
DE = постоянная таймера, единица счета = 2Øмс
Выход: CY=Ø в случае ввода номера несуществующего таймера

Запускается контрольный таймер пользователя с единицей счета 2Øмс.

RTMOUT - считывание значения таймера
Вход: FF44H
Параметры: A = номер таймера (нумерация начинается с нуля)
Выход: CY = Ø в случае ввода номера несуществующего таймера
Z = 1 при истечении контрольного времени
DE = значение таймера (показание текущего счета)

СКБ Вычислительной техники Института кибернетики АН ЭССР
"ЭКТА"

ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ИНТЕЛЛЕКТУАЛЬНОГО ТЕРМИНАЛА Е5104

ОПЕРАЦИОННАЯ СИСТЕМА

EKDOS 2.29

РУКОВОДСТВО ОПЕРАТОРА

353872.3Ø014-229

7 стр.

Таллинн 1988

СОДЕРЖАНИЕ

1. ОБЩИЕ СВЕДЕНИЯ О ПРОГРАММЕ

ЕКТА Tallinn *ЭКТА* Таллин КОМПОНЕНТ / КОМПЛЕКС ПО

EKDOS	353872.3Ø014-229	Версия: 2.29

ДИСКОВАЯ ОПЕРАЦИОННАЯ СИСТЕМА

ОБЪЕМ 7К байт

ТРЕБУЕМОЕ ОЗУ | Программа 6,5К байт | Данные Ø,5К байт

НОСИТЕЛЬ: ГМД
РЕГ. NO НОСИТЕЛЯ:

НАЗНАЧЕНИЕ И МЕТОД:

Предназначена для облегчения хранения информации на дисках и для создания множества дополнительных услуг. Определяет низкоуровневый интерфейс EKDOS с Е51Ø4. Обеспечивает взаимодействие оператора с системой и управление ресурсами системы, прежде всего дисковыми файлами. Создает возможности, обеспечивающие создание, удаление и изменение дисковых файлов, посимвольный обмен с "медленными" устройствами (клавиатура, дисплей и тд.), а также ряд других операций.

ССЫЛКИ:
Уэйт М.,Ангермейер Дж. Операционная система СР/М.
Пер. с англ. М.:Радио и связь, 1986. 312 с.

ТЕХН. ХАР-КИ:
EKDOS 2.29 совместима с версией 2.2 ОС СР/М

ОГРАНИЧЕНИЯ:
Резидентный монитор версии от 3.4 , в том числе загрузчик BOOTSTRAP от 4.1

ПРОЦЕССОР / ЭВМ: КР58ØИК8Ø / Е51Ø4

ВНЕШНИЕ УСТР-ВА:
НГМД, клавиатура, видеомонитор (телевизор)

НЕОБХ. ПРОГРАММЫ И ФАЙЛЫ:
комплект утилитов UTY

Ø5.Ø1.1988 | АРХ. МЛ: 3.ØØ | ПРОГРАММИСТ: Кауквер А.

2. ЗАГРУЗКА ОС EKDOS

Для загрузки ОС необходимо иметь соответствующую аппаратную конфигурацию E51Ø1 или E51Ø2 с резидентным монитором версии не менее 2.4 или 3.4, диск с версией 2.29 EKDOS.

Последовательность операций:

1) включить электропитание монитора/дисплея;
2) включить электропитание ЭВМ (на экране дисплея отображаются текст RomBios и номер версии монитора);
3) включить электропитание дисководов (если дисковод(ы) встроен(ы) в корпус ЭВМ, они включаются вместе с ЭВМ);
4) вставить в дисковод диск с EKDOS;
5) нажать на клавишу T. На экране появится сообщение вида:

```
System from <D>isk, <N>et?          ("Загрузить с диска/из сети")
```

6) для загрузки операционной системы с дискеты надо нажать на клавишу D.
7) на экране появится начальное сообщение вида:

```
BOOTSTRAP V4.1                      ("Загрузчик V4.1")

SYSTEM DISK TYPE (D/S/8) ?          ("Тип системного диска")
```

В первой строке сообщается версия загрузчика ОС и в третьей строке вопрос о типе диска.

8) При использовании дисков

- с двойной плотностью записи (DD) - нажать клавишу D
- с нормальной плотностью записи (SD) - нажать клавишу S
- размером 2Ø3 мм (8") - нажать клавишу 8

Процесс загрузки занимает несколько секунд, в зависимости от типа дисковода.

После этого на экране появится:

```
52K EKDOS 2.29/R

DRIVE ASSIGNMENTS:                  ("Распределение дисков")

<A> - X" (YYYK)
<B> - X" (YYYK)
<C> - RAM DISK (YYYK), где
A>
X   - тип дисковода (5 или 8 дюймов);
YYY - обьем диска в килобайтах;
```

Если ЭВМ инсталлирована с одним дисководом или на плате процессора нет дополнительного ОЗУ для логического диска, то соответствующие строки не выводятся.

Появление на экране дисплея сообщения-подсказки A> говорит о работоспособности ОС (готовности к функционированию).

При отказе в процессе загрузки на экране дисплея появится текст в виде:

```
XX ERROR
NON-SYSTEM DISK OR DISK ERROR
REPLACE AND STRIKE ANY KEY WHEN READY ,
```

где XX обозначает код ошибки (в 16-ым виде), где разряды шестнадцатиразрядного кода имеют следующие значения:

```
D7      - дисковод не готов          (DRIVE NOT READY)
D6      - всегда Ø
D5      - всегда Ø
D4      - сектор не найден           (SECTOR NOT FOUND)
D3      - ошибка контрольной суммы   (CRC ERROR)
D2...DØ - всегда Ø.
```

После нажатия любой клавиши повторяется процесс загрузки начиная с пункта 5.

3. КОМПОНЕНТЫ СИСТЕМЫ

Для удобства применения на компьютере произвольной конфигурации операционная система (ОС) EKDOS разделена на три части: Базовую Систему Ввода/Вывода (БСВВ,BIOS), Базовую Дисковую Операционную Систему (БДОС,BDOS) и Процессор Обработки Приказов (ПП,CCP).

БСВВ - модуль, определяющий низкоуровневый интерфейс EKDOS с конкретной ЭВМ. В начале БСВВ находится вектор переходов, обеспечивающий связь БСВВ и БДОС.

БДОС - аппаратно независимый модуль, обеспечивающий вместе с БСВВ взаимодействие оператора с системой и управление ресурсами системы. БДОС - логическое ядро EKDOS, которое с помощью механизма вызова системных функций создает стандартную среду для транзитных программ.

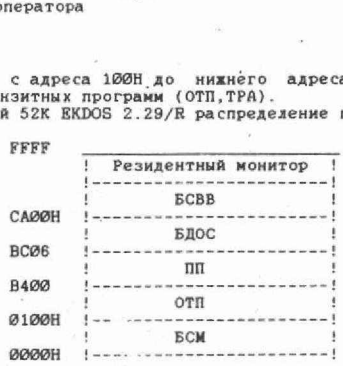
ПП - аппаратно независимый модуль, обеспечивающий выполнение резидентных (встроенных) команд (см. пп.9.) и запуск транзитных команд (программ).

Транзитные команды (утилиты) хранятся в файлах, расположенных на дисковых носителях. Для выполнения транзитной команды следует ввести имя файла и нажать на клавишу RETURN (см. руководство по утилитам UTY).

4. ОРГАНИЗАЦИЯ ПАМЯТИ

ПРК, БДОС и БСВВ занимают верхнюю часть памяти. Область памяти с адреса ØH по ØFFH называется базовой страницей памяти (БСП). Она включает несколько сегментов кодов и данных, обеспечивающих вход в БДОС и содержащих некоторые системные параметры.

Область с адреса 100H до нижнего адреса БДОС называется областью транзитных программ (ОТП,TPA).
В версий 52K EKDOS 2.29/R распределение памяти следующее:



5 РЕДАКТИРОВАНИЕ И УПРАВЛЕНИЕ ВЫВОДОМ

RETURN	Конец ввода строки
CTRL-X	Стереть строку
CTRL-Z	Конец ввода с консоли (для PIP и ED)
CTRL-P	Копировать весь дальнейший ввод с консоли на печать
CTRL-S	Временная приостановка вывода на консоль
CTRL-C	Горячий старт (перезагрузка EKDOS)

6. СПИСОК ДОПУСТИМЫХ ФИЗИЧЕСКИХ УСТРОЙСТВ

CON:	Системная консоль
LST:	Печатающее устройство
TTY:	Телетайп (медленная консоль)
CRT:	Дисплей (быстрая консоль)
UC1:	Консоль, заданная пользователем
UR1:	Ввод 1, определенный пользователем
UR2:	Ввод 2, определенный пользователем
UL1:	Устройство 1 вывода текста, определенное пользователем

7. ОБОЗНАЧЕНИЕ ФАЙЛОВ

Составное имя файлов состоит из двух частей:
filename.ext , где
filename (до 8 символов) - имя, а ext (до 3 символов) - расширение имени файла.

Символы, запрещенные в именах файлов < > . , ; : = ? * [ ]

В составном имени файла (файлов) допускаются следующие метасимволы:
* - используется вместо группы символов (например *.COM, M*.ASM),
? - используется вместо одного конкретного символа (например FILE????.???, A??.*).

8. ТИПЫ ФАЙЛОВ

ASM	Исходный файл для ассемблера ASM
PRN	Файл для печати листинга
HEX	Файл шестнадцатеричных машинных кодов
BAS	Исходный файл для компилятора BASIC
COM	Командный файл CCP
REL	Перемещаемый модуль
ERL	Перемещаемый модуль
PAS	Исходный файл для транслятора PASCAL/MT+
BAK	Вспомогательный файл для текстовых редакторов
FOR	Исходный файл для компилятора F(ORTRAN)80
SYM	Символический файл для SID
$$$	Временный файл

9. РЕЗИДЕНТНЫЕ КОМАНДЫ

6.1. ERA [X:] AFN	Стирание файлов с диска
6.2. DIR [X:] [AFN]	Вывод каталога диска или ряда файлов
6.3. REN [X:] UFN1=UFN2	Имя файла UFN2 заменяется на имя UFN1
6.4. SAVE N [X:] UFN	Запись из TPA N страниц по 256 байт начиная с адреса 100H в файл с именем UFN
6.5. TYPE [X:] UFN	Печать файла UFN в кодах ASCII
6.6. USER N	Установка номера пользователя

10. СООБЩЕНИЯ ОБ ОШИБКАХ EKDOS

BAD SECTOR	Ошибка при чтении/записи. Нажать на CTRL и C одновременно для акцептирования или RETURN для повторения,
SELECT	Неверная адресация дисковода (буква вне диапазона A-D). Нажать на CTRL и C одновременно.
READ ONLY	Данная дискета командой STAT назначена "только для чтения". Нажать CTRL и C одновременно.
DISK READ ERROR EXX Abort,Retry,Ignore?	Ошибка при чтении/записи, Нажать на R - для повторения, I - для игнорирования, A - для акцептирования ошибки.

СКБ Вычислительной техники Института кибернетики АН ЭССР
"ЭКТА"

ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ИНТЕЛЛЕКТУАЛЬНОГО ТЕРМИНАЛА Е5104

СЕТЕВАЯ ОПЕРАЦИОННАЯ СИСТЕМА
NETOS

РУКОВОДСТВО ОПЕРАТОРА

353872.3ØØ26-1Ø
18 стр.

Таллинн 1988

СОДЕРЖАНИЕ

1. ОБЩИЕ СВЕДЕНИЯ О ПРОГРАММЕ

ЕКТА Tallinn *ЭКТА* Таллинн КОМПОНЕНТ / КОМПЛЕКС ПО

NETOS	353872.30026-10	Версия: 2.0/1.0

СЕТЕВАЯ ОПЕРАЦИОННАЯ СИСТЕМА

ОБ'ЕМ 7/11 К байт | БАЗОВАЯ ОПЕРАЦ. СИСТЕМА: EKDOS

ТРЕБУЕМОЕ ОЗУ: | Программа 6,5/10,5К байт | Данные 1,2/3К байт

НОСИТЕЛЬ: ГМД
РЕГ. No НОСИТЕЛЯ:

НАЗНАЧЕНИЕ И МЕТОД:
Система осуществлена на базе операционной системы EKDOS, совместимой с CP(M), и средств локальной сети JANET.
Версия 2.0 предназначена для станции без НГМД. JANET должен находиться в ПЗУ. Об'емы памяти для этой версии указаны в числителе.
Версия 1.0 предназначена для станции с НГМД. JANET не обязательно в ПЗУ. Об'емы памяти для этой версии указаны в знаменателе.

ССЫЛКИ: Операционная система EKDOS. 353872.30014-225

ТЕХН. ХАР-КИ: 20 директив
18 функций

ОГРАНИЧЕНИЯ:
Максимальное число станций в сети - 31, по крайней мере одна из станций должна иметь НГМД.

ПРОЦЕССОР / ЭВМ: КР580ИК80 / Е5104

ВНЕШНИЕ УСТР-ВА: Видеомонитор, линия связи
Версия 1.0 : НГМД

НЕОБХ. ПРОГРАММЫ И ФАЙЛЫ:

NETR.COM - загрузочный модуль для станции без НГМД
NETD.COM - загрузочный модуль для станции с НГМД

АРХ. МЛ:	ПРОГРАММИСТ: Аменберг А.

2. ПУСК СИСТЕМЫ

2.1. Пуск системы в станции с НГМД

1. Вызвать программу NETD. Перед этим должна быть загружена и запущена базовая операционная система EKDOS.
Для вызова NETD 3 возможности:

```
>NETD                - таблица конфигурации (КТ) загружается из
                       сети
>NETD CT имя_файла - КТ загружается с диска
>NETD CT             - сеть запускается с пустой (заполненной нуля-
                       ми) КТ
```

После загрузки и активизации NETD выводится текст:

```
NETOS 1.Ø    (в случае задания параметров сети при помощи пере-
             ключателей на блоке ИПО) или
NETOS 1.Ø    (в случае интерактивного ввода параметров сети; сис-
N=           тема требует параметры, первый из которых N).
```

2. Ввести параметры сети

```
N=   число станций в сети (Ø1...1FH)
S=   индивидуальный адрес данной станции (Ø1...1FH), S=<N
P=   наличие процессора передачи данных (ППД)
          =ØØ    - ППД отсутствует
          >ØØ    - наличие ППД
```

3. После ввода параметров сети экран стирается, затем выводится заглавие сетевой операционной системы и выполняется тест сети. После этого на экране появляется одно из следующих сообщений:

```
NET o.k.     ("Сеть в норме") или
NET error    ("Отказ в сети")
```

4. После теста выполняется инициализация сети. Последовательность действий зависит от варианта вызова NETD.
(а) Вызов без параметров. Запрос таблицы конфигурации (КТ) передается в первую очередь в станции с минимальным адресом (но не себе); при отсутствии ответа запрос передается по следующему адресу и т.д. Если считывание таблицы из сети не дает положительных результатов, выводится сообщение:

```
CT error     ("Отказ КТ")
```

Это сообщение выводится, например, при отсутствии других станций в сети. Процесс поиска таблицы из сети можно прерывать нажатием на клавишу ESC.
(б) Вызов NETD CT E. Таблица заполняется автоматически нулями.

(в) Вызов с загрузкой КТ с диска. КТ считывается из файла, имя которого задано в строке вызова, и передается всем станциям. В случае каких-либо отказов при передаче таблицы на экран выводится сообщение об ошибке.

В случае успешной инициализации на экране появляется сообщение

CT o.k. ("КТ в норме")

5. Пуск командного процессора, выполняется автоматически после инициализации. На экран выводится приглашение в виде

Net ns>

где
n = индивидуальный адрес станции
s = имя дисковода (по умолчанию то же, что и при пуске NETD.COM под EKDOS.

2.2. Пуск системы в станции без НГМД

1. В ответ приглашению, выдаваемому резидентным монитором ввести команду "N". Перед этим сеть должна быть подготовлена, т.е.
- данная станция должна быть соединена со станцией, снабженной НГМД,
- в дисковод A станции с НГМД должна быть вставлена дискета с файлом NETR.COM и станция должна быть в активном состоянии.

2. Ввести параметры сети (если параметры не считываются с переключателей ИПО):

N = число станций в сети (Ø1...1FH)
S = индивидуальный адрес данной станции (Ø1...1FH)

3. Тест средств передачи данных. В зависимости от результатов теста выводится сообщение "NET o.k." или "NET error" (см. 2.1, п.3).

4. Инициализация станции. Запрос NETOS передается в первую очередь станции с минимальным адресом (но не себе). Если целевая станция не имеет дисководов или на дисководе A не нашелся файл NETR.COM, запрос повторяется по следующему адресу и т.д. Станция -источник ждет ответа от каждой опрашиваемой станции в течение 2Ø с.

В случае начала приема операционной системы из сети на экран выводится:
- адрес станции (в 16-ричном представлении), из которых передается операционная система,
- счетчик принятых 128-байтовых блоков (в 16-ричном представлении). Вместе с операционной системой передается и таблица конфигурации.

5. Пуск командного процессора. Выводится приглашение в виде

Net nE>
где n = индивидуальный адрес станции.

3. КОМАНДНЫЙ ПРОЦЕССОР

3.1. Индикация готовности

Выводимое на экран при пуске командного процессора приглашение имеет вид

```
NET ns>
```

n = индивидуальный адрес станции (в десятичном представлении, 1...31)
s = имя дисковода по умолчанию:
- в станции с дисководом допускаются значения A,...,P; обращение к этим устройствам выполняется как в EKDOS
- в станции без дисководов допускаются значения E,...,P; обращение к этим устройствам всегда выполняется через сеть

3.2. Резидентные функции

В описаниях функций приняты следующие обозначения:

к - адрес станции:
- индивидуальный = 1,...,31
- групповой Gn, номер группы n = 1,...,6
- глобальный * (для всех станций)

f - полное имя файла
[] - факультативные параметры; если заключенные в квадратные скобки параметры не вводятся, применяются значения по умолчанию
Примечание. Адресные параметры следует ввести в 16-ричной форме.

SGO к a - передать пусковой адрес
к - адрес целевой станции
a - пусковой адрес

Под действием этой команды в целевой станции управление передается на адрес "a" (см. функцию 3).

SFILE к f [G][a] - передать файл
к - адрес целевой станции
f - имя файла
G - пуск
a - целевой адрес файла; если задан пуск (G), следует задать и пусковой адрес; по умолчанию файл передается на адрес Ø1ØØH и не пускается

Под действием этой команды целевая станция переводится в режим приема, а станция-источник начинает передачу (см. функцию 13).

SMEM к р [G][a] - передача сообщения (содержимого памяти)
к - адрес целевой станции
р - число 256-байтовых блоков
G - пуск
а - целевой адрес файла; если задан пуск (G), следует задать и пусковой адрес; по умолчанию файл передается на адрес Ø1ØØH и не пускается

Целевая станция переводится в режим приема, станция-источник начинает передачу (см. функцию 12).

S к t - передача сообщения на экран
к - адрес целевой станции
t - текстовая строка (до 35 символов)

В целевой станции текст отображается на первой строке экрана (см. функцию 1).

R - отображать сообщение, поступившее последним
Номер перед текстом - индивидуальный адрес станции-источника, передавшего сообщение. Если приема сообщений не было, выводится текст

Buffer empty ("Буфер пуст")

SHOW - отображать таблицу конфигурации
Пример таблицы:

```
Device table:                    ("Таблица устройств")
E := A 12                        (Дисковод E в составе  станции 12)
F := no                          (Устройство F  отсутствует)
Station G=123456 Printer Status
                                 (Станция/Группы/Печат.устр./Блокир.)
   1.     +----+    8      -
   2.     +-----    -      +
   ...
```

В этом примере станция 1 входит в состав групп 1 и 6; печатающие устройство станции 1 входит физически в состав станции 8, прием в станции 1 не блокирован. Станция 2 входит в группу 2, печатающего устройства не имеет, прием сообщений блокирован (сообщения игнорируются).

Если станция в которой вызвали данную функцию, имеет печатающее устройство и это устройство занято, то выдается сообщение

Printer not ready ("Печатающее устройство занято")

SET v:=s[m] - переназначение сетевого дисковода
v - сетевой дисковод (E,...,P)
s - дисковод по EKDOS (A,...,D)
m - индивидуалый адрес станции; по умолчанию - адрес той станции, в которой ввели команду

SET P [-/n][m] - переназначение печатающего устройства
- - снятие печатающего устройства
n - индивидуальный адрес станции печатающего устройства
m - индивидуальный адрес станции; по умолчанию - станция, в которой ввели команду

SET S[+/-][m] - изменение состояния блокировки станции
- - разблокировка приема
+ - блокировка приема; в таком режиме станция игнорирует переданные ей сообщения, файлы, запросы ожидания и пуска
m - индивидуальный адрес станции; по умолчанию - станция, в которой ввели команду

RESET [T] - установить исходное состояние
T - в сеть (всем станциям) передается таблица конфигурации данной станции

После смены дискеток следует всегда ввести эту команду.

RUN [p1][p2] - пуск загруженной программы
p1,p2 - см. руководство оператора операционной системы EKDOS

Под действием этой команды управление передается на адрес 100H, параметры записываются в управляющие блоки файла.

SAVE p f [a] - запись содержимого памяти в файл
p - число 256-байтовых блоков
f - имя файла; если файл с таким именем уже существует, то этот файл стирается
a - начальный адрес области памяти; по умолчанию 0100H

LOAD f [a] - загрузка файла в ОЗУ
f - имя файла
a - начальный адрес загрузки; по умолчанию 0100H

TYPE f - отображение текстового файла
f - имя файла; файл должен быть символьным, в коде ASCII

DUMP f [a] - 16-ричное отображение содержимого файла
f - имя файла
a - начальный адрес; по умолчанию 0100H

Формат вывода: 8 байт на каждой строке, перед каждой строкой адрес ячейки.

REN f1=f2 - переименование файла
f1 - существующее имя файла
f2 - новое имя файла

ERA f - удаление файлов
f - имя файла; если вместо имени ввели *.* , то на экран выводится вопрос:

```
ALL FILES (Y/N)?          ("Все файлы? Да/Нет")
```

При ответе "Y" из каталога удаляются все файлы с атрибутом R/W (разрешены чтение и запись). При ответе "N" удаления не выполняются.

EXIT - выход из NETOS
На экран выводится вопрос

```
EXIT (Y/N)?               ("Выйти? Да/Нет")
```

При ответе "Y" имеет место выход из NETOS. В станции с НГМД это означает возврат в EKDOS, в станции без НГМД - возврат в монитор. При ответе "N" сохраняется активность командного процессора.

COPY f1f2 - копирование файла
f1 - имя файла-источника
f2 - имя целевого файла; если вместо f2 выводится только имя устройства (дисковода), то к целевому файлу прикрепляется имя f1

HELP - отображение списка резидентных функций NETOS

Примечания:

1. При применении резидентных функций ERA и DIR допускается имя файла задавать и неоднозначно, в виде универсального образца. В образце применяются универсальные символы "*" и "?" (см. руководство EKDOS).
2. Для прерывания выполнения резидентных функций следует нажать на клавишу ESC.
3. При выполнении функций DUMP, TYPE, SHOW, DIR для останова отображения текста следует нажать на клавиши CTRL S1; для продолжения отображения нажать на любую клавишу, кроме ESC.

4.СЕТЕВЫЕ ФУНКЦИИ

4.1. Обращение к функциям

Обращение к функциям выполняется через 8-ую ячейку памяти:

```
         ---------
ØØØ8H !  C3H  !  - Код команды перехода (JMP)
         ---------
ØØØ9H !  a1   !  - Младший байт адреса входной точки функции
         ---------
ØØØAH !  a2   !  - Старший байт адреса входной точки функции
         ---------
```

Номер функции следует перед обращением занести в регистр C (см. 4.3).

4.2. Код возврата

Код возврата записывается в регистр A. Если A=Ø, то функция была выполнена успешно, в противном случае при выполнении возникла ошибка; код возврата A>Ø представляет собой код ошибки:

Ø1 Синтаксическая ошибка
Ø2 Переполнение буферов для передачи
Ø3 Ошибка в системе передачи данных
Ø4 Переполнение приемных буферов
Ø5 Срабатывание контрольного таймера
Ø6 Ошибка при выборе устройства
Ø7 Файл не найден (только для функции 13)

4.3. Описание функций

В описании приняты следующие обозначения: C=...- номер функции, П - входные параметры, ВИ - выходная информация.

C=Ø Выйти из сети
П: -
ВИ: -
Если станция имеет дисковод, то при выходе имеет место возврат в операционную систему; при отсутствии дисковода выполняется возврат в монитор.

C=1 Передать краткое сообщение
П: B = целевой адрес
DE = адрес начала текста
ВИ: A = код возврата
Максимальная длина сообщения 32 байт, в конце текста должен быть символ "$" или ØØH. Если целевая станция не блокирована, сообщение выводится на первую строку экрана. Номер перед текстом - индивидуальный адрес станции-источника. Текст сообщения отображается в течение нескольких секунд, затем на экране восстанавливается содержимое строки.

C=2 Передать сообщение (содержимое памяти)
П: B = адрес целевой станции
DE = адрес начала текста в памяти станции-источника
HL = целевой адрес текста в ОЗУ целевой станции
A = длина сообщения (1-128 байт)
ВИ: A = код возврата
Если целевая станция находится в режиме приема, то после приема сообщения увеличивается значениее счетчика на экране.

C=3 Передать команду пуска
П: B = адрес целевой станции
DE = пусковой адрес
ВИ: A = код возврата
Экран целевой станции стирается, управление передается на заданный адрес, ситемные управляющие блоки файлов (адреса 5CH, 6CH) и буфер командного процессора (адрес 80H) заполняются нулями.

C=4 Передать команду переключения на режим приема
П: B = адрес целевой станции
DE = начальный адрес для приема сообщения
ВИ: A = код возврата
Целевая станция переводится в режим приема, т.е. станция начинает ожидание поступления сообщений на заданный начальный адрес, передаваемых при помощи функции 2.

C=5 Передать таблицу конфигурации
П: B = адрес целевой станции
ВИ: A = код возврата

C=6 Считывать адрес таблицы конфигурации
П: -
ВИ: HL = адрес начала таблицы конфигурации

C=7 Считывать адрес таблицы параметров станции
П: -
ВИ: HL = адрес начала таблицы параметров
Структура таблицы (каждый элемент = 1 байт):
0) индивидуальный адрес станции
1) число станций в сети
2) индивидуальный адрес станции с печатающим устройством (ПУ)
3) состояние ПУ: =00 - свободно, >0 - занято
4) состояние станции: =00 - разблокирована, >0 - блокирована

C=8 Переназначить сетевой дисковод
П: B = индивидуальный адрес станции (B=0 : данная станция)
E = имя сетевого дисковода (E,...,P)
D = имя дисковода в EKDOS (A,...,D); D=0 : снятие дисковода
ВИ: A = код возврата

```
C=9  Переназначить станцию с печатающим устройством (ПУ)
П:   B  = индивидуальный адрес станции (B=Ø : данная станция)
     E  = индивидуальный адрес станции с ПУ; E=Ø : снятие
ВИ:  A  = код возврата


C=1Ø  Изменить состояние блокировки
П:   B  = индивидуальный адрес станции (B=Ø : данная станция)
     E  = состояние:
          E=ØØH - разблокировка
          разряд Ø = 1  - игнорируется прием сообщений, команд
                          пуска и перевода в режим ожидания
          разряд 1 = 1  - игнорируется прием сообщения
ВИ:  A  = код возврата


C=11  Изменить принадлежность станции к группам
П:   B  = индивидуальный адрес станции (B=Ø : данная станция)
     E  = номер группы (1,...,6)
     D  = Ø  - включение в состав группы
        = 1  - удаление из состава группы
ВИ:  A = код возврата


C=12  Передать содержимое области памяти
П:   B  = индивидуальный адрес целевой станции
     HL = начальный адрес области памяти (в обеих станциях)
     DE = число 128-байтовых блоков
ВИ:  A = код возврата
Целевая станция переводится в режим приема,  в  станции-источнике
на  экран выводятся текст "Send" ("Передача") и  счетчик передан-
ных 128-байтовых блоков.


C=13  Передать файл
П:   B  = индивидуальный адрес целевой станции
     DE = адрес управляющего блока предаваемого файла
ВИ:  A  = код возврата


C=14  Считывать сообщение (содержимого памяти)
П:   B  = индивидуальный адрес станции-источника сообщения
     DE = начальный адрес для чтение сообщения
     HL = начальный адрес для записи сообщения
     A  = длина сообщения (1-128 байт)
ВИ:  A  = код возврата


C=15  Выполнить системные сетевые функции
П:   DE = адрес управляющего блока
ВИ:  -
Функция  применяется совместно с сетевой функцией HREAD  монитора
(см.п.5).
```

C=16 Выполнить тест сети
П: -
ВИ: A = код возврата (A=0 : тест прошел успешно)

C=17 Считывать сообщение, принятое последним
П: -
ВИ: B = адрес источника сообщения (B=00 : сообщений нет)
DE = начальный адрес текста сообщения
Максимальная длина сообщения 32 символа, в конце текста должен быть символ "$" или код 00H.

Примечание.

При выполнении функций 4, 12 и 13 целевая станция переводится в режим приема (ожидания):

1) экран стирается, на первую строку выводится текст "Load" ("Загрузить");

2) при выполнении функций 4, 12, 13 на экран выводиться счетчик принятых 128-байтовых блоков;

3) сообщения и команды пуска принимаются только от станции, выславшей команду перехода в режим приема; передачи от других станций игнорируются;

4) для прерывания режима ожидания следует нажать на CTRL ESC.

5. ПРИМЕНЕНИЕ СЕТЕВЫХ ФУНКЦИЙ МОНИТОРА

Для передачи и приема сообщений пользователь может использовать и сетевые функции резидентного монитора (см. руководство оператора монитора RomBios):

MWRITE - передача сообщения
HREAD - чтение идентификатора сообщения
MREAD - прием сообщения
MIGNORE - игнорирование поступившего сообщения

При применении этих функций следует учитывать следующие правила:

1) максимальная допустимая длина сообщения 160 байт;

2) длина идентификатора 7 байт; в первом байте пользователю не допускается применение значений 01H,...,20H;

3) если после выполнения функции HREAD в первом байте идентификатора значение из интервала 01H,...20H, то пользователь должен запускать сетевую функция 14, загрузив в DE адрес управляющего блока пользователя.

6. ТАБЛИЦА КОНФИГУРАЦИИ (КТ)

КТ содержит информацию о станциях, дисководах и печатающих устройствах, входящих в состав сети. Длина КТ - 117 байт. КТ состоит из двух подтаблиц - таблицы дисководов и таблицы станций.

6.1. Таблица дисководов

Длина этой таьлицы 2 x 12 = 24 байт. Через сеть возможно обращение до 12 различных дисководов (имена E,...,P).

Для каждого имеющегося в сети дисковода таблица содержит следующую инффомацию:

1) имя дисковода в EKDOS (1 байт, 00H-04H):

00H - дисковод отсутствует
01H - дисковод A
02H - дисковод B
03H - дисковод C
04H - дисковод D

2) индивидуальный адрес станции, к которой назначен дисковод (1 байт, 00H-1FH); 00H обозначает отсутствие дисковода.

Таким образом, длина элемента этой подтаблицы 2 байт. Первый элемент относится к сетевому устройству E, следующий к устройству F и т.д.

6.2. Таблица станций

Длина этой таблицы 3 x 31 = 96 байт. Максимальное число станций в сети 31.

Для а каждой станций таблица содержит следующую информацию:

1) принадлежность к группам (1 байт); кодирование групп следующее:

разряд 7 - не используется
разряд 6 - группа 6
разряд 5 - группа 5
... ...
разряд 1 - группа 1
разряд 0 - не используется

Принадлежности станции к какой-либо из групп 1-6 соответствует значение "1" соответствующего разряда. При значении 00H этого байта станция не входит в состав ни одной из групп;

2) индивидуальный адрес станции печатающего устройства (1 байт, 00H,...,1FH). 00H - отсутствие печатающего устройства;

3) состояние блокировки станции (1 байт):

00H - станция разблокирована
разряд 0=1 - игнорируются сообщения, приказы пуска и перехода в режиме приема
разряд 1=1 - игнорируются прием сообщений

Длина элемента этой подтаблицы 3 байт. Первый элемент относится к станции с индивидуальным адресам 01, второй к станции 02 и т.д.

7. ОГРАНИЧЕНИЯ

1. Не допускается присвоение равного значения индивидуального адреса двум или нескольким станциям.

2. Не допускается одновременная запись из нескольких станций на один и тот же дисковод файла с одним и тем же именам.

3. Не допускается несоответствие КТ действительной конфигурации сети (указание в таблице несуществующих дисководов и т.д.).

4. Максимальная скорость передачи достигается при глобальной адресации.

5. Скорость передачи сообщений тем больше, чем меньше число станций в сети.

6. При запуске через NETOS программ, использующих только функции монитора, системные сетевые функции не выполняются.

7. Если при передаче операционной системы поступает новый запрос, то это передается следующей станции (но не станции, выдавшей запрос на операционную систему).

8. Ответ на обращение к дисковым функциям и к функции чтения сообщения (содержимого памяти) ожидается в течение 20 с.

9. При начальном пуске вводят параметры сети в 16-ричной форме.

10. Не допускается одновременное обращение нескольких станций к одному и тому же дисководу посредством функций 17 и 18 EKDOS (т.е. поиска файла и поиска следующего файла).

11. Применение печатающего устройства допускается только через 5-ую функцию BDOS (CTRL P не действует).

8. СООБЩЕНИЯ СИСТЕМЫ NETOS

Disk full - переполнение каталога диска
File not found - указанный файл не найден
File exist - файл уже существует
No file(s) - файлов нет
File is R/O - файл только для чтения
Select error - выбор неправильного дисковода
Load error - ошибка при загрузке файла
Syntax error - несуществующая команда или ошибка в параметрах
W/buffer full - переполнение буферов передачи
Net error - ошибка при передаче данных
R/buffer full - переполнение приемных буферов
Time out - срабатывание контрольного таймера
Printer not exist - печатающее устройство отсутствует
Printer not ready - печатающее устройство не готово
Disk not ready - дисковод не готов к работе
Disk error - ошибка при выполнении функции EKDOS; соответствует стандартным сообщениям EKDOS:
BDOS ERROR: BAD SELECT - ошибка выбора
BDOS ERROR: FILE R/O - файл для чтения
BDOS ERROR: READ ERROR - ошибка при чтении
Bios not exist - отсутствует Bios базовой операц. системы
Bdos not exist - отсутствует Bdos базовой операц. системы
Nbdos error x (A/R/I) - ошибка при выполнении функции EKDOS, связанная с передачей данных (переполнение буферов, срабатывание контрольного таймера и т.д.).
x - описание ошибки
A - прекратить выполнение функции (возврат в командный процессор)
R - повторить
I - игнорировать ошибку

"ЭКТА"
СКБ вычислительной техники Института кибернетики АН ЭССР

ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ИНТЕЛЛЕКТУАЛЬНОГО ТЕРМИНАЛА Е5104

ПАКЕТ ТЕСТОВЫХ ПРОГРАММ "ДИАГНОСТИКА"

DIAGNOSTICS

РУКОВОДСТВО ОПЕРАТОРА

353872.30007-10

12 стр.

Таллин 1987

АННОТАЦИЯ

Пакет "Диагностика" работает под операционной системой CP/M и предназначен для проверки системы микроЭВМ, построенной на базе процессора типа КР580ВМ80А, "Интел 8080/8085" и Z80. Пакет позволяет тестировать работу ОЗУ, процессора, видеомонитора и внешнего накопителя (дисковода или магнитофона). Об'ем пакета 64К байт.

СОДЕРЖАНИЕ

1. ОБЩИЕ СВЕДЕНИЯ О ПРОГРАММЕ

ЕКТА Tallinn *ЭКТА* Таллин КОМПОНЕНТ / КОМПЛЕКС ПО

DIAGNOSTICS	353872.30007-10	Версия: 1.0

ПАКЕТ ТЕСТОВЫХ ПРОГРАММ

ОБЪЕМ: 16К байт	ОПЕРАЦ. СИСТЕМА: CP/M
ТРЕБУЕМОЕ ОЗУ	Программа: 52К байт / Данные - байт

НОСИТЕЛЬ: ГИД / Кассетная МЛ
РЕГ. но НОСИТЕЛЯ:

НАЗНАЧЕНИЕ И МЕТОД:
Проверка исправности ОЗУ, процессора, видеомонитора и внешнего накопителя. Состоит из файлов:

MTEST	- тест памяти, адреса 1500 ... FFFF
MTEST2	- тест памяти, адреса 0100 ... 14FF
CPU	- тест процессора
TERM	- тест видеомонитора
QRUN	- быстрый комплексный тест (ОЗУ, процессор, дисковод)
QDISK	- быстрый тест дисковода
TTEST	- магнитофон

ТЕХН. ХАР-КИ:
Время выполнения комплексного ускоренного теста 4 мин
Длительность поразрядной проверки памяти (16К) 13 часов

ПРОЦЕССОР / ЭВМ: 8080, 8085, Z80, K580 / E5104

ВНЕШНИЕ УСТР-ВА:
QRUN, QDISK - дисковод
MTEST, MTEST2, CPU - магнитофон или дисковод, если в командной строке указано внешнее ЗУ

НЕОБХ. ПРОГРАММЫ И ФАЙЛЫ:
Комплексный быстрый тест QRUN.COM запускает тесты QDISK, CPU, которые должны находиться на магнитном носителе

15.10.1987	АРХ. МЛ: 3.00	ПРОГРАММИСТ: Сийбак, С.
ИСХ. ТЕКСТ:	АРХ. МЛ: 3.00	ЯЗЫКИ: Паскаль, ASM

2. Управление пакетом

"Диагностика" - это пакет для комплексного испытания систем микроЭВМ, содержащих микропроцессор типа 8080, 8085, Z80 или К580ИК80А. Пакет даёт пользователю, не имеющему специальной подготовки, возможность выявить источник отказов (память, микропроцессор, дисковод, дисплей), которые затем могут быть устраньены специалистом по аппаратуре микроЭВМ. Разумеется, для применения тестов ЭВМ должна быть способна к загрузке и пуску программы.

2.1. Общие указания по применению

Пакет состоит из следующих программных и текстовых файлов:

MTEST	.COM	5K	Тест для памяти адреса 1500H...FFFFH
MTEST2	.COM	10K	Тест для памяти адреса 0100H...14FFH
CPU	.COM	19K	Тест для микропроцессора
TERM	.COM	17K	Тест для дисплея
QRUN	.COM	5K	Быстрый тест для памяти, процессора и дисковода *)
QDISK	.COM	8K	Быстрый тест для дисковода *)

Рекомендуется выполнять испытания, соблюдая следующий порядок:

1) память,
2) процессор,
3) дисковод *)
4) дисплей.

*) Тесты предназначены только для дисковода, они не применимы для магнитофона

Если память не функционирует нормально, то нельзя судить и о правильности результатов других тестов. После проверки выясняют исправность микропроцессора. Поскольку некоторые команды процессора применяются сравнительно редко, то может оказаться, что программы работают нормально и при неисправном процессоре.

2.2. Сбор тестовых данных на магнитный носитель

В находящийся на внешнем носителе файл A:DIAG.LOG могут выдать свои сообщения следующие тест-программы: MTEST, MTEST2 и CPU. Такой способ хранения сообщений весьма удобен в случае, если последовательность тестов оказывается продолжительной во времени (например, поразрядное тестирование памяти в об ёме 16K занимает около 13 часов). Занесение на внешний носитель не исключает отображения на экране. Внешним запоминающим устройством может быть дисковод или магнитофон. В файле DIAG.LOG перед сообщениями каждого теста остаются две пустые строки и символы "**". Позднее можно дополнить этот файл при помощи редактора текстов и вывести на печать. Важно, чтобы на магнитном носителе было обеспечено достаточное место для файла сообщений, иначе работа тест-программы может прерваться, а на экране появится сообщение операционной системы об ошибке.

2.3. Применение параметров командной строки

Для тест-программ MTEST, MTEST2 и CPU можно задать параметры посредством командной строки. Тест для дисплея и комплексный ускоренный тест (QRUN) не требуют параметров при вызове.

3. Проверка памяти

3.1. Набор тестов ЗУ

Имеются следующие тесты и возможности их применения:
- Быстрый тест (Quick test)
- Поразрядный тест (Walking bit test)
- Тест на восстанавливаемость (Burn-in test)
- Тест на скорость (Speed test)
- Переключение групп памяти
- Отображение содержимого проверяемой области
- При обнаружении ошибки: выдача адреса, занесённого и считанного значений
- Выдача перечня ошибок по двоичным разрядам

Быстрый тест состоит в том, что выбранная область памяти заполняется значением 00H, затем выполняется побайтовое считывание из этих ячеек, с тем, чтобы проверить нулевые значения всех двоичных разрядов. Потом такая же процедура выполняется со значением FFH, при котором все двоичные разряды должны содержать единицы. Наконец, в выбранную область записываются случайные числа, также с последующим контрольным чтением. Быстрый тест позволяет выяснить 90% из всех неисправностей памяти.

Поразрядный тест помогает выявить отказы, не обнаруживаемые быстрым тестом, например, ошибки в адресации. Этот тест отнимает значительно больше времени - оно пропорционально квадрату числа проверяемых байтов.

Тест на восстанавливаемость предназначен специально для динамических ЗУ. На заданной области памяти выполняется определённый цикл записи 1000H раз. Затем в эти ячейки записывают обратные коды применённых значений. Должна произойти инверсия всех двоичных разрядов.

Тест на скорость проверяет соответствие скорости работы памяти скорости процессора. На экран выдаётся сигнал времени с постоянным периодом. Этот период должен соответствовать тактовой частоте процессора следующим образом:

Тактовая частота процессора, МГц:	2	4	6
Сигналов времени в минуту:	1	2	3

Отсутствие такого соответствия указывает на неполадки в памяти.

Все тесты памяти позволяют выбрать для проверки любую группу памяти. В некоторых ЭВМ может применяться система ОЗУ, при которой передаваемый в определённый порт или определённую ячейку памяти специальный код вызывает переключение на другую группу памяти.

Для испытания памяти имеются две программы одинакового действия:

MTEST - для испытания диапазона 1500H...FFFFH
MTEST2 - для диапазона под MTEST (0100H...14FFH)

3.2. Пуск теста ЗУ

1) Вызов: MTEST или MTEST1 вызываются через операционную систему. На экране появится вопрос:

LOG TO DISK (Y/N) ("Занести на диск? Да/нет")

При отрицательном ответе все сообщения будут выводиться только на экран.

2) Начальный адрес вводится в ответ на сообщение

ENTER START ADDRESS (HEX) ("Введите начальный адрес")

Адрес вводится в виде 4-значного шестнадцатеричного числа. Допустимы только символы Ø...9 и A...F. В случае ошибки в процессе ввода можно вернуться к началу шага 2, вводя любой недопустимый символ.

По умолчанию (ввод RETURN вместо числа) начальным адресом будут считаться 1500H (в тесте MTEST2 100H).

Для возврата из теста следует нажать клавиши CTRL+C.

3) Конечный адрес вводят в ответ на текст

ENTER ENDING ADDRESS (HEX) ("Введите конечный адрес")

Адрес вводят в виде 4-значного шестнадцатеричного числа. Если вместо числа вводится RETURN, то используется значение по умолчанию: конец пользовательской памяти (в случае MTEST2 - 1500H).

Допустимость введенных адресов не проверяется, поэтому слишком малое (0100H) или слишком большое (>fbase) значение адреса может привести к записи тестовых данных в системную зону или к отказам теста. Значение адреса fbase хранится в ячейках 0005H, 0006H.

При испытании области памяти, содержащей несуществующие адреса, тест сообщает о таких адресах как о неверных.

4) Меню тестов появится на экране после ввода адресов.

PLEASE SELECT	("Выберите, пожалуйста:
Q - QUICK TEST	- быстрый тест
W - WALKING BIT TEST	- поразрядный тест
B - BURN IN TEST	- тест на восстанавливаемость
S - SPEED TEST	- тест на скорость

Тест выбирается нажатием соответсвующей клавиши.

5) Группа памяти выбирается на основе следующего диалога:

ENTER B FOR BANK SELECT ("Для переключения группы ввести B")

При нажатии на RETURN имеет место переход на шаг 6. При нажатии клавиши B появится вопрос:

POKE WHICH PORT? ("Указать вводно/выводной порт")

В ответ нужно ввести номер порта. "Порт" здесь надо понимать как номер порта, если введённое 16-ричное число не превышает 256, в ином случае - как специальный адрес памяти.

Следует вопрос:

```
WHAT VALUE?          ("Каково значение?")
```

В ответ нужно ввести в 16-ричной форме тот управляющий код, который применяется для переключения групп памяти.

Значения адреса и кода зависит от конкретной аппаратуры.

6) Число циклов теста - это последний вводимый параметр.

```
ENTER NUMBER OF ITERATIONS (DEFAULT = 1)
("Введите число итераций; по умолчанию: 1")
```

При нажатии RETURN тест выполняется только один раз.

После того как в ходе описанного диалога введены все параметры, пускается выбранный тест памяти. Работа продолжается до достижения желаемого числа повторений или до прерывания путем нажатия на какую-либо клавишу.

3.3. Параметры командной строки теста ЗУ

Ответы на все вопросы теста можно ввести с помощью командной строки при вызове теста. В зависимости от того, применяется переключение групп памяти или нет, возможны две различных формы вызова.

1) С переключение групп памяти:

```
>MTEST L S E T B P V R,
```

где:

```
L - место выдачи результатов (пЕлОГ)
    Л - в файл DIAG.LOG
    N - только на экран
S - начальный адрес области памяти, 16-ричное число (Start)
    Ø - значение по умолчанию
E - конечный адрес, 16-ричное число (End)
    Ø - значение по умолчанию
T - тип теста (Quick)
    Q - быстрый тест (Quick)
    W - поразрядный тест (Walking test)
    B - тест на восстанавливаемость (Burn test)
    S - тест на скорость (Speed)
B - выбор групп памяти (Bank select)
    B - переключение
    N - без переключения
P - номер порта или специальный адрес памяти, 16-ричное число
    (Port)
V - код выбора групп памяти, 16-ричное число (Value)
R - число циклов теста, 16-ричное (Repetitions)
```

Пример:

```
>MTEST L 2ØØØ 3ØØØ Q B 23 55 1
```

2) Без применения памяти:

MTEST L S E T N R

N - признак отсутствия переключения памяти
Остальные параметры: см. выше

Пример:
MTEST C 3000 30FF W N 3

Примечание. Параметры командной строки можно вводить только в привёденной выше последовательности.

4. Испытание процессора

4.1. Операции испытания

При испытании выполняются следующие операции:
- определение типа процессора,
- проверка скорости процессора,
- проверка набора команд процессора с извещением об ошибках.

Тест для процессора интерпретирует программу, предназначенную для проверки всех отдельных команд данного процессора, а также последовательностей, содержащих несколько команд. После выполнения каждой команды проверяется содержимое всех регистров процессора для контроля правильности изменения состояния регистров. Это позволяет обнаружить, например, таакой тип неисправности, при котором операция занесения в регистр A может оказать влияние на состояние регистра B.

4.2. Пуск теста процессора

1) Вызов: CPU, черз операционную систему.
2) Квитация: под заголовком теста выводится строка:

ABCDEFGHIJKLMNOPQRSTUWXYZ

Каждый символ в этой строке обозначает, что выполнен один краткий тест для процессора. Если строка не выведена или не содержит всех указанных букв, то процессор неисправен.
3) Тип процессора. После вывода первого сообщения тест выявляет тип процессора и выдаёт сообщение:

CPU IS 8080 (или CPU IS 8085 или CPU IS Z80)
("Процессор типа ...")

4) Скорость. В начале этого испытания выдаются сигнал времени и сообщение:

BEGIN TIMING TEST ("Начала теста времени")

Через определённый интервал времени выводятся второй сигнал времени и сообщение:

END TIMING TEST ("Конец теста времени")

Интервал между двумя сигналами времени зависит от типа процессора:

Тип процессора	8080	Z80	Z80A
Тактовая частота процессора, МГц	2	4	5
Интервал между сигналами времени	2 мин	1 мин	40с

Если интервал значительно отличается от приведённых значений, причиной этого может оказаться неисправность процессора.

Примечание. В операционной системе коллективного пользования или в системе, управляемой прерываниями, тест может дать результаты, отличающиеся от описанных здесь.

5) Проверка всех команд.

6) Конец теста. При нормальном исходе испытания появится текст:

```
CPU TESTS OK  ("Результат тестирования процессора нормальный")
```

В противном случае появятся сообщения об ошибках:

```
CPU FAILED TEST                 - отказ при испытании процессора
ERROR COUNT XXXXH               - счётчик ошибок
INSTRUCTION SEQUENCE WAS XXXXXXH
                                - последовательность команд была...
REGISTER XX CONTAINS XXH        - регистр XX содержит...
BUT SHOULD CONTAIN XXH          - а должен содержать...
REGISTER VALUE BEFORE INSTRUCTION SEQUENCE WAS XXH
                                - содержимое регистра перед
                                    выполнением команд было...
TEST NUMBER XXXXH               - номер теста
```

Если окажется, что процессор не смог правильно выполнить какую-либо последовательность команд, то, как правило, такой же отказ повторится и при следующей попытке. Поэтому для получения более точной информации о сущности отказа рекомендуется написать краткую ассемблерную программу и запустить эту программу при помощи отладчика SID.

4.3. Параметры командной строки теста процессора

При вызове теста процессора можно указать, желательна ли выдача сообщений об ошибках в файл на магнитном носителе. Если командная строка содержит параметр LOG, то сообщения выводятся в файл DIAG.LOG:

```
>CPU LOG
```

В иных случаях сообщения выводятся только на экран.

5. Проверка дисплея

5.1. Тесты для дисплея

Проверяются следующие операции:

- стирание всех данных с экрана,
- стирание строки,
- перемещение курсора в 4-х направлениях,
- адресация курсора,
- чтение позиции курсора,
- переключение режимов нормального и обращённого изображения.

5.2. Пуск тестов дисплея

1) Вызов: TERM, вызывается через операционную систему. На экране появится соответствующий текст.
2) Пуск последовательностей тестов автоматический. Перед каждым тестом выводится поясняющий текст. Пользователь имеет возможность выполнять тесты в другой последовательности или же только часть всех тестов. Для этого нужно выбрать подходящий символ из списка выводимого перед тестом:

Type "B" to bypass, "ESC" to exit or "RET" to test

и ввести "B" для пропускания, "ESC" для возврата, "RET" для пуска.
3) Конец тестирования: на всей площади экрана вычерчивается спираль.

Поскольку тесты для дисплея требуют активного участия пользователя, то выдавать сообщения в файл на магнитном носителе невозможно.

6. Комплексный ускоренный тест

6.1. Компонентные тесты

Компонентный тест состоит из трёх программ, выполняемых одна за другой (программы должны находиться в одном и том же внешнем ЗУ):

QRUN. COM - ускоренный тест для всей пользовательской памяти
QDISK. COM - ускоренный тест для диска *)
CPU. COM - тест для процессора

*) Тест для диска выполняются только при наличии дисковода, они не применимы для магнитной ленты.

Комплексный ускоренный тест позволяет быстро обнаружить основной источник ошибок. Испытания в полном об ёме (в том числе испытания диска) длятся около 4 мин при тактовой частоте процессора 2МГц.

6.2. Пуск комплексного ускоренного теста

1) Вызов: QRUN , через операционную систему
2) Тест памяти. После заголовка комплексного теста на экране появляется сообщение:

Memory test ("Тест памяти")

и пускается ускоренный тест пользовательской зоны памяти. При завершении этого испытания выдаётся сообщение:

Memory test complete ("Испытание памяти завершено")

Если тестированием обнаружены какие-либо ошибки, то об этом извещается следующим образом:

LOCATION	DATA WRITTEN	DATA READ
XXXXH	YYH	ZZH
("Адрес"	"Данные, запись"	"Данные, чтение")

При обнаружении ошибок желательно, для получения более полной информации, проверить память с помощью стандартного теста.
3) Тест для диска. После тестирования памяти проводится ускоренный тест для дисковода. На экране с интервалами в несколько секунд появятся следующие сообщения:

Disk test ("Проверка диска")
Read/write test ("Проверка записи и чтения")
Random seek test ("Проверка случайного обращения")

При завершении теста на экран выводятся сообщения:

X read/write errors detected ("Обнаружено х ошибок чтения/записи")
X seek errors dedected ("Обнаружено х ошибок обращения")
4) Тест для процессора - последний этап комплексного теста. Этот тест типовой, его функции и выводные формы описаны в 10.3.

7. Тест для магнитофона

чд7.1. Операции контроля .
Тест включает следующие операции:
- определение уровня записи магнитофона,
- проверка режима записи,
- проверка применяемой сформатизированной ленты.

7.2.Пуск теста

Для вызова теста черес операционную систему вводят имя теста TTEST.
Обязанности пользователя при выполнении теста заключаются в выполнении операций, сообщаемых программой, и не требуют подробного об´яснения.

"ЭКТА"
СКБ вычислительной техники Института кибернетики АН ЭССР

ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ИНТЕЛЛЕКТУАЛЬНОГО ТЕРМИНАЛА Е5104

УТИЛИТЫ К ОС EKDOS

UTY Ø.2

РУКОВОДСТВО ОПЕРАТОРА

353871.3ØØ13-Ø2

9 стр.

Таллинн 1988

СОДЕРЖАНИЕ

1.ОБЩИЕ СВЕДЕНИЯ О ПРОГРАММЕ

ЕКТА Tallinn *ЭКТА* Таллинн КОМПОНЕНТ / КОМПЛЕКС ПО

UTY	353872.3ØØ13-Ø2	Версия: Ø.2
УТИЛИТЫ К ОС EKDOS		
ОБЪЕМ байт	ОПЕРАЦ. СИСТЕМА:	EKDOS 2.29
НОСИТЕЛЬ: ГМД РЕГ. No. НОСИТЕЛЯ:		
НАЗНАЧЕНИЕ И МЕТОД: Обеспечение различных вспомогательных функций		
ССЫЛКИ: Уэйт М.,Ангермейер Дж. Операционная система CP/M. Пер. с англ. М.:Радио и связь, 1986, 312 с.		
ТЕХН. ХАР-КИ:		
ОГРАНИЧЕНИЯ:		
ПРОЦЕССОР / ЭВМ: КР58ØИК8Ø/Е51Ø1		
ВНЕШНИЕ УСТР-ВА: Видеомонитор		
НЕОБХ. ПРОГРАММЫ И ФАЙЛЫ: FORMAT, DOSGEN, STAT, LOAD, PIP, ED, SUBMIT, DUMP		
14.Ø9.1987	АРХ. МЛ: 3.ØØ	ПРОГРАММИСТ: Кауквер, А.

2. ТРАНЗИТНЫЕ КОМАНДЫ

2.1. FORMAT

Предназначена для форматирования односторонних 8Ø-дорожечных дисков при использовании дисководов СМ5640. После запуска на экран дисплея выводится текст:

```
*** FORMAT VER. 3.4 ***

FORMAT 386K DISK (CM564Ø)
SELECT DEVICE (A/B)?                    ("Выбрать дисковод A/B")
```

Нажать клавишу A или B, в зависимости от дисковода на котором будет проведено форматирование. После этого выводится текст:

```
INSERT NEW DISK TO DRIVE A (или B)      ("Вставить новый диск в A
AND STRIKE RETURN TO START FORMAT        (B), нажать RETURN для
                                         запуска форматизации")
```

Нажатием клавиши RETURN запускается программа форматирования. Во время работы программы на экран выводится информация о возможных обнаруженных ошибках.

Для выхода из программы нажать на ESC.

2.2. DOSGEN

Предназначена для генерирования ОС на системных дорожках диска.

После запуска на экран выводится текст:

```
*** DOSGEN VER.3.4 ***

SOURCE DRIVE NAME:                      ("Имя дисковода-источника")
```

Вставить в дисковод A или B исходный диск ОС и нажать соответствующую клавишу (A или B). На экран выводится текст

```
SOURCE ON A (B),                        ("Исходный диск в A (B).
THEN TYPE RETURN                         Нажать на RETURN")
```

Нажатием клавиши RETURN запускается считывание исходной системы. На экране появится сообщения:

```
FUNCTION COMPLETED                      ("Функция завершена")
DESTINATION DRIVE NAME                  ("Имя целевого дисковода:
(OR RETURN TO REBOOT):                   Для повторной загрузки RETURN")
```

Вставить в дисковод A или B диск, на котором генерируется ОС, и нажать соответствующую клавишу (A или B). На экране появится текст:

```
DESTINATION ON A (или B)          ("Целевой диск в A (B).
THEN TYPE RETURN                   Нажать на RETURN")
```

После нажатия клавиши RETURN генерируется (записывается) ОС на диске.

Для выхода из программы нажать RETURN.

2.3. MODE

Предназначена для изменения формата экрана. Запускается программа с параметрами:

```
MODE XX, где
XX=4Ø - выбрать формат экрана 4Ø x 24 (4Ø символов, 24 строки).
XX=53 - выбрать формат экрана 53 x 24.
XX=64 - выбрать формат экрана 64 x 2Ø.
```

2.4. STAT

Вывод состояния (STATUS) или характеристик (STATISTICS).

```
Команда              Сообщения

STAT                 D: R/W, SPACE: NNNK
                     D: R/O, SPACE: NNNK

STAT D:              BYTES REMAINING ON X: NNNK

STAT [D:] AFN[*]  [SIZE] RECS BYTS EXT D:FILENAME.TYP
                         RRRR BBBK EE  D:PPPPPPPP.SSS
```

```
SIZE - виртуальный размер файла
RRRR - число 128-байтных записей
BBB  - количество килобайт (BBB=RRRR*128/1Ø24)
EE   - количество экстентов по 16K (EE=BBB/16)
D    - имя диска, содержащего данный файл
PPPPPPPP.SSS - имя файла и расширение
AFN  - имя файла с расширением
UFN  - имя файла без расширения
```

```
STAT D:DSK:
     D: DRIVE CHARACTERISTICS          (Характеристики диска)
  312Ø: 128K BYTE RECORD CAPACITY      (Об´ем записи, байт)
   39Ø: KILOBYTE DRIVE CAPACITY        (Об´ем диска, байт)
   128: 32 BYTE DIRECTORY CAPACITY     (Об´ем каталога, байт)
   128: CHECKED DIRECTORY ENTRIES      (Макс.число записей в
                                         каталоге)
   256: RECORDS/EXTENT                 (Записей в экстенте)
    16: RECORDS/BLOCK                  (Записей в блоке)
     2: RESERVED TRACKS                (Резервированных дорожек)

STAT VAL: TEMP R/O DISK: D:=R/O        ("Диск только для чтения")
          SET INDICATOR: D:FILENAME.TYP $R/O $R/W $SYS $DIR
                                       ("Установить индикатор")
          DISK STATUS: DSK: D:DSK      ("Статус диска")
          USER STATUS: USR:            ("Статус пользователя")
          IOBYTE ASSIGN:               ("Список возможных назначений")

STAT D:=R/O                            Перевести диск в режим
                                       "только для чтения". Обрат-
                                       ный перевод возможен только
                                       при "горячем" старте.

STAT DEV:                              Печать текущих назначений.

STAT LD1=PD1,...,LDN=PDN               Задание списка назначений.
```

2.5. LOAD

```
LOAD [X:] UFN
```

Чтение файла с заданным именем в формате HEX и создание файла в формате COM.

2.6. PIP

Операции обмена с внешними устройствами.

(1) PIP
(2) PIP <Командная строка>

Формат (1) позволяет выполнить несколько командных строк, вводимых после *. Выход из PIP по CR (сразу после *).
Формат командной строки:

ПРИЕМНИК = ИСТОЧНИК £1, ..., ИСТОЧНИК £N параметры

Допустимые виды сокращений:

PIP X:=AFN	Копировать на X все файлы AFN с активного диска
PIP X:=Y:AFN	Копировать на X все AFN файлы с Y
PIP UFN=Y:	Эквивалентно PIP UFN=Y:UFN
PIP X:UFN=Y:	Эквивалентно X:UFN=Y:UFN

Дополнительные имена устройств:

NUL:	Вывод 4Ø нулей в коде ASCII
EOF:	Вывод конца файла (CTRL Z)
INP:	Специальное устройство ввода. Вызов: CALL 1Ø3H, данные в 1Ø9H
OUT:	Специальное устройство вывода. Вызов: CALL 1Ø6H, данные в регистре C
PRN:	Вывод на LST: с нумерацией строк, с таблицей по каждой восьмой колонке, с установкой страницы после каждых 6Ø строк

Параметры PIP, заключаемые в квадратные скобки:

B	Режим поблочной передачи
DN	Стирание символов после N-го (для узкой печати)
E	Эхо операций передачи
F	Определение форматов передачи
GN	Ввод файла пользователя с номером N (Ø-15)
H	Передача шестнадцатеричных данных с проверкой
I	Игнорирование ØØ при передаче (одновременно влечет H)
L	Перекодирование верхнего регистра в нижний
N	Нумерация строк (если N2, то включаются ведущие нули и устанавливается табуляция)
O	Передача об'ектного файла
PN	Перевод страницы после каждых N строк
QS+Z	Прекратить передачу после строки S
R	Чтение системного файла
SS+Z	Начать передачу со строки S
TN	Табуляция в каждой N-ой колонке
U	Перекодирование нижнего регистра в верхний
V	Проверка правильности копирования
W	Перезапись файла с индикатором R/O без вывода запроса
Z	Сбросить в Ø бит четности

2.7. ED

Редактирование текста

ED [X:] UFN [Y:]

Инициализируется командой:

ED имя файла . тип файла

Строка символов оканчивается <CR LF>

CR - указатель символа в буфере

Команды:

NA	Добавление N строк в буфер из редактируемого файла (если N=£, то заполнение всего буфера)
+-B	Перемещение CP в начало или конец буфера
+-NC	Перемещение CP по буферу на N символов вперед, если "+" и назад, если "-" (CR и LF воспринимаются как два отдельных символа)
+-ND	Стирает N символов перед CP, если "+" и за CP, если "-"
E	Конец редактирования, закрытие файла
NF C1 C2...CK	Поиск строки по образцу <C1 C2...CK>, CP передвигается за последним символом CK, если сравнение произошло. Ищется N-ое вхождение образца в буфере.
H	Окончание редактирования, закрытие и повторное открытие отредактированного файла в качестве исходного для ED.
I C1C2...CK	Ввод строк символов с клавиатуры (каждая строка оканчивается CR LF) до CTRL Z.
NI C1C2...CK	<CTRL Z> D1D2...DN <CTRL Z> E1E2...EQ <CTRL Z> Команда сопоставления ищет N раз ближайшее вхождение образца (C1C2...CK), затем за CK вводится последовательность символов (D1D2...DN) и стираются все символы от DN до образца (E1E2...EQ).
+-NK	Стирание N строк исходного текста в буфере. Если CP находится не в начале строки, то сохраняются символы перед CP, если "+" и после CP, если "-".
+-NL	Перемещает CP по строкам, если N=Ø, то в начало текущей строки; если N не равно Ø, то в начало текущей строки, а затем N строк вниз ("+") или вверх ("-").
NM C1C2...CK	Макрокоманда (C1, C2 и т.д. команды ED), выполняет строку команд N раз или если N=Ø/1, до ошибочного условия.
+-NN C1C2...CK	Поиск N-ого вхождения образца аналогичен F, но поиск происходит по всему исходному файлу.
O	Восстановление исходного файла, перезапуск ED и действия предыдущих команд аннулируются.

+-NP	Пересылка и печать страницы.
Q	Выход из E без изменения исходного файла.
R F1F2...FN	Чтение библиотечного файла в буфер в процессе редактирования (F1F2...FN файла тип LIB).
NS C1C2...CK (CTRL Z) D1D2...DH	Постановка второго образца вместо первого осуществляется N раз (аналогична F C1C2...CK и D1D2...DH).
+-NT	Печать строки если N=Ø, то начало текущей строки до CP если N=1, то конец текущей строки после CP если N>1, то текущая строка и N-1 строк за ней ("+") или перед ней ("-").
+-U	Перевод из нижнего регистра в верхний (+U) и отмена (-U).
NZ	
+-N	Перемещает CP вверх или вниз на N строк и печатает одну строку (эквивалентно +-NLT).

2.8. SUBMIT

SUBMIT UFN PAR1...PARN

Создание на активизированном диске файла типа SUB для дальнейшей работы в автоматическом режиме.

2.9. DUMP

DUMP UFN

Отображение на экране содержимого системного файла в шестнадцатеричном виде.

"ЭКТА"
СКБ вычислительной техники Института кибернетики АН ЭССР

ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ИНТЕЛЛЕКТУАЛЬНОГО ТЕРМИНАЛА Е5104

ИНТЕРПРЕТАТОР ЯЗЫКА БЕЙСИК

JBASIC

РУКОВОДСТВО ОПЕРАТОРА

353872.3Ø016-11

8 стр.

Таллинн 1988

СОДЕРЖАНИЕ

1.ОБЩИЕ СВЕДЕНИЯ О ПРОГРАММЕ

ЕКТА Tallinn *ЭКТА* Таллин КОМПОНЕНТ / КОМПЛЕКС ПО

JBASIC	353872.3Ø016-11	Версия: 1.1

ИНТЕРПРЕТАТОР ЯЗЫКА БЕЙСИК

ОБЪЕМ 8К байт | ОПЕРАЦ. СИСТЕМА: EKDOS

ТРЕБУЕМОЕ ОЗУ	Программа 8К байт	Данные 1К байт

НОСИТЕЛЬ: Расширитель ПЗУ / ГМД
РЕГ. NO НОСИТЕЛЯ:

НАЗНАЧЕНИЕ И МЕТОД:
Интерпретатор позволяет создать в ОЗУ программы на языке Бейсик, а при наличии внешних запоминающих устройств многократно использовать эти программы.
При использовании НГМД запускается командой JBASIC", а при использовании расширителя ПЗУ запускаются директивом монитора "A".

ТЕХН. ХАР-КИ:
Имеет примитивы для создания изображений. Максимальный номер программной строки: 65529.

ОГРАНИЧЕНИЯ:
Число символов для различения имен переменных: 2. Отсутствуют средства для работы с файлами.
В наименованиях команд Бейсика допускаются только заглавные буквы.

ПРОЦЕССОР / ЭВМ: КР580ВМ80А / Е5104

ВНЕШНИЕ УСТР-ВА:
Видеомонитор
НГМД или расширитель ПЗУ

НЕОБХ. ПРОГРАММЫ И ФАЙЛЫ: JBASIC.COM

14.Ø9.1987	АРХ. ИЛ: 3.ØØ	ПРОГРАММИСТ: Яаксоо К.
ИСХ. ТЕКСТ:	АРХ. ИЛ: 3.ØØ	ЯЗЫКИ: Ассемблер

Приведём краткое описание версии языка Бейсик, предназначенной для микроЭВМ, разработанных в СКБ вычислительной техники Института кибернетики АН ЭССР. Более подробные данные можно найти в руководящих материалах фирмы "Майкрософт".

2. СПЕЦИАЛЬНЫЕ СИМВОЛЫ

CTRL+C	Прерывает выполнение программы; транслятор переходит в командный режим, на экране появляется сообщение "READY" ("Готовность")
CTRL+H	Стирает последний введённый символ
CTRL+X	Стирает текущую вводимую строку
RETURN	Завершает введённую строку
:	Разделяет находящиеся на одной строке предложения
?	Равнозначен директиве PRINT
$	Обозначает символьную переменную

3. ТИПЫ ДАННЫХ

Символьная цепочка:	от Ø до 255 символов
Целое число:	от -32768 до 32767
Число с плавающей запятой:	от -1,7E+38 ДО 1,7E+38

4. КОМАНДЫ

Команда	Синтаксис/Описание	Пример
CHANGE	CHANGE Создаёт возможность замены ленты	CHANGE
CLEAR	CLEAR <выражение> Выделяет для буфера цепочки число байтов, определённое выражением	CLEAR 5ØØ
CLOAD	CLOAD "<имя файла>" Загружает с ленты в ОЗУ программу, находящуюся в текстовом файле <имя файла>.BAS	CLOAD "FILE"
CLS	CLS Гасит экран, перемещает курсор в левый верхний угол	CLS
COLOR	COLOR = <индекс> Определяет цвет (индексами от Ø до 7) для последующих графических операций	COLOR=Ø
CONT	CONT Продолжает выполнение программы, прерванное приказом STOP или нажатием клавиш CTRL C	CONT
CSAVE	CSAVE "<имя файла>" Записывает программу на Бейсике на ленту и присваивает ей идентификатор <имя файла>.BAS	CSAVE "ABC"
CUR	CUR<y><x> Перемещает курсор в позицию, определённую номерами строки (y) и столбца (x) на экране	CUR 1Ø,2Ø

DATA	DATA <постоянные> Записывает в память постоянные, чтение которых возможно после приказа READ	DATA 10,20 "ABC"
DFF FN	DFF FN <имя> (<аргумент>)= <выражение> Определяет арифметическую функцию одного аргумента	DFF FNSQ(x)= =X*X
DIM	DIM <массивы> Выделяет память для массивов, устанавливает максимальные значения индексов массивов	DIM A(3), X4(4,10)
END	END Завершает выполнение программы	END
FOR	FOR <переменная = <выражение> TO <выражение> STEP <выражение> Применяется вместе с NEXT для многократного выполнения отрезка программы. При каждом выполнении значение переменной увеличивается на величину STEP (по умолчанию 1)	FOR I=1 TO 10
GOSUB	GOSUB <номер строки> Вызывает подпрограмму, начинающуюся с указанной строки программы	GOSUB 1000
GOTO	GOTO <номер строки> Осуществляет безусловный переход на указанную строку .	GOTO 190
HGR	HGR Переводит транслятор в графический режим	HGR
IF/THEN IF/GOTO	IF <выражение> THEN <предложение> :<предложение>... <номерстроки> Если значение выражения ненулевое, выполняется директива после THEN или переход на строку, заданную номером. При нулевом значении выполняется переход к следующей строке	IF X>Y THEN M= IF/A<0 GOTO 3C
INPUT	INPUT "<текст>"; <переменная> , <переменная>... Считывает значения с клавиатуры и присваивает их соответствующим переменным	INPUT"NAME;A$
LET	LET <переменная> = <выражение> Присваивает значение переменной	LET A$="012"
LIST	LIST <номер строки> Отображает строки программы с первой строки (по умолчанию) или со строки, заданной номером	LIST 1000
NEW	NEW Удаляет программу из памяти	NEW
NEXT	NEXT <переменная> ,<переменная>... Фиксирует конец цикла типа FOR	NEXT I
ON/GOSUB	ON<выражение> GOSUB <строка>, <строка>... В зависимости от значения выражения выполняется одна из указанных подпрограмм (если выражение = 1, то первая и т.д.)	ON I+1 GOSUB 200,300

ON/GOTO	ON <выражение> GOTO <строка>, <строка>...	ON LEN (A$) GOTO 10,20,25
	Осуществляет переход на одну из указанных строк, в зависимости от значения выражения	
OUT	OUT<порт>,<байт>	OUT I,D(K)
	Выводит данные в выводной порт	
PLOT	PLOT TO <x>,<y> TO<x>,<y>...	PLOT 12,10 TO 120,100
	PLOT <x>,<y> - отображает на экране точки с координатами <x>,<y> PLOT TO <x>,<y> - отображает отрезок от графического курсора до точки <x>,<y1>TO<x2>,<x2> - отображает отрезок от точки <x1>,<y1> до точки <x2>,<y2>	
POKE	POKE <адрес>,<байт>	POKE 32000,255
	Записывает байт в ОЗУ по заданному адресу	
PRINT	PRINT <выражение> ; <выражение>...	PRINT A$, 1+2
	Выводит данные на экран	
READ	READ <переменная> ,<переменная>	READ I,J,LS
	Считывает значения, определённые приказом DATA, и присваивает их соответствующим переменным	
REM	REM <комментарий>	REM-SUBROUTINE
	Позволяет ввести и программу комментарии, не подлежащие обработке	
RESTORE	RESTORE	RESTORE
	Сбрасывает выполнение DATA, чтение начинается снова с первого определения DATA	
RETURN	RETURN	RETURN
	Завершает выполнение подпрограммы, возвращает управление в место вызова	
RUN	RUN	RUN
	Запускает выполнение программы	
STOP	STOP	STOP
	Прекращает выполнение программы, выводит сообщения об останове, переводит интерпретатор в командный режим	
SYSTEM	SYSTEM	SYSTEM
	Осуществляет возврат из Бейсика в опрационную систему или в монитор	

5. ОПЕРАТОРЫ

Символ	Функция
=	Присвоение или контроль равенства
-	Вычитание или отрицательное значение
+	Сложение или об единение цепочек
*	Умножение
/	Деление
^	Возведение в степень
NOT	Логическое НЕ
AND	Логическое И
OR	Логическое ИЛИ

<	Проверка соотношения, ре-
>	зультатом являются зна-
<=	чения TRUE=-1
>=	или
<>	FALSE=0

6. ФУНКЦИИ

X,Y,I,J обозначают числовые выражения
X$ и Y$ обозначают символьные выражения

Функция	Значение	Пример
ABS(X)	Абсолютное значение X	Y=ABS(A-3)
ASC(X$)	Код первого символа X$	?ASC("K")
ATN(X)	Арктангенс X	PRINT ATN(B)
CHR$(X)	Односимвольная цепочка, значение которой в кодовой таблице равно X	PRINT CHR$(7)
COS(X)	Косинус X	A=COS(3.14)
EXP(X)	e в степени X	B=EXP(U)
FRE(0)	Об ём свободной памяти	PRINT FRE(0)
FRE("")	Об ём свободного буфера цепочки	PRINT FRE("")
INKEY$(1)	Односимвольная цепочка, считываемая с клавиатуры (цепочка пуста, если ни одна из клавиш не была нажата)	G$=INKEY$(1)
INP(X)	Байт данных с вводного порта h	0=INP(21)
INT(X)	Наибольшее целое число меньше чем X	C=INT(RND(1)*100)
LEFT$(X$,Y)	Левые Y символов X$	PRINT LEFT$(X$,6)
LEN(X$)	Длина цепочки X$	PRINT LEN(B$)
LOG(X)	Натуральный логарифм X	G=LOG(Y-3)
MID$(X$,X,Y)	средняя часть цепочки X$,Y символов начиная с X-го символа; если Y не задан, то до конца цепочки	A$=MID$(X$,5,10)
PEEK(X)	Байт в ячейке памяти по адресу x	PRINT PEEK(32700)
POS(1)	Позиция курсора на экране	IF POS(1)>20...
RIGHT$(X$,Y)	Правые Y синволов цепочки X$	C$=RIGHT$(A$,10) C$=RIGHT$(A$,10)
RND(1)	Случайное число в диапазоне 0...1	?RND(1)*100
SGN(X)	0 при X = 0 ABS(X)/X при X<>0	A=SGN(1)
SIN(X)	Синус X	A=SIN(B)
SPC(X)	Вывод X пробелов в приказе PRINT	?SPC(5),A$
SQR(X)	Квадратный корень из X	D=SQR(C)
STR$(X)	X в символьной форме	PRINT STR$(28*1)
TAB(X)	Перемещение курсора в позицию X в приказе PRINT	PRINT TAB(10);1
TAN(X)	Тангенс X	A=TAN(3.14*1)

ЕКТА Tallinn
353872.3016-11

USER(X)	Вызов функции в машинном коде по адресу X; функция возвращает байтовое значение в регистре A	A=USR(5000)
VAL(X$)	Цепочка X в цифровой форме	X=VAL("3.14")

7. СООБЩЕНИЯ ОБ ОШИБКАХ

Код ошибки	Тип ошибки
1	NEXT без FOR
2	Синтаксическая ошибка
3	RETURN без GOSUB
4	Конец данных
5	Ошибочные данные
6	Переполнение
7	Память занята
8	Неопределённая строка
9	Индекс массива вне допустимых границ
10	Повторно декларированный (DIM) массив
11	Деление на нуль
12	Недопустимый приказ в командном режиме
13	Несоответствие типов
14	Переполнение буфера цепочки
15	Чрезмерная длина цепочки
16	Чрезмерно сложное символьное выражение
17	Продолжение невозможно (в приказе CONT)
18	Неопределённая функция
19	Файл отсутствует
20	Команда прямого режима в файле
21	Ошибка в имени файла
22	Ошибка в операции ввода/вывода

СКБ Вычислительной техники Института кибернетики АН ЭССР
"ЭКТА"

ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ИНТЕЛЛЕКТУАЛЬНОГО ТЕРМИНАЛА Е5104

АССЕМБЛЕР

ASM

РУКОВОДСТВО ОПЕРАТОРА

353872.3ØØ35-2Ø

22 стр.

Таллинн 1988

1. ОБЩИЕ СВЕДЕНИЯ О ПРОГРАММЕ

ЕКТА Tallinn *ЭКТА* Таллинн КОМПОНЕНТ / КОМПЛЕКС ПО

ASM	353872.3Ø035-2Ø	Версия: 2.Ø
АССЕМБЛЕР		
ОБЪЕМ 8К байт	ОПЕРАЦ. СИСТЕМА: EKDOS	
ТРЕБУЕМОЕ ОЗУ	Программа 8К байт	Данные 2К байт
НОСИТЕЛЬ: ГМД РЕГ.No НОСИТЕЛЯ:		
НАЗНАЧЕНИЕ И МЕТОД: Программа ASM позволяет пользователю транслировать программу, написанному на языке ассемблер, в машинный код микропроцессора К58ØИК8Ø. Результатам трансдирования могут быть машинный код в виде файла типа HEX и/или распечатка программы. Транслятор производит синтаксический анализ исходного текста и анализ корректности переходов.		
ССЫЛКИ: Уэйт М., Ангермейер Дж. Операционная система CP/M, Радио и связь, 1986		
ПРОЦЕССОР / ЭВМ: КР58ØИК8ØА / Е5104		
ВНЕШНИЕ УСТР-ВА: НГМД, видеомонитор		
5.1Ø.1988	АРХ. МЛ: 3.ØØØ	ПРОГРАММИСТ: Фрейберг Ю.

2. ВЫЗОВ И ПУСК

Ассемблер считывает исходный текст программы с магнитного диска и выводит на диск транслированную программу в виде файла типа HEX.

Для вызова и пуска ассемблера используется одна из директив:

ASM <имя файла>
ASM <имя файла..,параметр>

В обоих случаях выполняется поиск на дискетте файла <имя файла>.ASM, который должен содержать исходный текст на языке ассемблера. Вторая из указанных форм вызова позволяет ввести параметр, определяющий тип выходного файла - файл распечатки или файл типа HEX. По умолчанию на искетку записывается файл типа HEX, идентификатором которого будет <имя файла>.HEX.

Допустимые значения параметра:

P - на дискетку записывается файл распечатки, содержащий исходный текст, полученный машинный код и коды ошибок
X - на дискетку записывается об ектный файл, а на экран выводится распечатка

3. ФОРМАТ ПРОГРАММЫ

Команда на языке ассемблера, приемлемая для транслятора, может содержать следующие поля, наличие или отсутствие которых зависит от конкретных обстоятельств:

<Номер строки> <метка> <код операции> <операнд> ;<коментарий>

Ввод каждой строки завершается нажатием клавиш RETURN и LF. Команды,расположенные на одной строке, разделяются восклицательным знаком "!".

Номер строки может отсутствовать. Для его обозначения допускается применение любого числа в десятичной системе.

Метка - это состоящая из букв и цифр цепочка, содержащая до 16 символов. Первым символом в ней должна быть буква. Все элементы метки значащие, за исключением символа "$", который может использоваться для повышения мнемонической ясности метки.

Код операции может представлять собой директиву ассемблера, код псевдооперации или мнемонический код машинной команды.

Операнд - это в общем случае выражение, состоящее из числовых и символьных постоянных, идентификаторов меток и знаков арифметико-логических операций.

Комментарий начинается со знака ";" и может содержать любые символы. Допускается применение символа "*" вместо ";".

Программа состоит из предложений, оформленных по приведённым выше правилам. Последним обычно является предложение типа END. Транслятор не обрабатывает предложения, которые следуют за предложением END.

4. ПРЕДСТАВЛЕНИЕ ОПЕРАНДОВ

Операндом, как указывалось, может быть выражение, состоящее из меток, констант и зарезервированных слов, связанных между собой арифметико-логическими операторами. Значение выражения вычисляется в ходе трансляции и не должно превышать двоичного формата представления в 16 разрядов.

4.1. Числовые постоянные

Числовая постоянная выражается числом с двоичным форматом в 16 разрядов. Выбранная из четырёх возможных система счисления обозначается буквой в конце числа (описателем системы):

B - двоичная (binary)
O или Q - восьмеричная (octal)
D - десятичная (decimal)
H - шестнадцатеричная (hexadecimal)

Число, представленное без описателя системы счисления, считается десятичным. Первая цифра шестнадцатеричной постоянной должна быть десятичной, для различения с идентификаторами. Например, не FFH, а OFFH.

4.2. Зарезервированные слова

Ряд зарезервированных слов имеет в поле операнда определённое содержание и числовое значение. Такими словами являются обозначения регистров процессора:

A	B	C	D	E	H	L	M	SP	PSW
7	Ø	1	2	3	4	5	6	6	6

В правом столбце дано числовое значение, соответствующее имени.

Допускается и применение мнемокодов команд в поле операнда. Числовыми значениями являются соответствующие машинные коды.

Значением символа "$" в поле операнда является адрес следующей команды.

4.3. Символьные постоянные

В качестве символьных постоянных допускается применять любые цепочки символов кода КОИ-8. Максимальная длина постоянной 64 символа. Цепочка в начале и в конце ограничивается апострофами. Если апостроф оказывается элементом постоянной, то за ним вводится ещё один дополнительный апостроф. В большинстве случаев длина символьной постоянной составляет один или два символа, за исключением команды DB. Числовым значением символьной постоянной является последовательность соответствующих кодов КОИ-8.

4.4. Арифметико-логические операторы

Описанные выше выражения операндов могут содержать следующие арифметические и логические операторы:

```
a + в     арифметическая сумма без знака
a - в     арифметическая разность без знака
  + в     унарный плюс
  - в     унарный минус
a * в     арифметическое произведение без знака
a / в     целочисленное деление без знака
a MOD в   остаток целочисленного деления а/в
  NOT в   логическая инверсия
a AND     логическое поразрядное произведение (И)
a OR      логическое сложение (ИЛИ)
a XOR     исключающее ИЛИ
a SHL в   двоичный сдвиг а влево на в разрядов
a SHR в   двоичный сдвиг а вправо на в разрядов
```

При составлении выражений допускается применение скобок. Все вычисления выполняются над значениями в 16 двоичных разрядов.

4.5. Приоритетность операторов

Для упрощения программирования операторам присвоены приоритеты. Операторы с равными приоритетами обрабатываются по порядку слева направо.

Приоритеты операторов, начиная с наивысшего, следующие:

```
* / MOD SHL SHR
- +
NOT
AND
OR XOR
```

Например, выражение

```
a MOD в * c SHL d
```

эквивалентно выражению

```
((a MOD в) * c) SHL d
```

5. ДИРЕКТИВЫ АССЕМБЛЕРА

Директивы ассемблера предназначены для присвоения числовых значений меткам в ходе трансляции, для определения областей памяти, для установления начальных адресов программ и для выполнения условной трансляции.

Допустимые директивы (псевдооперации) следующие:

```
ORG   - за начальный адрес программы принять значение операнда
END   - завершить трансляцию программы
EQU   - присвоить идентификатору начальное значение
SET   - присвоить идентификатору новое значение
IF    - начать условную трансляцию
ENDIF - завершить условную трансляцию
DB    - присвоить байту данных значение операнда
DW    - присвоить слову данных значение операнда
DS    - определить память
```

Подробнее директивы описаны ниже.

5.1. ORG

Синтаксис этой директивы:

<метка>ORG<выражени>,

где меткой может быть любой идентификатор. Выражение имеет 16-разрядное двоичное значение, его операнды должны быть предварительно определены. На определённый выражением адрес транслятор занесёт следующую транслированную команду. Программа может содержать произвольное число директив ORG. Перекрываемость не проверяется. Метке присваивается значение, определённое выражением.

5.2. END

Эта директива в программе не обязательна. Появление директивы END определяет конец программы. Следующие за директивой END предложения не транслируются. Возможны два варианта формата директивы:

<метка>END
<метка>END<выражение>

Метка может быть выбрана произвольно. При первом варианте директивы трансляция завершается, в качестве пускового адреса программы устанавливается значение ØØØØ. При втором варианте пусковой адрес совпадает со значением выражения; это значение вводится в последнюю запись об ектной программы.

5.3. EQU

Эта директива применяется для присвоения числовых значений идентификаторам, её формат:

<метка>EQU<выражение>

Метка обязательна. Метка с таким же значением не должна встречаться в начале какого-либо другого предложения программы. Значение метки определяется выражением.

5.4. SET

Формат этой директивы:

<метка>SET<выражение>

Она аналогична директиве EQU, различие лишь в том, что та же метка может применяться и в начале других предложений программы. Значение метки действительно до переопределения её другим предложением SET. Эта директива часто применяется при условной трансляции.

5.5. IF, ENDIF

Эти директивы определяют группу предложений, которые в ходе трансляции могут быть включены в состав программы при определенных условиях.

```
Применение:
IF<выражение>
<предложение 1>
<предложение 2>
 ...

<предложение n>
ENDIF
```

Когда процесс трансляции доходит до предложения IF, вычисляется значение выражения. При ненулевом значении транслируются последующие предложения и вводятся в состав программы. При нулевом значении выражения предложения, оказавшиеся перед ENDIF, не обрабатываются.

5.6. DB

Директива DB позволяет задать значения отдельных байтов какой-либо области памяти и имеет формат:

<метка>DB<e1, e2, ..., en>,

где e1, ..., en означают выражения, определяющие 8-разрядные двоичные значения (старшие разряды операндов должны содержать нули), или же символьные постоянные размером до 64 символов. При трансляции вычисляются значения выражений и записываются в последующие байты об ектной программы. В случае символьных постоянных соответствующие ячейки будут содержать коды символов, входящих в постоянную.

5.7. DW

Эта директива аналогична предыдущей, но в об ектную программу записываются 2-байтовые значения. Формат:

<МЕТКА>DW<e1, e2, ..., en>,

где e1, ..., en имеют 16-разрядные двоичные значения. Цепочки длиной более двух символов не допускаются. В об ектную программу сперва записывается младший, а затем старший байт.

5.8. DS

Эта директива предназначена для резервирования памяти без присвоения начальных значений, её формат:

<метка>DS<выражение>

При трансляции ассемблер пропускает число байтов, равное значению выражения.

6. Структура файла типа HEX

Результат работы транслятора записывается на магнитную дискетку в одном из двух видов:
- как командный файл (тип = COM),
- как файл типа HEX (тип = OBJ).

В отличие от командного файл типа HEX может быть транслирован для любого адреса памяти и может загружаться на любой адрес. Об ектный файл состоит из блоков, длину которых определяет специальный байт в заголовке блока. Максимальная возможная длина 255 байтов.

Блок об ектного файла имеет следующую структуру:

```
    ----------------------
  0 !         :          ! - признак начала блока
    ----------------------
  1 !         n          ! - число байтов данных в блоке
    ----------------------
  2 !         a          ! - загрузочный адрес блока
  3 !                    !
    ----------------------
  4 !        до          ! - данные
... !       255         ...
n+3 !      байтов        !
    ----------------------
```

Признаком конца файла является блок, указанная в заголовке длина которого равна нулю. Обычно трансляторы формируются в блоки длиной 20 байтов.

Приложение 1.

НАБОР КОМАНД МИКРОПРОЦЕССОРА К580ВМ80

Приводимые далее описания команд образуют краткую сводку иллюстративного характера, при программировании желательно использовать более подробные руководящие материалы. Параметрическая часть кода команды обозначена буквами X и Y. Соответствующие коды представлены только в двоичной форме.

Для открытия содержания мнемоники соответсвующие элементы английских наименований кодов представлены в виде прописных букв.

Имя регистра, заключенное в скобки, обозначает содержимое этого регистра.

Мнемокод	Машинный код: 16-ричный	Машинный код: Двоичный	Длина команды в байтах	Описание
Команды для изменения флажка переноса				
CMS	3F	0011 1111	1	CoMplement Carry Инверсия флажка переноса CY:=CY + 1
STC	37	0011 0111	1	SeT Carry CY:=1 Установка флажка переноса CY:=1
Команды для изменения содержания ргистров или памяти				
INR r		00XX X100	1	INcRement register Увеличение содержимого указанного регистра на единицу
DCR r		00XX X101	1	DeCRement register Уменьшение содержимого указанного регистра на единицу
INR M		00XX X100	1	INcRement Memory Увеличение содержимого указанной ячейки памяти на единицу
DCR M		00XX X101	1	DeCRement Memory Уменьшение содержимого указанной ячейки памяти на единицу

Эти команды изменяют флажки Z, S, P, AC

Безадресные однобайтовые команды.

CMA	2F	0010 1111	1	CoMplement A Инверсия содержимого регистра A

Пример: (A) = 01011011 = 5BH (до CMA)
(A) = 10100100 = A4H (после CMA)

DAA	27	0010 0111	1	Decimal Adjust A Десятичная коррекция, применяется после арифметической операции над десятичными числами

8-разрядное двоичное число в A преобразовывается в двоично-кодированное десятичное число по следующим правилам:
1) если содержимое младшего полубайта > 9 или AC = 1 (перенос в старший полубайт), то содержимое регистра A увеличивают на 6
2) если теперь в старшем полубайте число > 9 или CY = 1 (перенос из самого старшего разряда), то содержимое старшего полубайта увеличивают на 6

NOP	00	0000 0000	1	No OPeration Содержимое счётчика команд увеличивают на единицу

Команды пересылки одного байта

MOV r1, r2		01XX XYYY	1	MOVe register to register Пересылка байта из регистра r1 в r2
MOW M, r		01XX XYYY	1	MOVe register to Memory Пересылка байта из регистра r в ячейку, адрес которой в регистрах H & L
MOV r, M		01XX XYYY	1	MOVe Memory to register Аналогичная пересылка из памяти в регистр Rr
STAX B	02	0000 0010	1	STore A indirect via B Пересылка содержимого регистра A в ячейку памяти, адрес которой в регистрах BC
STAX D	12	0001 0010	1	STore A indirect via D То же, но вместо BC здесь DE
LDAX B	0A	0000 1010	1	LoaD A indirect via B Пересылка в регистр A содержимого ячейки, адрес которой в регистрах BC
LDAX D	1A	0001 1010	1	LoaD A indirect via D То же, но вместо BC здесь DE

Эти команды не меняют флажков.

Арифметико-логические команды

ADD r — 1000 0XXX — 1 — ADD r to A
Сложение содержимых регистров r и A

Пример:
Если (D) = 2EH и (A) = 6CH, то в результате ADD D :
(A) = 2EH + 6CH = 9AH, Z = CY = 0, P = S = AC = 1

ADC r — 1000 1XXX — 1 — ADd r to A with Carry
Сложение содержимых регистров r и A, и значения флажка CY

SUB r — 1001 0XXX — 1 — SUBtract r from A
Вычитание содержимого регистра r из содержимого регистра A

Вычитание осуществляется путём сложения дополнительного кода регистра r с содержимым регистра A. При отсутствии перевода (заимствования) из наиболее старшего разряда устанавливается CY = 1, в противном случае CY = 0.

Пример: (A) = 3EH, SUB A
3EH = 00111110
+
-(3EH)= 11000001 (обратный код)
+ 1 (дополнительный код)
---
(1) 00000000

Так как был перенос, то CY = 0; AC = P = Z = 1, S=0; (A) = 0

SBB r — 1001 1XXX — 1 — SuBtract r from A with Borrow
Вычитание из содержимого регистра A суммы содержимого регистра r и флажка CY

Пример: (L) = 2, (A) = 4, CY = 1, SBB L
1) 02H + C = 03H
2) дополн. код 03H : 11111101
04H = 00000100
---
(1) 00000001 = 01H

Ввиду переноса CY = 1; P = Z = S = 0, AC = 1, (A) = 01H

ANA r	1Ø1Ø ØXXX	1	ANd,register r with A Операция И над содержимыми регистров r и A, результат находится в регистре A
XRA r	1Ø1Ø 1XXX	1	eXclusive oR, register r with A Исключающее ИЛИ над содержимыми регистров r и A
ORA r	1Ø11 ØXXX	1	OR, register r with A Операция ИЛИ над содержимыми регистров r и A

ИЛИ при (r)=1 даёт "1", а при (r)=Ø не меняет значения, поэтому операция ORA часто применяется для установки групп двоичных разрядов в состояние "1".

CMP r	1Ø11 1XXX	1	CoMPare register r with A Сравнение содержимых регистров r и A; содержимое A не меняется

Сравнение выполняется путём внутреннего вычитания, поэтому при равенстве Z = 1, если не было переноса, т.е. при (r)>(a), устанавливается CY = 1, в противном случае CY = Ø.

Примечание. Во всех арифметико-логических командах можно вместо параметра r использовать пару H & L, т.е. операнд может находиться и в ячейке памяти с адресом (H & L).

Команды циклического сдвига содержимого регистра A

RLC	Ø7	0000 0111	1	Rotate A Left in Carry Сдвиг влево, содержимое наиболее старшего разряда перейдёт к флажку CY
RRC	ØF	0000 1111	1	Rotate A Right in Carry Сдвиг вправо, содержимое самого младшего разряда перейдёт в флажок CY
RaL	17	0001 0111	1	Rotate A Left through carry Аналогично RLC, но CY будет расширением регистра A, т.е. прежнее значение CY перейдёт в младший разряд регистра A
RAR	1F	0001 1111	1	Rotate A Right through carry Аналогичная операция, но вправо

Команды для оперирования с двумя байтами

PUSH rp		11XX Ø1Ø1	1	PUSH register pair on stack Пересылка в стек содержимого пары регистров rp и уменьшение указателя стека на 2; значениями gr могут быть BC, DE, HL
PUSH PSW		11XX Ø1Ø1	1	PUSH Program Status Word on stack Пересылка в стек содержимого регистра A и признаков (CY, AC, Z, S, P)
POP rp		11XX ØØ Ø1	1	POP register pair rp off stack Пересылка двух байтов из стека в пару rg (BC,DE,HL) и увеличение указателя стека на 2
POP PSW		11XX ØØ Ø1	1	POP Program Status Word off stack Восстановление содержимого регистра A и признаков путём пересылки из стека
DAD rp		ØØXX 1ØØ1	1	Double ADd Сложение чисел двойной длины: содержимого пары регистров rp (BC, DE,SP) с содержимым пары HL

Пример: (B) = 33H, (C) = 9FH, (H) = ØA1H; (L) = 7BH; DAD B

```
(BC)= 339F
+(HL)= A17B
----------
(HL)= Ø5A1 ; т.к. переноса не было, то CY = Ø
```

INX rp		ØØXX ØØ11	1	INcrement register pair Увеличение содержимого пары регистров на единицу
DCX rp		ØØXX 1Ø11	1	DeCrement register pair Уменьшение содержимого пары регистров на единицу
XCHG	EB	1110 1Ø11	1	eXCHanGe DE and HL Обмен содержимым пар регистров HL и DE
XTHL	E3	1110 ØØ11	1	eXchange Top of stack to HL Обмен содержимым пары регистров HL и послед них двух байтов стека
SPHL	F9	1111 1ØØ1	1	Load SP from HL Пересылка содержимого пары регистров HL в указатель стека

Последние пять команд не оказывают влияния на значение флажков.

Команды с непосредственным операндом

LXI rp, v		ØØxx ØØØ1 value	3	Load Immediate value to register pair Загрузка непосредственного операнда v (2байта) в пару регистров rp
MVI r, p		ØØXX X11Ø value	2	MoVe Immediate value to register Следующий за кодом команды байт загружается в регистр r; если r = HL, то в память
Признаки не изменяются.				
ADI v	C6 v	11ØØ Ø11Ø value	2	ADd Immediate value to A Следующий за кодом команды байт складывается с содержимым регистра A
ACI v	CE v	11ØØ 111Ø value	2	Add Carry and Immediate value to A То же, но к результату прибавляется значение флажка CY
SUI v	D6 v	11Ø1 Ø11Ø value	2	SUbtract Immediate from A Следующий за кодом команды байт вычитается из содержимого регистра A
SBI v	DE v	11Ø1 111Ø value	2	Subtract with Borrow Immediate value from A То же, но из результата вычитается значение флажка CY
ANI v	E6 v	1110 Ø11Ø value	2	AND, Immediate value with A Операция И над содержимым регистра A и байтом, следующим непосредственно за кодом команды
XRI v	EE v	1110 111Ø value	2	eXclusive oR, Immediate with A Исключающее ИЛИ над содержимым регистра A и байтом, следующим непосредственно за кодом команды
ORI v	F6 v	1111 Ø11Ø value	2	OR, Immediate with A ИЛИ над содержимым регистра A и байтом, следующим непосредственно за кодом команды

CPI v	FE v	1111 1110 value	2	ComPare Immediate with A Сравнение содержимого регистра A и байта, непосредственно следующего за кодом команды

Изменяются флажки AC (за исключением AN1, XR1, OR1), CY, S, Z, P.

Команды с прямой адресацией

STA ad	32 ad	0011 0010 address	3	STore A direct Содержимое регистра A записывается ячейку памяти по адресу ad
LDA ad	3A ad	0011 1010 address	3	LoaD A direct В регистр A загружается байт с адреса ad
SHLD ad	22 ad	0010 0010 address	3	Store HL Direct Содержимое регистра L записывается в ячейку памяти по адресу ad, а содержимое H - по адресу ad+1
LHLD ad	2A ad	0010 1010 address	3	Load HL Direct Загрузка в регистры HL байтов с адресов ad+1, ad

Признаки не изменяются.

Команды перехода.

HLT	76	0111 0110	1	HaLT Прекращение выполне ния программы до приёма прерывания или RST; счётчик команд устанавливается на следующую команду
PCHL	E9	1110 1001	1	Load PC from HL Переход по адресу, содержащемуся в паре регистров HL
JMP ad	C3 ad	1100 0011 address	3	JuMP unconditional Безусловный переход по адресу ad

Команды условного перехода выполняются в зависимости от значения соответствующего флажка:
1) при значении флажка, указанном в об яснении к команде, осуществляется переход по адресу, заданному в виде операнда ad
2) при ином значении - переход по адресу (PC)+3

JC ad	DA ad	1101 1010 address	3	Jump if Carry Переход по адресу ad при CY = 1
JNC ad	D2 ad	1101 0010 address	3	Jump if No Carry Переход по адресу ad при CY = 0
JZ ad	CA ad	1100 1010 address	3	Jump if Zero Переход по адресу ad при Z = 1
JNZ ad	CA ad	1100 0010 address	3	Jump if Not Zero Переход по адресу ad при Z = 0
JM ad	FA ad	1111 1010 address	3	Jump if Minus Переход по адресу ad при S = 1
JP ad	F2 ad	1111 0010 address	3	Jump if Positive Переход по адресу ad при S = 0
JPE ad	EA ad	1110 1010 address	3	Jump if Parity Even Переход по адресу ad при P = 1
JPO ad	E2 ad	1110 0010 address	3	Jump if Parity Odd Переход по адресу ad при P = 0

CALL ad	CD ad	11ØØ 11Ø1 address	3	CALL unconditional Безусловный переход к подпрограмме по адресу ad; PC (адрес возврата) загружается в стек

Команды условного перехода в подпрограмму выполняются в зависимости от значения соответствующего флажка:
1) при значении флажка, указанном в об яснении к команде, осуществляется переход по адресу ad, а значение PC загружается в стек,
2) при ином значении - переход по адресу (PC)+3

CC ad	DC ad	1101 1100 address	3	Call if Carry При CY = 1 переход по ad
CNC ad	D4 ad	1101 Ø100 address	3	Call if No Carry При CY = Ø переход по ad
CZ ad	CC ad	1100 1100 address	3	Call if Zero При Z = 1 переход по ad
CNZ ad	C4 ad	1100 Ø100 address	3	Call if Not Zero При Z = Ø переход по ad
CM ad	FC ad	1110 1100 address	3	Call if Minus При S = 1 переход по ad
CP ad	F4 ad	1111 Ø100 address	3	Call if Positive При S = Ø переход по ad
CPE ad	EC ad	1110 1100 address	3	Call if Parity Even При P = 1 переход по ad
CPO ad	E4 ad	1110 Ø100 address	3	Call if Parity Odd При P = Ø переход по ad

RET	C9	1100 1001	1	RETurn Безусловный возврат по адресу, содержащемуся в двух последних байтах стека

Команды условного возврата из подпрограммы выполняются в зависимости от значения флажка:

1) при значении флажка, указанном в об яснении к команде, осуществляется возврат по адресу, содержащемуся в двух последних байтах стека,
2) при ином значении - переход по адресу (PC)+3

RC	D8	1101 1000	1	Return if Carry Возврат при CY = 1
RNC	D0	1101 0000	1	Return if No Carry Возврат при CY = 0
RZ	C8	1100 1000	1	Return if Zero Возврат при Z = 1
RNZ	C0	1100 0000	1	Return if Not Zero Возврат при Z = 0
RM	F8	1111 1000	1	Return if Minus Возврат при S = 1
RP	F0	1111 0000	1	Return if Positive Возврат при S = 0
RPE	E8	1110 1000	1	Return if Parity Even Возврат при P = 1
RPO	E0	1110 0000	1	Return if Parity Odd Возврат при P = 0
RST n		11XX X111	1	ReSTart on level n PC записывается в стек, управление передаётся программе обработки прерывания по адресу 8n, 0<n<7, т.е. 00000000000NNN000B

Команды ввода-вывода и управления прерываниями

EI	FB	1111 1011	1	Enable Interrupts Разрешение прерываний
DI	F3	1111 0011	1	Disable Interrupts Запрещение прерываний
IN ad	DB ad	1101 1011 address	2	INput Ввод байта из порта ad в регистр A
OUT ad	D3 ad	1101 0011 address	2	OUTput Вывод байта из регистра A в порт по адресу ad

Приложение 2.

ПРИМЕРЫ ПРОГРАММИРОХАНИЯ

Пример 1. Многобайтовое сложение и вычитание

Применение команды ADC и признака переноса CY позволяет выполнить сложение чисел произвольной длины. Например:

```
    32A F8A
  + 84BA 90
  -------
    B76A 90
```

Сперва следует сложить с помощью команды ADD младшие байты, затем продолжать, применяя ADC:

```
MADD  LXI  B, FIRST;   загрузка адреса числа FIRST
      LXI  H, SEC  ;   загрузка адреса числа SEC
      XRA  A       ;   сброс CY
      MVI  E,3     ;   начальное значение счётчика
LOOP: LDAX B       ;   цикл
      ADC  M
      STAX B       ;   запись FIRST
      DCR  E
      JZ   DONE
      INX  B
      INX  H       ;   следующий байт
      JMP  LOOP
DONE:              ;   конец
FIRST:DB   90H
      DB   0BAH
      DB   84H
SEC   DB   8AH, OAFH, 32H
```

Пример 2. Умножение

Для умножения двух однобайтовых чисел нужно загрузить эти числа соответственно в регистры D и C, а для размещения результата - выделить регистры B (старший байт) и C (младший байт).

```
MULT:  MVI B,0
       MVI E,9
MULTO: MOV A,C
       RAR
       MOV C,A
       DCR E
       JZ  DONE
       MOV A,B
       JNC MULT1
       ADD D
MULT1: RAR
       MOV B,A
       JMP MULTO
DONE:
```

Пример 3. Вычитание 16-разрядных двоично-десятичных чисел

Уменьшаемое находится по адресу MINU (младший байт впереди), вычитаемое по адресу SBTRA. Результат записывается на место первого операнда.

```
DSUB: LXI D,MINU   ; DE = уменьшаемое
      LXI H,SBTRA
      MVI C,8      ; на каждом цикле вычитаются 2 десят. разряда
      STC          ; перенос, CY := 1
LOOP: MVI A,99H
      ACI 0        ; сложение, +CY
      SUB M
      XCHG
      ADD M
      DAA          ; десятичная коррекция
      MOV M,A
      XCHG
      DCR C
      JZ  DONE
      INX D
      INX H        ; адресация след. операндов
      JMP LOOP     ; следующие 2 цифры
DONE:
```

Приложение 3.

СООБЩЕНИЯ ОБ ОШИБКАХ

При обнаружении в программе ошибки транслятор обозначает её кодом ошибки, находящимся в файле распечатки перед командой. Строка, содержащая ошибку, выводится на экран.

Коды ошибок следующие:

D - "Данные": значение выражения больше допустимого
E - "Выражение": вычислить значение выражения невозможно
L - "Метка": метка недопустима в данном контексте (например, повторяющаяся метка)
O - "Переполнение": выражение чрезмерно сложно
P - "Фаза": в следующей фазе метка имеет другое значение
R - "Регистр": значение регистра не соответствует коду операции
V - "Значение": в выражении содержится недопустимый операнд

На экран выводятся также следующие сообщения об ошибках, вызывающих прекращение трансляции:

CANNOT OPEN SOURCE FILE - на дискете нет заданного файла с исходным текстом
NO DIRECTORY SPACE - в каталоге дискета нет места для файла
SOURCE FILE NAME ERROR - синтаксическая ошибка в имени исходного файла
SOURCE FILE READ ERROR - ошибка при чтении исходного файла
OUTPUT FILE READ ERROR - ошибка при записи в выходной файл
CANNOT CLOSE FILE - закрытие выходного файла невозможно

СКБ Вычислительной техники Института кибернетики АН ЭССР
"ЭКТА"

ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ИНТЕЛЛЕКТУАЛЬНОГО ТЕРМИНАЛА Е5104

ТРАНСЛЯТОР ЯЗЫКА ПАСКАЛЬ

MTPLUS

РУКОВОДСТВО ОПЕРАТОРА

353872.30021-561

14 стр.

Таллинн 1988

СОДЕРЖАНИЕ

1. ОБЩИЕ СВЕДЕНИЯ О ПРОГРАММЕ

ЕКТА Tallinn *ЭКТА* Таллинн КОМПОНЕНТ / КОМПЛЕКС ПО

MTPLUS	353872.30021-561	Версия: 5.6.1
ТРАНСЛЯТОР ЯЗЫКА ПАСКАЛЬ		
ОБ'ЕМ 120К байт	ОПЕРАЦ. СИСТЕМА: CP/M, EKDOS	
ТРЕБУЕМОЕ ОЗУ: 36К байт		
НОСИТЕЛЬ: ГМД РЕГ. No НОСИТЕЛЯ:		
НАЗНАЧЕНИЕ И МЕТОД: Программа создает из файла типа PAS файл типа ERL. Вызов транслятора: MTPLUS <имя файла> Для создания об'ектной программы из транслированных модулей следует использовать редактор связей LINKMT		
ССЫЛКИ: Перминов О.Н. Программирование на языке Паскаль. М.: Радио и связь, 1988, 224 с. К.Йенсен, Н.Вирт. Паскаль. Руководство для пользователя и описание языка. М.: Финансы и статистика, 1982, 152 с.		
ПРОЦЕССОР / ЭВМ: КР580ВМ80А / Е5104		
ВНЕШНИЕ УСТР-ВА: Видеомонитор, НГМД		
НЕОБХ. ПРОГРАММЫ И ФАЙЛЫ: MTPLUS.000 (16K) MTPLUS.001 (12K) MTPLUS.002 (8K) MTPLUS.003 (8K)	MTPLUS.COM (12K) MTPLUS.004 (20K) MTPLUS.005 (12K) MTPLUS.006 (8K) MTERRS.TXT LINKMT.COM (12K)	
20.09.1987	АРХ. МЛ: 3.000	ПРОГРАММИСТ: ГЛАДИН М.

2. РАБОТА С ТРАНСЛЯТОРОМ

2.1. Общие указания

Транслятор имеет имя MTPLUS.COM и исползует четыре оверлейных файла. Входные файлы могут быть размещены на любом диске и иметь произвольные имена. Входной файл может иметь любое расширение, но если указано, что он имеет в качестве расширения пробел и он с этим расширением не найден, то транслятор будет искать файл с расширением .SRC, а затем будет искать файл с расширением .PAS. Если никакой подходящий файл не будет найден, то будет выдано сообщение об ошибке: 'Unable to open input file' (нельзя открыть входной файл). Транслятор создаёт перемещаемый файл <имя> .ERL, который с помощью редактора связей должен быть связан с пакетом периода выполнения.

2.2. Вызов транслятора

Командная строка для вызова транслятора:

MTPLUS <имя файла> $ <факультативные параметры, см.2.6>

Загрузка и запуск транслятора подтверждаются сообщением
Pascal / MT + 5.6
После этого следуют сообщения отдельных фаз трансляции.

2.3. Фаза 0

В ходе этой фазы после синтаксического посмотра каждых 16 строк исходного текста на экран выводится символ '+'.
Отображаемый блок сообщений фазы 0 имеет вид:

```
Code Gen:                    (Генерация кода)
+++++++
Source lines:                (Число строк исходного текста)
```

2.4. Фаза 1

В начале фазы 1 выводится используемое пространство памяти. Это - число байт памяти (в десятичном представлении) перед генерацией таблицы символов. Приблизительно 3K байт из пространства для таблицы используется на заранее определенные идентификаторы. При нахождении процедуры или функции на экран выдается символ '£'. При завершении фазы 1 на экран выводится в десятичной форме число свободных байтов в памяти.

Сообщения фазы 1:

```
Phase 1                          (Фаза 1)
Available Memory: nnnnn          (Имеющаяся память)
User Table Space: nnnnn          (Область таблицы пользователя)
£££££££
Remaining Memory: nnnnn          (Свободная память)
```

2.5. Фаза 2

В этой фазе генерируется об´ектный код. При входе в тело каждой процедуры имя этой процедуры выводится на экран. Транслятор устанавливает для обратываемого модуля относительные адреса. Список абсолютных адресов выдается при обработке модулей редактором связей. По завершении трансляции выводятся число транслированных строк исходного текста (в десятичной форме), число обнаруженных ошибок, число байт сгенерированного об´ектного кода (в десятичной форме) и число байт, зарезервированных для данных (в десятичной форме).

Сообщения фазы 2:

```
Phase 2                          (Фаза 2)
8080
<процедура 1>                    (Список процедур)
   ...
<процедура n>

Lines:  <число строк>
Errors: <число ошибок>
Code:   <число байт об´ектного кода>
Data:   <число байт для данных>
Compilation Completed            (Трансляция закончена)
```

2.6. Факультативные управляющие параметры командной строки

Параметры располагаются в командной строке (см. 2.2), за именем исходного файла, в вид строки, перед которой стоит символ '$', и представляют собой отдельные буквы, за которыми могут следовать нуль или более символов-параметров. Строка параметров продолжается от $ до конца данный строки, причем пробелы игнорируются. Параметры перечислены ниже, звёздочкой (*) обозначены значения по умолчанию.

Параметр	Значение
Rd	Поместить файл .ERL на дисковод 'd:'. * Файл .ERL помещается на тот же самый диск, на котором находится исходный файл.
Nd	Файл .OVL с номером n(n=1...4) на дисководе 'd:'. * Файлы .OVL находится на диске, используемом по умолчанию.
Pd	Поместить файл .PRN на дисковод 'd:'. * Файл .PRN не генерируется.
X	Сгенерировать расширенный перемещаемый файл, включающий записи дизассемблера. * Генерируется нерасширенный файл.
D	Сгенерировать в об'ектном коде отладочную информацию и записать файл .PSY на дисковод, задаваемый ключом R. * В об'ектном файле не генерируется никакой отладочной информации и никакого файла .PSY не записывается.
Ed	Файл MTERRS.TXT на 'd:'. * Файл MTERRS.TXT находится на диске, используемом по умолчанию.
Td	Поместить файл PASTEMP.TOK на дисковод 'd:'. * Файл PASTEMP.TOK помещается на диск, используемый по умолчанию.
Q	Покой (Qulet), подавляет выдачу на экран любых сообщений, которые не являются необходимыми. * Сообщения выводятся.
C	Продолжать в случае ошибки; умолчание заключается в паузе и диалоге с оператором по поводу каждой ошибки, по одной за раз. * Транслятор останавливается и запрашивает при каждой ошибке.
A	Автоматический вызов редактора связей в конце трансляции и редактирование связей файла .ERL только со стандартной библиотекой. Файл .COM будет помещен на тот же диск, что и файл .ERL. * Транслятор не сцеплен автоматически с редактором связей.

Продолжение

Параметры	Значение
B	Использовать для вещественных чисел двоично-десятичное представление (BCD), а не плавающую форму. * По умолчанию вещественные числа представляются с плавающей запятой.
Z	Генерировать оптимизированный код микропроцессора Z80 - только для версии 8080 (Z). * Генерируется код только для версии 8080 (Z80).

Значения d:

d=X	Вывод на экран.
d=P	Вывод на печать
d= ,A,...,O	Вывод на дисковод

3. ИСХОДНЫЙ ТЕКСТ ПРОГРАММЫ. ЛЕКСИЧЕСКИЕ ЭЛЕМЕНТЫ ЯЗЫКА

3.1.Ключи управления транслятором в исходном тексте

Ключи транслятора		По умолчанию
$E+/-	Управление генерацией точек входа.	$E+
$S+/-	Управление рекурсивным или статическим размещением переменных.	$S-
$I<имя>	Включает во входной поток другой исходный файл (например, $I XXX.LIB).	
$R+/-	Управление кодами контроля за диапазонами.	$R-
$T+/-		$T-
$W+/-	Управление строгой проверкой типов и генерацией предостерегающих сообщений.	$W-
$X+/-	Управление кодами контроля за исключениями.	$X-
$P	Ввод смещения формы (FF) (переход на новую страницу) в файл .PRN .	
$L+/-	Управление выдачей листинга исходного текста.	$L+
SCn	Использование команд RSTn для операций с данными типа REAL (по умолчанию используется команда CALL).	
$Kn	Предусмотрен для удаления встроенных процедур с целью экономии места в таблице символов (n=0...7).	

Ключи $Kn используются для удаления из таблицы символов определений встроенных процедур, которые не являются необходимыми, с тем чтобы оставить больше места для символов пользователя. Значение (0...6) используется для упрвления различными группами процедур. Эти значения могут быть использованы в любой комбинации, но чтобы они учитывались, их необходимо помещать перед словом PROGRAM или MODUIE. Значения n выделяют сле-дующие группы:

Группа	Удаляемые процедуры
0	ROUND, TRUNC, EXP, LN, ARCTAN, SQRT, COS, SIN
1	COPY, INSERT, POS, DELETE, LENGTH, CONCAT
2	GNB, WNB, CLOSEDEL, OPENX, BLOCKREAD, BLOCKWRITE
3	CLOSE, OPEN, PURGE, CHAIN, CREATE
4	WRD, HI, LO, SWAR, ADDR, SIZEOF, INLINE, EXIT, PACK, UNPACK
5	IORESULT, PAGE, NEW, DISPOSE
6	SUCC, PRED, EOF, EOLN
7	TSTBIT, CLRBIT, SETBIT, SHR, SHL

Пользователь должен понимать, что при этом удаляются только имена из таблицы заранее определённых символов, с тем чтобы оставить больше места для символов пользователя. Сами эти процедуры включаются в программу пользователя редактором связей только в том случае, если они используются в данной программе.

3.2.Сводка операторов

Оператор	Операция	Тип операндов	Тип результата
:=	Присваивание	Любой кроме файла	-
+(унарный)	Эквивалентность	Целочисл./Вещ.	Ц/В
-(унарный)	Инверсия	Целочисл./Вещ.	Ц/В
+	Сложение	Целочисл./Вещ./Указ.	Ц/В/У
-	Вычитание	Целочисл./Вещ./Указ.	Ц/В/У
*	Умножение	Целочисл./Вещ.	
dir	Целочисл.деление	Целочисл.	Целочисл.
/	Вещ.деление	Целочисл./Вещ.	Веществ.
mod	Остаток	Целочисл.	Целочисл.
=	Равенство	Скаляр / Строка /	
<>	Неравенство	Множество / Указатель / Запись	
<	Меньше	Скаляр / Строка	
>	Больше	Скаляр / Строка	
<=	Меньше или равно	Скаляр / Строка	
>=	Больше или равно	Скаляр / Строка	
IN	Принадлежность к множеству	I: скаляр, II: множество	
NOT	НЕ	Логический	Логический
OR	ИЛИ	Логический	Логический
AND	И	Логический	Логический
- ?	НЕ		
!	ИЛИ	Целочисл./Указатель	Ц/У
&	И	Целочисл./Указатель	Ц/У
+	Об'единение	Множество	Множество
-	Разность множеств	Множество	Множество
*	Пересечение	Множество	Множество

3.3. Зарезервированные слова

MOD, NIL, IN, OR, AND, NOT, IF, THEN, ELSE, CASE, OF, REPEAT, UNTIL, WHILE, DO, FOR, TO, DOWNTO, BEGIN, END, WITH, GOTO, CONST, VAR, TYPE, ARRAY, RECORD, SET, FILE, FUNKTION, PROCEDURE, LABEL, PACKED, PROGRAM, ABSOLUTE, EXTERNAL

3.4 Встроенные процедуры и функции

Символьные массивы:

```
PROCEDURE FILLCHAR  (DESTINATION, LENGTH, CHARACTER);
PROCEDURE MOVELEFT  (SOURCE, DESTINATION, NUM_BYTES);
PROCEDURE MOVERIGHT (SOURCE, DESTINATION, NUM_BYTES);
```

Бит, байт:

```
PROCEDURE CLRBIT (BASIC_VAR, BIT_NUM);
FUNCTION HI      (BASIC_VAR)           : INTEGER;
FUNCTION LO      (BASIC_VAR)           : INTEGER;
PROCEDURE SETBIT (BASIC_VAR, BIT_NUM);
FUNCTION SHL     (BASIC_VAR, NUM)      : INTEGER;
FUNCTION SHR     (BASIC_VAR, NUM)      : INTEGER;
FUNCTION SWAP    (BASIC_VAR)           : INTEGER;
FUNCTION TSTBIT  (BASIC_VAR, BIT_NUM)  : BOOLEAN;
```

Строки:

```
FUNCTION CONCAT  (SOURCE1, SOURCE2,..., SOURCEn) : STRING;
FUNCTION COPY    (SOURCE, LOCATION, NUM_BYTES)   : STRING;
PROCEDURE DELETE (TARGET, INDEX, SIZE);
PROCEDURE INSERT (SOURCE, DESTINATION, INDEX);
PROCEDURE LENGTH (STRING)                        : INTEGER;
FUNCTION POS     (PATTERN, SOURCE)               : INTEGER;
```

Пересылка:

```
PROCEDURE PACK   (ARRAY, INTEGER, ARRAY);
PROCEDURE UNPACK (ARRAY, ARRAY, INTEGER);
FUNKTION CHR(SC) : SCAR;
FUNKTION ODD(SC) : BOOLEAN;
FUNKTION ORD(SC) : INTEGER;
FUNKTION ROUND   (REAL) : INTEGER;
FUNKTION TRUNC   (REAL) : INTEGER;
FUNKTION WRD(SC) : WORD
```

Арифметика:

```
FUNKTION ABS    (INTEGER or REAL) : INTEGER or REAL;
FUNKTION ARCTAN (INTEGER or REAL) : INTEGER or REAL;
FUNKTION COS    (INTEGER or REAL) : INTEGER or REAL;
FUNKTION EXP    (INTEGER or REAL) : INTEGER or REAL;
FUNKTION LN     (INTEGER or REAL) : INTEGER or REAL;
FUNKTION SIN    (INTEGER or REAL) : INTEGER or REAL;
FUNKTION SQR    (INTEGER or REAL) : INTEGER or REAL;
FUNKTION SQRT   (INTEGER or REAL) : INTEGER or REAL;
```

Динамическое распределение:

```
PROCEDURE DISPOSE  (P,  case variant selectors );
FUNKTION MAXAVAIL  : INTEGER;
FUNKTION MEMAVAIL  : INTEGER;
PROCEDURE NEW      (P,  case variant selectors );
```

Файлы:

```
PROCEDURE ASSIGN      (FILE, NAME);
PROCEDURE BLOCKREAD   (FILE, BUF, IOR, NUMBYTES, RELBLK);
PROCEDURE BLOCKWRITE  (FILE, BUF, IOR, NUMBYTES, RELBLN);
PROCEDURE CLOSE       (FILE, RESULT);
PROCEDURE CLOSEDEL    (FILE, RESULT);
FUNKTION  EOF          (FILE) ;
FUNKTION  EOLN         (FILE) ;
PROCEDURE GET         (FILE);
FUNKTION  GNB         (FILE)  : CHAR;
PROCEDURE IORESULT             : INTEGER;
PROCEDURE OPEN        (FILE, TITLE, RESULT);
PROCEDURE PAGE        ;
PROCEDURE PUT         (FILE);
PROCEDURE PURGE       (FILE);
PROCEDURE READ(LN)    ( FILE, VARIABLE(S));
PROCEDURE READHEX     (FILE, VAR, SIZE);
PROCEDURE RESET       (FILE);
PROCEDURE REWRITE     (FILE);
PROCEDURE SEEKREAD    (FILE, RECORD_NUMBER);
PROCEDURE SEEKWRITE   (FILE, RECORD_NUMBER);
FUNKTION  WNB         (FILE, CHAR) : BOOLEAN;
PROCEDURE WRITE(LN)   ( FILE, DATA);
PROCEDURE WRITEHEX    (FILE, EXPRESSION, SIZE);
```

Разное:

```
FUNKTION  QCMD    : PTR_TO_STRING;
FUNKTION  ADDR    (VARIABLE REFERENCE) : INTEGER;
PROCEDURE EXIT    ;
FUNKTION  PRED    (<scalar except real>) : <same as parameter>;
FUNKTION  RIM85   : BYTE;
PROCEDURE SIM85   (VAL : BYTE);
FUNKTION  SIZEOF  (VARIABLE OR TYPE NAME) : INTEGER;
FUNKTION  SUCC    (<scalar except real>) : <same as parameter>;
PROCEDURE WAIT    (PORTNUM, MASK, POLARITY);
```

4. СООБЩЕНИЯ ОБ ОШИБКИХ

1: Ошибка в простом типе.
2: Ожидается идентификатор.
3: Ожидается 'PROGRAM'.
4: Ожидается ')'.
5: Ожидается ':'.
6: Незаконный символ (возможно, что в предыдущей строке не хватает ';').
7: Ошибка в списке параметров.
8: Ожидается 'OF'.
9: Ожидается '('.
10: Ошибка в типе.
11: Ожидается '['.
12: Ожидается ']'.
13: Ожидается 'END'.
14: Ожидается ';'(возможно в предыдущей строке).
15: Ожидается целое.
16: Ожидается '='.
17: Ожидается 'BEGIN'.
18: Ошибка в разделе описаний.
19: Ошибка в < списке полей>.
20: Ожидается '.'.
21: Ожидается '+'.
50: Ошибка в константе.
51: Ожидается ':='.
52: Ожидается 'THEN'.
53: Ожидается 'UNTIL'.
54: Ожидается 'DO'.
55: Ожидается 'TO' или 'DOWNTO' в операторе FOR.
56: Ожидается 'IF'.
57: Ожидается 'FILE'.
58: Ошибка в <множителе> (неправильное выражение).
59: Ошибка в переменной.
99: Ожидается MODEND.
101: Идентификатор описан дважды.
102: Нижняя граница превосходит верхнюю границу.
103: Идентификатор не принадлежит соответствующему классу.
104: Неописанный идентификатор.
105: Знак запрещен.
106: Ожидается число.
107: Несовместимые отрезки типов.
108: Файлы здесь не допускаются.
109: Тип не должен быть вещественным.
110: Тип <поля признака> должен быть скалярным или отрезком типа.
111: Несовместимо с разделом <поле признака>.
112: Тип индекса не может быть вещественным.
113: Тип индекса должен быть скалярным или отрезком типа.
114: Тип базы не должен быть вещественным.
115: Тип базы должен быть скалярным или отрезком типа.
116: Ошибка в типе параметра стандартной процедуры.
117: Неопределенная ссылка вперед.

118: Ссылающийся вперед идентификатор типа в описании переменных.
119: Переопределенные параметры не подходят для описанной позднее процедуры.
120: Тип результата функции должен быть скалярным, отрезком типа или указателем.
121: Передача файлов в качестве параметров по значению не допускается.
122: Тип результата описываемых позднее функций не может быть переопределен
123: В описании функции отсутствует тип результата.
125: Ошибка в типе параметра стандартной процедуры.
126: Число параметров не согласуется с описанием.
127: Незаконная подстановка параметра.
128: Тип результата не согласуется с описанием.
129: Конфликт типов операндов.
130: Выражение не имеет тип множества.
131: Допускается только проверка на равенство.
133: Сравнение файлов не допускается.
134: Незаконный тип операнда (операндов).
135: Тип операнда должен быть булевым.
136: Тип элементов множеств должен быть скалярным или отрезком типа.
137: Типы элементов множеств должны быть совместимым.
138: Тип переменной не является массивом.
139: Тип индекса несовместим с описанием.
140: Тип переменной не является записью.
141: Тип переменной должен быть файлом или указателем.
142: Незаконный параметр.
143: Незаконный тип параметра цикла.
144: Незаконный тип выражения.
145: Конфликт типов.
146: Присваивание файлов не допускается.
147: Тип метки несовместим с селекторным выражением.
148: Границы отрезка типа должны быть скалярными.
149: Тип индекса должен быть целым.
150: Присваивание значения стандартной функции не допускается.
151: Присваивание значения функции, являющейся формальным параметром не допускается.
152: В этой записи нет такого поля.
153: Ошибка типа при чтении.
154: Фактический параметр должен быть переменной.
155: Параметр цикла не может быть формальным параметром или нелокальной переменной.
156: Многократно определенная метка выбора.
157: Слишком много альтернатив в операторе выбора.
158: Отсутствует описание соответсвующего варианта.
159: Вещественные или строки в качестве полей признаков не допускаются.
160: Предыдущее описание было не вперед.
161: Снова описано вперед.
162: Размер параметра должен быть постоянным.
163: Отсутствует вариант в описании.
164: Подстоновка стандартной процедуры или функции не допускается.
165: Многократно определенная метка.
166: Многократно описанная метка.

167: Неописанная метка.
168: Неопределенная метка.
169: Ошибка в базе множества.
170: Ожидается параметр, передаваемый по значению.
171: Стандартный файл был переописан.
172: Неописанный внешний файл.
174: Ожидается функция или процедура Паскаля.
183: Внешнее описание на этом уровне вложения не допускается.
187: Неудачная попытка открыть библиотеку.
191: Никаких приватных файлов.
193: Для этой операции не хватает места.
194: Комментарий должен появиться в вершине программы.
201: Ошибка в вещественном числе - ожидается цифра.
202: Строчная константа не должна превосходить строки исходного текста.
203: Целая константа выходит за границы диапазона.
250: Слишком много областей действия вложенных идентификаторов.
251: Слишком много вложенных процедур или функций.
253: Процедура слишком длинная.
259: Выражение слишком сложное.
397: Слишком много операторов FOR или WITH в процедуре.
400: Недопустимый символ в тексте.
401: Неожиданный конец входа.
402: Ошибка при записи программного файла, не хватает места.
403: Ошибка при чтении включаемого файла.
404: Ошибка при записи файла листинга, не хватает места.
405: Недопустимое обращение к отдельно транслируемой процедуре.
406: Незаконный включаемый файл.
407: Переполнение таблицы символов.
497: Ошибка при закрытии программного файла.

СКБ вычислительной техники Института кибернетики АН ЭССР
"ЭКТА"

ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ИНТЕЛЛЕКТУАЛЬНОГО ТЕРМИНАЛА Е5104

РЕДАКТОР СВЯЗЕЙ ДЛЯ ЯЗЫКА ПАСКАЛЬ
LINKMT

РУКОВОДСТВО ОПЕРАТОРА

353872.3Ø022-55

5 стр.

Таллинн 1988

1.ОБЩИЕ СВЕДЕНИЯ О ПРОГРАММЕ

ЕКТА Tallinn *ЭКТА* Таллин КОМПОНЕНТ / КОМПЛЕКС ПО

LINKMT	353872.30022-55	Версия: 5.5

РЕДАКТОР СВЯЗЕЙ ДЛЯ ЯЗЫКА ПАСКАЛЬ

ОБ'ЕМ 12К байт | ОПЕРАЦ. СИСТЕМА: CP/M, EKDOS

РЕБУЕМОЕ ОЗУ : 12 К байт

НОСИТЕЛЬ: ГМД
РЕГ. но НОСИТЕЛЯ:

НАЗНАЧЕНИЕ И МЕТОД:
Создание об'ектной программы из транслированных при помощи MTPLUS модулей.

ОГРАНИЧЕНИЯ:
Макс.число связываемых файлов - 32

ПРОЦЕССОР / ЭВМ: КР 580 ВМ 80А / Е5104

ВНЕШНИЕ УСТР-ВА:
Видеомонитор НГМД

НЕОБХ. ПРОГРАММЫ И ФАЙЛЫ:

PASLIB.ERL - ввод/вывод, сравнения, арифметика и пр.
FPREALS.ERL - арифметика с плавающей запятой
TRANCEND.ERL - подпрограммы для SIN, COS, ARCTAN, SQRT, LN, EXP, SQR

20.09.88	АРХ. №:	ПРОГРАММИСТ: Гладин М.

2. ВЫЗОВ

Для использования редактора связей LINKMT нужно напечатать его имя, за которым через один пробел должно следовать имя ведущей программы и разделяемые запятыми имена модулей, которые должны быть связаны. Если пользователь не укажет, куда направить выходной файл, поместив имя выходного файла, за которым следует знак равенства, перед именем ведущей программы, то выход будет направлен на тот же самый диск, где находится ведущая программа.

Структура командной строки:

LINKMT <ведущая программа>, <файлы .ERL и библиотеки /S > ,
PASLIB <ключи>

Редактор связей воспринимает в командной строке (или входном командом файле) до 32 имен файлов, которые должны быть связаны.

3. СВОДКА КЛЮЧЕЙ КОМАНДНОЙ СТРОКИ

/S - Рассматривать предшествующее имя как библиотеку, извлекая только требуемые процедуры.

/L - Выдавать список модулей во время их редактирования.

/M - Выдать список всех точек входа в табличной форме.

/E - В дополнение к другим точкам входа выдать точки входа процедур, имена которых начинаются с символов $, ? и .

/P:nnnn - Переместить об´ектный код в nnnnH.

/D:nnnn - Переместить область данных в nnnnH.

/W - Записать совместимый с символическим диалоговым отладчиком (SID) файл .SYM записывается на тот же самый диск, что и файл .COM .

H:nnnn - Записать выход как файл .HEX с nnnnH в качестве начального адреса для шестнадцатеричного формата. Этот ключ совершенно независим от ключа P; если используется этот ключ, то не создаётся никакого файла .COM .

/F - Рассматривать предшествующее имя файла как файл .CMD , содержащий имена файлов (по одному на строку).

4. СООБЩЕНИЯ ОБ ОШИБКАХ

Сообщения выводятся на нормальном английском языком и не требуют раз´яснений.

Например:

Unable to open input file - невозможно открытие входного файла
Duplicate symbol - дублирующийся символ

СКБ Вычислительной техники Института кибернетики АН ЭССР
"ЭКТА"

ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ИНТЕЛЛЕКТУАЛЬНОГО ТЕРМИНАЛА Е5104

ИНТЕРПРЕТАТОР ЯЗЫКА BASIC

В80 5.21

РУКОВОДСТВО ОПЕРАТОРА

353872.3Ø037-52

17 стр.

Таллинн 1988

1.ОБЩИЕ СВЕДЕНИЯ О ПРОГРАММЕ

ЕКТА Tallinn *ЭКТА* Таллин КОМПОНЕНТ / КОМПЛЕКС ПО

В8Ø	353872.3Ø037-52	Версия: 5.21
ИНТЕРПРЕТАТОР ЯЗЫКА BASIC		
ОБЪЕМ 24К байт	ОПЕРАЦ. СИСТЕМА: EKDOS	
ТРЕБУЕМОЕ ОЗУ	Программа 24К байт	Данные байт
НОСИТЕЛЬ: ГМД РЕГ. но НОСИТЕЛЯ:		
НАЗНАЧЕНИЕ И МЕТОД: Программа В8Ø позволяет пользователю написать программы на языке BASIC, корректировать запускать их в интерактивном режиме.Простота конструкции языка делает его доступным для широкого круга пользоват лей, а большое количество операторов и функций (числовых строковых) позволяет создавать на языке Бейсик достаточ сложные и эффективные программы.		
ССЫЛКИ: Уэйт М., Ангермейер Дж. Операционная система СР/М, Радио и связь, 1986		
ПРОЦЕССОР / ЭВМ: К58ØИК8Ø / Е51Ø4		
ВНЕШНИЕ УСТР-ВА: НГМД, видеомонитор		
5.1Ø.1988	АРХ. МЛ: 3.ØØØ	ПРОГРАММИСТ: Мяртин К.

2. ОПИСАНИЕ ЯЗЫКА BASIC.

В описании языка Бейсик принята следующая система обозначений:

1) элемент, заключённый в квадратные скобки [ ], считается необязательным и может отсутствовать;
2) три точки (...) означают, что элемент, стоящий перед ними, может быть неоднократно повторен;
3) / означает альтернативу;
4) элемент в угловых скобках со строчными буквами заполняется пользователем.

2.1. Правила записи программ.

Каждая строка начинается с метки и заканчивается признаком конца строки - управляющим символом "RERURN". Метка от оператора должна отделяться по крайней мере одним пробелом. Одна строка языка Бейсик может занимать несколько строк терминала. В этом случае признаком продолжения служит управляющий символ "LE".

2.2. Символы.

Набор символов языка Бейсик состоит из цифр от 1 до 9, прописных букв латинского алфавита от A до Z и следующих специальных знаков:

!,",#,$,%,',(,),*,+,,,-,.,/,:,;,<,=,>,?,^,_,(пробел).

Символ пробела - это отсутствие какого-либо графического изображения в данной позиции. Символ пробела, кроме специально оговоренных случаев, не является значащим и поэтому может свободно использоваться для улучшения наглядности программ.

Специальные управляющие символы:

CTRL+A - Переход в режим редактирования
CTRL+C - Прерывает выполнение программы транслятор переходит в командный режим, на экране появляется сообщение "READY" ("ГОТОВНОСТЬ")
CTRL+H - Стирает последний введённый символ
CTRL+I - Табулятор, останов через каждые 8 позиции
CTRL+O - Останов/восстановление операции вывода программ
CTRL+S - Останов выполнения программы
CTRL+Q - Восстановление выполнения программы
CTRL+U - Стирает текущую вводимую строку
CTRL+X - Стирает текущую вводимую строку
RETURN - Завершает вводимую строку
: - Разделяет находящиеся на одной строке предложения
? - Равнозначен директиве PRINT
& - Обозначает символьную переменную

&o or & - Префикс для восьмеричной константы
&H - Префикс для шестнадцатеричной константы

3. ЭЛЕМЕНТЫ ЯЗЫКА.

3.1. Типы данных.

В языке Бейсик имеются два типа данных - числовые и строковые. Числовый тип подразделяется на три вида: целые, вещественные с одинарной точностью и вещественные с двойной точностью.

& - строковые
% - целые
! - одинарная точность
- двойная точность

Символьная цепочка: от Ø до 255 символов
Целое число: от -32768 до 32767
Вещественное число: от -1,7E+38 до 1,7E+38

3.2. Задание формата в операторе PRINT USING.

! Указывает, что из данной строки выводится только первый символ

N пробелов Указывает, что из данной строки выводятся только первые N+2 символа. Если выводится строка длиннее поля вывода, то лишние символы игнорируются, если же строка короче поля вывода, то она дополняется справа пробелами

& Задаёт переменную длину поля вывода. В этом случае строка выводится полностью

"Знак числа" используется для задания позиции каждой цифры. Эти позиции всегда заполняются. Если выводимое число имеет меньше цифр, чем задано "знаками числа", то оно дополняется слева пробелами

. "Десятичная точка" может стоять в поле вывода любой позиции. Числа при необходимости округляются

+ Знак "плюс" показывает, что перед числом должен явно стоять его знак (+ или -)

- Если описываемый формат оканчивается знаком "минус", то знак "минус" будет выводиться в конце отрицательных чисел

** Две "звёздочки", стоящие в начале формата, указывают, что вместо ведущих пробелов должны стоять "звёздочки". Они одновременно задают две дополнительные позиции для цифр

$$ Два "знака денежной единицы" вызывают печать одного знака денежной единицы перед выводимым числом

$$$ Такое сочетание знаков в начале формата производит комбинированное действие: ведущие пробелы заполняются "звёздочками" и перед числом выводится "знак денежной единицы"

, Запятая, стоящая в формате слева от десятичной точки, вызывает вывод запятой через каждые три пози-

ции, считая влево от десятичной точки. Запятая, стоящая в конце формата выводится как знак после вывода числа

^^^^ Четыре "стрелки вверх" могут располагаться после позиции цифр. Они указывают на вывод числа в экспоненциальном формате. При этом резервируется место для вывода E+xx

_ Знак подчеркивания в формате выводит следующий за ним символ в графическом изображении

% Если выводимое число не помещается в заданном формате, то перед ним ставится знак %

3.3. Операции.

= Присваивание или тест на равно
- Вычитание
+ Сложение
* Умножение
/ Деление (результат вещественный)
^ Возведение в степень

Арифметические операции сложения и вычитания могут быть одноместными и двухместными.

Целочисленное деление обозначается символом \. Операнды округляются до целых чисел, после чего производится деление. От частного отбрасывается дробная часть.

Оператор MOD - это вычисление остатка от деления одного целого числа на другое.

В Бейсике имеются следующие логические операции:

NOT отрицание
AND и
OR или
XOR исключающие или
IMP импликация
EQU эквивалентность

Отношения состоят из двух выражений одного типа, раделённых знаком операции отношения. TRUE=-1, FALSE=Ø.

Знаки операций отношения:

< меньше
> больше
= равно
<> не равно
<= не больше
>= не меньше.

3.4. Операторы и выражения.

AUTO AUTO[<line number>[,<increment>]]

Предназначен для перехода в режим генерирования номеров строк автоматически после каждого RETURN

CALL CALL<variable name>[<argument list>]

Оператор CALL позволяет обращаться к подпрограммам, написанным на ассемблере

CHAIN[MERGE]<filename>[,[line number exp>]
[,ALL][,DELETE<range>]]

Оператор CHAIN позволяет из данной программы загрузить в память другую программу и передать ей управление

CLEAR[,[<expression1>][,<expression2>]]

Оператор CLEAR предназначен для обнуления всех числовых и символьных переменных и по необходимости для определения адреса конца памяти и выделенной для стэка памяти

CLOSE[[£]<file number>[,[£]<file number...>]]

Этот оператор завершает операции ввода/вывода с указанными файлами. Если CLOSE не содержит аргументов, то он закрывает все открытые файлы. При выводе информации в файл с последовательным доступом CLOSE записывает содержимое буфера в файл и закрывает его

COMMON<list of varibles>

Оператор предназначен для передачи программе переменных вызываемых при помощи CHAIN

CONT

Оператор предназначен для продолжения выполнения программы, прерванной при помощи STOP или END

DATA<list of constants>

Записывает в память постоянные, чтение которых возможно после приказа READ

DEF FN<name>[(<parameter list>)]=<function definition>

Определяет арифметическую функцию одного аргумента

DEF INT/SNG/DBL/STR
DEF<type><range(s)of letters>
где <type> is INT.SNG,DBL или STR

Определяет тип переменной как целое, вещественное одинарной точности, вещественное двойной точности и строковое

DEF USR[<digit>]=<integer expression>

Предназначен для задания начального адреса программы, написанной на ассемблере

DELETE[<line number>][-<line number>]

Предназначен для уничтожения строк программы

DIM<list of subscripted variables>

Выделяет память для массивов, устанавливает максимальные значения индексов массивов

EDIT<line number>

Переход в режим редактирования на данной строке

END

Завершает выполнение программы, закрывает все файлы и возвращает управление на командный уровень

ERASE<list of array variables>

Оператор ERASE предназначен для удаления массивов из программы

ERR и ERL переменные

Переменная ERR содержит код ошибки, переменная ERL - номер строки программы, при выполнении которой появилась ошибка.

ERROR <integer expression>

Оператором ERROR 1) симулируется ошибка Бейсика и 2) задаётся код ошибки

FIELD[#]<file number>,<field with>AS<string variable>...

С помощью этого оператора в буфере файла с прямым доступом выделяются области для переменных

FOR...NEXT

```
FOR<variable>=xTOy[STEP Z]
.
.
.
NEXT[variable>][,<variable>...]
```

Действие операторов FOR...NEXT позволяет выполнить последовательность операторов циклически с проверкой на окончание цикла.

```
GET[#]<file number>[,<record number>]
```

Этот оператор читает в буфер одну запись из открытого файла с данным номером. Если номер записи опущен, то читается запись со следующим номером

GOSUB...RETURN

```
GOSUB<line number>
.
.
.
RETURN
```

Предназначены для передачи управления в подпрограмму и возврата управления

```
GOTO<line number>
```

Оператор предназначен для безусловной передачи управления на заданную строку (прерывается нормальный порядок выполнения).

IF...THEN[...ELSE] и IF...GOTO

```
IF<expression>THEN<statement(s)>I<line number>
    [ELSE<statement(s)>I<line number>
IF<expression>GOTO<line number>
    [ELSE<statement(s)>I<line number>
```

Операторы предназначены для принятия решения на последовательность выполнения операторов программы в зависимости от значения выражения.

```
INPUT[;][<"promt string">;]<list of variables>
```

С помощью этого оператора можно читать данные с терминала во время выполнения программы.

```
INPUT#<file number>,<variable list>
```

С помощью этого оператора можно читать данные из файла с последовательным доступом

```
KILL<file name>
```

Оператор уничтожает файл на диске

```
LET<variable>=<expression>
```

Оператор присваивает операнду значение вычисленного выражения

```
LINE INPUT[;][<"pronit string">;]<string variable>
```

С помощью этого оператора можно читать целую строку (до 254 символов) без использовании разделителей.

```
LINE INPUT#<file number>,<string variable>
```

С помощью этого оператора можно читать целую строку (до 254 символов) без разделителей из последовательного файла

```
LIST[<line number>]
```

Оператор отображает строки программы (в памяти) с первой строки (по умолчанию) или со строки, заданной номером

```
LLIST[<line number>[-[line number>]]]
```

Оператор предназначен для вывода на принтер части или всей программы (в памяти).

```
LOAD<file name>[,R]
```

Оператор предназначен для чтения программы с диска в память

```
LPRINT и LPRINT USING
   LPRINT[<list of expressions>]
   LPRINT USING<string exp>;<list of expressions>
```

С помощью операторов производится вывод информации на принтер

```
LSET и RSET
   LSET<string variable>=<string expression>
   RSET<string variable>=<string expression>
```

Эти операторы используются для пересылки данных в буфер файла с прямым доступом (для подготовки к выполнению оператора PUT)

```
MERGE<filename>
```

С помощью оператора текст заданного файла на диске добавляется к программе в памяти.

MID&(<string exp1>,n[,m])=<string exp2>

Оператор MID& позволяет заменить часть одной строки на часть другой строки.

NAME <old filename>AS<new filename>

Оператор предназначен для переименования файла на диске

NEW

Оператор NEW уничтожает текущую программу в памяти удаляет все переменные

ON ERROR GO <line number>

Оператор определяет на какую строку программы передаётся управление в случае ошибки.

ON<expression>GOTO<list of line numbers>

В зависимости от значения выражения управление передаётся на одну из нескольких строк с указанными номерами.

ON<expression>GOSUB<list of line numbers>

В зависимости от значения выражения управление передаётся на одну из нескольких строк с указанными номерами

OPEN <mode>,[#]<file number>,<file name>,[reclen>]

Оператор открывает файл для операций ввода/вывода. При выполнении оператора OPEN для заданного файла назначается буфер ввода/вывода, а также определяется вид операции ввода/вывода

OPTION BASE N

Определяет минимальное значение индекса при индексации массивов (Ø или 1)

OUT I,J

Этот оператор выводит число J на порт устройства с номером I (I и J - целочисленные выражения в диапазоне Ø...255)

POKE I,J

Этот оператор записывает в ячейку памяти с адресом I целое число J. I в диапазоне от Ø до 65536, J в диапазоне от Ø до 255

PRINT[<list of expressions>]

Оператор печати PRINT передаёт значения результатов вычислений и пояснительных текстов из внутренней памяти на внешнее устройство. Вместо слова PRINT может быть использован вопросительный знак (?).

PRINT USING <string exp>;<list of expressions>

Этот оператор используется для вывода строк или чисел в специальном формате (см. также 3.2.)

PRINT#<filename>,[USING<string exp>;]<list of exps>

Оператор предназначен для вывода информации в последовательный дисковый файл.

PUT[#]<file number>[,<record number>]

Этот оператор помещает одну запись из буфера в файл с прямым доступом. Если номер записи опущен, то она получает очередной номер (после предыдущего PUT).

RANDOMIZE [<expression>]

Оператор выдаёт последовательность псевдослучайных чисел в качестве значений для функций RND. Моделируется равномерное распределение чисел от Ø до 1.

READ <list of variables>

Оператор присваивает переменным и элементам массива значения, объявленные в операторах DATA.

REM<remark>

Оператор даёт возможность комментировать программу

RENUM[[<new number>][,[<old number>][,<increment>]]]

Оператором перенумеруется программа

RESTORE [<line number>]

Оператор восстановления RESTORE устанавливает указатель блока данных на заданный элемент и тем самым позволяет осуществить повторную выборку данных

```
RESUME                (1)
RESUME Ø              (2)
RESUME NEXT           (3)
RESUME<line number>   (4)
```

Оператор определяет точку продолжения выполнения программы после работы процедуры обработки ошибки (с оператора, где была ошибка (1) и (2), со следующего оператора после ошибки (3) или начиная с определённой строки (4)).

```
RUN[<line number>]
```

Передаёт управление программе в памяти

```
RUN<filename>[,R]
```

Загружает программу в память и передаёт ей управление

```
SAVE<filename>[,A|,P]
```

Предназначен для записи программы на диск.

```
STOP
```

Оператор STOP останавливает выполнение программы и передаёт управление на командный уровень.

```
SWAP<variable>,<variable>
```

Этот оператор осуществляет обмен значений переменных. Допускаются любые типы переменных, но обе переменные должны быть одного типа.

```
TRON/TROFF
```

Эти операторы используются для трассировки участков программы.

```
WAIT<port number>,I[,J]
```

Этот оператор подавляет выполнение программы до тех пор, пока не изменится статус заданного порта.

```
WHILE...WEND

WHILE<expression>
.
.
[loop statements]
.
.
WEND
```

Пока значение выражения есть "истина", выполняется список операторов, заключённых между WHILE и WEND.

```
WIDTH [LPRINT]<integer expression>
```

Этот оператор задаёт длину строки вывода на консоль или на устройство печати. Если LPRINT опущен, то длина устанавливается для консоли, а иначе - для устройства печати.

```
WRITE[<list of expressions>]
```

Этот оператор выводит данные на консоль

```
WRITE#<file number>,<list of expressions>
```

Этот оператор выводит данные в последовательный файл.

3.4.1. Подкоманды режима редактирования (EDIT Mode).

a	Восстановление исходной строки и начало редактирования строки с начала
[i]C<ch>	Изменение i следующих символов на ch (текст)
[i]D	Уничтожение i следующих символов начиная с текущей позиции
E	Конец редактирования и сохранение текста без последующей части строки
H<string><ESCAPE>	Уничтожение последующей части текста и добавление string
I<string><ESCAPE>	Добавление string на текущей позиции
[i]K<ch>	Исключение всех символов до i-того повторения <ch>
L	Выводит последующую часть строки и переходит на начало строки
Q	Выход с редактирования без изменения строки
[i]S<ch>	Поиск в строке до i-го повторения символа <ch>
X<string><ESCAPE>	Переход в конец строки и добавить string
<RUBOUT>	Переход через символ; в режиме добавления уничтожение символа
<RETURN>	Конец редактирования и сохранение изменений
<SPACE>	Переход на следующий символ

3.5. Встроенные элементы (функции).

```
ABS(x)
```

Возвращает абсолютное значение выражения x.

```
ASC(x$)
```

Возвращает числовую величину, равную значению кода (КОИ7).

ATN(X)

Возвращает значение арктангенса в радианах.

CDBL(X)

Преобразует X в число с удвоенной точностью.

CHR$(I)

Возвращает строку, содержащую один элемент. Этот элемент есть код КОИ-7, соответсвующий числу 1.

CINT(X)

Преобразует число X в целое с округлением дробной части.

COS(X)

Возвращает косинус числа X, выраженного в радианах

CSNG(X)

Преобразует число x в число одинарной точностью.

CVI,CVS,CVD

CVI(2-х байтовая строка);
CVS(4-х байтовая строка);
CVD(8-х байтовая строка);

CVI преобразует двухбайтовую строку в целое число;
CVS преобразует четырёхбайтовую строку в число с одинарной точностью;
CVD преобразует восьмибайтовую строку в число с двойной точностью.

Эти функции нужны при чтении данных из файла с произвольным доступом, т.к. там данные хранятся в виде строк.

EOF(NF)

Возвращает значение "истина"(-1), если достигнут конец файла с последовательным доступом с номером NF.

EXP(X)

Возвращает значение "E" в степени X.

FIX(X)

Возвращает усеченную целую часть числа X. Дробная часть отбрасывается.

FRE(Ø)

Возвращает число оставшихся свободных байтов. Аргумент функции FRE - фиктивный.

HEX$(X)

Возвращает строку, которая представляет шестнадцатеричное значение десятичного аргумента. Перед вычислением HEX$(X) число X округляется.

INP(I)

Возвращает байт, введённый с порта I.

INPUT$(X[,[#]Y])

Возвращает строку, содержащую x символов, введённых с консоли или из файла с номером Y.

INSTR([I,]X$,Y$)

Эта функция ищет первое вхождение образца Y$ в строке X$ и возвращает номер позиции вхождения.

INT(X)

Возвращает наибольшее целое, не превосходящее X.

LEFT$(X$,I)

Возвращает строку, содержащую I первых символов строки X$, начиная с первого.

LEN(X$)

Возвращает число символов строки X$. Считаются также непечатные символы и пробелы.

LOC(NF)

Для файла с произвольным доступом с номером NF эта функция возвращает номер следующей записи, используемый в операторах GET или PUT, если этот номер не будет в этих операторах задан явно. Для файла с последовательным доступом LOC возвращает число секторов, прочтённых или записанных с момента открытия файла.

LOG(X)

Возвращает натуральный логарифм числа X.

LPOS(X)

Возвращает позицию печатающего узла принтера внутри буфера принтера.

MID&(X&,I[,J])

Возвращает строку, содержащую J символов строки X$, начиная с 1-того символа. I и J должен быть в диапазоне от Ø до 255.

MKI$,MKS$,MKD$

MKI$(<integer expression>)
MKS$(<single pecision expression>)
MKD$(<double precision expression>)

Эти функции преобразуют числа в строки. Любое число, помещаемое в буфер файла с произвольным доступом операторами LSET и KSET, должно быть преобразовано в строку.

OST&(X)

Возвращает строку, содержащую восьмеричное представление десятичного аргумента. Число X перед преобразованием округляется в целое.

PEEK(I)

Возвращает текущую позицию курсора. Самая левая позиция-1. I - фиктивный параметр.

RIGHT$(X$,I)

Возвращает строку, содержащую I правых символов строки X$. Если I=LEN(X$), то возвращается вся строка X$. Если I=Ø, то возвращается пустая строка.

RND[(X)]

Возвращает случайное число в диапазоне между Ø и 1.

SGN(X)

Эта функция "знак числа" 1, если X>Ø, если X=Ø и -1 если X<Ø.

SIN(X)

Возвращает значение SIN(X) от аргумента в радианах.

SPACE$(X)

Возвращает строку, содержащую X пробелов.

SPC(I)

Функция выводит на консоль I пробелов

SQR(X)

Возвращает значение квадратного корня из числа X.

STR$(X)

Возвращает строковое представление числа X.

STRING$(I,J) или
STRING$(I,X$)

Возвращает строку, содержащую I символов - кодов числа J или строку символов, эквивалентных первому символу строки X$.

TAB(I)

Функция табуляции. Она выставляет на консоли пробелы до позиции с номером I. Если текущая позиция вывода превышает номер I, то позиция I вставляется на следующем строке.

TAN(X)

Возвращает TG(X) от аргумента в радианах.

VAL(X$)

Возвращает числовое значение строки X,$. Если первый символ строки отличен от:+,-,& или цифры, то VAL(X$)=0

VARPTR(V) или
VARPTR(NF)

где V - имя переменной или NF - имя файла. При первом формате возвращается адрес первого байта области памяти, отведённой для указанной переменной, при втором формате - адрес буфера ввода/вывода незначенного заданному файлу.

СКБ Вычислительной техники Института кибернетики АН ЭССР
"ЭКТА"

ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ИНТЕЛЛЕКТУАЛЬНОГО ТЕРМИНАЛА Е5104

ТРАНСЛЯТОР ЯЗЫКА BASIC

BASCOM

РУКОВОДСТВО ОПЕРАТОРА

353872.30043-00

7 стр.

Таллинн 1988

1.ОБЩИЕ СВЕДЕНИЯ О ПРОГРАММЕ

ЕКТА Tallinn *ЭКТА* Таллин КОМПОНЕНТ / КОМПЛЕКС ПО

BASCOM	353872.30043-00	Версия:
ТРАНСЛЯТОР ЯЗЫКА BASIC		
ОБЪЕМ 32K байт	ОПЕРАЦ. СИСТЕМА: EKDOS	
ТРЕБУЕМОЕ ОЗУ	Программа 31K байт	Данные байт
НОСИТЕЛЬ: ГМД РЕГ. но НОСИТЕЛЯ:		
НАЗНАЧЕНИЕ И МЕТОД: Компилятор языка BASIC компилирует программы, написанные на языке BASIC. Из исходной программы компилятор создаёт перемещаемый модуль. Впоследствии этот модуль может быть скомпонован с другими модулями, загружен в память и выполнен.		
ССЫЛКИ: Уэйт М., Ангермейер Дж. Операционная система CP/M, Радио и связь, 1986		
ПРОЦЕССОР / ЭВМ: К580ИК80 / Е5104		
ВНЕШНИЕ УСТР-ВА: НГМД, видеомонитор		
НЕОБХ. ПРОГРАММЫ И ФАЙЛЫ: LB0.COM, BASLIB.REL		
5.10.1988	АРХ. НЛ: 3.000	ПРОГРАММИСТ: Мяртин К.

2. ВЫЗОВ И ПУСК.

Обращение к компилятору можно осуществлять двумя способами:

```
A>BASCOM K или
A>BASCOM
*K
```
где K - командная строка.

При первом способе после выполнения компиляции управление передаётся процессору консольных команд(ПКК). При втором способе после загрузки компилятор выводит на экран запрос - знак * и ждёт ввода командной строки. Выход из компилятора в ПКК производится после успешного завершения компиляции или после ввода управляющего символа CTRL+C

Формат командной строки:

```
[["имя 1",["имя 2]]="имя 3"
```
где

"имя 1" - это имя файла (перед которым может быть задано имя знака), в который будет записан перемещаемый модуль. Тип этого файла всегда REL;

"имя 2" - это устройство вывода листинга. Таким устройством может быть консоль (TTY:), или файл на диске. В последнем случае задаётся полное имя файла;

"имя 3" - это имя файла, содержащего исходную программу.

2.1. Ключи компиляции.

Процессом компиляции можно управлять с помощью специальных паранетров, задаваемых в командной строке - ключей компиляции.

Ключ	Действие
/N	Отменить вывод листинга сгенерированных кодов в символической нотации. Если этот ключ отсутствует, то в листинге после каждого предложения будут стоять об'ектные коды, сгенерированные компилятором
/E	Этот ключ указывает компилятору, что в программе, написанной на языке Бейсик, есть оператор ON ERROR GOTO. Чтобы обработать оператор ON ERROR GOTO ком- пилятор генерирует дополнительные коды для операто- ров GOSUB и RETURN. Кроме того, в файл типа REL включаются дополнительно номера строк. Поэтому ключ /E надо использовать только в том случае, когда есть ON

еррор гото. Если оператор RESUME имеет форму, отличную от RESUME "номер строки", то следует ис- пользовать ключ /X (см.ниже)

/X Этот ключ сообщает компилятору, что в программе есть один или несколько операторов RESUME, RESUME NEXT или RESUME 0. Ключ /X включает также действие ключа /E

/D Этот ключ вызывает генерацию специальных отладочных кодов во время выполнения программы. Ключ следует задавать, если в программе используются операторы TRON/TROFF. Тогда компилятор создаёт коды для следующих проверок в период выполнения программы:

1) Арифметическое переполнение. Все арифметические операции с фиксированной и плавающей точкой проверяются на переполнение и потерю точности;
2) Границы массивов. Производится проверка значений индексов во всех ссылках на массивы. Индексы должны находиться в пределах размерности массива, заданной в операторе DIM;
3) В об'ектный файл включаются номера строк, поэтому ошибки, выявленные в процессе прохождения программы, выводятся с указанием номера предложения;
4) Для всех операторов RETURN проверяется наличие соответствующих операторов GOSUB или ON GOSUB

/C Этот ключ указывает компилятору, что не следует производить проверку правильности нумерации предложений. Если задан ключ /C, то допускается произвольная нумерация предложений или отсутствие номеров строк вообще. При этом следует учитывать, что номера должны присутствовать у тех предложений, на которые ссылаются операторы IF, GOTO и GOSUB.

Примечание.

Правильная нумерация предполагает расположение номеров строк в порядке возрастания. Повторяющиеся номе- ра будут восприниматься как ошибка.

3. СООБЩЕНИЯ, ВЫДАВАЕМЫЕ КОМПИЛЯТОРОМ.

При программировании могут возникнуть ошибки двух типов: синтаксические ошибки и ошибки в логике программы. Синтаксические ошибки выявляются компилятором в период компиляции программы и сразу же выводятся на консоль. Логические ошибки могут выявиться только в процессе выполнения программы. Некоторые из этих ошибок как, например, переполнение, деление на 0 и т.п. диагностируются подпрограммами, прикомпонованными к основной программе редактором связей L80. Сообщения о таких ошибках выводятся во время выполнения программы.

В процессе компиляции при обнаружении ошибок компилятор выводит на консоль или в файл типа PRN двухбуквенный код ошибки и знак ^, указывающий на то место в строке, где произошла ошибка.

СЕРЬЕЗНЫЕ ОШИБКИ

Код ошибки	Возможный источник ошибки
SN	Ошибка в синтаксисе
	Недопустимое имя в аргументе
	Неверное присвоение
	Неверный формат константы
	Неверное задание символа в операторе DEFXXX
	Неверный синтаксис в выражении
	Неверный список аргументов в функции
	Неверное имя функции
	Неверный формальный параметр в функции
	Неверный разделитель
	Неверный формат номера предложения
	Неверный синтаксис подпрограммы
	Неверное имя функции
	Неверный формальный параметр функции
	Неправильный разделитель
	Неверный формат номера оператора
	Неверный синтаксис в подпрограмме
	Недопустимый символ
	Пропущен знак равенства
	Пропущен GOTO или GOSUB
	Пропущена запятая
	Пропущен INPUT
	Нет номера строки
	Пропущена левая скобка
	Пропущен знак минус
	Пропущен операнд в выражении
	Пропущен AS
	Пропущена правая скобка
	Пропущены точка с запятой
	Слишком длинное имя
	Ожидается GOTO или GOSUB
	Требуется строковое выражение
	Здесь должна быть строковая переменная

	Неверный синтаксис
	Здесь должна быть переменная
	Неверное число аргументов
	Формальный параметр должен быть уникальным
	Допускается только одна переменная
	Пропущено TO
	Неверная индексная переменная для цикла FOR
	Пропущено THEN
	Пропущено BASE
	Неверное имя подпрограммы
OM	Превышен размер памяти
	Слишком большой массив
	Переполнение области данных
	Слишком много операторов
	Переполнение памяти, отведённой программе
SQ	Неверная последовательность
	Одинаковые номера строк
	Нарушена последовательность номеров строк
TM	Несоответствие типов
	Несоответствие типов данных
	Переменные должны быть однотипными
TC	Слишком сложное выражение
	Слишком много аргументов в вызове функции
	Слишком много размерностей в массиве
	Слишком много переменных для INPUT
BS	Неверная индексация
	Неверное значение размерности
	Неверное число индексов
LL	Слишком длинная строка
UC	Неопознаваемая команда
	Оператор неопознан
	Оператор пока не реализован
OV	Математическое переполнение
/Ø	Деление на ноль
DD	Массив с этим именем уже описан
FN	Ошибка FOR/NEXT
	Индексная переменная цикла FOR уже задействована
	FOR без NEXT
	NEXT без FOR
FD	Функция с этим именем уже определена
UF	Функция не определена

WE	Ошибка WHILE/WEND WHILE без WEND WEND без WHILE
/E	Пропущен ключ /E
/x	Пропущен ключ /X

Предупреждения

ND	Не была задана размерность массива
SI	Оператор игнорируется

В конце работы компилятор выводит следующие сообщения:

NNNNNN FATAL ERROR(S)	где	NNNNNN - количество серьёзных ошибок;
MMMMMM WARNING(S)	где	MMMMMM - количество предупреждений;
KKKKK BYTES FREE	где	KKKKK - количество оставшейся свободной памяти

В процессе работы компилятор может выдать также следующие сообщения:

INTERNAL ERROR MEMORY OVERFLOW	внутренняя ошибка, переполнение памяти;
COMMAND ERROR	неверная команда;
FILE NOT FOUND	файл не найден;
CAN'T ENTER FILE	нельзя создать файл.

СКБ Вычислительной техники Института кибернетики АН ЭССР
"ЭКТА"

ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ИНТЕЛЛЕКТУАЛЬНОГО ТЕРМИНАЛА Е5104

ТРАНСЛЯТОР ЯЗЫКА ФОРТРАН

F80

РУКОВОДСТВО ОПЕРАТОРА

353872.30038-34

30 стр.

Таллинн 1988

1.ОБЩИЕ СВЕДЕНИЯ О ПРОГРАММЕ

ЕКТА Tallinn *ЭКТА* Таллин КОМПОНЕНТ / КОМПЛЕКС ПО

F8Ø	353872.3ØØ38-34	Версия: 3.44
ТРАНСЛЯТОР ЯЗЫКА ФОРТРАН		
ОБЪЕМ 28К байт	ОПЕРАЦ. СИСТЕМА: EKDOS	
ТРЕБУЕМОЕ ОЗУ	Программа 26К байт	Данные байт
НОСИТЕЛЬ: ГМД РЕГ. но НОСИТЕЛЯ:		
НАЗНАЧЕНИЕ И МЕТОД: Программа F8Ø позволяет пользователю транслировать программу, написанному на языке фортран, в перемещаемый модуль, готовый к обработке компоновщиком. Транслятор осуществляет синтактический разбор программы, анализ корректности переходов, построение программы в машинных командах, а в случае необходимости, в виде псев-доассемблерного текста.		
ССЫЛКИ: Уэйт М., Ангермейер Дж. Операционная система СР/М, Радио и связь, 1986		
ПРОЦЕССОР / ЭВМ: К58ØИК8Ø / Е51Ø4		
ВНЕШНИЕ УСТР-ВА: НГМД, видеомонитор		
НЕОБХ. ПРОГРАММЫ И ФАЙЛЫ: L8Ø.COM, FORLIB,REL		
5.1Ø.1988	АРХ. МЛ: 3.ØØØ	ПРОГРАММИСТ: Мяртин К.

2. ВЫЗОВ И ПУСК

2.1. Условия применения программы компилятора.

При работе с компилятором языка ФОРТРАН на диске необходимо иметь следующие файлы:

F8Ø.COM 26КБ > компилятор языка ФОРТРАН

FORLIB.REL 27КБ. > библиотека периода выполнения.

2.2. Обращение к компилятору, входные и выходные данные.

Обращение к компилятору ФОРТРАН может осуществляться двумя способами:

1) вводом с консоли командной строки:
   F8Ø

После чего загружается компилятор ФОРТРАН. Идентифицируя себя выводом сообщения о своей версии и размере, и с новой строки выводит запрос:

*

Оператор должен ввести командную строку компилятора;

2) вводом с консоли:
   F8Ø "командная строка"

Входными данными для компилятора служит файл на диске, содержащий текст исходной программы на языке ФОРТРАН.
Выходными данными компилятора являются:

- файл листинг исходной программы;
- файл перемещаемых машинных кодов.

2.3. Синтаксис командной строки.

Командная строка компилятора имеет такой вид:

OBJ, LST = SFILE [/KEY] [/KEY]...[/KEY]

где
OBJ - имя файла перемещаемых машинных кодов;
LST - имя файла листинга программы;
SFILE - имя файла с программой на ФОРТРАНЕ;
/KEY - ключи управления работой компилятора.

Имя файла должно быть записано в виде принятом в EKDOS, а именно:

DEV: FILENAME.EXT

где:

DEV - имя устройства;
FILENAME - имя файла;
EXT - тип файла.

По умолчанию предполагаются типы для

файла перемещаемых машинных кодов - REL,
листинга программы - PRN,
исходной программы - FOR.

В командной строке могут быть опущены вместе или отдельно OBJECT и LIST. Если не нужен ни листинг программы, ни файл перемещаемых машинных кодов, их имена можно опустить, оставив разделяющую их запятую.

2.4. Ключи компилятора.

В командной строке можно задать ключи, управляющие режимами работы компилятора. Каждый ключ представляет собой дробную черту, за которой следует буква. Перечисление ключей и описания их действия приведено в приложении 1.

По умолчанию, во время работы компилятора действуют следующие ключи: /H, /R, /L.

2.5. Сообщения об ошибках периода компиляции.

Компилятор обнаруживает в компилируемой программе ошибки двух типов: предупреждения и недопустимые ошибки.
При обнаружении ошибки, дающей компилятору возможность продолжить компиляцию, выдаётся предупреждение, и компиляция продолжается со следующей распознаваемой синтаксической единицы. Если же ошибка в программе грубая, то компилятор выдаёт сообщение о недопустимой ошибке, игнорирует оставшуюся часть оператора, включая и строки продолжения. Предупреждениям предшествует символ (?), а недопустимым ошибкам вопросительный знак (?), затем печатается номер строки программы и сообщение об ошибке.

Например:

? LINE 176 ILLEGAL STATEMENT FOLLOWING LOGICAL IF

% LINE 21 ILLEGAL DO TERMINATION

Если при компиляции программы компилятором были обнаружены ошибки любого типа, программа должна быть исправлена и скомпилирована снова.

Список выдаваемых компилятором предупреждений и сообщений о недопустимых ошибок периода компиляции приведен в приложении 2.

2.6. Сообщения об ошибках периода выполнения.

Во время программы возможно возникновение различных ошибок. Сообщения о них выводятся на консоль в виде:

```
**XX** AT ADDRESS HHHH
```

где

XX - двухбуквенный код ошибки;
AT ADDRESS HHHH - по адресу HHHH (в шестнадцатиричном виде.)

Список сообщений об ошибках периода выполнения и их пояснения приведены в приложении 3.

При возникновении грубой ошибки выполнение программы завершается и управление передаётся операционной системе.
Если ошибка - предупреждение, то выполнение программы продолжается. Однако после предупреждений общим числом 20, выполнение программы завершается.

2.7. Компоновка программы, написанной на языке ФОРТРАН.

Для компоновки скомпилированной програмы используется редактор связей L80, который преобразует файл перемещаемых машинных кодов в файл типа .COM, готовый к выполнению.

3. ОПИСАНИЕ ЯЗЫКА ФОРТРАН.

3.1. Введение.

Язык ФОРТРАН представляет собой реализацию ФОРТРАНА, принятого в качестве стандарта со следующими расширениями:

1) при задании операторов "STOP C" и "PAUSE C" может быть не только десятичным числом, но и любой цепочкой, содержащей до 6 символов в коде, принятом в ОС 18ØØ;
2) в операторах READ и WRITE реализована возможность передачи управления при ошибке (ERR=) и концу файла (END=);
3) в операторной функции можно использовать переменные с индексами;
4) во всех случаях возможно использование вместо целых констант шестнадцатеричных констант;
5) допустимо использование в выражениях литералов вместо целых констант;
6) не накладываются ограничения на количество строк продолжения;
7) в выражениях и операторах присваивания возможно использование данных разных типов.

Ниже перечислены ограничения языка ФОРТРАН по сравнению со стандартным:

1) не реализованы данные типа COMPLEX;
2) операторы спецификации должны располагаться в таком порядке;
   - операторы PROGRAM, SUBROUTINE, FUNCTION, BLOC DATA;
   - операторы спецификации типа EXTERNAL и DIMENSION;
   - операторы COMMON;
   - оператор EQUIVALENCE;
   - операторы DATA;
   - операторные функции;
3) знак равенства в операторе присваивания и первая запятая в операторе DO не могут быть помещены в строке продолжения;
4) целые переменные типа INTEGER*4 не могут быть использованы при вычислении индексов, а также в качестве переменной цикла DO и его параметров.

3.2. Компоненты программы на языке ФОРТРАН.

Программа на языке ФОРТРАН состоит из операторов и необязательных комментариев. Операторы об единяются в логические блоки, называемые программными единицами. Программная единица является последовательностью операторов, определяющей некоторую процедуру вычисления. Программная единица может быть или программой, или подпрограммой, или подпрограммой-функцией, или подпрограммой BLOCK DATA. Выполняемая программа состоит из одной основной программы и любого количества подпрограмм и подпрограмм-функций. Компилятор различает следующие типа программных единиц:

1) программу - последовательность операторов, начинающуюся с PROGRAM и заканчивающуюся END;

2) подпрограмму, начинающуюся с SUBROUTINE и кончающуюся END;
3) подпрограмму-функцию, начинающуюся с FUNCTION и кончающуюся END.

Компилятор языка ФОРТРАН может обрабатывать файл, содержащий любое количество различных программных единиц.

Все операторы делятся на два класса: выполняемые и невыполняемые операторы. Невыполняемые операторы описывают размещение и характеристики данных, а также содержат информацию, необходимую для редактирования и преобразования данных.

Оператор состоит из одной или нескольких строк. Строкой называется последовательность символов до 8Ø знаков длиной, заканчивающаяся управлением символов CR. Если размер оператора превышает допустимую длину строки, он может быть продолжён на следующей строке или на нескольких последующих строках, называемых строками продолжения. Строка продолжения идентифицируется наличием символа продолжения в шестой позиции этой строки.

Оператор может иметь метку, по которой другие операторы получат возможность обращаться к нему либо за информацией, либо для передачи ему управления. Метка оператора является целым числом, записанным в первых пяти позициях первой строки оператора.

Наличие комментария не влияет на выполнение программы; он служит вспомогательной информацией для программиста.

3.3. Набор символов языка ФОРТРАН.

При записи программ на языке ФОРТРАН используются следующие символы:

- прописные буквы латинского алфавита от "A" до "Z";
- цифры от "Ø" до "9";
- следующие специальные знаки:

	пробел;
=	равно;
+	плюс;
-	минус;
*	звёздочка;
/	косая (дробная) черта;
(	круглая скобка левая;
)	круглая скобка правая;
,	запятая;
.	точка;
'	апостроф;
$	знак денежной единицы.

В набор символов языка ФОРТРАН не входят, но используются компилятором следующие управляющие символы:

SI	вход;
SO	выход;
HT	горизонтальная табуляция;
LF	перевод строки;
CR	возврат каретки.

В нечисловых литералах и комментариях могут использоваться все графические символы в частности русские буквы.

3.4. Использование бланка программирования на языке ФОРТРАН.

Метка оператора (или номер оператора) состоит из одной или нескольких (не более пяти) десятичных цифр, расположенных в первых пяти позициях начальной строки оператора. Пробелы и левые нули игнорируются. Запрещается использование метки оператора, состоящей только из нулей.

Любой оператор, на который существует ссылка из другого оператора, должен иметь метку. Никакие два оператора в программной единице не должны иметь одинаковых меток.

В позиции 1 может быть помещена буква "c". Это означает, что данная строка ивляется комментарием. Компилятор выводит содержимое этой строки на листинг исходной программы, после чего эта строка игнорируется.

Позиция 6 строки ФОРТРАНА зарезэрвирована для признака продолжения. Любой символ, за исключением нуля или пробела, помещённый в эту позицию, распознается как признак продолжения. Символы, начинающиеся с позиции 7 строки продолжения, рассматриваются как следующие за последним символом предыдущей строки.

4. ЭЛЕМЕНТЫ ДАННЫХ ЯЗЫКА ФОРТРАН.

Основными элементами операторов языка ФОРТРАН являются:

Константа - представляет фиксированную величину, которая не меняется в ходе выполнения программы;

Переменная - символическое имя (идентификатор), представляющее некоторую величину, хранящуюся в памяти;

Массив - группа однотипных величин, расположенных в памяти последовательного друг за другом. Отдельные величины, составляющие массив, называются элементами массива. Можно обращаться не только к массиву, но и к любому элементу массива. Для обращения к массиву используется идентификатор;

Выражение - может быть простой константой, переменной, элементом массива или обращением к функции. Выражение также может быть комбинацией вышеперечисленных элементов и элементов, называемых операциями. Операции определяют вид вычислений, которые надо выполнить над элементами для того, чтобы получить значение выражения;

Обращение к функции
- идентификатор функции, за которым следует список аргументов (параметров). Появление в программе обращения к функции вызывает выполнение вычислений над параметрами согласно определению функции.

Идентификатор
- это последовательность букв и цифр, обязательно начинающаяся с буквы.

4.1. Типы данных.

Каждый основной элемент представляет данные одного из нескольких различных типов. Тип данных элемента может определяться самой конструкцией элемента, может задаваться по умолчанию, либо явным образом.

Типы данных, допустимые в ФОРТРАНЕ, следующие:

Целый	- целое число;
Вещественный	- число с десятичной точкой, это может быть целое число, десятичная дробь или комбинация этих значений;
С двойной точностью	- аналогично вещественным, но с повышенной точностью представления числа;
Логический	- данное, принимающее логическое значение "истина" или "ложь".

Важной характеристикой каждого типа данных является объем памяти, требующийся для хранения величины этого типа. Различная внутри основных типов выражаются либо точностью представления величины, либо диапазоном допустимых значений.

Текстовые константы и литералы не имеют типа данных. Они принимают тип данных в зависимости от того контекста, в котором они встречаются.

Для оптимального использования памяти и повышения производительности компилятора допускается использование дополнительных типов данных.

Распределение памяти по типам данных следующее:

INTEGER (INTEGER*2)=	- занимает 2 байта памяти;
INTEGER*1	- отводится 1 байт памяти;
REAL (REAL*4)	- отводится 4 байта памяти;
DOUBLE PRECISION (REAL*8)	- занимает 8 байтов памяти;
LOGICAL	- отводится 1-байтавая область памяти, которая может содержать логические личины "истина" или "ложь", один буквенно-цифровой символ или целые величины от минус 128 до +127;
INTEREG*4	- занимает 4 байта памяи;
LOGICAL*2	- отводится 2-байтовая область памяти. Эквивалентно INTEGER*2;
BYTE	- занимает один байт.

4.2. Константы.

Константа представляет неизменяемую величину: логическую, числовое значение или последовательность символов.

4.3. Переменные.

Переменная - это величина, обращение к которой осущест-вляется с помощью её идентификатора. Идентификатор переменной может превышать по размеру шесть знаков. Однако используются только первые шесть знаков, которые должны быть уникальны среди имён переменных данной программной единицы. Значение переменной может быть изменено присваиванием ей нового значения. Переменные классифицируются по типу данных точно так же, как и константы. Тип переменной определяет, какой тип данных она представляет. Если переменной присваиваются данные другого типа, то они преобразуются к типу переменной. Тип переменной может быть установлен операторами описания типа, либо неявным заданием типа.

4.3.1. Спецификация типа данных.

Операторы явного описания типа определяют, что переменные будут представлять указанные типы данных.

Все переменные, идентификаторы которых начинаются с букв I, J, K, L, M, N, представляют переменные целого типа. Переменные, идентификаторы которых начинаются с любой другой буквы, представляют вещественные переменные.

4.3.2. Оператор задания типа данных по первой букве имени (IMPLICIT).

Оператор используется для переопределения неявного типа данных (определяемого по первой букве имени). Оператор имеет вид:

```
IMPICIT T1 (D1), T2 (D2),...
```

где

T1, T2, ... - тип переменной - INTEGER, REAL, LOGICAL, DOUBLE PRECISION, BYTE, INTEGER*1, INTEGER*2, REAL*4, INTEGER*4, REAL*8;

D1, D2, ... - диапазон - последовательность букв, разделённых запятыми или знаком минус.

Все переменные, за исключением явно определённых, имена которых начинаются с букв A, X, Y, Z, быдут типа INTEGER. Переменные, имена которых начинаются с букв B, C, D, E, будут типа REAL, а с букв F, G, N и W - типа LOGICAL.

Оператор IMPLICIT задаёт тип, который будет присваиваться переменные по умолчанию. Любой оператор IMPLISIT должен находиться в программе перед любым оператором спецификации типа.

Если оператор IMPLICIT встретится в программе после операторов явной спецификации типа или оператора DIMENSION, типа переменных, уже определённых, не будет переопределяться.

4.4. Массивы.

Массив - это множество последовательно расположенных ячеек памяти, имеющих общее название - идентификатор (имя) массива. Обращение к отдельным областям памяти, которые называются элементами массива, осуществляется по идентификатору массива, к которому добавляется список индексов.

Массив имеет размерность в зависимости от количества индексов. Индексов может быть от одного до трёх. Примером масси- ва, имеющего размерность единица, является столбец чисел. Несколько столбцов чисел представляет собой двумерный массив. Для обращения к определённой величине из этого массива должны указываться номер строки и номер столбца, в которых находится эта величина. Для обращения к величине из трёхмерного массива должны быть заданы номер строки, номер столбца и номер страницы.

Массивы описываются при помощи следующих операторов ФОРТРАНА:

- операторы описания типа переменной;
- DIMENSION;
- COMMON.

Каждый из этих операторов содержит описатель массива и указывает идентификатор массива, его размерность и число элементов в каждом измерении.

Описатель массива имеет следующий формат:

```
A(D1[,D2]...)
```

где

A - идентификатор массива;
D1, D2, ... - индексы массива.

Число индексов определяет размерность массива. Максимальная размерность массива равна единице, а максимальная - трём.

5. ВЫРАЖЕНИЯ.

Выражение может быть константой, переменной, либо комбинацией из этих компонент и одного или более знаков операций.

Знаки операций показывают, какие вычисления над основными компонентами должны быть выполнены для того, чтобы получить значение выражения.

В ФОРТРАНЕ различаются три класса выражений:

- арифметические выражения;
- выражения отношения;
- логические выражения.

Результатом вычисления арифметического выражения является число. Выражения отношений и логические выражения могут быть истинными или ложными (т.е. их результатом являются логические величины).

5.1. Арифметические выражения.

Арифметические выражения формируются из арифметических элементов и знаков арифметических операций. Значением такого выражения является число.
К арифметическим элементам относятся следующие:

- константа;
- переменная;
- элемент массива;
- арифметическое выражение;
- обращение к арифметической функции.

Знаки арифметических операций определяют, какие вычисления надо выполнить над значениями арифметических элементов; в резуль- тате выполнения этих операций получается числовое значение. В языке ФОРТРАН используются следующие арифметические операции:

** (возведение в степень);
* (умножение);
/ (деление);
+ (сложение и одноместный плюс);
- (вычитание и одноместный минус).

5.2. Выражения отношений.

Выражение отношения состоит из двух арифметических выражений, разделённых операцией отношения. Значение данного выражения есть либо "истина", либо "ложь" в зависимости от того, выполняется или нет указанное отношение.

Операции отношения следующие:

Знак операции	Отношение
.LT.	меньше
.LE.	меньше или равно
.EQ.	равно
.NE.	не равно
.GT.	больше
.GE.	больше или равно

Ограничивающие точки - это часть знака операции отношения и их наличие в тексте обязательно.

5.3. Логические выражения.

Логическое выражение может быть отдельным логическим элементом или комбинацией логических элементов и логических операций. Логическое выражение имеет значение "истина" или "ложь".

Логический элемент может быть одним из следующих:

- целая или логическая константа;
- целая или логическая переменная;
- элемент целого или логического массива;
- выражение отношения;
- логическое выражение, заключённое в скобки;
- обращение к целой или логической функции.

Логические операции следующие:

.AND. - логическая кон'юнкция (логическое "и"); выражение A.AND.B истинно тогда и только тогда, когда и "A", и "B" истинны;
.OR. - логическая диз'юнкция (логическое "или"); например, "A.OR.B"; выражение истинно тогда и только тогда, когда истинно либо "A", либо "B", либо оба истинны;
.XOR. - исключающее или; например, "A.XOR.B"; выражение истинно, если "A" истинно, а "B" ложно, либо, если "B" истинно, а "A" ложно; выражение ложно, если оба элемента имеют одинаковое значение;
.NOT. - логическое отрицание (логическое "НЕ"); например, "NOT.A"; выражение истинно тогда и только тогда, когда ложно "A".

Ниже приведён перечень всех операции, которые могут встречаться в логическом выражении, и порядок, в котором они выполняются.

Операция	Выполняется
**	первой
*,/	второй
+,-	третьей
Отношения	четвёртой
.NOT.	пятой
.AND.	шестой
.OR.,.XOR.	седьмой

Операции с одинаковым приоритетом выполняются слева направо.

6. ОПЕРАТОРЫ.

Ниже приведен перечень операторов языка ФОРТРАН, дан основной формат каждого оператора и краткие пояснения.

Ассерт
- см. READ, форматный последовательный ввод;
- см. READ, ввод под управлением списка.

Арифметическое/логическое присваивание W=A

W - идентификатор переменной или идентификатор элемента массива;
A - выражение.

Переменной W присваивается значение арифметического или логического выражения A.

Арифметический оператор-функция F(P1, P2. ..., PN)=E

```
F                    - идентификатор;
PI (I=1, 2, ..., N) - идентификатор;
E                    - выражение.
```

Строится функция пользователя, в которой "P" является формальным параметром. При обращении к ней выполняется вычисление выражения с использованием значений фактических параметров.

ASSIGN S TO W

```
S   - метка выполняемого оператора;
W   - идентификатор целой переменной.
```

Устанавливается соответствие между меткой оператора "S" и целой переменной "W" для последующего использования в операторе перехода по предписанию GO TO.

BACKSPACE U

```
U   - целая переменная или целая константа.
```

Возврат к предыдущей записи в файле, открытом в текущий момент времени на логическом устройстве "U".

BLOCK DATA [NAM]

```
NAM   - идентификатор.
```

Определяет следующую за ним подпрограмму как подпрограмму данных.

```
CALL SE(P1, P2, .... PN)]

S                    - идентификатор подпрограммы;
PI (I=1, 2, ..., N) - выражение, идентификатор процедуры или
                      идентификатор массива.
```

Вызывается подпрограмма с идентификатором "S". При этом передаются фактические параметры "P" для замены формальных параметров в определении подпрограммы.

COMMON [/[A1]/ K1][[,]/[A2]/K2]...

```
AI (I=1, 2, ...) - идентификатор общего блока;
KI (I=1, 2, ...) - список, состоящий из одного или несколь-
                   ких идентификаторов переменных, идентифи-
                   каторов массивов или описателей массивов,
                   разделённых запятыми.
```

Резервирует участок оперативной памяти, помеченный идентифи-

катором блока COMMON, для размещения переменных, связанных с этим блоком.

CONTINUE

Передаёт управление на следующий выполняемый оператор.

DATA K1/C1/[[,] K2/C2/]...

KI (I=1, 2,...) - список, состоящий из одного или нескольких идентификаторов переменных, идентификаторов массивов или идентификаторов элементов массивов, разделённых запятыми. Индексные выражения должны быть константами;
CI (I=1, 2,...) - список, состоящий из одной или нескольких констант, разделённых запятыми, причём каждой константе может предшествовать необязательная форма J* (J - ненулевая целая константа без знака).

Вызывает присваивание начальных значений из списка констант соответствующим элементам списка идентификаторов переменных.

DIMENSION A1(D1), A2(D2),..., AN(DN)

AI(DI) (I=1, 2, ..., N) - описатель массива.

Определяет об'ём памяти, требующейся для массивов.

DO S I=E1, E2[,E3]

S - метка последнего выполняемого оператора в цикле;
I - идентификатор переменной цикла;
E1,E2[,[3]- целые положительные переменные или беззнаковые целые константы.

Выполняются следующие действия:

1) установка I=E1;
2) выполнение операторов до оператора с номером "S" включительно;
3) вычисление I=I+E3;
4) повторение пунктов 2) и 3) "M" раз.
(M = MAX(1,INT((E2-E1)/E3)+1))

END

Завершает программную единицу.

END FILE U

U - целая переменная или константа.

На логическое устройство "U" записывается признак конца файла.

END=S, ERR=S

S - метка выполняемого оператора.

Передача управления по концу файла или ошибочному состоянию, являющаяся необязательным элементом любого оператора ввода/вывода. Этот элемент позволяет программе перейти к оператору с номером "S" при условии конца файла (END=) или при ошибочном состоянии (ERR=).

EQUIVALENCE (K1),(K2),...,(KN)

KI (I=1, 2, ...N) - список, состоящий из двух или более идентификаторов переменных, идентификаторов массивов или идентификаторов элементов массивов, разделённых запятыми. Индексные выражения должны быть константами.

Каждый из идентификаторов, принадлежащих списку, размещается в одной и той же области памяти.

EXTERNAL V1, V2, ..., VN

VI (I=1, 2, ..., N) - идентификатор программы.

Описывает идентификаторы, как идентификаторы внешней процедуры.

FORMAT (спецификация преобразования)

Описывает формат, в котором должна быть передана одна или несколько записей. Оператор должен иметь метку.

[TYP] FUNCTION NAME[*N] [(P1, P2, ..., PN)]

TYP - указатель типа данных;
NAM - идентификатор;
*N - указатель длины данных;
PI (I=1, 2, ..., N) - идентификатор.

Является первым оператором подпрограммы-функции, указывая её идентификатор и формальные параметры "PI". Может включать необязательный указатель типа.

GO TO S

S - метка выполняемого оператора.

Безусловный переход. Передаёт управление оператору с меткой "S".

GO TO (K1, K2, ..., KN), M

K1, K2, ..., KN - список, состоящий из одной или нескольких

меток выполняемых операторов, разделённых запятыми;

M - целое выражение.

Вычисляемый оператор перехода; передаёт управление оператору с меткой, определяемой значением выражения "M"; если "M" равно 1, управление передаётся оператору с первой меткой из списка; если "M" равно 2, управление передаётся оператору со второй меткой и т.д.; если "M" меньше единицы или больше, чем имеющееся число меток, передача управления происходит на следующий выполняемый оператор.

GO TO V [[,](K1, K2, ..., KN)]

V - идентификатор целой переменной;
K1, K2, ..., KN - список, состоящий из одной или нескольких меток выполняемых операторов, разделённых запятыми.

Переход по предписанию; передаёт управление оператору, который был последним установлен в соответствие переменной "V" оператором "ASSIGN".

IF (M) K1, K2, K3

M - выражение;.
K1, K2, K3 - метки выполняемых операторов.

Арифметический оператор условного перехода; передаёт управление оператору с номером K1, K2 или K3 в зависимости от значения выражения;
Если значение выражения меньше нуля, то выполняется переход на "K1";
Если значение выражения равно нулю, то выполняется переход на "K2";
Если значение выражения больше нуля, то выполняется переход на "K3".

IF (E) S

E - выражение;
S - любой выполняемый оператор, за исключением оператора "DO" и логического условного оператора "IF".

Логический условный оператор. Выполняет оператор "S" при условии истинности выражения.

IMPLICIT T1(D1), T2(D2), ...

TI (I=1, 2, ...) - тип данного;
DI (I=1, 2, ...) - список букв, разделённых запятыми или знаком минус.

PAUSE [N]

N - последовательность от 1 до 6 любых символов.

Приостанавливает выполнение программы и выводит на консоль строку "N", если "N" задано.

```
PROGRAM NAM
```

NAM - идентификатор.

Определяет идентификатор программы.

```
READ (U,F[,END=S1] [,ERR=S2])[K]
READ F[,K]
```

```
U       - целая переменная или целая константа;
F       - метка оператора "FORMAT" или идентификатор массива;
Si,S2   - метки выполняемых операторов;
K       - список ввода.
```

Форматный последовательный ввод. Считывает одну или более логических записей с устройства "U" и присваивает элементам списка ввода значения, преобразованные в соответствии со спецификацией формата в операторе "F".

```
READ (U[,END=S1][,ERR=S2])[K]
```

```
U       - целая переменная или целая константа;
S1, S2  - метки выполняемых операторов;
K       - список ввода.
```

Неформатный последовательный ввод. Считывает одну неформатную запись с устройства "U" и присваивает значения элементам списка.

```
RETURN
```

Возвращает управление в вызывающую программу из выполняемой подпрограммы.

```
REWIND U
```

U - целая переменная или целая константа.

Вызывает возврат к началу файла, открытого в настоящий момент на логическом устройстве "U".

```
STOP [N]
```

N - последовательность от 1 до 6 символов.

Прекращает выполнение программы и выводит на консоль строку "N", если "N" задано.

```
SUBROUTINE NAM [(P1, P2, ..., PN)]
```

```
NAM                  - идентификатор;
PI (I=1, 2, ..., N) - идентификатор.
```

Начинает подпрограмму, указывая её идентификатор и идентификаторы формальных параметров "PI".

```
TYPE
```

```
- см. WRITE, форматный последовательный вывод;
- см. WRITE, вывод под управлением списка.
```

```
TYP V1, V2, ..., VN

TYP                  - указатель типа данных;
VI (I=1, 2, ..., N) - идентификатор переменной, идентификатор массива, идентификатор функции или точки входа в функцию, или описатель массива. За идентификатором может следовать необязательный указатель длины типа данных (*N).
```

Идентификаторам "VI" в данной программой единице присваивается указанный тип. Тип может быть одним из следующих:

```
DOUBLE PRECISION
REAL
REAL*4
REAL*8
INTEGER
INTEGER*1
INTEGER*2
INTEGER*4
BYTE
LOGICAL
LOGICAL*1
LOGICAL*2
```

```
WRITE (U,F[,ERR=S])[K]

U - целая переменная или целая константа;
F - метка оператора "FORMAT" или идентификатор массива;
S - метка выполняемого оператора;
K - список вывода.
```

Форматный последовательный вывод; записывает на устройство "U" одну или несколько записей, содержащих значения элементов списка, преобразованных согласно спецификации формата "F".

```
WRITE (U[,ERR=S])[K]

U - целая переменная или целая константа;
S - метка выполняемого оператора;
K - список вывода.
```

Неформатный последовательный вывод. Записывает на устройство "U" одну неформатную запись, содержащую значения элементов спис-

ка.

7. БИБЛИОТЕЧНЫЕ ФУНКЦИИ ЯЗЫКА ФОРТРАН.

7.1. Функции преобразования.

ABS(X)
- вещественное абсолютное значение; тип параметра - вещественный , тип функции - вещественный.

IABS(X)
- целое абсолютное значение; тип параметра - целый, тип функии - целый.

DABS(X)
- абсолютное значение с двойной точностью; тип параметра - с двойной точностью, тип функции - с двойной точностью.

FLOAT(I)
- преобразование целой величины в вещественную; тип параметра - целый, тип функции - вещественный.

IFIX(X)
- преобразование вещественной величины в целую; IFIX(X) эквивалентна INT(X); тип параметра - вещественный, тип функции - целый.

SNGL(X)
- преобразование величины с двойной точностью в вещественную величину; тип параметра - с двойной точностью, тип функции - вещественный.

DBLE(X)

- преобразование вещественной величины в величину с двойной точностью; тип параметра - вещественный, тип функции - с двойной точностью.

7.2. Функции выделения целой части.

Результатом функции выделения целой части является наибольшее целое число, не превышающее по абсолютной величине аргумент функции. Функции присваивается знак аргумента.

AINT(X)
- выделение целой части вещественной величины; тип параметра - вещественный, тип функции - вещественный.

IDINT(X)
- выделение целой части вещественной величины; тип параметра - вещественный, тип функции - вещественный.

INT(X)
- выделение целой части величины с двойной точностью; тип

параметра - двойной точности, тип функции - целый.

7.3. Функции остатка.

Результатом функции остатка является остаток от деления первого аргумента на второй.

AMOD(X,Y)
- остаток вещественный; тип параметров - вещественный, тип функции - вещественный.

MOD(I,J)
- остаток целый; тип параметров - целый, тип функции - целый

DMOD(X,Y)
- остаток с двойной точностью; тип параметров - двойной точности, тип функции - двойной точности.

7.4. Функции выбора максимального значения.

Результатом функции выбора максимального значения является наибольшее значение из аргументов списка, состоящего из двух или более аргументов.

AMAXO(I, J, ...)
- максимальное вещественное значение из списка целых аргументов; тип параметров - целый, тип функции - вещественный

AMAX1(X, Y, ...)
- максимальное вещественное значение из списка вещественных аргументов; тип параметров - вещественный, тип функции - вещественный.

MAXO(I, J, ...)
- максимальное целое значение из списка целых аргументов; тип параметров - целый, тип функции - целый.

MAX1(X, Y, ...)
- максимальное целое значение из списка вещественных аргументов; тип параметров - вещественный, тип функции - целый.

DMAX1(X, Y, ...)
- максимальное значение двойной точности из списка аргументов с двойной точностью; тип параметров - с двойной точностью, тип функции - с двойной точностью.

7.5. Функции выбора минимального значения.

Результатом функции выбора минимального значения является наименьшее значение из списка аргументов, состоящего из двух или более аргументов.

AMINO(I, J, ...)
- минимальное вещественное значение из списка целых аргументов; тип параметров - целый, тип функции - вещественный.

AMIN1(X, Y, ...)
- минимальное вещественное значение из списка вещественных аргументов; тип параметров - вещественный, тип функции - вещественный.

MINO(I, J, ...)
- минимальное целое значение из списка целых аргументов; тип параметров - целый, тип функции - целый.

MIN1(X, Y, ...)
- минимальное целое значение из списка вещественных аргументов; тип параметров - вещественный, тип функции - целый.

DMIN1(X, Y, ...)

- минимальное значение с двойной точностью из списка параметров с двойной точностью; тип параметров - двойной точности, тип функции - двойной точности.

7.6. Функции присваивания знака.

Результатом функции присваивания знака является абсолютное значение первого аргумента с присвоенным ему знаком второго аргумента.

SIGN(X, Y)
- присваивание знака вещественной величине; тип параметров - вещественный, тип функции - вещественный.

ISIGN(I, J)
- присваивание знака целой величине; тип параметров - целый, тип функции - целый.

DSIGN(X, Y)
- присваивание знака величине с двойной точностью; тип параметров - с двойной точностью, тип функции - с двойной точностью.

7.7. Функции положительной разности.

Результатом функции положительной разности является разность первого аргумента и минимального из двух аргументов.

DIM(X, Y)
- положительная разность вещественных величин; тип параметров - вещественный, тип функции - вещественный.

IDIM(I, J)

- положительная разность целых величин; тип параметров - целый, тип функции - целый.

7.8. Экспоненциальные функции.

Результатом экспоненциальной функции является значение "E", возведённое в степень, равную аргументу.

EXP(X)
- E**X. тип параметра - вещественный, тип функции - вещественный.

DEXP(X)
- E**X. тип параметра - с двойной точностью, тип функции - с двойной точностью.

7.9. Логарифмические функции.

ALOG(X)
- натуральный логарифм вещественного аргумента; тип параметра - вещественный, тип функции - вещественный.

ALOG1Ø(X)
- десятичный логарифм вещественного аргумента.

DLOG(X)
- натуральный логарифм аргумента с двойной точностью; тип параметра с двойной точностью, тип функции - с двойной точностью.

7.1Ø. Квадратный корень.

SQRT(X)
- корень квадратный из вещественного аргумента; тип параметра - вещественный, тип функции - вещественный.

DSQRT(X)
- корень квадратный из аргумента двойной точностью, тип функции - с двойной точностью.

7.11. Тригонометрические функции.

SIN(X)
- синус вещественного аргумента; тип параметра - вещественный, тип функции - вещественный.

DSIN(X)
- синус аргумента с двойной точностью; тип параметра - с двойной точностью, тип функции - с двойной точностью.

COS(X)

- косинус вещественного аргумента; тип параметра - вещественный, тип функции - вещественный.

DCOS(X)
- косинус аргумента с двойной точностью; тип параметра - с двойной точностью, тип функции - с двойной точностью.

ATAN(X)
- арктангенс вещественного аргумента; тип параметра - вещественный, тип функции - вещественный.

DATAN(X)
- арктангенс аргумента с двойной точностью; тип параметра - с двойной точностью, тип функции - с двойной точностью.

ATAN2(X, Y)
- арктангенс вещественного аргумента от "X/Y"; тип функции - вещественный.

DATAN2(X, Y)
- арктангенс аргумента с двойной точностью от "X/Y"; тип функции - с двойной точностью.

7.12. Дополнительные математические функции.

TAN(X)
- тангенс гиперболический; тип параметра - вещественный, тип функции - вещественный.

Приложение 1.

КЛЮЧИ КОМПИЛЯТОРА.

Ключ	Действие
/O	Печатать все адреса и т.п. в листинге в виде восьмеричных чисел
/H	Печатать все адреса и т.п. в листинге в виде шестнадцатеричных чисел
/N	Не печатать листинг псевдоассемблерного текста скомпилированной программы
/R	Генерировать файл перемещаемых машинных кодов
/L	Выдавать листинг программы
/P	Резервировать дополнительный об'ём памяти в 100H байтов для стека компилятора. Возможно использование нескольких P. Например, /P/P - резервирует дополнительные 200H байтов для стека
/M	Генерируемые машинные коды могут быть помещены в ПЗУ. Когда задаётся ключ /M, то генерируемые коды отличаются от обычных тем, что: 1) операторы FORMAT помещаются в CSEG вместе с командами JMP, обходящими эти операторы; 2) блоки параметров для вызовов подпрограмм, имеющих более 3 формальных параметров, инициализируются во время работы программы, а не при компиляции.

По умолчанию, во время работы компилятора действуют следующие ключи: /H, /R, /L.

Приложение 2.
СООБЩЕНИЯ ОБ ОШИБКАХ ПЕРИОДА КОМПИЛЯЦИИ.

ILLEGAL STATEMENT NUMBER
- недопустимая метка оператора;

STATEMENT UNRECOGNIZABLE OR MISSNELLED
- оператор нераспознаваем или неверно записан;

ILLEGAL STATEMENT COMPLETION
- недопустимое завершение оператора;

ILLEGAL DO NESTING
- недопустимая вложенность операторов DO;

ILLEGAL DATA CONSTANT
- неверная константа в данных;

MISSING NAME
- имя опущено;

ILLEGAL PROCEDURE NAME
- недопустимое имя процедуры;

ILLEGAL DATA CONSTANT OR REPEAT FACTOR
- недопустимая константа или коэффициент повторения;

INCORRECT INTEGER CONSTANT
- неправильная целая константа;

INCORRECT NUMBER OF DATA CONSTANTS
- неверное число констант в DATA;

INVALID STATEMENT NUMBER
- неверная метка оператора;

NOTA VARIABLE NAME
- не имя переменной;

ILLEGAL LOGICAL FORM OPERATOR
- недопустимый логический оператор;

DATA POOL OVERFLOW: OUT OF MEMORY
- переполнение пула констант: не хватает памяти;

LITERAL STRING TOO LARGE
- слишком длинный литерал;

INVALID DATA LIST ELEMENT IN I/Ø
- неверный элемент данных в списке ввода/вывода;

UNBALANCED DO NEST
- несбалансированные вложенные области DO;

IDENTIFIER TOO LONG
- имя слишком длинно;

ILLEGAL OPERATOR
- недопустимый оператор;

MISINATCHED PARENTHESIS
- несбалансированные скобки;

CONSECUTIVE OPERATORS
- последовательные операторы;

IMPROPER SUBSCRIPT SYNTAX
- неверная запись индекса;

ILLEGAL INTEGER QUANTITY
- недопустимая целая величина;

ILLEGAL HOLLERITIN CONSTRUCTION
- неверная запись символьной строки;

BACKWARDS DO REFERENCE
- последний оператор области DO расположен раньше оператора DO;

ILLEGAL STATEMENT FUNCTION NAME
- недопустимое имя операторной функции;

ILLEGAL CHARACTER FOR SYNTAX
- недопустимый знак;

STATEMENT OUT OF SEQUENCE
- неправильный порядок операторов;

MISSING INTEGER QUANTITY
- пропущено целое;

INVALID LOGICAL OPERATOR
- недопустимый логический оператор;

ILLEGAL ITEM FOLLOWING INTEGER OR REAL OR LOGICAL
- недопустимая синтаксическая единица, следующая за INTEGER, REAL или LOGICAL;

PREMATURE END OF FILE ON INPUT DEVICE
- на устройстве ввода преждевременно обнаружен конец файла;

ILLEGAL MIXED MODE OPERATION
- недопустимая операция над данными разных типов;

FUNCTION CALL WHITH NO PARAMETRS
- в обращении к функции нет фактических параметров;

STACK OVERFLOW
- переполнение стека;

ILLEGAL STATEMENT FOLLOWING LOGICAL IF
- недопустимый оператор в логическом IF;

WRONG NUMBER OF SUBSCRIPTS
- неверное число индексов.

Предупреждения:

DIPLICATE STATEMENT NUMBER
- многократное определение метки оператора;

ILLEGAL DO TERMINATION
- область действия DO завершается недопустимым оператором;

ARRAY NAME MISUSE
- неверное использование имени массива;

ARRAY MULTIPLY EQUIVALENCED WITHIN A GROUP
- массив сэквивалентирован сам себе в группе;

MULTIPLE EQUIVALENCE OF COMMON
- многократное эквивалентирование COMMON;

COMMON BASE COWERED
- расширение области COMMON в сторону отрицательных адресов;

NON-COMMON VARIABLE IN BLOCK DATA
- переменные в подпрограмме BLOCK DATA не находятся в COMMON;

EMPTY LIST FOR UNFORMATED WRITE
- пустой список переменных в операторе бесформатной записи (WRITE);

NON-INTEGER EXPRESSION
- выражение не целого типа;

OPERATOR MODE NOT COMPATIBLE WITH OPERATOR
- операция недопустима в операндом данного типа;

MIXING OF OPERAND MODES NOT ALLOWED
- недопустимое смещение операторов разных типов;

MISSING INTEGER VARIABLE
- пропущена целая переменная;

MISSING STATEMENT NUMBER ON FORMAT
- в операторе FORMAT отсутствует метка;

ZERO REPEAT FACTOR
- нулевой коэффициент повторения;

ZERO FORMAT VALUE
- нулевая длина в спецификации формата;

STATEMENT NUMBER NOT FORMAT ASSOCIATED
- оператор с данной меткой не является оператором FORMAT;

INVALID STATEMENT NUMBER USAGE
- недопустимое использование метки оператора;

NO PATH TO THIS STATEMENT
- отсутствует передача управления на этот оператор;

MISSING DO TERMINATION
- отсутствует оператор, завершающий оператор DO;

CODE OUTPUT IN BLOCK DATA
- в подпрограмме BLOCK DATA встретились выполняемые операторы;

UNDEFINED LABELS OCCURRED
- имеются ссылки на несуществующие операторы;

RETURN IN MAIN PROGRAM
- оператор RETURN недопустим в главной программе;

STATUS ERROR ON READ
- ошибка при чтении;

INVALID OPERAND USAGE
- недопустимое использование операнда;

FUNCTION WITH NO PARAMETR
- функция без формальных параметров;

NEX CONSTANT OVERFLOW
- значение шестнадцатеричной константы слишком велико;

DIVISION BY ZERO
- деление на ноль;

ARRAY NAME EXPECTED
- здесь должно быть имя массива;

Приложение 3.
СООБЩЕНИЯ ОБ ОШИБКАХ ПЕРИОДА ВЫПОЛНЕНИЯ.

Предупреждения:

IB	переполнение буфера ввода;
TL	слишком много левых скобок в операторе FORMAT;
OB	переполнение буфера вывода;
DE	порядок действительного числа при вводе больше 99;
IS	величина целого слишком велика;
BE	переполнение порядка при действиях с плавающей точкой;
IN	вводимая запись слишком длинна;
OV	переполнение при арифметических действиях;
CN	при преобразовании действительного числа в целое, последнее слишком велико;
SN	аргумент в функции SIN(X) слишком велик;
AZ	оба аргумента в функции ATANZ равны нулю;
IO	недопустимая операция ввода/вывода;
BI	переполнение буфера при выполнении операции бесформатного чтения/записи;
RC	отрицательная величина коэффициента кратности в операторе FORMAT.

Грубые ошибки:

ID	недопустимая спецификация в операторе FORMAT;
FO	поле длины в спецификации оператора FORMAT равно нулю;
MP	пропущена точка в спецификации оператора FORMAT;
FW	поле длины в спецификации слишком мало;
IT	ошибка при выполнении передачи данных во время ввода/вывода;
ML	отсутствует левая скобка в операторе FORMAT;
DZ	деление на ноль действительного или целого;
LG	аргумент в функции LOG(X) отрицателен или равен нулю;
SQ	отрицательный аргумент X в функции SQRT(X);
DT	тип данного и спецификация в операторе FORMAT противоречат друг другу;
EF	встретился конец файла при чтении.

СКБ Вычислительной техники Института кибернетики АН ЭССР
"ЭКТА"

ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ИНТЕЛЛЕКТУАЛЬНОГО ТЕРМИНАЛА Е5104

РЕДАКТОР СВЯЗЕЙ

L8Ø

РУКОВОДСТВО ОПЕРАТОРА

353872.3ØØ39-34

4 стр.

Таллинн 1988

1.ОБЩИЕ СВЕДЕНИЯ О ПРОГРАММЕ

ЕКТА Tallinn *ЭКТА* Таллин КОМПОНЕНТ / КОМПЛЕКС ПО

L8Ø	353872.3ØØ39-34	Версия: 3.44
РЕДАКТОР СВЯЗЕЙ		
ОБЬЕМ 12К байт	ОПЕРАЦ. СИСТЕМА: EKDOS	
ТРЕБУЕМОЕ ОЗУ	Программа 12К байт	Данные байт
НОСИТЕЛЬ: ГМД РЕГ. но НОСИТЕЛЯ:		
НАЗНАЧЕНИЕ И МЕТОД: Программа L8Ø преобразует файл перемещаемых машинных кодов в файл типа .COM, готовый к выполнению.		
ССЫЛКИ: Уэйт М., Ангермейер Дж. Операционная система CP/M, Радио и связь, 1986		
ПРОЦЕССОР / ЭВМ: К58ØИК8Ø / Е5104		
ВНЕШНИЕ УСТР-ВА: НГМД, видеомонитор		
5.1Ø.1988	АРХ. МЛ: 3.ØØØ	ПРОГРАММИСТ: Мяртин К.

2. ВЫЗОВ И ПУСК

2.1. Командная строка редактора связей.

Редактор связей L80 можно вызвать одним из двух способов:

1) вводом с консоли командной строки:

```
L80
```

После чего загружается редактор связей, который идентифицирует себя выводом сообщения о своей версии и с новой строки выводит запрос:

```
- *
```

Оператор должен ввести командную строку редактора связей;

2) Вводом с консоли:

```
L80 "командная строка"
```

где "командная строка" редактора связей представляет собой последовательность имён файлов и ключей, разделённых запятыми. В приложений приведено перечень и краткое описание основных ключей редактора связей.

2.2. Примеры командных строк.

1) `L80 RKT,FORLIB/S/G/E`

Компонует все перемещаемые модули из файла RKT.REL и требуемые из файла FORLIB.REL в программу, размещаемую в оперативной памяти, и передаёт ей управление;

2) `L80 RKT/N,RKT,FORLIB/S/E`

Компонует программу, помещая её в файл RKT.COM, из модулей файлов RKT.REL и FORLIB.REL.

ПРИЛОЖЕНИЕ

ОСНОВНЫЕ КЛЮЧИ РЕДАКТОРА СВЯЗЕЙ.

FILE/N	указывает редактору связей, что надо сформировать на диске программу, готовую к выполнению, в файле FILE.COM
/E	указывает редактору связей на конец командной строки
/G	после редактирования передать управление скомпонованной в памяти программы
FILE/S	из файла с именем FILE редактор связей выбирает лишь те перемещаемые модули, которые необходимы для разрешения глобальных переменных
/M	печать таблицы глобальных символов
/U	печать таблицы неопределённых глобальных символов

СКБ Вычислительной техники Института кибернетики АН ЭССР
"ЭКТА"

ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ИНТЕЛЛЕКТУАЛЬНОГО ТЕРМИНАЛА Е51Ø4

ОТЛАДЧИК ПРОГРАММ

SID

РУКОВОДСТВО ОПЕРАТОРА

353872.3ØØ36-14

11 стр.

Таллинн 1988

1.ОБЩИЕ СВЕДЕНИЯ О ПРОГРАММЕ

ЕКТА Tallinn *ЭКТА* Таллин КОМПОНЕНТ / КОМПЛЕКС ПО

SID	353872.300036-14	Версия: 1.4
ОТЛАДЧИК ПРОГРАММ		
ОБЪЕМ 8К байт	ОПЕРАЦ. СИСТЕМА: EKDOS	
ТРЕБУЕМОЕ ОЗУ	Программа 8К байт	Данные байт
НОСИТЕЛЬ: ГМД РЕГ. но НОСИТЕЛЯ:		
НАЗНАЧЕНИЕ И МЕТОД: Программа SID позволяет пользователю загружат в оперативную памят , просматриват , тестировать , а также изменят и отблаживат на языке ассемблера любую программу, представленную на машинном коде. Программа работает в динамическом режиме, что позволяет пол - зователю запускат находящуюся в памяти отлаживаемую программу и отлаживать каждый шаг ее выполнения. С ее помощью можно также вносить небольшие изменения в существующие программы или реассемблировать их ц тем, чтобы понять, как они работают. Программа SID содержит также встроенный мини-ассемблер.		
ССЫЛКИ: Уэйт М., Ангермейер Дж. Операционная система CP/M, Радио и связь, 1986		
ТЕХН. ХАР-КИ: 15 директивов, программа загружается во верхний часть поля памяти пользователя.		
ПРОЦЕССОР / ЭВМ: К580ИК80 / Е5104		
ВНЕШНИЕ УСТР-ВА: НГМД, видеомонитор		
5.10.1988	АРХ. МЛ: 3.000	ПРОГРАММИСТ: Томмингас, Т.

2. ВЫЗОВ И ПУСК

```
SID          - пускается отладчик, обрабатываемая программа не
               загружается
SID x.COM    - программа x.COM загружается начиная с адреса 100H
SID x.OBJ    - то же, но файл в об ектном коде
SID x.y u.v  - загружаются x.y (в общем случае u.SYM) и
               таблица меток u.v
```

Пример: SID SORT.COM SORT.SYM

После пуска отладчик выдаёт следующие сообщения:

```
£       - отладчик ожидает ввод директивы
SYMBOLS - таблица меток загружена; при отказе в начале следующей
          строки появится "?"

NEXT    PC    END
nnnn    pppp  eeee   - программа и метки загружены
 !       !      !
 !       !      !- конечный адрес свободной зоны памяти
 !       !- значение счётчика команд
 !- начальный адрес свободной зоны памяти
```

3. ОБЩИЕ ПОНЯТИЯ

3.1. Директивы

```
A (Assembler)    - Построчная трансляция
C (Call)         - Вызов подпрограммы
D (Display)      - Распечатка памяти
F (Fill memory)  - Заполнение памяти
G (Go)           - Пуск программы
H (Hex)          - 16-ричная арифметика
I (Input line)   - Ввод директивы операционной системы
L (List)         - Обратная трансляция
M (Move)         - Пересылка содержимого памяти
P (Pass point)   - Задание контрольных точек
R (Read)         - Загрузка
S (Set memory)   - Запись в память
T (Trace)        - Трассировка
U (Untrace)      - Пошаговая работа
X (eXamine)      - Вектор состояния программы
```

Директиву вводят с клавиатуры в ответ на сигнал готовности отладчика "£". Каждая директива состоит из мнемонической буквы и факультативных параметров в виде выражений. Максимальная длина директивы 64 символа. Ввод директивы завершается нажатием клавиши RETURN, в качестве разделителей могут применяться запятая или пробел. Для возврата из отладчика используются клавиши CTRL C или директива G0.

3.2. Числа

По умолчанию все числа воспринимаются как шестнадцатеричные в диапазоне 0H...0FFFFH. Если число содержит более 4 цифр, отладчик воспринимает последние четыре цифры.

Примеры: 30 3F FF3F F3

3.3. Десятичные числа

Признаком десятичного числа является символ "£" перед числом. Из чисел, превышающих 65535, воспринимаются 16 правых двоичных разрядов.

Примеры: £48 £9999 £655535 £0

3.4. Символы

SID воспринимает символы КОИ-8, заключённые между апострофами. Из цепочек, содержащих более 2 символов, воспринимаются 2 правых. Правый символ записывается в младший байт 16-разрядного двоичного слова. Наличие только одного символа означает нулевое содержание старшего байта. Цепочки нулевой длины недопустимы. Апостроф, являющийся элементом цепочки, вводится в виде двух апострофов ('...''...').

Примеры: 'a' 'A' 'xy' '£'

3.5. Ссылки на метки

При наличии таблицы меток символьные выражения могут содержать ссылки на метки:

```
.s   - адрес, соответствующий метке s
 s   - 16-разрядное слово на адресе .s
=s   - байт на адресе .s
```

Величина s - элемент таблицы меток.

3.6. Символьные выражения

Символьные выражения представляет собой комбинацию из шестнадцатеричных и десятичных чисел, цепочек символов и ссылок на метки, связанных между собой операторами "+" и "-". В зависимости от связующих операторов, числовые значения компонентов складываются или вычитаются, контроля за переполнением нет. Результатом является 16-разрядное слово.

Синтаксис:

```
-x      = 0-x
+x      = x+x, где x = значение выражения, предшествующего знаку "+"
^^...^  = n-е слово в стеке; символ пишется n раз подряд
```

Примечание. Пробелы внутри выражения не допускаются.

4. Описание директив

4.1. A (Assemble) - Построчный ассемблер

As - Включение режима транслирования строки с адреса s. Следующий возможный адрес для новой ассемблерной команды выводится на экран автоматически после ввода предыдущей. Для возврата из этого режима вводят пустую строку (только RETURN) или ".".

A - То же, но пусковой адрес определяется по последнему адресу, обработанному директивами A, L или T.

-A - Удаляет из отладчика модуль ассемблера и обратного ассемблера и таблицу меток. Одновременно дезактивируются директивы A и L. Директива T теперь выдаёт только машинный код в 16-ричной форме.

Примеры:

```
A100          AGAMMA+X-=1
A#256         A+30
A. CTRF+5
```

4.2. C (Call) - Вызов подпрограммы

Cs - Вызов из отладчика, по адресу s, причём содержимое регистров исследуемой программы сохраняется. При входе в подпрограмму BC=0000 и DE=0000

Cs, b - То же, BC = b, DE=0000, где b - выражение

Cs, b, d - То же, BC = b, DE = d, где b,d - выражения

Примеры:

```
C100          CJMPVEC+=X
C#4096        C. CRLF,#34
C. DISPLAY    C. CRLF, X,+=X
```

4.3. D (Display memory) - Распечатка памяти

Содержимое памяти выводится в виде байтов, если же директива содержит W, то 2-байтовыми словами. В конце каждой строки приводятся соответствующие символы КОИ-8.

Ds DWs - Начиная с адреса S, до заполнения половины экрана

Ds,f DWs,f - В диапазоне адресов s...f

D DW - Начиная с адреса , следующего за последним выведенным байтом, или же с адреса, содержащегося в регистрах HL после вывода регистров (см.директиву X), до заполнения половины экрана.

d,F DW,f - То же, но до адреса F

Примеры:

```
DF3F                  D,. GAMMA
D#256,#512            DW  ALPHA,+#100
D. gamma,.DELTA+#30
```

4.4. F (Fill memory) - Заполнение памяти

F s,f.d - Заполнение области памяти на адресах s...f значением d (1 байт)

Примеры:

```
F100,3FF,ff
F.gamma,+#100,#23
F ALPHA,+=1, =X
```

4.5. G (Go to program) - Пуск программы

```
G          -G
Gp         -Gp
G,a        -G,a
Gp,a       -Gp,a
G,a,b      -G,a,b
@B,A,B     -@B,A,B
```

Исследуемая программа выполняется в реальном времени; управление передаётся из программы в отладчик только в точках останова или в контрольных точках, если они заданы (см. директиву P), или при RST 7.

Параметры:

p - пусковой адрес, записывается в счётчик команд PC перед пуском программы; если директива не содержит этот параметр, программа запускается по значению PC, содержащемуся в её векторе состояния

a,b - определение точек останова; если определение содержит символ "^", то управление передаётся по адресу, содержащемуся в самом верхнем 2-байтовом слове в стеке, поэтому такой вариант удобен для останова на адресе возврата из подпрограммы

минус - отображение состояния в контрольных точках, заданных директивой P, блокируется до достижения счётчиком контрольных точек нуля.

Сообщения:

При достижении точки останова или в случае внешней RST 7 выдаётся адрес точки останова в виде

```
*nnnn
```

и точка останова аннулируется.

Примеры:

```
G                GCRLF,.PRINT,£1024
G100             GJMPVEC+=1,.ENDC,.ERRC
G100,103         G..errsub
G,.ERRSUB.+30    -G100,+10,+10
```

4.6. H (Hexadecimal values) - Шестнадцатеричная арифметика

Ha,b - Выдача суммы (a + b) и разности (a-b) в шестнадцатеричной форме

Ha - Выдача значений a: в шестнадцатеричной системе, в десятичной системе, в виде символа 'c'(КОИ-8); следует метка, если она имеется:
hhhhЂdddd'c'.ssss

H - Выдача всей таблицы меток с шестнадцатеричными значениями элементов. Для прекращения вывода нажимают на любую клавишу.

Примеры:
```
H 100,200
H#1000,965
H,GAMMA+=1,ALPHA-#10
```

4.7. I (Input line) - Ввод директивы операционной системы

Ic1c2...cn - Подготовка областей памяти для директивы R или для испытуемой программы, как если бы символы c1...cn были введены через операционную систему. Устанавливается исходное состояние УБФ, вводно-выводной буфер размещают на стандартном месте (B00T+80H). c1...cn - символы КОИ-8, которые в командной строке операционной системы следовали бы за именем испытуемой программы.

Примеры:

```
Ix dat                  ITEST.COM
Ix.inp y.out            ITEST.OBJ TEST SYM
Ia.x.inp b:y.out $-p
```

4.8. L (List code) - Обратный ассемблер

Выполняется обратная трансляция машинного кода, на экране отображаются адрес команды в 16-ричной системе и мнемокод команды. Если директива начинается со знака "минус", то адреса и метки не отображаются.

Ls - Ls - Начиная с адреса s, до заполнения половины экрана

Ls,f - Ls,f - В диапазоне адресов s...f нажатие любой клавиши прекращает вывод

L -L - Начиная с последнего адреса, обработанного директивами A,L или T, до заполнения половины экрана

Коды, не входящие в набор команд КР580ВМ80, выводятся в виде: ?? = nn, где nn = шестнадцатеричное значение байта.

Примеры:

```
L1Ø0            LICALL,+3Ø
L#1024,#1034    -L.PRBUFF+=1,+'A'
L.CRLF
```

4.9. M (Move memory) - Пересылка содержимого памяти

M s,h,d - Находящиеся на адресах s...h данные пересылаются в область памяти с начальным адресом d. Исходная и целевая области могут перекрываться.

Примеры:

```
M1Ø0,1FF,3Ø0      M.GAMMA,+FF,.DELTA
M.X,.Y,.Z         Malpha+=X,+#5Ø,+1Ø0
```

4.10 P (Pass counter) - Задание контрольных точек

Контрольная точка - это значение счётчика команд PC, при достижении которого проверяется содержимое регистров и выводится при желании на экран. К каждой контрольной точке принадлежит счётчик, значения которого находятся в пределах Ø...FFH. Значение счётчика уменьшается на единицу при каждом достижении контрольной точки. Если счётчик принимает значение 1, контрольная точка автоматически превращается в постоянную точку. В отличие от временной точки останова (см. директиву G) контрольная точка прерывает выполнение программы после выполнения команды, находящейся на заданном адресе.

```
Pp      - Контрольная точка устанавливается на адресе p; значение
          счётчика точки равно 1
Pp,c    - То же, значение счётчика равно c
P       - На экран выводятся адреса всех контрольных точек и значе-
          ния их счётчиков
-Pp     - Аннулирование контрольной точки на адресе p
-P      - Аннулирование всех контрольных точек
```

Одновременно могут существовать не более 8 контрольных точек. При каждом достижении контрольной точки выводится текст:

```
nn PASShhhh.ssss, где
nn      - значение счётчика точки
hhhh    - адрес
.ssss   - метка, если она существует.
```

Директивы -G и -U блокируют вывод содержимого регистров до приобретения счётчиком значения 1. Выполнение программы можно прекратить, нажав на любую клавишу.

Примеры:

```
P1Ø0,ff     PICALL+3Ø,#2Ø
P.BDOS      -P.CRLF
```

4.11. R (Read code/symbols) - Загрузка

Перед загрузкой задаётся имя (имена) загружаемого файла (файлов) директивой I. В зависимости от параметров этой директивы выполняется загрузка программы и/или таблицы меток.

Базовые адреса зон загрузки:
программа: 100H,
таблица меток: конец свободной зоны памяти.

```
R   - загрузка начинается с базового адреса
Rd  - загрузка со смещением d,т.е. начальный адрес будет база + d
```

Варианты применения:

```
Ix,y     - файл машинного кода (в нормальном случае y=COM) загру-
R          жается на адрес 100H; если y=OBJ, то файл должен быть
           в об ектном формате LOS
Ix,y u.v- кроме файла машинного кода загружается таблица меток
R          (в нормальном случае v=SYM)
I*u.v    - начальная часть ОЗУ на изменяется, загружается только
R          таблица меток
```

При загрузке таблицы меток на экран выводится текст

```
Symbols
```

Сообщения об ошибках:

```
?Symbols - отказ при загрузке машинного кода
Symbols? - отказ при загрузке таблицы меток
```

Примеры:

```
ICOPY.COM            Imest.com mest.sym
R                    R1000
ISORT.OBJ SORT.SYM   I*test.sym
R                    R-*256
```

4.12. S (Set memory) - Запись в память

```
Ss    - ввод байтов, запись начинается с адреса s
SWs   - ввод 2-байтовых слов, запись начинается с адреса s
```

В обоих случаях на экране отображаются адрес и содержимое ячеек. При вводе пустой строки (только RETURN) содержимое ячеек ОЗУ не изменяется, значение адреса увеличивается на единицу записи (байт или 2 байта). При вводе нового текста изменяется содержимое ОЗУ и значение адреса увеличивается как описано выше. Ввод завершается нажатием клавиши RETURN. Цепочки символов вводят, применяя формат ввода байтов, цепочка должна быть заключена в кавычки.

Примеры (введённые пользователем значения выделены жирным шрифтом):

```
S100              SW.X+#30
0100 C3 34        2300   006D   44F
0101 24 #254      2302   4F32   GAMMA
0102 CF           2304   33E2
0103 4B"ASCII"    2306   FF11 0+.X=1- '20
0108 6E = X+5     2308   348F.
0109 B4.
```

4.13. T (Trace mode) - Режим трассировки

Перед выполнением трассируемой команды на экран выводятся содержимое регистров и мнемокод команды, полученный путём обратной трансляции. Если перед директивой знак "минус", то обратная трансляция операндов и меток не осуществляется и программа выполняется быстрее.

Для возврата из режима трассировки следует нажать на любую клавишу.

```
Tn  -Tn     - выполняются n машинных команд
T   -T      - выполняется одна машинная команда
TWn -TWn    - n команд, без трассировки подпрограмм
TW  -TW     - одна команда, без трассировки подпрограмм
```

Примеры:

```
T100
-TW#10
```

4.14. U (Untrace mode) - Пошаговый режим

Режим предусматривает те же функции, что и T, но без отображения содержимого регистров. Для возврата из этого режима нажимают на любую клавишу.

```
U...
```

Примеры:

```
Uffff
UW#10
```

4.15. X (eXamine CPU state) - Вектор состояния программы

Эта директива позволяет проверить и изменить содержимое регистров и значения флажков.

Приняты следующие обозначения флажков:

C - перенос
Z - нулевой результат
M - отрицательный результат
X - вектор состояния, выводится в виде:

E - чётный результат
I - декадный перенос

<f>A=<a>B=<b>D=<d>H=<h>S=<s>P=<p><i><s>,

где

f - значения флажков (1: имя флажка, Ø: минус)
a - содержимое регистра A
b - содержимое регистров BC
d - содержимое регистров D,E
h - содержимое регистров H,L
s - значение указателя стека SP
p - значение счётчика команд PC
i - код последней машинной команды
s - метка, если она существует: в случае команд с косвенной адресацией (напр. INR M, ADD M) на месте s выводится содержимое соответствующей ячейки памяти перед выполнением команды

Xf - отображение значения флажка f (=C,Z,M,E или I);
для изменения значения ввести Ø или 1
Xr - Изменение содержания регистра r (=A,B,D,H,S или P)

Примеры (вводимые пользователем данные обозначены жирным шрифтом):

```
XM      XB              XP
MO      3EO4 3EFFF      446E .CRLF+1Ø
```

"ЭКТА"
СКБ вычислительной техники Института кибернетики АН ЭССР

ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ИНТЕЛЛЕКТУАЛЬНОГО ТЕРМИНАЛА Е5104

СИСТЕМА УПРАВЛЕНИЯ БАЗАМИ ДАННЫХ

DBASE

353872.3Ø023-24

Руководство оператора

19 стр.

Таллинн 1988

СОДЕРЖАНИЕ

1. ОБЩИЕ СВЕДЕНИЯ О ПРОГРАММЕ

ЕКТА Tallinn *ЭКТА* Таллинн КОМПОНЕНТ / КОМПЛЕКС ПО

DBASE	353872.30023-24	Версия: 2.4
СИСТЕМА УПРАВЛЕНИЯ БАЗАМИ ДАННЫХ		
ОБ'ЕМ 128К байт	ОПЕРАЦ.СИСТЕМА:	CP/M , EKDOS
ТРЕБУЕМОЕ ОЗУ 48К байт		
НОСИТЕЛЬ: ГМД РЕГ. No НОСИТЕЛЯ:		
НАЗНАЧЕНИЕ И МЕТОД: Создание возможности манипулирования базами данных небольшого и среднего об'ема. Система запускается на уровне операционной системы командой DBASE		

ОРГАНИЧЕНИЯ:

Число записей в файле данных	65535
Число полей символов в записи	32/1000
Число символов в поле	254
Представление чисел	+-1,8 x 10 ...+-1,0 x 10
Разрядность чисел	10
Длина строки символов/команды	254/250
Длина заголовка отчета	254
Число выражений в команде SUM	5
Длина индексного ключа	100

ПРОЦЕССОР / ЭВМ : КР580ВМ80 / Е5104

ВНЕШНИЕ УСТР-ВА: Видеомонитор, НГМД

НЕОБХ. ПРОГРАММЫ И ФАЙЛЫ:

DBASE.COM	DBASEMSG.TXT
DBASEOVR.COM	DBINST.COM

АРХ. МЛ:	ПРОГРАММИСТ:	ЗАРАНС

2. ОСНОВНЫЕ ФУНКЦИИ СИСТЕМЫ

Система DBASE является средством управления базами данных, которые позволяет легко манипулировать базами данных небольшого и среднего об'ема с помощю команд, использующих органиченный английский язык. Система представляет следующие возможности:

- создание полных систем баз данных;
- простые для использования средства добавления, исключения, редактирования, вывода на экран и печать данных из базы данных; при этом обеспечивается минимальное дублирование информации в файле;
- значительную степень независимости от данных и программ, так что при изменении данных отсутствует необходимость модификации прикладных программ и наоборот;
- генерацию отчетов из одной или нескольких баз данных с автоматическим выполнением операций умножения, деления, вычисления промежуточных и окончательных итогов, а также других аналогичных операции;
- средства экранного редактирования, обеспечивающие возможность установки форматов экрана для вывода и ввода информации.

3. ВЫЧИСЛИТЕЛЬНАЯ СРЕДА

Для функционирования системы DBASE необходимо следующая програмная среда и оборудование:

1. ЭВМ на базе микропроцессора 8080, 8085, Z80 или соответствующего, с операционной системой, совместимой с CP/M.
2. Не менее 48 килобайт ОЗУ (система DBASE использует адрес от 5CH до A400H).
3. Один или несколько НГМД.
4. Дисплей (видеомонитор), поддерживающий операции в экранном режиме, с применением курсора.
5. Печатающее устройство (факультативно).

4. ТИПЫ ДАННЫХ И ФАЙЛОВ

Каждое поле в записи должно содержать одиноковой тип данных. В системе допускаются следующие типы данных:

- символьный: все видимые символы кода ASCII , включая числа, специальные символы и пробелы;
- числовый: положительные и отрицательные числа в диапазоне от 1,0 x 10 до 1,8 x 10 ; Максимальная разрядность составляет 10 цифр, а максимальной формат при включении знака доллара равен 99.999.999.99;
- логический: значения истина/ложь (или да/нет), занимающие один байт; Система распознавает как истинные следующие значения: T, t, Y, y, а как ложные F, f, N, n.

Файл системы DBASE представляет собой просто набор информации однородного типа. Система поддерживает следующие типы файлов:

.DBF - файлы баз данных
Это те файлы, в которых хранится информация. Во время выполнения команда CREATE. Система DBASE автоматически присваивает это расширение имени новому файлу.

.FRM - файлы форм отчетов
Такие файлы автоматически создаются системой во время выполнения команды REPORT. Файлы содержат заголовки отчетов, колонок и другие спецификации отчета и могут быть модифицированы с помощью текстового редактора ; тем не менее не рекомендуется использование редактора - лучше использовать для этой цели систему DBASE.

.CMD - командные файлы
Эти файлы содержат последовательность команд системы DBASE, предназначенные для выполнения наиболее часто используемых функций. Командные файлы могут создаваться с помощью текстового редактора или команды MODIFY COMMAND.

.NDX - индексные файлы
Эти файлы автоматические создаются системой DBASE с помощью команды INDEX. Индексирование обеспечивает очень быструю локализацию данных в больших базах данных.

.MEM - файлы переменных
Эти файлы автоматически создаются системой DBASE во время выполнения команды SAVE, записывающей на дискете результаты вычислений, константы или переменные, которые нужно использовать позже. Возможно сохранение до 254 символов; восстановить их можно с помощью команды RESTORE.

.TXT - текстовые выводные файлы
Эти файлы автоматически создаются во время выполнения команды SET ALTERNATE, которая записывает на дискете всю информацию, появляющуюся на экране. Такая функция может использоваться в целях регистрации выполненных операции. Сохраненная информация может быть затем отредактирована и напечатана. Текстовые файлы создаются также с помощью команды COPY...SDF.

5. ОБЗОР ОПЕРАЦИЙ СИСТЕМЫ

Арифметические операторы (генерируют арифметические результаты)

```
( )        - скобки для группирования
 *         - умножение
 /         - деление
 -         - вычитание
```

Операторы сравнения (генерируют логические результаты)

```
 <         - меньше чем
 >         - больше чем
 =         - равно
 # или <>  - не равно
 <=        - меньше чем или равно
 >=        - больше чем или равно
```

Логические операторы (генерируют логические значения T/F)

```
 ()        - скобки для группирования
 .NOT.     - булевское "отрицание"
 .AND.     - булевское "и"
 .OR.      - булевское "или"
 $         - логический оператор поиски подстроки
```

Строковые операторы (генерируют строковую результаты)

```
 +         - сцепление (соединение) строк
 -         - сцепление строк с удалением пробелов
```

6. ОБЗОР ФУНКЦИЙ СИСТЕМ

Здесь и далее внастоящем разделе применяются следующие условные обозначения:

<выр> - выражение
<пер> - переменная
<стр> % строка
<пстр> - подстрока
<дл> - длина
<нач> - начало
<сфд> - сфера действия
<коор> - координаты
<точ> - точность

Система поддерживает следующие функции:

#	- номер записи
*	- запись исключена?
EOF	- конец файла
!(<пер/стр>)	- перевод в прописные символы
TYPE(<выр>)	- тип данных
INT(<пер/выр>)	- целая цункция
VAL(<пер/стр/пстр>)	- преобразование строки и число
STR(<выр/пер/число>,<дл>,<точ>)	- преобразование числа в строку
LEN(<пер/стр>)	- длина строки
$(<выр/пер/стр>,<нач>,<дл>	- выборка подстроки
(<пер1/стр1>,<пер2/стр2>)	- поиск подстроки
CHR(<число>)	- преобразование числа в код ASCII
&(<пер>)	- макроподстоновка
FILE(<"имя файла"/пер/выр>)	- файл существует?
TRIM(<стр>)	- удаление завершающих пробелов
RANK(<стр>)	- получение значения символа в коде ASCII

7. ОБЗОР КОМАНД СИСТЕМЫ

При описаний команд системы символы <...> определяют элементы, указываемые пользователем. Квадратные скобки [...] ограничивают необязательные элементы, которые, в свою очередь, также могут содержать [...]

? <выр[список]>
Вывод значения выражения или списки элементов, разделенных запятыми.

<коор>[SAY<выр>[USING'формат']] [GET<пер>[PICTURE'формат']]
Форматирование экрана терминала или печатного выводе

ACCEPT['сообщение'] TO <пер>
Ввод символьной строки, указанной без апострофов, с консоли

APPEND [BLANK]
APPEND FROM <имя файла> [SDF] [FOR <выр>]
[DELIMITED] [FOR <выр>]
Добавление данных в базу данных

CANCEL
Прекращение выполнении командного файла

CHANGE [<фд>] FIELD<список> [FOR <выр>]
Выполнение многих изменений базы данных

CLEAR
Установка файлов данных и начальное состояние и удаление переменных

CONTINUE
Продолжение команды LOCATE

COPY [<фд>] TO <имя файла> [SDF]
[STRUCTURE] [FIELD <список>][FOR <выр>]
[DELIMITED] [WITH <разделитель>]
Копирование данных из базы данных в другой файл

COPY TO <имя файла> STRUCTURE EXTENDED
Создание нового файла .DBF, записи которого описывают структуру <старого файла> (См. также команду CREATE <новый файл> FROM <старый файл>)

COUNT [<сдф>] [FOR <выр>] [TO <пер>]
Подсчет числа записей, удовлетворяющих некоторому условию

CREATE [<имя файла>]
Создание новой базы данных

CREATE <новый файл> FROM <старый файл>
Создание <нового файла>, структура которого описывается записями <старого файла>.(См. также команду COPY STRUCTURE EXTENDED

DELETE [сфд] [FOR <выр>]
Установление признака исключения для указанных записей

DELETE FILE <имя файла>
Удаление из системы указанного файла

DISPLAY [сфд] [FOR <выр>] [OFF]
Вывод данных по запросу пользователя

DISPLAY [сфд] [<поле>,[<список>]]
Вывод указанных полей

DISPLAY STRUCTURE
Вывод структуры открытой базы данных

DISPLAY MEMORY
Вывод значений временных переменных

DISPLAY FILES [ON дисковое устройство]
Вывод справочника диска

DISPLAY STATUS
Вывод перечня открытых файлов данных, индексных файлов и ключей и параметров GET

DO <имя файла>
Выполнение командного файла

DO WHILE <выр>
Многократное выполнение группы команд

EDIT
Изменение данных в базе данных

EDIT [число]
Редактирование указанной записи

EJECT
Перевод страницы на принтере

ELSE
Альтернативный путь выполнения в команде IF

ENDDO
Конец команды DO WHILE

ENDIF
Конец команды IF

ENDTEXT
Конец команды TEXT

ERASE
Очистка экрана

FIND <ключ>
Локализация записи по значению ключа в индексированной базе данных (апострофы для символьных ключей не требуются)

GO или GOTO [RECORD], или [TOP], или [BOTTOM]
Позиционирование в базе данных

HELP [<команда>]
Вывод краткой справочной информации о команде системы

IF <выр>
Выполнение команд по условию

INDEX ON <ключ> TO <имя файла>
Создание для базы данных индексного файла

INPUT ['сообщение'] TO [пер]
Ввод данных пользователя и сохранение их значений во временных переменных

INSERT [BEFORE]
[BLANK]
Включение в базу данных новой записи

JOIN TO <имя файла> FOR <выр> [FIELDS <список>]
Создание новой базы данных с помощью операции соединания двух других баз данных.

LIST
Вывод содержимого записей

LOCATE [сфд] [FOR <выр>]
Поиск записи по условию

LOOP
Механизм выхода из группы команд, организированной с помощью DO WHILE

NOTE или *
Комментарий, включаемый в командный файл и не выводимый при его выполнении

MODIFY COMMAND <имя файла>
Модификация файла в режиме системы DBASE II

MODIFY STRUCTURE
Модификация структуры базы данных, разрушающая все данные

PACK
Физическое удаление, помеченных для исключения записей

QUIT [TO список команд операционной системы или командных файлов]
Завершение выполнения системы DBASE II и выполнение последовательности команд или программ. Каждая команда должна указываться в апострофах, а команды разделяться запятыми.

READ
Инициация операции редактирования на полном форматированном экране. Ввод данных по командам GET.

RECALL [сфд] [FOR <выр>]
Удаление признаков исключенных записей

RELEASE [<пер>,[список]] или [ALL]
Удаление ненужных временных переменных

REMARK
Комментарии, выводимые на экран во время выполнения командного файла.

RENAME <старый файл> TO <новый файл>
Переименование файла

REPLACE [сфд]<поле> WITH <выр>,[<поле> WITH <выр>...][FOR <выр>]
Изменение данных в базе данных. Вы должны иметь копию базы данных до выполнения этой команды во избежание возможных пользовательских ошибков.

REPORT [сфд] [FORM <имя файла>] [TO PRINT] [FOR <выр>]
Генерация отчета

RESET
Информация для системы DBASE о том, что может быть выполнена смена дискеты.

RESTORE FROM <имя файла>
Восстановление значения сохраненных временных переменных. Все существующие переменные разрушаются.

RETURN
Завершение текущего командного файла и возврат в вызвавший его командный файл

SAVE TO <имя файла>
Запись значений временных переменных в файл для последующего использования

SELECT [PRIMARY] или [SECONDARY]
Переключение рабочих областей

SET параметр [ON], или [OFF], или [TO <условие, имя файла>]
Динамическая реконфигурация системы DBASE

SKIP <выр/число>
Перемещение в базе данных вперед и назад

SORT ON <ключ> TO <имя файла> [ASCENDING]
[DESCENDING]
Генерация базы данных, отсортированой по значению поля

STORE <выр> TO <пер>
Помещение значения во временную переменную

SUM [сфд] <поле [,список]> [TO <пер [,список]>[FOR <выр>]
Суммирование полей в базе данных

TEXT
Вывод блока текста вплоть до команды ENDTEXT без специального форматирования

TOTAL TO <имя файла> ON <ключ> [FIELDS <поле [,список]>
Генерация базы данных с вычислением для записей подитогов

UPDATE FROM <имя файла> ON <ключ> [ADD <поле[,список]>]
[REPLACE <поле [,список]>]
Модификация базы данных на основе информации другой базы данных

USE <имя файла> [INDEX <имена файлов>]
Открытие базы данных

USE
Закрытие всех открытых файлов баз данных

WAIT [TO<пер>]
Прерывание выполнения программы и ожидание ввода

8. ОБЗОР КОМАНД ПО ВЫПОЛНЯЕМЫМ ФУНКЦИЯМ

8.1. Структура файлов

CREATE
Определение полностью новой структуры файла

CREATE <новый файл> FROM <старый файл>
Создание нового файла, структура которого описывается записями старого файла

USE <старый файл>
COPY TO <новый файл> STRUCTURE
Две команды, создающие новый файл такой же структуры, как и старый файл

USE <старый файл>
COPY TO <новый файл> STRUCTURE EXTENDED
Создание нового файла, записи которого содержат описание структуры старого файла

CREATE <новый файл> FROM <старый файл>
Создание нового файла, структура которого описана в записях старого файла

DISPLAY STRUCTURE
Вывод структуры открытого файла

MODIFY STRUCTURE
Модификация имени файла, его размера и общей структуры. Разрушение всех данных.

8.2. Изменение структуры базы данных, содержащей данные:

```
USE <старый файл>
COPY TO <новый файл>
USE <новый файл>
MODIFY STRUCTURE
APPEND FROM <старый файл>
COPY TO <старый файл>
USE <старый файл>
DELETE FILE <новый файл>
```

8.3. Переименование полей базы данных, содержащих данные:

```
USE <старый файл>
COPY TO <новый файл> SDF
MODIFY STRUCTURE
APPEND FROM <новый файл> .TXT SDF
DELETE FILE <новый файл>
```

8.4. Операции над файлами:

USE <имя файла>
Открытие файла

USE <новый файл>
Закрытие старого файла и открытие нового

USE
Закрытие всех файлов

RENAME <старое имя> TO <новое имя>
Переименование файла (не должно выполняться для открытого файла)

COPY TO <имя файла>
Создание резервной копии

CLEAR
Закрытие всех файлов и исключение всех временных переменных

SELECT [PRIMARY] [SECONDARY]
Обеспечение открытия двух файлов независимо друг от друга. Данные могут пересылаться из одной рабочей области в другую с использованием префиксов P. и S.

DISPLAY FILES [ON <диск>]
Вывод перечень файлов, расположенных на указанном или такующем устройстве. Может также использоваться команда LIST

DISPLAY FILES LIKE <маска> [ON <диск>]
Вывод перечня файлов, расположенных на диске, по маске

QUIT
Завершение выполнения системы DBASE , закрытие активных рабочих областей и всех файлов

8.5. Сортировка и индексирование

SORT ON <ключ> TO <новый файл>
INDEX ON <ключ> TO <новый файл>
Для обеих команд можно использовать несколько ключей

8.6. Об'единение баз данных

COPY TO <новый файл>
Создание копии открытого файла

APPEND FROM <другой файл>
Добавление записей в открытый файл

UPDATE FROM <другой файл> ON <ключ>
Добавление или замещение данных в открытом файле. Оба файла должны быть отсортированы по <ключу>

JOIN
Соединение двух файлов в третий файл

8.7. Редактирование, обновление и изменение данных

DISPLAY, LIST, BROWSE
Вывод и просмотр содержимого записей

DELETE
Установка признака исключенной записи

RECALL
Удаление признака исключением записи

PACK
Физическое удаление записей, помеченных для исключения

EDIT
Обновление отдельных записей

REPLACE <поле WITH данные>
Глобальное изменение содержимого полей данных. Команда может спецификацию условия, как и большинство других команд системы DBASE

@ <коор> GET <пер>
READ
Вывод, ввод и изменение значении переменных

INSERT [BEFORE] [BLANK]
Включение в файл новой записи

UPDATE FROM <другой файл> ON <ключ>
Продавление или замещение данных в открытом файле из другого файла

MODIFY COMMAND <имя файла>
Модификация командного файла в режиме системы DBASE

8.8 Использование переменных

Допускается использование до 64 переменных и любое число имен полей.
LIST MEMORY, DISPLAY MEMORY
Выводит значения переменных, а также их тип данных

STORE <значение TO <пер>
Присваивание значений временным переменным

RELEASE <пер>
Исключает указанную переменную

SAVE MEMORY TO <имя файла>
Сохраняет значения временных переменных в указанном файле, имеющем расширение .MEM

RESTORE FROM <имя файла>
Чтение значения временных переме

8.9. Интерактивный ввод данных.

WAIT
Остановка выполнений программы и ожидание нахатия любого клавиши

WAIT TO <пер>
Ввод символа во временную переменную

INPUT [´сообщение´] TO <пер>
Ввод данных любого типа во временную переменную и ее создание, если она несуществовала до выполнения команды. Символьный ввод должен указываться в апострофах

ACCEPT [´сообщение´] TO <пер>
Выполняет такие же действия, как и команда INPUT. Вводимые символьные данные не требуют апострофов

<коор> SAY [´сообщение´] GET <пер> [PICTURE]
READ
Вывод значений переменных и замещение старах значений на новые

8.10. Поиск данных

SKIP [+-<выр>]
Пропуск определенного числа записей "вперед" и "назад"

GO [TO] <число>, GO TOP, GO BOTTOM
Позиционирование на указанную запись, первую или последнюю запись базы данных

FIND <стр>
Быстрая локализация записи в индексированной базе данных

LOCATE FOR <выр>
CONTINUE
Поиск в полной базе данных

8.11. Вывод данных

?, DISPLAY, LIST
Вывод значений выражений, записей, переменных и структур

```
REPORT [FROM <имя формы>]
     Создание формата отчета и его вывод

 <коор> SAY <пер/выр/стр>
     Форматирование экрана или принтера для вывода информации.
     Может быть добавлена спецификация [USING <формат>], обес-
     печивающая формат PICTURE для печати

TEXT
     Вывод блока текста без специального редактирования
```

8.12. Программирование

Программы хранятся в командных файлах, имеющих расширение .CMD

```
DO <имя файла>
     Вызов программы для выполнения

IF <условия>                          Выбор одной или нескольких
     выполнение команд                альтернатив

ELSE
     Выполнение других команд

ENDIF

DO WHILE <условия>                    Организация повторяющегося
     Выполнение команд                процесса (процесс должен
                                      модифицировать заданные ус-
                                      ловия)

ENDDO

DO CASE                               Выполнение многих альтернатив
   CASE <выражение>
        <команды>
   CASE <выражение>
        <команды>
OTHERWISE
        <команды>
ENDCASE
```

"ЭКТА"
СКБ Вычислительной техники Института кибернетики АН ЭССР

Пакет таблицных расчетов

MULTIPLAN

353872.30004-15

18 стр.

Таллинн 1987

СОДЕРЖАНИЕ

1.ОБЩИЕ СВЕДЕНИЯ О ПРОГРАММЕ

ЕКТА Tallinn *ЭКТА* Таллин КОМПОНЕНТ / КОМПЛЕКС ПО

MULTIPLAN	353872.30004-15	Версия: 1.05

ПАКЕТ ДЛЯ ТАБЛИЧНЫХ РАСЧЕТОВ

ОБЪЕМ 124К байт | ОПЕРАЦ. СИСТЕМА: CP/M

ТРЕБУЕМОЕ ОЗУ | Программа 30К байт | Данные 13К байт

НОСИТЕЛЬ: ГМД
РЕГ. No НОСИТЕЛЯ:

НАЗНАЧЕНИЕ И МЕТОД:
Создание и обработка расчетных таблиц в диалоговом режиме, для решения различных экономических, инженерных, научных и других задач. При изменение содержания какой-либо клетки возможен автоматический пересчет всей таблицы. Таблица может содержать числовые, логические и символьные переменные, в том числе текст. Созданные таблицы можно записывать на ГМД. Возможны мехтабличные операции.

ССЫЛКИ:
Пакет совместим с одноименным пакетом фирмы "Майкрософт"

ТЕХН. ХАР-КИ:
40 функций, в том числе
24 математических
20 функций общего назначения

ОГРАНИЧЕНИЯ:
не более 255 строк
63 столбцов

ПРОЦЕССОР / ЭВМ: КР580ВМ80А/Е5104

ВНЕШНИЕ УСТР-ВА:
Видеомонитор
НГМД

СОСТОИТ ИЗ ФАЙЛОВ:
MP.COM (20К)
MP.OVR (44К)
MP.HLP (40К)

ВЫЗЫВАЕТ МОДУЛИ ИЗ MP.OVR, MP.HLP

22.10.1987	АРХ. №: 3.	ПРОГРАММИСТ: Заранс, Н

2. СВОДКА ДИРЕКТИВ ОПЕРАТОРА

2.1. Перемещение указателя, прокрутка экрана

CTRL E (вверх) CTRL X (вниз) CTRL S (налево) CTRL D (направо)	(Клавиши направления) Перемещение указателя клетки таблицы в соответствующем направлении. После перемещения указателя до края экрана (окна) начинается прокрутка экрана для ввода в окно следующих клеток.
CTRL Q	("Начало") Возврат указателя в левую верхнюю позицию (строка 1, столбец 1)
CTRL Z	("Конец") Перемещение указателя в нижний правый угол активной части расчетной таблицы.
CTRL W	("Следующее окно") Перемещение указателя в следующее окно.
CTRL F LINE FEED	("Следующая неблокированная клетка") Перемещение указателя на следующую неблокированную непустую клетку.
CTRL R +клавиша направления	("Прокрутка страницы") Прокрутка для показания в окне следующей области таблицы, находящейся в указанном направлении.

См. также команды перехода (GOTO)

2.2. Операционные клавиши

CTRL C	("Аннулирование") Аннулирование текущей операции и возврат в основное командное меню.
RETURN	Запуск команды выбранной из меню, или выполнение завершенной команду.
Пробел	Выбор следующей услуги в меню.
Клавиша возврата на один шаг, или CTRL H	Выбор предыдущей услуги в меню.
TAB, CTRL A, CTRL I	Переход на следующее поле командный строки и выбор содержания этого поля.

?	Запрос информации о выбранной команде или о текущей выполняемой команде.
!	Пересчет всей расчетной таблицы. Если была введена формула, то она заменяется результатом расчета по этой формуле.
=, +, Ø, 1...9	Активизация команду ввода значения (VALUE).

2.3. Клавиши редактирования

Клавиши редактирования могут быть использованы всегда после ввода параметров команд. Текст может вставляться перед выбранной позиции.

Исключения:

1. После ввода таб (смотри выше) существующий текст заменяется новым вводимым текстом.
2. На полях ссылок вводимые знаки добавляются к существующим при возможности.

Клавиша возврата на один шаг	Стирание знака левее курсора
CTRL Y DELETE	("Стирание") Стирание выбранных знаков. Возможен ввод заменяющего текста.
CTRL K	("Шаг налево") Выбор знака левее текущей позиции.
CTRL L	("Шаг направо") Выбор знака правее текущей позиции.
CTRL O	("Слово налево") Выбор слова левее текущей позиции.
CTRL P	("Слово направо") Выбор слова правее текущей позиции.
@	("Ссылка") Замена относительных ссылок абсолютными.
Клавиши направления	1) Могут быть исползованы для вставления относительной ссылки на указанную клетку; 2) Изменение значения параметра команды в допустимых пределах; 3) Вставление имени в формуле (после).

2.4. Команды

ALPHA:...
Переход на ввод текста в клетки таблицы (и на редактирование).

BLANK клетки:...
Стирание содержания указанной клетки (указанных клеток).

COPY .....

COPY RIGHT число клеток:.... начиная с:....
Копирование содержания заданного числа клеток, начиная с клетки справа от указанной.

COPY DOWN число клеток:.... начиная с:....
Копирование содержания заданного числа клеток, начиная с клетки под указанной.

COPY FROM клетки:..... к клетки:.....
Копирование содержания с исходной клетки (исходных клеток) в другую клетку (другие клетки).

DELETE .....

DELETE ROW число строк:.... начиная с:....
между столбцами:.... и:....
Удаление строк(и) между указанными столбцами.

DELETE COLUMN число строк:.... начиная с:....
между столбцами:.... и:....
Удаление столбцов между указанными строками.

EDIT .....
Пересылка содержания активной клетки на командную строку, для редигирования (текст обозначен кавычками). При нажатии на клавишу RETURN исходный текст в клетке заменяется отредактированным.

FORMAT .....

FORMAT клетки:.... Выравнивание: Def Ctr Gen Left Right -
Код формата: Def Cont Exp Fix Gen Int $ * % -
Число десятичных знаков:....
Определение описания формата указанной клетки (указанных клеток) На поле "число десятичных знаков" влияет параметры Exp, Fix и %. Числа будут округляны до заданной для отображения точности.

Выравнивание

Def	("По умалчанию") Выравнивание согласно параметрам команды FORMAT DEFAULT.
Ctr	Центрирование изображения клетки в пределах столбца.
Gen	("Общее") Выравнивание текста налево, чисел направо (начальная система выравнивания по-умолчанию).
Left	("Левое") Левое выравнивание изображения клетки.
Right	("Правое") Правое выравнивание изображения клетки.
-	Не влияет на выравнивание.

Кодирование формата

Def	("По умолчанию") Формат согласно параметрам команды FORMAT DEFAULT.
Cont	("Продолжение") Продолжение длинного текста через границы столбцов, при условии, что смежные клетки пусты и форматированы также по коду CONT.
Exp	("Экспонента") Числа отображаются в десятичной форме с множителям в виде степени десяти.
Fix	("Фиксирование") Отображением с числом десятичных знаков, заданным в поле "число десятичных знаков").
Gen	("Основной") Числа отображаются в наиболее подходящей форме, с учетом величины клетки и длины числа (Начальная система форматирования по умолчанию).
Int	("Целочисленный") Числа отображаются в виде целых чисел.
$	Отображение чисел начинается со знака $, которому следуют два десятичных знака. Отрицательные числа заключены в скобки.
*	("Гистограмма") Значение числа отображается в виде гистограммы, столбцы которой составлены из звездочек.

%	("Проценты") Число отображается в процентном выражении, т.е. умноженным на 100 и со знаком %.
-	Не влияет на формат.

FORMAT DEFAULT ..... ("Формат по умолчанию")

FORMAT DEFAULT CELLS Выравнивание: Ctr Gen Left Right
Код формата:Cont Exp Fix Gen Int $ * %
Число десятичных знаков:.....

Установление формата и выравнивания для клеток типа "Def".

FORMAT DEFAULT Ширина символов:.....

Установление ширины всех столбцов типа "d".

FORMAT OPTIONS запятые: Да/Нет Формулы: Да/Нет
("Факультативные элементы формата")
Запятые: Отображение всех чисел (за исключением типов Exp и Gen) с запятыми за каждой третей цифрой.
Формулы: Отображение формул вместо значений формул.

FORMAT WIDTH Символов или по умолчанию(Д):... Столбец:... до:...
("Ширина формата")
Установление ширины заданного диапазона столбцов.
Параметр "d" обозначает применение ширины, заданной командой "FORMAT DEFAULT Ширина".

HELP: ("Вспомогательные указания")

Resume Start Next PrEvious Applications Commands Editing Formulas Keyword

HELP Resume ("Возврат") Возврат в состояние расчетной таблицы, в котором дали запрос на вспомогательные указания.

HELP Start ("Начало") Начало текста вспомогательных указаний.

HELP Next ("Следующая") Переход на следующую страницу вспомогательного текста.

HELP PrEvious ("Предыдущая") Переход на предыдущую страницу вспомогательного текста.

HELP Applications ("Приложения") Перечисление ряда общих задач с соответствующими командами.

HELP Commands ("Команды") Описание выбора команд и перечисление всех команд.

HELP Editing ("Редактирование") Описание редактирования.

HELP Formulas ("Формулы") Перечисление всех формул и правил составления формул.

HELP Keyboard ("Клавиатура")
Описание клавиатуры

GOTO: Name Row-col Window ("Переход:Имя Строка, столбец окно")

GOTO Имя:....
Перемещение указателя клетки в верхний левый угол области, заданной именем.

GOTO Строка:.... Столбец:....
Перемещение указателя клетки в заданную клетку.

GOTO Номер окна:.... Строка:.... Столбец:....
Перемещение указателя клетки и самой клетки в верхний левый угол указанного окна.

INSERT: Row Column ("Вставка: Строка Столбец")

INSERT ROW Число строк:.... Перед строкой:....
Между столбцами:.... и:....
Вставление новой строки (новых строк) между указаннами столбцами.

INSERT COLUMN Число столбцов:.... Перед строкой:....
Между строками:.... и:...
Вставление нового столбца (новых столбцов) между указаннами строками.

LOCK: Cells Formulas ("Блокировка: Клетки Формулы")

LOCK Клетки:.... Состояние: Блокировка/Разблокировка
Блокирование или разблокирование указанной клетки (указанных клеток).

LOCK Формулы:....
Блокирование всех клеток, содержащие формулы или тексты. Клетки содержащие цифровые постоянные, останутся неизменными.

MOVE: Row Column ("Перемещение: Строка Столбец")

MOVE ROW От строки:.... В позицию перед строками:....
Число строк:....
Перемещение строк(и) целиком с одной позиции в таблице в другую.

MOVE COLUMN От столбца:.... В позицию левее столбца:....
Число столбцов:....
Перемещение столбцов целиком с одной позиции в таблице в другую.

NAME: Установить имя:.... для:....

Определение клетки или группы клеток.

OPTIONS Пересчет: Да/Нет Глушение: Да/Нет
("Факультативные параметры")
Пересчет: автоматический пересчет таблицы при изменений содержания какой-либо клетки.
Глушение: выключение звукового сигнала.

PRINT: Print File Margins Options ("Печать")

PRINT ON PRINTER:
Запуск печати на печатающем устройстве.

PRINT ON FILE:....
Вывод вместо печатающего устройства в файл.

PRINT MARGINS: Левое:.... Верхнее:.... Ширина печати:....
Длина печати:.... Длина страницы:....
("Поля при печати")
Левое: Левое поле, символов
Верхнее: Верхнее поле, строк
Ширина печати: Высота области печати на странице, символов
Длина печати: Высота области печати на странице, строк
Длина страницы: Длина страницы, строк

PRINT OPTIONS: Область:.... Конфигурация:....
Формулы: Да/Нет Строка-столбец: Да/Нет
("Факультативные параметры")
Область: Указание области таблицы для вывода на печать.
Конфигурация: Возможность выбора параметров печатающего устройства.
Другие факультативные элементы позволяют печатать формулы вместо их значений, или же печатать всю таблицу вместе с номерами строк и столбцов.

QUIT: ("Выход из программы ")

Завершение сеанса работы с "Мультипланом", при утверждении.

SORT Столбец:.... Между строками:.... и:....
---- Направление: > <
("Сортировка")
Сортировка в указанном диапазоне строк заданного столбца в возрастающей (>) или убывающей (<) последовательности значений; возможна сортировка как цифровых, так и текстовых значений.

TRANSFER..... ("Пересылка")

TRANSFER LOAD мя файла:.... ("Загрузка")
Загрузка расчетной таблицы с дискового файла. Для просмотра каталога и для выбора файла применяются клавиши управления курсором.

TRANSFER SAVE:Имя файла:... ("Спасение")
Запись расчетной табляцы в указанной файл.

TRANSFER CLEAR: ("Отирание")
Стирание цехой расчетной таблицы после подтверждения.

TRANSFER DELETE Имя файла:.... ("Удаление")
Удаление названного файла.

TRANSFER OPTIONS Режим: нормальный/символьный/прочий
Конфигурация....
("Факультативные параметры")
Модифицирование контекста следующей операции пересылки. "Режим" выбирает формат файла. "Конфигурация" определяет каталог или дисковод, для выбора файла.

TRANSFER RENAME Имя файла:.... ("Переименование")
Переименование расчетной таблицы на указанное имя.

VALUE:.... ("Значение")

Ввод значения или формулы в активную клетку. Значение вводится также при вводе символов =, +, -, ., ", или (: или цифр 0,1,...,9.

WINDOW:.... ("Окно")

WINDOW SPLIT: Горизонтальные Вертикальные Заголовки
("Разделение окна")

WINDOW SPLIT HORIZONTAL по строке:.... Соединение: Да/Нет
("Горизонтальное разделение")
Разделение окна вдоль экрана по указанной строке. Созданные два окна могут быть соединены для синхронной прокрутки.

WINDOW SPLIT VERTICAL По столбцу:.... Соединение: Да/Нет
("Вертикальное разделение")
Разделение окна вертикально по указанному столбцу.

WINDOW SPLIT TITLES: Число строк:.... Число столбцов:....
("Выделение окна заголовком")
Выделение вертикального окна содержащего заданное число столбцов и выделение горизонтального окна содержащего заданное число строк. Прокрутка синкронизована соответствующим образом.

WINDOW change border in window number:....
("Переключение обрамления окна номер")
Обрамление окна или удаление обрамления.

WINDOW CLOSE номер окна ("Закрытие окна")
Удаление окна с экрана.

WINDOW LINK Окно NO. .... с окном NO. .... ("Соединение")
Соединение: Да/Нет
Установление или удаление связи между окнами для синкранизованной прокрутки.

XTERNAL:..... ("Внешнее")

XTERNAL COPY С таблицы:.... Имя группы клеток:....
в:..... Соединение: Да/Нет
("Внешняя копия")
Копирование данных с позванный группы клеток внешней таблицы в активную таблицу. Факультативно может быть создано постоянное соединение между активной таблицей и источником данных.

XTERNAL LIST: ("бнешний список")
Отображение списка внешних таблиц, поддерживающих активную таблицу и зависящих от нее.

XTERNAL USE имя файла:.... вместо:....
Установление заменяющего имени для поддерживающей таблицы.

2.5. Формулы

Формулы могут состоять из постоянных, ссылок на клетки, и функций.

Численные постоянные

Могут быть написаны в обычном представлении (напр.,3.1416) или в "научном" представлении (напр.,1.5E6).

Строки текста

В составе формул должны быть заключены в кавычки (напр. "$").

Абсолютные ссылки

Rn или Cn	Указание номера строки n (от 1 до 255) или номера столбца n (от 1 до 63).
Rn:m или Cn:m	Диапазон строк или столбцов.

Относительные ссылки

R или C	Активная строка или активный столбец.
R[+n] или C[+n]	Строка n под активной строкой или столбец n правее активного столбца. Знак "+" может быть пропущен.
R[-n] или C[-n]	Строка n над активной строкой или столбец n левее активного столбца.

Формулы с указанием строки (R) и столбца (C) могут быть обьединены для обозначения пересечения ссылок; напр. рНцМ - абсолютная ссылка на одну клетку; RC [-1] - клетка левее активной клетки.

Имена

Должны начинается с буквы; могут содержать буквы, цифры, точки и чрточки для подчеркивания. Имена могут быть определяны для ссылания на любую клетку или группу клеток.

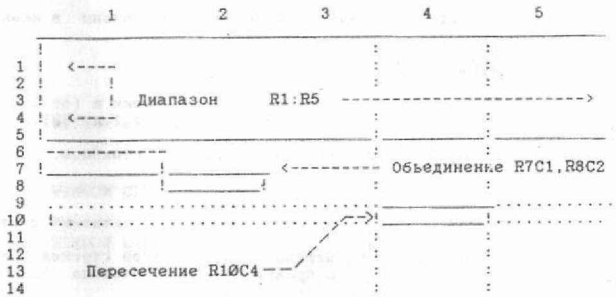
Операции над группами клеток

:	Задание диапазона: наименший прямоугольник, который содержит оба операнды (напр. R1:R5 обозначает строки от 1 до 5).

, Задание объединения (напр. R7C1, R8C2 обозначает клетку на строке 7, в столбце 1 и клетку на строке 8, в столбце 2).

(пробел) Задание пересечения: клетка (клетки) принадлежащая (-ие) обоим операндам (напр.R1ØC4 - единая клетка, в месте пересечения строки 1Ø со столбцом 4).

Пример



2.6. Функции

Операция над числовыми и текстовыми значениями

+	Сложение
-	Вычитание
/	Деление
*	Умножение
	Возведение в степень
%	Процент, тоже ,что "/1ØØ"
&	Соединение строк символов

Функции групп клеток

AND (Список) ("И")
Истинна тогда (и только тогда), когда все значения истинны, иначе ложна.

AVERAGE (Список) ("Среднее")
Среднее значение (=SUM/COUNT).

COUNT (Список) ("Счет")
Число значений, данных в виде аргументов или ссылок.

MAX (Список) ("Максимум")
Наибольшее из значений.

MIN (Список) ("Минимум")
Наименьшее из значений.

NPV (Тариф; Список) ("Суммарное значение")
Текущее суммарное значение денежных величин, представленных значениями, входящими в список.

OR (Список) ("Или")
Истинна тогда (и только тогда), когда любое из значений истинно, иначе ложно.

STDEV (Список) ("Среднеквадратное отключение")
Среднеквадратное отклонение значений.

SUM (Список) ("Сумма")
Сумма значений.

Математические, логические и текстовые функции

ABS (N) Абсолютное значение величины N.

ATAN (N) Арктангенс N, в радианах.

COS (N) Косинус угла N, в радианах.

COLUM () Текущий номер столбца.

DOLLAR (N) Строка текста, представляющая значение N в формате $.

EXP (N) e в степени N.

FALSE () Логическое значение "Ложно".

FIXED (N,m) Текст содержащий N в формате фиксированной запятой, с m десятичными разрядами, при m=0 формат соответствует целочисленному.

IF (Логическое выражение, Значение "истинно", Значение "ложно")
Результатом является Значение "истинно" при истинности Логического выражения, и Значение "ложно" при ложности Логического выражения.

INDEX (Область, Индексы)
Результатом является значение из клетки в области указанной индексами.

INT (N) Целочисленная часть N, усеченная к Ø.

ISERROR (Значение) Результатом является "истинно" тогда (и только тогда), когда значение = код ошибки.

ISNA (Значение) Результатом является "истинно" тогда (и только тогда), когда значение = £N/A.

LET (T) Длина текста T, символов.

LN (N) Натуральный логарифм N.

LOG 1Ø (N) Десятичный логарифм N.

LOOKUP (Значение, Подтаблица)
Если высота подтаблицы (прямоугольной группы клеток) больше ее ширины, программа начинает поиск последней такой строки, значение в первом столбце которой меньше или равно заданному значению. Результатом является значение в последнем столбце найденной строки. Если ширина подтаблицы больше ее высоты, то поиск выполняется вдоль строки.

Таблица столбцов

Значение между	! 5ØØ	x	x	5	!
1ØØØ и 1499 ---->	! 1ØØØ	x	x	7	! <---- Результат
	! 15ØØ	x	x	9	!
	! 2ØØØ	x	x	13	!
	! 25ØØ	x	x	17	!
	! 3ØØØ	x	x	22	!
	! 35ØØ	x	x	27	!
	! 4ØØØ	x	x	33	!
	! 45ØØ	x	x	38	!

MID (T,s,c) Длина (c) текста (T), в символах, начиная с (C).

MOD (Делитель, Делимое)
Остаток целочисленного деления "Делимое/Делитель".

NA () Результатом является код отсутствия (£н/а)

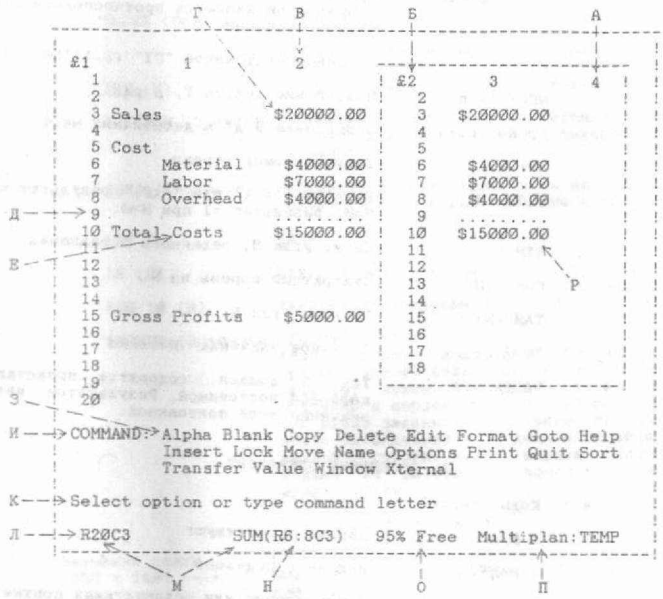
NOT (Логическая переменная)	Результатом является противоположное логическое значение.
PI ()	Значение постоянной "PI" (3.14159...).
REPT (T,n)	Повторение текста T, n раз.
ROUND (N,m)	Округление N до m десятичных мест.
ROW ()	Текущий номер строки.
SIGN (N)	Результат =-1 при N<Ø, результат =Ø при N=Ø, результат =1 при N>Ø.
SIN (N)	Синус угла N, заданного в радианах.
SQRT (N)	Квадратный корень из N.
TAN (N)	Тангенс угла N.
TRUE ()	Логичное значение "Истинно".
VALUE (T)	Текст T должен содержать представление цифровой постоянной. Результатом является значение этой постоянной.

3. СООБЩЕНИЯ ОПЕРАТОРУ

3.1. Коды ошибок

£ N/A	Данные отсутствуют
£ NAME?	Имя не определено
£ NUM!	Переполнение или недопустимая аритметическая операция
£ DIV/Ø	Деление на Ø
£ REF!	Ссылка на несуществующую клетку
£ NULL!	Пересечение несвязанных областей
£ VALUE!	Неправильный тип значения
£££££	Число слишком длинное для размещения в столбце

3.2. Пример отображения операции над таблицей



А - столбцы (1...63)
Б - активное обрамленное окно NO.2
В - кадр
Г - формат "$"
Д - строки (1...255)
Е - отображаемые данные
З - выбор по меню
И - командная строка
К - сообщение ("Выбрать вариант или ввести в букву команды")
Л - состояние
М - координаты (строка 20, столбец 3) и соодержание активной клетки
Н - абсолютная ссылка (строки 6:8, столбец 3)
О - свободная память
П - имя таблицы
Р - указатель клетки

СКБ вычислительной техники Института кибернетики АН ЭССР
"ЭКТА"

ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ИНТЕЛЛЕКТУАЛЬНОГО ТЕРМИНАЛА Е51Ø4

ЭКРАННЫЙ ТЕКСТОВЫЙ РЕДАКТОР
SED 6.1

РУКОВОДСТВО ОПЕРАТОРА

353872.3ØØØ5-6.1

17 стр.

Таллин 1987

АННОТАЦИЯ

Экранный текстовый редактор предназначен для ввода и редактирования различных текстовых документов, с возможностью записи документов на ГМД или кассетную магнетную ленту, и вывода на печать.

Программа предназначена для работы на Е5104, СМ-1800, "Robotron 1715", "Videoton", "Mostek" или "Labtam" в среде операционных систем CP/M или LOS. Созданные файлы могут обрабатываться при помощи других редакторов.

Основные функции и команды совместимы с системой "WORDSTAR". 3 режима работы, 55 основных команд.

СОДЕРЖАНИЕ

1.ОБЩИЕ СВЕДЕНИЯ О ПРОГРАММЕ

ЕКТА Tallinn *ЭКТА* Таллинн КОМПОНЕНТ / КОМПЛЕКС ПО

SED 6.1	353872.30005-61	Версия: 6.1
ЭКРАННЫЙ ТЕКСТОВЫЙ РЕДАКТОР		
ОБ'ЕМ 10К байт	ОПЕРАЦ. СИСТЕМА: CP/M, LOS, EKDOS	
ТРЕБУЕМОЕ ОЗУ	Программа 10к байт	Данные 352 байт
НОСИТЕЛЬ: ГМД / Кассетная МЛ РЕГ. No НОСИТЕЛЯ:		
НАЗНАЧЕНИЕ И МЕТОД: Ввод и редактирование текстовых документов, в том числе исходных текстов программ. Содержит команды для поиска, замещения, перемещения, стирания, печати текста, а также для объединения и разделения файлов, и для обработки блоков (отрезков) текста. Созданные файлы могут дополнительно обрабатываться любыми другими редакторами. В среде указанных ЭВМ и ОС редактор работает без необходимости переопределения структурных параметров.		
ССЫЛКИ: По основным функциям и командам совместим с WORDSTAR 3 режима, 55 основных команд		
ОГРАНИЧЕНИЯ: Обрабатываемый текст должен полностью разместиться в ОЗУ		
ПРОЦЕССОР / ЭВМ: КР580, Z80 / Е5104, СМ-1800, Mostek, Labtam, Robotron 1715, Videoton		
ВНЕШНИЕ УСТР-ВА: Видеомонитор НГМД или кассетный магнитофон Печатающее устройство		
НЕОБХ. ПРОГРАММЫ И ФАЙЛЫ: 1) Драйвер печатающего устройство 2) Файл ARUS - при отсутствии резидентного русского алфавита		
14.09.1987	АРХ. МЛ: 3.00	ПРОГРАММИСТ: Аменберг,А.
ИСХ. ТЕКСТ:	АРХ. МЛ: 3.00	ЯЗЫКИ: Ассемблер

2. ОБЩИЕ УКАЗАНИЯ ПО ПРИМЕНЕНИЮ

2.1. Режим работы программы

2.1.1.ЭКРАННЫЙ РЕЖИМ является основным режимом ввода и редактирования. Для выполнения различных функций вводят команды типа CTRL < символ >. Первая строка на экране (управляющая строка, УС) предназначена для отображения служебной информации - координат курсора и т.п. Знак "*" в УС обозначает режим замены.

Если соответствующим параметром команды SET не задана длина страницы, в управляющей строке отображаются
- номер строки, с отсчетом от начала документа,
- номер текущего столбца (позиции на строке).

Если же длина страницы задана, то УС будет содержать:
- номер страницы,
- номер строки, с отсчетом от начала страницы,
- номер текущего столбца (позиции на строке).

Знак "+" в конце строки обозначает продолжение строки, т.е. длина строки документа больше длины строки отображаемого кадра.

При нажатии на клавишу ESC в экранном режиме выводится перечень допустимых команд.

2.1.2. В РЕЖИМ ОСНОВНОГО МЕНЮ входят из других режимов командой CTRL Q.

Это меню содержит команды для загрузки файлов, для записи текста в файл, для поиска текста и.т.д. На экране выделено окно, в котором отображаются
- имя редактируемого файла,
- имя устройства по умолчанию
- об´ем текста в буфере, байт ("U="),
- об´ем свободной области буфера ("F=").

2.1.3. РЕЖИМ ОБРАБОТКИ БЛОКОВ содержит команды для маркировки, стирания, скопирования, записи блоков и т.д.

В этот режим входят командой CTRL P.

2.2. Общие указания по вводу текста

При вводе имен файлов, цепочек текста и параметров применяются следующие команды общего назначения:

RETURN - завершение ввода
ESC - прекращение ввода
CTRL H - стирание последнего введенного символа

2.3. Совместимость текстовых файлов

Создан.ые редактором SED текстовые файлы можно обрабатывать другими редакторами, без особых ограничений.

При загрузке текстовых файлов, созданных другими редакторами из этих файлов удаляют служебные коды (CTRL...). При загрузке документных файлов, созданных системой вордстар необходимо в связи с этим соблюдать некоторые меры предосторожности.

3. УПРАВЛЕНИЕ ПРОГРАММОЙ

3.1. Запуск редактора

После запуска редактора вычисляется контрольная сумма. При неправильном значении выводится сообщение

ERROR ("Ошибка")

При повторном запуске редактора командой RUN эту ошибку можно игнорировать.

Возможны следующие варианты запуска редактора:

1) A> SED RETURN

При отсутствии имени файла на командной строке выводится вопрос

EDIT FILE ("Редактировать файл")

требующий ввод имени файла редактируемого текста или имени нового создаваемого текстового файла. При нажатии на клавишу RETURN появится вопрос о выходе из редактора. При отрицательном ответе снова запрашивается имя редактируемого файла.

2) A> SED имя_файла Return

Файл с заданным именим загружается в буфер редактора. При отсутствии файла с данным именем выводится вопрос

Имя_файла IS NEW (Y/N) ("Новый файл? Да/Нет")

При положительном ответе начинается редактирование нового файла, при отрицательном следует снова ввести имя файла.

3) A> RUN Return ("Запуск")

При повторном запуске редактора вычисляемая контрольная сумма будеть неверной (это можно игнорировать)

4) A> RUN имя_файла Return

Загружается редактируемый файл и запускается редактор. Смотри также (3).

5) A> RUN $ Return

Повторный запуск. Сохраняется последний редактированный текст.

6) A> SED !

Применение функции операционной системы для вывода символов. По умолчанию для этого используется функция резидентного монитора.

7) A> SED .

Инициализация для ЭВМ "Videoton".

Для выхода из редактора переходят командой CTRL Q к основному меню и выбирают там функцию "X"(при выходе текст записывается в файл) или "Q"(без записи текста).

3.2. Система команд основного меню

R - TOP ("Начало")

Перемещение маркера в начало текста

C - BOTTOM ("Конец")

Перемещение маркера в конец текста

F - FIND ("Найти")

Поиск заданной текстовой цепочки (длиной до 20 символов). Если при вводе искаемого текста нажать на клавишу ESC, то выполнение команды прекращается. Для поиска можно вводить два параметра:

A - поиск начинается с начала текста
B - поиск начинается с позиции маркера и ведется в направлении начала текста

По умолчанию поиск выполняется начиная со следующего маркеру символа и ведется в направлении конца текста. Если искаемая текстовая цепочка не найдется, то выводится сообщение

NOT FOUND ("Не найдена")

и местонахождение маркера не меняется (при применении параметра A маркер остается в начале текста). При нахождении искаемой цепочки маркер будет находиться в начале этой цепочки.

A - REPLACE ("Заменить")

Замещение заданной текстовой цепочки другой заданной цепочкой. Если при вводе текста нажать на клавишу ESC, то выполнение команды прекращается. Для поиска можно ввести те же параметры, что и при команде "F". По умолчанию поиск ведется начиная с маркера в направлении конца текста, а при нахождении текста, появится вопрос в строке состояния

REPLACE (Y/N) ("Заменить Да/Нет")

При положительном ответе состоится замена. Для этой команды возможны два параметра:

N - автоматическая замена, без вопроса REPLACE (Y/N)?
G - глобальная замена всех найденных текстов

Если параметр "N" не введен, то всегда выводится вопрос о замещении. Если введены оба параметра ("N" и "G"), то состоится глобальное автоматическое замещение. После каждого замещения экран заполняется новым текстом. Если в этом режиме нажать на клавиши CTRL Y, то вывод текста на экран заканчивается (замещение выполняется быстрее). Клавиша ESC прекращает замещения.

G - GOTO ("Перейти")

Перевод курсора на заданную строку или страницу. Вводят номер строки/страницы. Если эта строка/страница не будет найдена, выводят сообщения:

NOT FOUND ("Не найдена")

и позиция курсора не меняется

N - NEW ("Новый")

Загрузка нового файла для редактирования. Вводят имя файла. При вводе ESC процесс прекращается, существующий текст сохраняется. Если существует файл с заданным именем, то загружается этот файл, иначе появится вопрос:

Имяфайла IS NEW (Y/N) ("Новый? Да/Нет")

При положительном ответе создается новый файл. При отрицательном ответе необходимо ввести имя файла снова. Новое имя файла отображается в окне основного меню.

I - FILENAME ("Имя файла")

Изменение имени указанного файла в окне основного меню. При изменении имени в окне файл записывается командами "X", "S" или "Z" в файл с новым именем.

M - MODE ("Режим")

Команда выполняется только на E51Ø4. Изменение числа символов в отображаемой строке и числа строк на экране. Возможны режимы:

24 x 4Ø символов (ESC M Ø)
24 x 53 символов (ESC M 1)
24 x 64 символов (ESC M 2)

Если вывод цепочек ESC... допускается только посредством функций операционной системы, то редактор должен быть запущен приказом

A> SED ! (См. 3.1)

L - LOAD ("Загрузка")

Загрузка файла с заданным именем в создаваемый текст. Загружаемый текст размещается начиная с текущей позиции курсора. Если при загрузке выводится текст: NO SPACE ("Нет места"), то это означает, что текстовый буфер полон. При нажатии на клавишу ESC загруженный текст останется в буфере. Иначе выполнение команды аннулируется.

S - SAVE ("Спасение")

Запись текста в файл. Имя файла задано в основном меню. После завершения записи курсор будет находиться в начале текста. При записи командой "Z" позиция маркера не меняется. Если редактор работает под ленточной операционной системой, то после каждой записи спрашивается:

TAPE CLOSED (Y/N)? ("Лента закрыта? Да/Нет)

При положительном ответе на магнитную ленту записывается каталог файлов.

X - SAVE & EXIT ("Запись и выход")

Запись текста как при команде "S", и выход из редактора.

D - DIRECTROY ("Каталог")

Отображение каталога файлов. Возможность ввода кода устройства (A...F). Отображения останавливается клавишами CTRL S и прекращается клавишей ESC.

O - OPEN ("Открытие")

Открытие каталога магнитной ленты. Спрашивается:

OPEN CATALOG (Y/N)? ("Каталог открыть? Да/Нет")

При положительном ответе считывается каталог магнитной ленты.

T - TYPE ("Отображение")

Отображение файла с заданным именем. Отображение останавливают клавишами CTRL S и прекращают клавишей ESC. Отредактированный текст сохраняется, позиция курсора не меняется.

Q - EXIT ("Выход")

Выход из редактора. Выводится вопрос:

EXIT (Y/N) ("Выйти? Да/Нет")

При положительном ответе имеет место выход из редактора, при отрицательном ответе редактирование продолжается.
Внимание! Эта команда не записывает текст на внешний носитель.

CTRL-E

Ввод и вывод произвольной ESC-последовательности длиной до 128-и символов. Эта команда позволяет ввести редактор в режим плавной прокрутки и т.д.

CTRL-L

Команда выполняется только на микроЭВМ "Labtam". Редактор переключается в экранный режим, в котором длина строки 80 или 40 символов (нормальный или дважды увеличенный размер символов соответственно).

/ - SET (Изменение параметров)

Для изменения параметров применяются следующие команды:

RET или <стрелка вниз>	- переход к следующему параметру
CTRL Y	- стирание параметра
ESC	- переход в экранный режим
CTRL H	- восстановление символа

P - PRINT ("Печать")

При печати заголовок страницы состоит из следующих частей:
- строка заглавия страницы с последующими пустыми строками
- строка номера страницы с последующими пустыми строками

Цепочка символов ".pa" в первой позиции текстовой строки обозначает конец страницы. Следующая строка начинается при печати всегда с новой страницы. Для управления форматом печати возможно кроме параметров команды SET ввести еще следующие параметры:

PAGE HEADING ("Заглавие страницы")

Заглавие страницы, до 25 символов. Если заглавие не введено, то строка надписи и следующие пустые строки не выводятся. Параметрами команды SET можно задать позицию заглавия; если позиция=0, то заглавие автоматически размещается в середине страницы.

PAGE NUMBER (Y/N/p) ("Номер страницы? Да/Нет/Номер")

N - Страницы не нумеруются, строка номера страницы и следующие пустые строки не выводятся

Y - Нумерация страниц начинается с 1; параметром команды SET можно задать позицию номера, а также текст перед и за номером

p - Нумерация страниц начинается с введенного значения

LINE NUMBER (Y/N/1) ("Номер строки? Да/Нет/Номер")

Возможные варианты аналогичны случаю ввода номера страницы.

PAGE EJECT ("Межстраницный промежуток")

Число пустых строк между страницами.

PAGE STOP (Y/N) ("Останов? Да/Нет)

При положительном ответе после распечатки каждой страницы появится вопрос:

CONTINUE (Y/N) ("Продолжать? Да/Нет")

При отрицательном ответе печать прекращается.

INTERSECTION (Y/N) ("Разделяющая линия? Да/Нет")

При положительном ответе выводится разделяющая линия (---) каждой страницы.

REPEAT ("Повторение")

Число требуемых экземпляров распечатки документа или блока.

После ввода перечисленных выше параметров можно выбрать одно из следующих продолжений:

CTRL C - вместо печати вывод на экран. При отображении игнорируются некоторые параметры печати:
- перепечатывание строк,
- останов в конце страницы.

ESC - прекращение; переход в экранный режим

... - (любая иная клавиша:) пуск печати

При выводе текста на печатающее устройство:

ESC - прекращение

CTRL S - останов; для продолжения печати нажать на любую клавишу (за исключением ESC)

3.3. Параметры команды SET

3.3.1. Параметры отображения и форматизации

Text page (Ø-255) - Длина страницы документа в строках. Если число строк отличается от нуля, то в управляющей строке отображаются номер страницы, номер строки страницы и номер столбца (позиции курсора)

Screen page (Ø-255) - Длина страницы кадра в строках. По умолчанию равна числу строк на экране. Этот параметр устанавливает число строк, через которое происходит перемещение курсора командами CTRL R и CTRL C в экранном режиме

Auto CRLF (8-248) - Автоматический перевод строки при редактировании текста. Максимальная длина строки - 248 символов. Автоматический перевод строки состоится при вводе в случае отсутствия символов за курсором в данной строке

Scroll (8-32) - ("Прокрутка") Длина горизонтального сдвига строки в символах. Возможны значения сдвига: 8, 16, 24 и 32 символа

Time (Ø-255) - ("Время") Постоянная задержки, задающая скорость вывода текста при отображении текста, блока и файла. В случае значения 255 задержка при выводе каждой строки составляет 5 секунд

Read (Ø-255) - Считывание файла, начиная с n-того Кбайта. Значение Ø показывает, что считывание файла начинается с первой записи. Изменением этого параметра создают возможность редактирования документов, не размещающихся полностью в оперативной памяти

3.3.2. Параметры печати

Page number position (Ø-128) - позиция номера страницы
overprint (1-1Ø) - кратность перепечатания строки номера страницы

skip lines (Ø-1Ø) - число пустых строк после строки номера страницы

text < - текст перед номером
text > - текст после номера

Page heading position (Ø-128) - позиция надписи страницы
overprint (1-1Ø) - кратность перепечатания строки надписи страницы

skip lines (Ø-1Ø)- число пустых строк после строки надписи страницы

Text lines position (Ø-128) - позиция текстовой строки
Overprint (1-1Ø) - кратность перепечатания каждой текстовой строки

skip lines (Ø-1Ø) - число пустых строк после каждой строкой текста

3.4. Команды для обработки блоков

Операции выполняются в режиме меню обработки блоков. Если блок обозначен, то в окне меню отображаются номера первой и последней строк блока.

B - BEGIN ("Начало")

Строка, на которой находится курсор, обозначается как начало блока. Если не обозначен конец блока, то эта же строка автоматически обозначается как конец блока.

K - END ("Конец")

Строка, на которой находится курсор, обозначается как конец блока. Если не обозначено начало блока, то эта же строка автоматически обозначается как начало блока.

H - HIDE ("Спрятать")

Аннулирование обозначений блока.

T - TOP ("Верх")

Перевод курсора в начало блока.

E - BOTTOM ("Низ")

Перевод курсора в конец блока.

W - WRITE ("Запись")

Запись блока в файл. Вводят имя файла. Если файл с таким именем уже существует, выводится вопрос

OVERWRITE (Y/N) ("Перезаписать? Да/Нет")

При положительном ответе файл перезаписывается. При отрицательном ответе следует снова ввести имя файла.

L - LIST ("Вывод")

Отображение блока. Для останова движения текста нажать CTRL S, для прекращения вывода нажать ESC. Скорость вывода задается параметрами команды SET.

P - PRINT ("Печать")

Вывод блока на печатающее устройство. Параметры те же, что и при печати всего документа.

M - MOVE ("Перемещение")

Перемещение блока на позицию, следующую за курсором. Перемещение блока в область самого блока невозможно.

C - COPY ("Копирование")

Скопирование блока на позицию, следующую за курсором. Скопирование блока в область самого блока невозможно.

Y - DELETE ("Удаление")

Удаление строк, входящих в состав блока.

3.5. Экранный режим

В экранном режиме возможен ввод следующих команд:

ESC - отображение указаний для пользователя

CTRL Q - обращение к основному меню

CTRL P - обращение к меню работы с блоками

CTRL E - перемещение курсора на одну позицию вверх; если предыдущая строка короче, то курсор будет находиться в конце строки

CTRL X - перемещение курсора на одну позицию вниз; если следущая строка короче, то курсор будет находиться в конце строки

CTRL S - перемещение курсора на одну позицию направо; с конца строки курсор переводится в начало следующей строки

CTRL D - перемещение курсора на одну позицию налево; с конца строки курсор переводится в начало предыдущей строки

CTRL C - перемещение курсора на один кадр вверх

CTRL R - перемещение курсора на один кадр вниз; длина кадра задается соответствующим параметром команды SET

CTRL F - перемещение курсора на одно слова направо; с конца строки курсор переводится в начало следующей строки

CTRL A - перемещение курсора на одно слова налево; с конца строки переводится в конец предыдущей строки

CTRL G - стирание символа

CTRL U - стирание текста начиная с курсора до конца строки

CTRL Y - стирание строки

CTRL W - стирание слова справа курсора

CTRL O - перенос курсора на первую/последнюю строку кадра

CTRL B - перенос курсора в начало/в конец строки кадра; с начала строки курсор переводится в конец строки и наоборот

CTRL T - перенос курсора в начало/в конец строки документа; с начала строки курсор переводится в конец строки и наоборот

CTRL] - замена строчной буквы прописной и наоборот

CTRL Z - копирование строки на которой находится курсор, на следующую строку; местонахождение курсора не меняется

CTRL V - переключение режимов вставления/замены; режим замены обозначается символом "*" в управляющей строке

CTRL N - поиск/замещение следующего отрезка текста; перед вводом этой команды должен быть введен образец искаемого текста (в режиме основного меню, командой "F" или "A")

CTRL ^ - отображение, для прекращения нажать на клавишу CTRL

TAB - табулятор (1, 9, 17, 25, 33, 41, 49 ...)

BACK - стирание символа перед курсором

4. СООБЩЕНИЯ ОПЕРАТОРУ

BLOCK NOT FOUND - блок не отмечен

BLOCK ERROR - ошибка в операции с блоком (попытка перемещения/скопирования блока в тот же блок)

READ ERROR - ошибка считывания каталога. При загрузке файла эта ошибка встречается только в случае отсутствия маркера 1AH в конце текстового файла

WRITE ERROR - ошибка при записи каталога; при записи файла: нет места на ГМД/ленте или каталог полон

NO TEXT - текста нет. При печати или записи: текст не введен в буфер текста

NO SPACE - буфер текста редактора полон

NOT FOUND - заданная цепочка текста или заданная строка/страница не найдена

FILE NOT FOUND - заданный файл не найден

LINE TOO LONG - недопустимая длина строки

FILE IS R/O - файл только для считывания; ошибка встречается при попытке записи в файл предназначенный только для считывания; для спасения отредактированного текста следует командой "I" менять имя файла и записать текст в виде другого файла

ERROR - ошибка в контрольной сумме при загрузке редактора

NO SPACE <Esc> to ignore
- переполнение буфера при загрузке текста; при нажатии на ESC текст в буфере сохраняется, в противном случае загрузка аннулируется.

READ ERROR <Esc> to ignore
- ошибка при загрузке текста: не найден признак конца текстового файла (1AH).

"ЭКТА"
СКБ вычислительной техники Института кибернетики АН ЭССР

ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ИНТЕЛЛЕКТУАЛЬНОГО ТЕРМИНАЛА Е51Ø4

РЕДАКТОР ГРАФИКИ И ТЕКСТОВ GTR
353872.3ØØØ1-25

РУКОВОДСТВО ОПЕРАТОРА

11 стр.

Таллин 1987

АННОТАЦИЯ

Управляемая манипулятором программа позволяет создать и редактировать графическо-текстовые документы, с записью их на внешний носитель (ГМД или кассетну магнитную ленту) и с выводом на матричное графическое печатающее устройство. Формат изображения 320 x 230 точек, длина текстовой строки 40 символов. 42 функций. 1 рисунок.

СОДЕРЖАНИЕ

1.ОБЩИЕ СВЕДЕНИЯ О ПРОГРАММЕ

ЕКТА Tallinn *ЭКТА* Таллин КОМПОНЕНТ / КОМПЛЕКС ПО

GTR	353872.30001-25	Версия: 2.5B
РЕДАКТОР ГРАФИКИ И ТЕКСТА		
ОБЬЕМ 28K байт	ОПЕРАЦ. СИСТЕМА: LOS	
ТРЕБУЕМОЕ ОЗУ	Программа 28K байт	Данные 15K байт

НОСИТЕЛЬ: / ГМД / Кассетная МЛ
РЕГ. No НОСИТЕЛЯ:

НАЗНАЧЕНИЕ И МЕТОД:
Создание и редактирование графическо-текстовых изображений на экране при помощи манипулятора, с возможностью записи изображения на магнитном носителе и вывода на печать.
Вид операции ("инструмент") выбирается клавишами, операция выполняется при помощи манипулятора.

ССЫЛКИ:

ТЕХН. ХАР-КИ:
16 операций над графикой
18 операций над текстом
8 операций общего назначения

ОГРАНИЧЕНИЯ:
Формат изображения: 320 x 230
Длина текстовой строки: 40 символов

ПРОЦЕССОР / ЭВМ: К580/Е5104

ВНЕШНИЕ УСТР-ВА:
Видеомонитор
Манипулятор Е4701
НГМД или кассетный магнитофон
Матричный принтер УВВПЧ-30-004 или соотв.

НЕОБХ. ПРОГРАММЫ И ФАЙЛЫ:
1) BELEX/CMR/CENT - драйвер печатающего устройства
2) ARUS - файл русского алфавита (только при отсутствии в ПЗУ)

14.09.1987	АРХ. МЛ: 3.	ПРОГРАММИСТ: Гладин М.
ИСХ. ТЕКСТ:	АРХ. МЛ: 3.	ЯЗЫКИ: Паскаль, Ассемблер

2. ВЫПОЛНЕНИЕ ПРОГРАММЫ

2.1. Подготовка к работе. Загрузка

Собрать конфигурацию терминала E5104 согласно стр.4. Вставить магнитный носитель, содержащий GTR, драйвер печатающего устройства и алфавитный файл ARUS (при отсутствии русского алфавита в ПЗУ), и имеющий свободный объем для записи создаваемых файлов. Загрузить операционную систему.

Если предусматривается вывод изображения на печать, подготовить и включить печатающее устройство, после этого загрузить драйвер путем ввода его имени. Драйвер выбирается в зависимости от типа печатающего устройства:

BELEX - для УВВПЧ-30-004,
CMR - для CM6329.02-M ("Robotron"),
CENT - для "Centronics HPC136-2B".

При отсутствии резидентного в ПЗУ руссского алфавита загрузить файл ARUS путем ввода его имени.

Перед загрузкой программы GTR манипулятор E4701 должен быть подключен к разъему X1 терминала E5104 (или к соответствующему разъему блока расширения E6501). Для загрузки и записи ввести имя GTR. После запуска программы на экране появится краткое меню, содержащее основные функции редактора. После выбора функции меню исчезнет и экран свободен для операций. В нижней части экрана появится служебная строка, квитирующая команды оператора и содержащая дополнительную информацию.

2.2. Графические функции

Все функции выбираются нажатием одной буквенной клавиши. Выполнение операций описано ниже. При этом для сокращенной записи последовательности приняты следующие обозначения:

*0 - нажата левая кнопка манипулятора
0* - нажата правая кнопка манипулятора
00 - исходное состояние (возврат кнопок) манипулятора
<-> - перемещение манипулятора

Выбор и отображение графических точек выполняется по наивысшей точке графического курсора.

A - прямоугольник:
    *0,00 - фиксирование угловой точки
    0* <-> - создание прямоугольника
    00 - фиксирование прямоугольника

B - стирание "резинкой"
    *0 <-> - стирание
    00 - конец стирания

C - окружность
    *0,00 - фиксирование центра
    0* <-> - создание окружности
    00 - фиксирование окружности

D - рисование от руки
    *0 <-> - рисование
    00   - конец линии

E - стирание всего экрана

F - закраска замкнутого контура:
    1,2,...,9 или 0   - выбор узора
    *0,00   - закраска (верхняя точка курсора должна находиться в контуре)
    Обновление узора номер n:
    n(=1,...,9 или 0) - выбор номера
    <->   - выбор узора на экране
    ESC   - замена узора номер n

G - увеличение части изображения:
    увеличивается прямоугольная область, верхний левый угол которой определен положением курсора
    *0,00   - инвертирование точки изображения
    любая клавиша - возврат из режима увеличения

H - вызов краткого меню

I - инвертирование изображения на экране

K - вызов каталога файлов изображений

L - прямая линия (отрезок)
    *0,00 - фиксирование начала отрезка
    0* <-> - создание отрезка
    00   - фиксирование отрезка

N - вывод изображения на печать
    (соответствующий драйвер должен быть загружен!)

O - включение/выключение отображения цифровых значений координат курсора
    Значения координат отображаются в правой части служебной строки в одном из следующих видов:
    1) x=... y=...   - координаты точки
    2) x=... y=...   x=... y=...   - координаты начальной и конечной (диагональной) точек (функции A,L)
    3) x=... y=...   Radius=   - центр и радиус окружности (функция C)

P - выбор цвета линии для последующих операций
    0   - черный
    1,...,7 - белый

Q - выход из редактора, возврат в операционную систему

R - чтение изображения из файла, находящегося на внешнем магнитном носителе
После появления сообщения "Read screen from" вводят имя файла (до 8-и символов, без атрибута)
Примечание. Сообщение на экране содержит также последнее введенное имя файла. Для чтения из этого файла достаточно нажатия на RETURN. Это касается и операции записи.

S - запись изображения в файл, находящийся на внешнем магнитном носителе
После появления сообщения "Save screen as" вводят имя файла (до 8-и символов, без атрибута)
См. "R", примечание

T - переход в текстовой режим

U - стирание всех результатов последней операции типа A,C,D,L или F

Z - операции со спрайтами

C - Выделение спрайта (части изображения):
*0,00 - фиксирование угла спрайта
<-> - выбор границ спрайта
0*,00 - фиксирование спрайта

P - Копирование спрайта на экране
<-> - перемещение копии
0*,00 или *0,00 - фиксирование копии

S - Запись спрайта в файл
После появления текста "Save sprite as" вводят имя файла (см. операцию "R", примечание)

R - Чтение спрайта из файла
После появления текста "Read sprite from" вводят имя файла (см. операцию "R", примечание)
После считывания спрайта возможно копирование (P)

U - аннулирование последней операции над спрайтом

RETURN - возврат из функций F,G,M,K,Z

2.3. Текстовые функции

Для перехода в текстовый режим из исходного состояния редактора или из графического режима вводят букву "T". Переход квитируется заменой графического курсора (черточки) текстовым курсором (прямоугольником). В текстовом режиме возможен ввод всех алфавитно-цифровых символов, в том числе прописных и строчных букв русского и латинского алфавитов. Если конкретный вариант терминала не имеет резидентного русского алфавита, следует перед запуском GTR загрузить с магнитного носителя файл ARUS.

Для редактирования текста используются следующие управляющие символы.

CTRL A - перемещение строки налево

CTRL B - первод курсора на правый/левый край экрана

CTRL C - центрирование строки

CTRL D - перемещение курсора направо на 1 шаг
(то же: CTRL L или ->)

CTRL E - перемещение курсора вверх на 1 шаг
(то же: CTRL K или стрелка вверх)

CTRL F - перемещение текста направо

CTRL G - стирание символа

CTRL H - перемещение курсора налево на 1 шаг
(то же: CTRL S или <-)

CTRL I - инверсия текста/нормальное отображение

CTRL J - перемещение курсора вниз на 1 шаг
(то же: CTRL X или стрелка вниз)

CTRL K - см. CTRL E

CTRL L - см. CTRL D

CTRL M - перевод курсора в начало следующей строки

CTRL N - переход в режим увеличения символа
Возврат из этого режима автоматический, после завершения одной из следующих операций:
1) hv - ввод масштабов увеличения
h - горизонтальный масштаб (1,...,9)
v - вертикальный масштаб (1,...,9)
2) CTRL N - увеличение символа, на котором находится курсор

Примечание. Масштабные множители сохраняются в памяти до ввода новых значений, поэтому увеличение группы символов в одинаковое число раз выполняется при помощи следующей последовательности: CTRL N, h, v, CTRL N, <->, CTRL N, CTRL N, <->, CTRL N и т.д.

CTRL O - включение/выключение индикации координат (см. п. 1.2., функция "O"

CTRL Q - возврат из текстового режима

CTRL R - инвертирование отрезка текста (от курсора до пробела)

CTRL S - см. CTRL H

CTRL V - включение/выключение режима вставки текста

CTRL W - удаление строки

CTRL X - см. GTRL J

CTRL Y - стирание строки

CTRL Z - вставка пустой строки (=RETURN)

ESC - определение алфавитного символа как графического (символ удаляется из текстовой памяти и образует теперь часть графического изображения)
Примечание. При считывании текстово-графического изображения с магнитного носителя в первую очередь выводится графика, а на графику - текст. При этом белое поле буквенной матрицы перекрывает находящиеся под ним элементы графики. Для предотвращения этого явления можно соответствующие символы при помощи ESC переопределить в графику.

RETURN - см. CTRL Z

Стрелки - перемещение курсора, 4 направления
Примечание. Для перемещения курсора можно использовать манипулятор, а также управляющие символы (CTRL D, CTRL E, CTRL H, CTRL J, или CTRL K, CTRL L, CTRL S, CTRL X)

3. СООБЩЕНИЯ, ВЫДАВАЕМЫЕ ОПЕРАТОРУ

3.1. При запуске программы

На экран выводится краткое меню функций (см. рис.1). Движущаяся взад-вперед фигурка под меню является признаком готовности редактора.

3.2. В графическом режиме

Функция	Сообщения	Координаты (функция "O")	
A	F*RECTANGLE ("Прямоугольник")	x=... y=... (Исходный угол)	x=... y=... (Диагональный угол)
C	F* CIRCLE ("Окружность")	x=... y=... (Центр)	Radius=... (Радиус)
D	F*DRAW ("Рисование")	x=... y=... (Текущая точка)	
F	F*FILL ("Заполнение")	Ø<yØ> 1<y1> ... 9<y9> yØ,...,y9 - образцы узоров Номер выбранного узора инвертирован	

```
G       Magnification                x=... y=...
        ("Увеличение")

H       (см. рис.1)

K       <Имя файла 1> <Атрибут 1>
        <Имя файла 2> <Атрибут 2>
        и т.д.
        Значениями атрибутов являются PCC, PCD, PIC
        При отсутствии файлов  с такими  атрибутами  выво-
        дится сообщение:
        No *.PIC *.PCD *.PCC files
        ("Нет файлов типов *.PIC *.PCD *.PCC")

L       F*LINE                       x=... y=... x=... y=...
        ("Линия")                    (Начальная  (Конечная
                                      точка)      точка)

R       Read screen from <имя файла>
        ("Считывать изображение  из ...")
        WAIT
        ("Ждать" - во время считывания)

S       Save screen as <имя файла>
        ("Спасти изображения как ...")
        Packed picture ... bytes
        ("Упакованное изображение - ... байт" - после  по-
        готовки изображения к записи)
        WAIT
        ("Ждать" - во время записи)

T       F*TEXT      Exit [CTRL/Q]
        ("Текст")   ("Выход: CTRL Q")

Z       Cut          Paste      Undo       Save      Read
        ("Вырезать"          "Аннулировать"       "Считывать")
                  "Наклеить"            "Спасти"
        Наименование выбранной функции инвертируется
        При выборе записи (Save) выводится текст:

        Save sprite as <имя файла>
        ("Спасти спрайт как ...")
        WAIT

        ("Ждать")
        При выборе считывания (Read) выводится текст:
        Read sprite from <имя файла>
        ("Считывать спрайт из ...")
        WAIT
        ("Ждать")
```

```
*** Turbo GTR 2.5b ***

A  rectAngl    use left and right buttons
C  Circle      use left and right buttons
E  Erase       clear all screen
F  Fill        fill area
H  Help        quick reference
L  Line        use left and right buttons
N  priNt       copy from screen to printer
O  cOordinat   coordinates ON/OFF
P  Palette     set aktive colour
Q  Quit        terminates the program
R  Read        get screen from file
S  Save        store screen to file
T  Text        text from keyboard
Y  stYle       change mode Free or Stright
B  ruBout      use left button to erase
D  Draw        free style
G  maGnify     picture magnification
I  Invert      inverts all screen
Z              operations whis sprites
```

Press any key when you are ready

Рис.1. Краткое меню редактора

1.ОБЩИЕ СВЕДЕНИЯ О ПРОГРАММЕ

ЕКТА Tallinn *ЭКТА* Таллин КОМПОНЕНТ / КОМПЛЕКС ПО

SNAKE	353872.3Ø018-13	Версия: 1.3
ИГРА "Змейка"		
ОБЬЕМ 5К байт	ОПЕРАЦ. СИСТЕМА:	
ТРЕБУЕМОЕ ОЗУ	Программа 5К байт	Данные 1К байт
НОСИТЕЛЬ: ПЗУ / Расширитель ПЗУ / ГМД / Кассетная МЛ РЕГ. No НОСИТЕЛЯ:		

НАЗНАЧЕНИЕ И МЕТОД:

Игрок стремится поймать движущуюся змейку за кончик хвоста, при этом змейка потеряет одно звено. Столкновении игрока с головой змейки увеличивает длину змейки на одно звено. Игра кончается при нажатии на клавишу <ESC> или если змейка достаточно короткая.

Игрок может выбрать уровень трудности игры и расположение клавиш управления.

Указания по игре отображаются на экране.

ТЕХН. ХАР-КИ:

ОГРАНИЧЕНИЯ:

ПРОЦЕССОР / ЭВМ: КР58ØВМ8ØА / Е51Ø1, Е51Ø2		
ВНЕШНИЕ УСТР-ВА:		
Видеомонитор		
16.11.1987	АРХ. МЛ: 3.ØØ	ПРОГРАММИСТ: Гладин М.
ИСХ. ТЕКСТ:	АРХ. МЛ: 3.ØØ	ЯЗЫКИ: Ассемблер

1.ОБЩИЕ СВЕДЕНИЯ О ПРОГРАММЕ

*ЕКТ. Tallinn *ЭКТА* Таллин КОМПОНЕНТ / КОМПЛЕКС ПО

R.EM	353872.3Ø017-11	Версия: 1.1

ДЕМОНСТРАЦИОННАЯ ПРОГРАММА

ОБЪЕМ 32К байт | ОПЕРАЦ. СИСТЕМА: CP/M , LOS

ТРЕБУЕМОЕ ОЗУ	Программа 32К байт	Данные - байт

НОСИТЕЛЬ: ГМД / Кассетная МЛ
РЕГ. No НОСИТЕЛЯ:

НАЗНАЧЕНИЕ И МЕТОД:

Программа демонстрирует некоторые возможности терминалов Е51Ø1, Е51Ø2 и дает информацию о технических характеристиках. Отображение непрепывное серия кадров образует замкнутый цикл. Любой кадр можно остановить нажатием клавиши "T". Для продолжения следует нажать на любую клавишу. Для выхода из программы нажимают клавишу "Q". Для быстрого перехода к следующему кадру нажать на любую другую клавишу. Иначе происходит некоторая задержка при каждом кадре.

ТЕХН. ХАР-КИ:
Продолжительность отображения одного кадра около 3Øс
Число кадров - 1Ø

ПРОЦЕССОР / ЭВМ: КР58ØВМ8ØА/Е51Ø1, Е51Ø2

ВНЕШНИЕ УСТР-ВА:
Видеомонитор
НГМД или кассетный магнитофон

15.07.1987	АРХ. МЛ: 3.ØØ	ПРОГРАММИСТ: Гладин М.
ИСХ. ТЕКСТ:	АРХ. МЛ: 3.ØØ	ЯЗЫКИ: Паскаль, Ассемблер

1.ОБЩИЕ СВЕДЕНИЯ О ПРОГРАММЕ

ЕКТА Tallinn *ЭКТА* Таллин КОМПОНЕНТ / КОМПЛЕКС ПО

СМ6329	353872.3Ø04Ø-1Ø	Версия: 1.Ø

ДРАЙВЕР ПЕЧАТАЮЩЕГО УСТРОЙСТВО СМ6329

ОБ'ЕМ байт ! ОПЕРАЦ. СИСТЕМА: EKDOS

ТРЕБУЕМОЕ ОЗУ ! Программа байт ! Данные байт

НОСИТЕЛЬ: НГМД
РЕГ. NO НОСИТЕЛЯ:

НАЗНАЧЕНИЕ И МЕТОД:
Вывод текстовой и графической информации на печатающее устройство. Вывод графической информации возможен в 2-х форматах:
1 - в виде копии экрана
2 - в виде копии экрана в двухкратном увеличении

ССЫЛКИ:
Паспорт печатающего устройства СМ6329

ОБРАЩЕНИЕ: Через функции монитора
PRINTCHAR (вывод текста)
HARDCOPY (распечатка содержание экрана в графическом режиме)
При обращении к HARDCOPY регистры должны иметь следующее содержание HL=5555H, DE=7777H, BC=AAAAH

СООБЩЕНИЯ:
При запузке драйвера выводится меню для вывода стандартной или эстонской символьной таблицы.
При обращении через HARDCOPY в нижней части экрана появится меню для выбора формата: NORMAL (нормальный), EXPANDED (2-х кратный)

ПРОЦЕССОР / ЭВМ: КР58ØВМ8ØА, Е51Ø4

ВНЕШНИЕ УСТР-ВА: видеомонитор
печатающее устройство СМ6329
НГМД

НЕОБХ. ПРОГРАММЫ И ФАЙЛЫ: СМ6329.COM

28.1Ø.88	АРХ. МЛ:	ПРОГРАММИСТ: Заранс И.
ИСХ. ТЕКСТ:	АРХ. МЛ:	ЯЗЫКИ: Ассемблер

1.ОБЩИЕ СВЕДЕНИЯ О ПРОГРАММЕ

ЕКТА Tallinn *ЭКТА* Таллин КОМПОНЕНТ / КОМПЛЕКС ПО

FX800	353872.30045-10	Версия: 1.0

ДРАЙВЕР ПЕЧАТАЮЩЕГО УСТРОЙСТВА FX800

ОБЪЕМ байт | ОПЕРАЦ. СИСТЕМА: EKDOS

ТРЕБУЕМОЕ ОЗУ | Программа байт | Данные байт

НОСИТЕЛЬ: НГМД
РЕГ. NO НОСИТЕЛЯ:

НАЗНАЧЕНИЕ И МЕТОД:
Вывод текстовой и графической информации на печатающее устройство. Вывод графической информации возможен в 2-х форматах:
1 - в виде копии экрана
2 - в виде копии экрана в двукратном увеличении
Вывод текстовой информации возможен в 2-х вариантах:
1 - со стандартной символьной таблицей
2 - с эстонской символьной таблицей

ВЫЗОВ:
Распечатка печатающих устройств CPF-H80, FX-800

ОБРАЩЕНИЕ: Через функции монитора
PRINTCHAR (вывод текста)
HARDCOPY (распечатка содержания экрана в графическом режиме)
При обращении к HARDCOPY регистры должны иметь следующее содержание HL=5555H, DE=7777H, BC=AAAAH

СООБЩЕНИЯ:
При запуске драйвера выводится меню для вывода стандартной или эстонской символьной таблицы.
При обращении через HARDCOPY в нижней части экрана появится меню для выбора формата: NORMAL (нормальный), EXPANDED (2-х кратный)

ПРОЦЕССОР / ЭВМ: КР580ИК80А, Е5104

ВНЕШНИЕ УСТР-ВА: видеомонитор
печатающее устройство CPF-H80 или FX-800
НГМД

НЕСБХ. ПРОГРАММЫ: ФАЙЛЫ: FX800.COM

28.10.88	АРХ. МЛ:	ПРОГРАММИСТ: Заранс И.
ИСХ. ТЕКСТ:	АРХ. МЛ:	ЯЗЫКИ: Ассемблер