

ВМЕСТО ЭПИГРАФА

СОБИРАЙТЕ "ОКЕАН" !
ДЕФИЦИТА В НЕМ - НА ГРАН,
НО ЗАТО ЕСТЬ ЦВЕТ И ЗВУК,
(БЫЛ БЫ МОЗГ, ДА ПАРА РУК!).
С НИМ НЕ БУДЕТ СКУЧНО ВАМ:
СОБИРАЙТЕ "ОКЕАН" !
ОН НЕ СИНКЛЕР, НЕ РК
- ИХ ПОРА СПУСТИТЬ В <ВК> !
ОН НЕ ВЫРВЕТ ВАМ КАРМАН;
СОБИРАЙТЕ "ОКЕАН" !
ЧТОБЫ ЧЕРЕЗ ПАРУ ЛЕТ
МЫ С ПО НЕ ЗНАЛИ БЕД
И ДРУЖНОЮ КАМПАНИЕЙ
УСТРОИЛИ БРАТАНИЕ.
ПОСЕМУ - ОДИН ЛИШЬ ПУТЬ;
БРОСИВ ВСЕ, СЛЕГКА РИСКНУТЬ
И, СОБРАВШИ "ОКЕАН"
РАЗОГНАТЬ В МОЗГАХ ТУМАН !

ГАВРИЛЕНКОВ В. В.
05.12.88 СЕВЕРОМОРСК,

A>TYPE: BOOK.TXT

МАРУШЕВ О.Г.
ГРОМАЗИН А.Р.

**СПРАВОЧНЫЕ МАТЕРИАЛЫ
ПОЛЬЗОВАТЕЛЯМ ПЭВМ
" О К Е А Н - 2 4 0 "**

1. ДИРЕКТИВЫ МОНИТОРА
2. ПОДПРОГРАММЫ МОНИТОРА
3. ВСТРОЕННЫЕ КОМАНДЫ ОС СР/М
4. УПРАВЛЯЮЩИЕ ESC
5. ПРЯМОЕ (БЫСТРОЕ) ИСПОЛЬЗОВАНИЕ ГРАФИЧЕСКИХ ПОДПРОГРАММ
6. ФУНКЦИИ BIOS
7. ОРГАНИЗАЦИЯ СИСТЕМНОЙ СТРАНИЦЫ ОЗУ
8. ФУНКЦИИ ВВОЗ
9. НЕКОТОРЫЕ ПОЯСНЕНИЯ
10. СТРУКТУРА КАТАЛОГА
11. УПРАВЛЕНИЕ РЕГИСТРОМ ЦВЕТА
12. НЕКОТОРЫЕ СТАНДАРТНЫЕ УТИЛИТЫ
13. ИСПОЛЬЗУЕМЫЕ РАСШИРЕНИЯ ФАЙЛОВ
14. КРАТКОЕ ОПИСАНИЕ BASIC MICROSOFT
15. МОДИФИКАЦИЯ ЭЛЕКТРОННОГО ДИСКА
16. ПРИМЕРНЫЙ СОСТАВ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

1 9 8 9
С Е В Е Р О М О Р С К

ДИРЕКТИВЫ MONITOR «ОКЕАН-240»

D'ADR' - ВЫВОД СОДЕРЖИМОГО 128-МИ БАЙТ ОЗУ!
S'ADR' - ВЫВОД И МОДИФИКАЦИЯ СОДЕРЖИМОГО ЯЧЕЙКИ ОЗУ1
G'ADR' - ЗАПУСК ПРОГРАММЫ С ADR
M'ADR 1'ADR2'ADR3' - ПЕРЕСЫЛКА ОБЛАСТИ ADR1-ADR2
 В ОБЛАСТЬ С ADR3
F'ADR1'ADR2'##' - ЗАПОЛНЕНИЕ ОБЛАСТИ ADR1-ADR2 КОДОМ ##
X'&' - ВЫВОД И МОДИФИКАЦИЯ РЕГИСТРА &
W - ВЫВОД НА МАГНИТОФОН СОДЕРЖИМОГО ОЗУ1
R - ЧТЕНИЕ С МАГНИТОФОНА В ОЗУ1
A'N' - ПЕРЕСЫЛКА СЕКТОРА С НОМЕРОМ 'N' ИЗ ОЗУ1 В ОЗУ2
B'N' - ПЕРЕСЫЛКА СЕКТОРА С НОМЕРОМ 'N' ИЗ ОЗУ2 В ОЗУ1
L - ВВОД С ИНТЕРФЕЙСА RS-232

ТОЧКИ ВХОДА MONITOR 'ОКЕАН-240'

E000 - ГОРЯЧИЙ СТАРТ
E003 - ХОЛОДНЫЙ СТАРТ
E006 - СТАТУС КОНСОЛИ (РЕГ. А: FF-НАЖАТА. 00-НЕТ)
E009 - ВВОД СИМВОЛА С КОНСОЛИ (РЕГ. А)
E00C - ВЫВОД СИМВОЛА НА ЭКРАН (РЕГ. С)
E00F - ВОВД СИМВОЛА С RS-232 (РЕГ. А)
E012 - ВЫВОД НА RS-232 (РЕГ. С)
E015 - ВЫВОД НА ПЕЧАТЬ (РЕГ. С)
E018 - ЧТЕНИЕ БЛОКА С МАГ. (BC - НОМЕР СЧИТАННОГО БЛОКА)
E01B - ЗАПИСЬ БЛОКА НА МАГ. (DE - НОМЕР БЛОКА)
E01E - ЗАПИСЬ БЛОКА (ОЗУ2 В ОЗУ1),
 (NL - АДРЕС БУФЕРА, DE - НОМЕР БЛОКА)
E021 - ЧТЕНИЕ БЛОКА (ОЗУ1 В ОЗУ2),
 (NL - АДРЕС БУФЕРА, DE - НОМЕР БЛОКА)
E024 - ЧТЕНИЕ ОЗУ2 С МАГНИТОФОНА
E027 - ЗАПИСЬ ОЗУ2 НА МАГНИТОФОН
E02A - WAIT (ОЖИДАНИЕ ПАУЗЫ С МАГ.)
E02D - SIGNAL (ПРОВЕРКА МАРКЕРА БЛОКА С МАГ.)
E030
E033 - ДРАЙВЕРЫ РАБОТЫ С КОНТРОЛЛЕРОМ НГМД (ДЛЯ ВЕРСИИ REL 5')

ВСТРОЕННЫЕ КОМАНДЫ ОС СР/М

EXIT - ВЫХОД В MONITOR (ОТСУТСТВУЕТ В REL 8'
SAVE NN ИМФАЙЛ - СОХРАНЕНИЕ NN СЕКТОРОВ ОЗУ1 В ФАЙЛЕ С ИМЕНЕМ 'ИМФАЙЛ'
READ - ЧТЕНИЕ ФАЙЛОВ С МАГНИТОФОНА.
WRITE - ЗАПИСЬ ВСЕХ ФАЙЛОВ ОЗУ2 НА МАГНИТОФОН,
 (ВОЗМОЖЕН ПАРАМЕТР: 'ИМФАЙЛ' - ЗАПИСЬ ФАЙЛА 'ИМФАЙЛ')
 ! В REL.4 АНАЛОГИЧНА DUMP
DIR - ВЫВОД КАТАЛОГА ДИСКА (ОЗУ2)
REN 'НОВ.ИМЯ'='СТАР.ИМЯ' - ПЕРЕИМЕНОВАНИЕ ФАЙЛА
ERA 'ИМФАЙЛ' - СТИРАНИЕ ФАЙЛА ИЗ КАТАЛОГА,
 ДОПУСТИМО ИСПОЛЬЗОВАНИЕ МЕТАСИМВОЛОВ
 (ТОЛЬКО ДЛЯ REL.5)
TYPE 'ИМФАЙЛ' - ВЫВОД СОДЕРЖИМОГО ФАЙЛА НА ЭКРАН
USER N - ОБ'ЯВЛЕНИЕ ОБЛАСТИ ПОЛЬЗОВАТЕЛЯ С
 НОМЕРОМ N (0...15)

УПРАВЛЯЮЩИЕ ESC-ПОСЛЕДОВАТЕЛЬНОСТИ

- КООРДИНАТА X НАЧАЛЬНОЙ ТОЧКИ
\$\$ - КООРДИНАТА Y НАЧАЛЬНОЙ ТОЧКИ
%% - КООРДИНАТА X КОНЕЧНОЙ ТОЧКИ
&& - КООРДИНАТА Y КОНЕЧНОЙ ТОЧКИ
 КООРДИНАТЫ HEX: (0...F)

<ESC>0###\$\$ - ПЕЧАТЬ С ТОЧКИ ЭКРАНА
<ESC>1###\$\$%&& - ИЗОБРАЖЕНИЕ ПРЯМОУГОЛЬНИКА
<ESC>2###\$\$%&& - ИЗОБРАЖЕНИЕ ЛИНИИ
<ESC>3###\$\$ - ИЗОБРАЖЕНИЕ ТОЧКИ
<ESC>4N - ВЫБОР ЦВЕТА ИЗОБРАЖЕНИЯ <N=1...4>
 (N=4: ЦВЕТ СОВПАДАЕТ С ФОНОМ)
*** <ESC>0...<ESC>4 : ТОЛЬКО ДЛЯ РЕЖИМА ЭКРАНА <ESC>60,<ESC>68,
<ESC>5@@** - ПЕЧАТЬ С ПОЗИЦИИ КУРСОРА:
 @@: HEX НОМЕР СТРОКИ
 **: HEX НОМЕР СТОЛБЦА
(ДЛЯ ПРОГРАММНОГО РЕЖИМА, С КОНСОЛИ НЕ ВВОДИТСЯ)
ДЛЯ РЕЖИМА ЭКРАНА В 64 СИМВОЛА:
 СТОЛБЕЦ НУМЕРУЕТСЯ ОТ -32(E0H) ДО 32(1FH)
ДЛЯ РЕЖИМА ЭКРАНА В 80 СИМВОЛОВ:
 СТОЛБЕЦ НУМЕРУЕТСЯ ОТ -96(A0H) ДО -16(EFH)
ДЛЯ REL 5' ФУНКЦИЯ РАБОТАЕТ ПО ДРУГОМУ:
 ПАРАМЕТРАМИ ЯВЛЯЮТСЯ НОМЕРА СТОЛБЦА И СТРОКИ, УВЕЛИЧЕННЫЕ НА 20H
 ДЛЯ ВСЕХ РЕЖИМОВ ЭКРАНА,
<ESC>6N - РЕЖИМ ЭКРАНА N=0...9
 N= 0: 32СИМВ/18СТР С КУРСОРОМ
 ДЛЯ REL 5' - 40СИМВ/23СТР
 1,2: 64СИМВ/23СТР С КУРСОРОМ
 3: 80СИМВ/23СТР С КУРСОРОМ 4
 ДЛЯ REL 5' - 40СИМВ/23СТР БЕЗ КУРСОРА
 5,6: 64СИМВ/23СТР БЕЗ КУРСОРА
 7: 80СИМВ/23СТР БЕЗ КУРСОРА
 8: РЕЖИМ ГРАФИКИ, (БЕЗ СВИТКА)
<ESC>7N - ВЫБОР НАБОРА СИМВОЛОВ N=0,1,2
 N= 0: ЛАТИНСКИЕ, ПРОПИСНЫЕ/СТРОЧИЙЕ
 1: РУССКИЕ, ПРОПИСНЫЕ/СТРОЧИЙЕ
 2: ПРОПИСНЫЕ, ЛАТИНСКИЕ/РУССКИЕ
<ESC>8@@ - ВЫБОР ЦВЕТОВОЙ ПАЛИТРЫ, @@ - HEX КОД,
 ОПРЕДЕЛЯЮЩИЙ ЦВЕТ ИЗОБРАЖЕНИЯ И ФОНА
<ESC>;###\$\$@@%&& - ИЗОБРАЖЕНИЕ ОКРУЖНОСТИ
 ###\$\$: КООРДИНАТЫ ЦЕНТРА X И Y, HEX
 @@: РАДИУС (00...7F), HEX
 %&&: КОЭФИЦИЕНТЫ СЖАТИЯ ПО X И Y, HEX

<<<<<<< КОНЕЦ >>>>>>>

ИСПОЛЬЗОВАНИЕ ГРАФИЧЕСКИХ ФУНКЦИЙ В ПРОГРАММНОМ РЕЖИМЕ

ДЛЯ ВЫЗОВА ГРАФИЧЕСКИХ ПОДПРОГРАММ ИЗ ПРОГРАММ НА АССЕМБЛЕРЕ НЕОБХОДИМО ПОДГОТОВИТЬ ПАРАМЕТРЫ АНАЛОГИЧНО ПАРАМЕТРАМ СООТВЕТСТВУЮЩИХ ESC-ПОСЛЕДОВАТЕЛЬНОСТЕЙ. ПАРАМЕТРЫ В HEX ФОРМЕ ЗАПИСЫВАЮТСЯ В СПЕЦИАЛЬНЫЙ БУФЕР. АДРЕС НАЧАЛА БУФЕРА: 0BFE2H. ПОСЛЕ ЭТОГО СЛЕДУЕТ ВЫЗОВ ПОДПРОГРАММЫ. ИСКЛЮЧЕНИЕ СОСТАВЛЯЕТ ФУНКЦИЯ "ПЕЧАТЬ С ТОЧКИ ЭКРАНА" (ESC0). ДЛЯ ЕЕ РЕАЛИЗАЦИИ СЛЕДУЕТ ЗАПИСАТЬ КОД 02H В СПЕЦИАЛЬНЫЙ ФЛАГ ПО АДРЕСУ: 0BFD7H И ПРОИЗВОДИТЬ ВЫВОД СИМВОЛОВ НА ДИСПЛЕЙ ЛЮБЫМ СТАНДАРТНЫМ СПОСОБОМ (НАПРИМЕР ПОДПРОГРАММОЙ 0E00CH). ПАРАМЕТРЫ (КООРДИНАТЫ ТОЧКИ) ДОЛЖНЫ НАХОДИТЬСЯ В БУФЕРЕ, КАК ДЛЯ ВСЕХ ПОДПРОГРАММ.

ЕСЛИ ЕСТЬ НЕОБХОДИМОСТЬ ИСПОЛЬЗОВАТЬ ПАРАМЕТРЫ В ASCII ФОРМЕ, ВЫЗОВ ПРОИСХОДИТ ЧЕРЕЗ ЕДИНУЮ ТОЧКУ ВХОДА: 0E321H (ДЛЯ REL 5' - 0E32BH) ПРЕДВАРИТЕЛЬНО В БУФЕР ЗАПИСЫВАЮТСЯ ПАРАМЕТРЫ В КОДАХ ASCII. ЧИСЛО ЗАПИСЫВАЕМЫХ В БУФЕР БАЙТ ПРИ ЭТОМ В ДВА РАЗА БОЛЬШЕ. КРОМЕ ТОГО, НЕОБХОДИМО ПО АДРЕСУ 0BFE0H ЗАПИСАТЬ HEX НОМЕР ESC-ФУНКЦИИ, А ПО АДРЕСУ 0BFE1H - ЧИСЛО ЗАПИСЫВАЕМЫХ В БУФЕР БАЙТ.

ТАБЛИЦА АДРЕСОВ ПОДПРОГРАММ

НОМЕР DEC	ESC HEX	КЛАВИША	ЧИСЛО HEX	ПАР. ASCII	АДРЕС П/ПРГ-ММЫ	АДРЕС ДЛЯ REL5'
0	0	"0"	2	4	СМ. ОПИСАНИЕ	
1	1	"1"	4	8	0EC1BH	0ED5EH
2	2	"2"	4	8	0ECB4H	0EDF7H
3	3	"3"	2	4	0EDFBH	0EF3EH
4	4	"4"	1	1	0EB79H	0ECBCH
5	5	----	2	----	0EA8FH	0EC2BH
6	6	"6"	1	1	0EB4DH	0EC8BH
7	7	"7"	1	1	0E4F7H	0E501H
8	8	"8"	1	1	0E4DEH	0E4E8H
9	9	ИСПОЛЬЗУЕТСЯ СИСТЕМОЙ			0E3E8H	0E3F2H
10	A	ИСПОЛЬЗУЕТСЯ СИСТЕМОЙ			0E3F1H	0E3FBH
11	B	";"	5	10	0EE1FH	0EF62H
12	C	"<"	1	1	РЕЗЕРВ	
13	B	"="	1	1	РЕЗЕРВ	
14	E	">"	1	1	РЕЗЕРВ	
15	F	ИСПОЛЬЗУЕТСЯ СИСТЕМОЙ			0E3D3H	0E3DDH

<<<<<< КОНЕЦ >>>>>>

Ф У Н К Ц И И В И О С

АДРЕС	ИМЯ	ФУНКЦИЯ	ПАРАМЕТР	ВОЗВРАТ
D600H	BOOT	ХОЛОДНЫЙ СТАРТ	-	C=00
D603H	WBOOT	ГОРЯЧИЙ СТАРТ	-	C=ИМЯ ДИСКА
D606H	CONST	СТАТУС КОНСОЛИ	-	A=00: НЕ/ГОТ A=FF: ГОТОВ A=СИМВОЛ
D609H	CONIN	ВВОД С КОНСОЛИ	-	-
D60CH	CONOUT	ВЫВОД СИМВОЛА НА CON	C=СИМВОЛ	-
D60FH	LIST	ВЫВОД СИМВОЛА НА LST	C=СИМВОЛ	-
D612H	PUNCH	ВЫВОД СИМВОЛА ИА PUN	C=СИМВОЛ	-
D615H	READER	ВВОД С RDR	-	A=СИМВОЛ
D618H	HOME	УСТАНОВИТЬ ДОРОЖКУ 00	-	-
D61BH	SET_DSK	ВЫБРАТЬ ДИСК	C=НОМ. ДИСКА	HL=АДР. БЛОК HL=0-ДИСК ПОЛОН, ОШИБКА
D61EH	SET_TRK	УСТАНОВИТЬ ДОРОЖКУ	C=НОМЕР	-
D621H	SET_SEK	УСТАНОВИТЬ СЕКТОР	C=НОМЕР	-
D624H	SET_DMA	УСТАНОВИТЬ АДРЕС DMA	BC=АДР. DMA	-
D627H	READ	ЧИТАТЬ ЗАДАННЫЙ СЕКТОР	-	A=00: НОРМА A=01: ОШИБКА
D62AH	WRITE	ПИСАТЬ ЗАДАННЫЙ СЕКТОР	-	A=00: НОРМА A=01: ОШИБКА
D62DH	LISTST	ПОЛУЧИТЬ СТАТУС LST	-	A=00: НОРМА A=01: НЕ/ГОТ
D630H	SECTRAN	ТРАНЛЯЦИЯ ЛОГ. СЕКТОРА (ЛС) В ФИЗИЧЕСКИЙ (ФС)	-	HL=НОМ.ФС DE=АДР. ТАБ- ЛИЦЫ ТРАНС.
D633H		РЕЗЕРВ		
D636H		РЕЗЕРВ		
D639H	*****	ЧТЕНИЕ БЛОКА С МАГНИТ.	HL=АДР. DMA	A=00: НОРМА A=04: BREAK ИНАЧЕ: СБОЙ BC=НОМ. БЛОК
D63CH	*****	ЗАПИСЬ БЛОКА НА МАГНИТ.	HL=АДР. DMA DE=НОМ. БЛОК	-
D63FH	*****	ПОИСК ПАУЗЫ ПРИ ЧТЕНИИ С МАГНИТОФОНА	A=ДЛИТ. ПАУЗ	A=00: ПАУЗА НАЙДЕНА A=04: BREAK

**ОРГАНИЗАЦИЯ СИСТЕМНОЙ ОБЛАСТИ
(ЗАРЕЗЕРВИРОВАННЫЕ ЯЧЕЙКИ НУЛЕВОЙ СТРАНИЦЫ)**

АДРЕС	НАЗНАЧЕНИЕ
0000H-0002H	ПЕРЕХОД НА WBOOT
0003H	БАЙТ КОНФИГУРАЦИИ ВВОДА/ВЫВОДА
0004H	ТЕКУЩИЙ НОМЕР ПОЛЬЗОВАТЕЛЯ (USER)
0005H-0007H	ТОЧКА ВХОДА VDOS (ИЗМЕНЯЕТСЯ ОТЛАДЧИКОМ)
0008H-0037H	ЗАРЕЗЕРВИРОВАННЫЕ ПРЕРЫВАНИЯ
0038H-003AH	ТОЧКА ВХОДА В ОТЛАДЧИК ПО ПРЕРЫВАНИЮ 7
003BH-003FH	РЕЗЕРВ
0040H-004FH	РЕЗЕРВ BIOS
0050H-005AH	РЕЗЕРВ
005CH-007CH	СТАНДАРТНЫЙ FSB ТРАНЗИТНЫХ ПРОГРАММ
007DH-007FH	ПОЗИЦИИ FSB ДЛЯ ПРЯМОГО ОБРАЩЕНИЯ К ФАЙЛУ (R0-R2)
0080H-00FFH	СТАНДАРТНЫЙ БУФЕР DMA (ПО УМОЛЧАНИЮ) БУФЕР КОМАНДНОЙ СТРОКИ ПРИ ЗАПУСКЕ ТРАНЗИТНОЙ ПРОГРАММЫ ОБЛАСТЬ ВТОРОГО FSB ДЛЯ ТРАНЗИТНЫХ ПРОГРАММ

<<<<<< КОНЕЦ >>>>>>

BDOS - СИСТЕМНЫЕ ВЫЗОВЫ

ВЫЗОВ: CALL 0005

[РЕГИСТР C - НОМЕР ФУНКЦИИ]

[ПАРА DE - ПАРАМЕТР]

ВОЗВРАТ: ОДНОБАЙТОВЫЙ РЕЗУЛЬТАТ - РЕГИСТР A

ДВУХБАЙТОВЫЙ РЕЗУЛЬТАТ - ПАРА HL

HEX	ФУНКЦИЯ	ПАРАМЕТРЫ	ВОЗВРАТ
00	СБРОС СИСТЕМЫ	--	--
01	ВВОД СИМВОЛА С КОНСОЛИ	--	A=<СИМВОЛ>
02	ВЫВОД СИМВОЛА НА КОНСОЛЬ	E=<СИМВОЛ>	--
03	ВВОД СИМВОЛА С RDR	--	A=<СИМВОЛ>
04	ВЫВОД СИМВОЛА НА PUN	E=<СИМВОЛ>	--
05	ВЫВОД СИМВОЛА НА LST	E=<СИМВОЛ>	--
06	ПРЯМОЙ ВВОД/ВЫВОД НА КОНСОЛЬ	1. E=FE: ВВОД СТАТУСА 2. E=FF: ВВОД СИМВОЛА 3. E=<СИМВОЛ>: ВЫВОД СИМВОЛА	A=<СТАТУС> A=<СИМВОЛ> A=0: НЕГОТОВ --
07	ПОЛУЧИТЬ БАЙТ КОНФИГУРАЦИИ ВВОДА/ВЫВОДА	--	A=I/O БАЙТ
08	УСТАНОВИТЬ БАЙТ КОНФИГУРАЦИИ ВВОДА/ВЫВОДА	E=I/O БАЙТ	--
09	РАСПЕЧАТАТЬ БУФЕР (\$ - ПРИЗНАК КОНЦА)	DE=АДР.БУФЕРА	--
0A	ЧТЕНИЕ С КОНСОЛИ В БУФЕР: ФОРМАТ БУФЕРА DE: 0 -1 -2 -3...-N MX NC C1 C2...CN MX - РАЗМЕР БУФЕРА [1...256], УКАЗЫВАЕТСЯ ПОЛЬЗОВАТЕЛЕМ, NC - ЧИСЛО ЗАПИСАННЫХ СИМВОЛОВ, УСТАНОВЛИВАЕТСЯ BDOS C1... - СИМВОЛЫ, СЧИТАННЫЕ С КОНСОЛИ	DE=АДР.БУФЕРА	--
0B	ПОЛУЧИТЬ СТАТУС КОНСОЛИ	--	A=FF: ГОТОВ A=00: НЕГОТОВ H1...<НОМЕР>
0C	ПОЛУЧИТЬ НОМЕР ВЕРСИИ	--	--
0D	СБРОС ДИСКОВОЙ СИСТЕМЫ: УСТАНОВЛИВАЕТСЯ СОСТОЯНИЕ R/W ВСЕХ ДИСКОВ, АКТИВНЫЙ ДИСК A> СТАНДАРТНЫЙ АДРЕС НЕПОСРЕДСТВЕННОГО ДОСТУПА К ПАМЯТИ (DMA В ОБЛАСТЬ BOOT + 80H)	--	--
0E	ВЫБРАТЬ ДИСК (0-A, 1-B, 2-C, ...)	E=<КОД ДИСКА>	--
0F	ОТКРЫТЬ ФАЙЛ. BDOS ПРОСМАТРИВАЕТ ОГЛАВ- ЛЕНИЕ ДО СОВПАДЕНИЯ 14-ТИ ПОЗИЦИЙ FCB И ПРИ УСПЕШНОМ ПОИСКЕ КОПИРУЕТ В ПАМЯТЬ БАЙТЫ С D0 ПО DN FCB.	DE=<АДРЕС FCB>	A=0...3: КОД ОГЛАВЛЕНИЯ A=FF: ОШИБКА
10	ЗАКРЫТЬ ФАЙЛ	DE=<АДРЕС FCB>	A=<КОД ОГЛ.> A=FF: ОШИБКА
11	ПОИСК ПЕРВОГО. ПРОСМАТРИВАЕТСЯ ОГЛАВЛЕНИЕ ДО СОВПАДЕНИЯ С ФАЙЛОМ, ЗАДАННЫМ FCB. ПРИ УСПЕШНОМ ПОИСКЕ ЗАПИСЬ С ЭЛЕМЕНТАМИ ОГЛАВЛЕНИЯ СЧИТЫВАЕТСЯ В БУФЕР ПО ТЕКУЩЕМУ АДРЕСУ В DMA, И СМЕЩЕНИЕ ОТ НАЧАЛА БУФЕРА РАВНО A+32	DE=<АДРЕС FCB>	A=<КОД ОГЛ.> A=FF: ОШИБКА
12	ПОИСК СЛЕДУЮЩЕГО. АНАЛОГИЧНО ФУНКЦИИ 11, НО ПРОСМОТР ОГЛАВЛЕНИЯ ДИСКА ПРОДОЛЖАЕТСЯ С ПОСЛЕДНЕГО СОВПАВШЕГО ЭЛЕМЕНТА ОГЛАВЛЕНИЯ.	--	A=<КОД ОГЛ.> A=FF: ОШИБКА
13	УДАЛИТЬ ФАЙЛ	DE=<АДРЕС FCB>	A=0: НОРМА A>0: ОШИБКА
14	ПОСЛЕДОВАТЕЛЬНОЕ ЧТЕНИЕ. ДЛЯ ОТКРЫТОГО ИЛИ АКТИВИРОВАННОГО ФАЙЛА ПРОИСХОДИТ ЧТЕНИЕ 128-МИ БАЙТОВОЙ ЗАПИСИ В ПАМЯТЬ ПО АДРЕСУ DMA. НОМЕР ЗАПИСИ УКАЗЫВАЕТСЯ В ПОЛЕ <CR> И ПОСЛЕ ЧТЕНИЯ ЗАПИСИ УВЕЛИЧИВАЕТСЯ НА 1.	DE=<АДРЕС FCB>	A=0: НОРМА A>0: ОШИБКА
15	ПОСЛЕДОВАТЕЛЬНАЯ ЗАПИСЬ. 128 БАЙТ ПАМЯТИ ПО АДРЕСУ В DMA ЗАПИСЫВАЮТСЯ В ФАЙЛ ПО FCB. ПОЛОЖЕНИЕ ЗАПИСИ ОПРЕДЕЛЯЕТСЯ ПОЛЕМ <CR>, КОТОРОЕ УВЕЛИЧИВАЕТСЯ ИА 1.	DE=<АДРЕС FCB>	A=0: НОРМА A>0: ДИСК ЗАПОЛНЕН
16	СОЗДАТЬ ФАЙЛ. АНАЛОГИЧНО ФУНКЦИИ 0F, НО FCB ДОЛЖЕН ДАТЬ ИМЯ ФАЙЛУ, КОТОРОГО НЕТ В ОГЛАВЛЕНИИ ДИСКА.	DE=<АДРЕС FCB>	A=0: <КОД ОГЛ.> A=FF0: ОГЛ. ЗАПОЛНЕНО

17	ПЕРЕИМЕНОВАТЬ ФАЙЛ. СТАРОЕ ИМЯ = ПЕРВЫЕ 16 БАЙТ FCB, НОВОЕ = ВТОРЫЕ 16 БАЙТ	DE=<АДРЕС FCB>	A=<КОД ОГЛ.> A=FF: ОШИБКА
18	ПОЛУЧИТЬ ВЕКТОР ПОДКЛЮЧЕННЫХ ДИСКОВ	--	HL=<ВЕКТОР>
19	ПОЛУЧИТЬ НОМЕР АКТИВНОГО ДИСКА	--	A=<НОМЕР>
1A	УСТАНОВИТЬ АДРЕС DMA	DE=<АДРЕС DMA>	--
1B	ПОЛУЧИТЬ АДРЕС ВЕКТОРА РАСПРЕДЕЛЕНИЯ ПАМЯТИ ДИСКА	--	HL=<АДРЕС>
1C	ЗАЩИТИТЬ ДИСК ОТ ЗАПИСИ	--	--
1D	ПОЛУЧИТЬ ВЕКТОР R/O ДИСКОВ	--	HL=<ВЕКТОР>
1E	УСТАНОВИТЬ АТТРИБУТЫ ФАЙЛА	DE=<АДРЕС FCB>	A=<КОД ОГЛ.>
1F	ПОЛУЧИТЬ АДРЕС БЛОКА ПАРАМЕТРОВ ДИСКА	--	HL=<АДРЕС>
20	УСТАНОВИТЬ/ПОЛУЧИТЬ КОД ПОЛЬЗОВАТЕЛЯ	E=<КОД ПОЛЬЗ.> E=FF	-- A=<КОД ПОЛЬЗ.>
21	ПРЯМОЕ ЧТЕНИЕ. АДРЕС ЗАПИСИ В R0, R1.	DE=<АДРЕС FCB>	A=<КОД ОШИБКИ>
22	ПРЯМАЯ ЗАПИСЬ.	DE=<АДРЕС FCB>	A=<КОД ОШИБКИ>
23	ВЫЧИСЛИТЬ РАЗМЕР ФАЙЛА	DE=<АДРЕС FCB>	R0, R1: РАЗМЕР
24	УСТАНОВИТЬ НОМЕР ЗАПИСИ, ПЕРЕХОД ОТ ПОСЛЕДОВАТЕЛЬНОГО К ПРЯМОМУ ВВОДУ/ВЫВОДУ	DE=<АДРЕС FCB>	R0, R1: РАЗМЕР
25	СБРОС ЗАДАННОГО ДИСКА	DE=<ВЕКТОР>	A=0: НОРМА A=FF: ОШИБКА
26	ПРЯМАЯ ЗАПИСЬ. УСТАНОВИТЬ В 0 НЕЗАДЕЙСТВОВАННЫЙ СЕКТОР В ТОМ-ЖЕ БЛОКЕ.	DE=<АДРЕС FCB>	A=<КОД ОШИБКИ>

<<< THE END >>>

СПРАВОЧНАЯ ИНФОРМАЦИЯ

ПО ПРОЦЕДУРАМ ОТКРЫТИЯ, ЗАКРЫТИЯ, ЧТЕНИЯ, ЗАПИСИ ФАЙЛОВ, А ТАКЖЕ НЕКОТОРЫЕ ПОЯСНЕНИЯ К СИСТЕМНЫМ ВЫЗОВАМ **BDOS**.

1. ОПРЕДЕЛЕНИЕ НЕКОТОРЫХ ПОНЯТИЙ.

D M A - 128 БАЙТОВЫЙ БУФЕР С КОТОРЫМ РАБОТАЕТ **BDOS**.
F S B - БУФЕР В КОТОРОМ СОХРАНЯЕТСЯ КОПИЯ АКТИВНОЙ ЗАПИСИ О ФАЙЛЕ И ЕГО ПАРАМЕТРАХ ИЗ КАТАЛОГА.
<CR> - 33-Й БАЙТ F S B
R1, R0 - 34,35,36 БАЙТЫ F S B

2. СТРУКТУРА И ФОРМИРОВАНИЕ F S B

ОСОБОЕ ВНИМАНИЕ СЛЕДУЕТ УДЕЛИТЬ ЗАДАНИЮ ИМЕНИ ФАЙЛА. СТРУКТУРА ЗАПИСИ О ФАЙЛЕ ДОЛЖНА БЫТЬ ПОЛНОСТЬЮ ИДЕНТИЧНА ЗАПИСИ В КАТАЛОГЕ, В ПРОТИВНОМ СЛУЧАЕ ПРИ ПОПЫТКЕ ОТКРЫТИЯ ФАЙЛА ПРОИЗОЙДЕТ ОШИБКА. ИЗ ЭТОГО СЛЕДУЕТ, ЧТО РАЗМЕР БУФЕРА FSB БУДЕТ ПО КРАЙНЕЙ МЕРЕ НЕ МЕНЬШЕ РАЗМЕРА ЗАПИСИ В КАТАЛОГЕ (32 БАЙТА). ПРИ РАБОТЕ С ОПЕРАЦИЯМИ ЧТЕНИЯ ИЛИ ЗАПИСИ ФАЙЛОВ В FSB ДОЛЖНО ВКЛЮЧАТЬСЯ ПОЛЕ <CR>. УЧИТЫВАЯ ТАК ЖЕ, ЧТО НЕКОТОРЫЕ ФУНКЦИИ ИСПОЛЬЗУЮТ R1, R0 ПОЛУЧАЕМ НЕОБХОДИМЫЙ РАЗМЕР FSB (36 БАЙТ).

КАК УЖЕ ОТМЕЧАЛОСЬ СТРУКТУРА ЗАПИСИ В FSB ДОЛЖНА БЫТЬ ИДЕНТИЧНА ЗАПИСИ В КАТАЛОГЕ. ОДНАКО УДОБНЕЕ ЗАДАВАТЬ ИМЯ ФАЙЛА В ВИДЕ **IMFILE.TIP** ДЛЯ ПРЕОБРАЗОВАНИЯ ЭТОГО ИМЕНИ В ЗАПИСЬ МОЖНО ПОЛЬЗОВАТЬСЯ СТАНДАРТНОЙ ПРОГРАММОЙ **OS CP/M** РАСПОЛОЖЕННОЙ ПО АДРЕСУ **B65EH**. ДЛЯ ЭТОГО НУЖНО:

- В ЯЧЕЙКУ ПАМЯТИ **B488H** ПОМЕСТИТЬ АДРЕС БУФЕРА ГДЕ ЗАПИСАНО ИМЯ ФАЙЛА **IMFILE.TIP**
- ВЫЗВАТЬ ПРОГРАММУ **B65EH**
- ПОСЛЕ ЕЕ РАБОТЫ НАЧИНАЯ С АДРЕСА **B5CDH** 32 БАЙТА БУДУТ ЯВЛЯТЬСЯ ПРИВЕДЕННОЙ К ВИДУ КАТАЛОГА ЗАПИСЬЮ О ФАЙЛЕ.

ЧИСЛО ОДНОВРЕМЕННО ОТКРЫТЫХ ФАЙЛОВ НЕ ОГРАНИЧИВАЕТСЯ И ОПРЕДЕЛЯЕТСЯ ТОЛЬКО ЧИСЛОМ АКТИВНЫХ FSB.

3. ПРИМЕРНЫЙ ПОРЯДОК РАБОТЫ ПО ОТКРЫТИЮ И ЧТЕНИЮ ФАЙЛА.

- 1 ПРИНЯТЬ ИМЯ ФАЙЛА В БУФЕР
- 2 ПРЕОБРАЗОВАТЬ ИМЯ ФАЙЛА В ВИД ПОНЯТНЫЙ **BDOS**
- 3 ОТКРЫТЬ ФАЙЛ (ФУНКЦИЯ **0FH**)
- 4 ВЫЧИСЛИТЬ РАЗМЕР ФАЙЛА (ЧИСЛО БЛОКОВ) (ФУНКЦИЯ **23H**)
- 5 УСТАНОВИТЬ СЧЕТЧИК БЛОКОВ
- 6 УСТАНОВИТЬ АДРЕС DMA (ФУНКЦИЯ **1AH**)
- 7 СЧИТАТЬ БЛОК ПО АДРЕСУ В DMA (ФУНКЦИЯ **14H**)
- 8 УМЕНЬШИТЬ СЧЕТЧИК
- 9 ЕСЛИ ВСЕ БЛОКИ СЧИТАНЫ, ТО ВЫПОЛНЯТЬ ПУНКТ 10, ЕСЛИ НЕТ, ТО УВЕЛИЧИТЬ АДРЕС DMA НА **80H** И ПОВТОРИТЬ ПУНКТЫ 6...9
- 10 ЗАКРЫТЬ ФАЙЛ (ФУНКЦИЯ **10H**)

ПРИ ЗАПИСИ В ФАЙЛ ВЫПОЛНЯЮТСЯ ТЕ ЖЕ ДЕЙСТВИЯ ЗА ИСКЛЮЧЕНИЕМ ПУНКТА 4, А В ПУНКТЕ 7 ФУНКЦИЯ **15H**.

СПРАВОЧНАЯ ИНФОРМАЦИЯ О СТРУКТУРЕ ЗАПИСЕЙ В КАТАЛОГЕ

- ИНФОРМАЦИЯ О ФАЙЛАХ В КАТАЛОГЕ НАХОДИТСЯ В 32-Х БАЙТОВЫХ ЗАПИСЯХ
- МАКСИМАЛЬНЫЙ РАЗМЕР ФАЙЛА, КОТОРЫЙ УМЕЩАЕТСЯ В ОДНУ ЗАПИСЬ 16 КБАЙТ (ЗАПИСЬ СОДЕРЖИТ 16 ГРУПП)
- МИНИМАЛЬНЫЙ РАЗМЕР ФАЙЛА, КОТОРЫЙ МОЖЕТ ХРАНИТЬ ОДНА ЗАПИСЬ 1 КБАЙТ
- ИНФОРМАЦИЯ О ФАЙЛЕ БОЛЕЕ 16 КБАЙТ ХРАНИТСЯ В НЕСКОЛЬКИХ ЗАПИСЯХ

СТРУКТУРА ОДНОЙ ЗАПИСИ

БАЙТ	ЗНАЧЕНИЕ
00	E5-ФАЙЛ УДАЛЕН, 00...0F – ЗОНА ПОЛЬЗОВАТЕЛЯ
01...07	КОД ASCII – ИМЯ ФАЙЛА
08	ПРОБЕЛ - РАЗДЕЛИТЕЛЬ
09	1-Й БАЙТ РАСШИРЕНИЯ ФАЙЛА, ЕСЛИ 7-Й БИТ УСТАНОВЛЕН, ТО ФАЙЛ ЗАЩИЩЕН ОТ ЗАПИСИ (R/O)
0A	2-Й БАЙТ РАСШИРЕНИЯ ФАЙЛА, ЕСЛИ 7-Й БИТ УСТАНОВЛЕН, ТО ФАЙЛ ОБЪЯВЛЕН СИСТЕМНЫМ (SYS)
0B	3-Й БАЙТ РАСШИРЕНИЯ ФАЙЛА
0C...0E	НОМЕР ЗАПИСИ О ФАЙЛЕ
0F	РАЗМЕР ЗАПИСИ В БЛОКАХ
10...1F	НОМЕРА КИЛОБАЙТ (ГРУПП) В КОТОРЫХ НАХОДЯТСЯ ДАННЫЕ ФАЙЛА

УПРАВЛЕНИЕ ЦВЕТОВЫМИ ВОЗМОЖНОСТЯМИ "ОКЕАНА-240"
 ЧЕРЕЗ ПОРТ E1H (241D)

ДАННЫЕ СВЕДЕНИЯ ДОСТОВЕРНЫ ДЛЯ НЕИНВЕРСНОГО ПОДКЛЮЧЕНИЯ ВИДЕОУСИЛИТЕЛЕЙ ТЕЛЕВИЗОРА К ЦВЕТОВЫМ КАНАЛАМ "ОКЕАН-240" СОГЛАСНО СХЕМЕ, (ПО СБРОСУ УСТАНОВЛИВАЕТСЯ ЧЕРНЫЙ ФОН, ЗЕЛЕНое ИЗОБРАЖЕНИЕ).

ДАННЫЕ СВЕДЕНИЯ ОТЛИЧАЮТСЯ ОТ ПРИВЕДЕННЫХ В МПСС N4 ЗА 1986Г., СТР. 76, ТАБ. 2.

НАЗНАЧЕНИЕ РАЗРЯДОВ :

- 0... 2 - ВЫБОР ЦВЕТА
- 3... 5 - ВЫБОР ФОНА
- 6 - РЕЖИМ: 0-МОНОХРОМ, 1-ЦВЕТОВОЙ
- 7 - СТРАНИЦА ВИДЕООЗУ: 0-ОСНОВНАЯ
1- ДОПОЛНИТЕЛЬНАЯ

ДИАПАЗОН ЗНАЧЕНИЙ УПРАВЛЯЮЩЕГО КОДА:

- 0... 63 (DEC) - МОНОХРОМНЫЙ РЕЖИМ
- 64... 127 (DEC) - ЦВЕТОВОЙ РЕЖИМ
- 128...255 (DEC) - ВТОРАЯ СТРАНИЦА

ТАБЛИЦЫ СООТВЕТСТВИЯ КОДОВ ЦВЕТАМ

МОНОХРОМНЫЙ РЕЖИМ					
ЦВЕТ			ФОН		
DEC	БАЙТ: 67543210	ЗНАЧЕНИЕ	БАЙТ: 67543210	ЗНАЧЕНИЕ	
0	00XXX000	- БЕЛЫЙ	0000XXX	- ЧЕРНЫЙ	
1	00XXX001	- ЗЕЛЕНый	00001XXX	- ТЕМНО-КРАСНЫЙ	
2	00XXX010	- СИНИЙ	00010XXX	- СИНИЙ	
3	00XXX011	- КРАСНЫЙ	00011XXX	- ТЕМ-СИРЕНЕВый	
4	00XXX100	- СИРЕНЕВый	00100XXX	- ТЕМ-ЗЕЛЕНый	
5	00XXX101	- ГОЛУБОЙ	00101XXX	- КОРИЧНЕВый	
6	00XXX110	- ЧРН. (ФОН:ЯРК.СИН)	00110XXX	- ТЕМНО-СИНИЙ	
7	00XXX111	- НЕТ ИЗОБРАЖЕНИЯ	00111XXX	- СЕРый	

ПРИМЕЧАНИЕ: УСТАНОВКА ЦВЕТА ЧЕРЕЗ <ESC>4:

- <ESC>41, <ESC>42 - НОРМАЛЬНЫЕ СИМВОЛЫ
- <ESC>43 - СДВОЕННЫЕ СИМВОЛЫ
- <ESC>44 - ЦВЕТ СОВПАДАЕТ С ФОНОМ

ЦВЕТОВОЙ РЕЖИМ					
ЦВЕТ (ФОН УСТАНОВЛИВАЕТСЯ ТЕМЖЕ КОДОМ)					ФОН
DEC	БАЙТ: 76543210	<ESC>41	<ESC>42	<ESC>43	
0	01XXX000	ЗЕЛЕНый	СИНИЙ	КРАСНЫЙ	ЧЕРНЫЙ***
1	01XXX001	ЗЕЛЕНый	СИНИЙ	КРАСНЫЙ	БЕЛый
2	01XXX010	СИНИЙ	РОЗОВый	ГОЛУБОЙ	ЯРКО ЗЕЛЕНый
3	01XXX011	ЗЕЛЕНый	ЖЕЛТый	БЕЛый	ЧЕРНЫЙ***
4	01XXX100	ЗЕЛЕНый	ГОЛУБОЙ	КРАСНЫЙ	ЧЕРНЫЙ***
5	01XXX101	КРАСНЫЙ	СИНИЙ	ГОЛУБОЙ	ЧЕРНЫЙ***
6	01XXX110	БЕЛый	ГОЛУБОЙ	КРАСНЫЙ	ЯРКО СИНИЙ
7	01XXX111	НЕТ ИЗОБРАЖЕНИЯ	НЕТ ИЗОБРАЖЕНИЯ	НЕТ ИЗОБРАЖЕНИЯ	--

ПРИМЕЧАНИЕ: *** - ЦВЕТ ФОНА ОПРЕДЕЛЯЕТСЯ РАЗРЯДАМИ 3,4,5
 АНАЛОГИЧНО МОНОХРОМНОМУ РЕЖИМУ.

<<<< КОНЕЦ >>>>

КОМАНДЫ УПРАВЛЯЮЩИХ СИМВОЛОВ

КОМАНДЫ УПРАВЛЯЮЩИХ СИМВОЛОВ ИСПОЛЬЗУЮТСЯ В ОСНОВНОМ КОМАНДНОМ РЕЖИМЕ ОС СР/М И ИНОГДА В ПРОЦЕССЕ ВЫПОЛНЕНИЯ ОПРЕДЕЛЕННЫХ ПРОГРАММ; ВВОД СПЕЦИАЛЬНЫХ КОМАНД ОСУЩЕСТВЛЯЕТСЯ ПРИ ОДНОВРЕМЕННОМ НАЖАТИИ КЛАВИШИ CTRL И ТОЙ, КОТОРАЯ СООТВЕТСТВУЕТ КОНКРЕТНОЙ КОМАНДЕ.

CTRL/C РЕИНИЦИАЛИЗАЦИЯ СИСТЕМЫ

CTRL/M ВОЗВРАТ КАРЕТКИ

CTRL/J ПЕРЕВОД СТРОКИ

CTRL./H ВОЗВРАТ КУРСОРА НАЗАД НА ОДНУ ПОЗИЦИЮ

DELETE ИЛИ RUBOUT УДАЛИТЬ РАСПОЛОЖЕННЫЙ СЛЕВА ОТ КУРСОРА СИМВОЛ

CTRL/U УДАЛИТЬ ТЕКУЩУЮ КОМАНДНУЮ СТРОКУ

CTRL/X УДАЛИТЬ И СТЕРЕТЬ ТЕКУЩУЮ КОМАНДНУЮ СТРОКУ

CTRL/E ПЕРЕХОД ТЕКУЩЕЙ КОМАНДНОЙ СТРОКИ НА СЛЕДУЮЩУЮ СТРОКУ

CTRL/R ПЕРЕПЕЧАТАТЬ ОТРЕДАКТИРОВАННУЮ КОМАНДНУЮ СТРОКУ НА СЛЕДУЮЩЕЙ СТРОКЕ И ОПУСТИТЬ ВСЕ СИМВОЛЫ, УДАЛЕННЫЕ С ПОМОЩЬЮ КОМАНДЫ DELETE ИЛИ RUBOUT

CTRL/S ОСТАНОВИТЬ/ПРОДОЛЖИТЬ ПЕРЕМЕЩЕНИЕ ТЕКСТА НА ЭКРАНЕ КОНСОЛИ

CTRL/P ВЫВЕСТИ НА УСТРОЙСТВО ПЕЧАТИ

(LST) ЛЮБОЙ ТЕКСТ, РАСПЕЧАТЫВАЕМЫЙ НА КОНСОЛИ

ОСНОВНЫЕ ПРОГРАММЫ ОС СР/М

STAT (СТАТУС ДЛЯ ХАРАКТЕРИСТИКИ ДИСКОВ И ФАЙЛОВ)

STAT ВЫВЕСТИ ОБ'ЕМ СВОБОДНОГО ПРОСТРАНСТВА НА ВСЕХ ДИСКАХ

ПРИМЕЧАНИЕ: ОБЫЧНО ПО ЭТОЙ КОМАНДЕ ВЫВОДИТСЯ ОБ'ЕМ НЕИСПОЛЬЗОВАННОГО ПРОСТРАНСТВА НА ТЕКУЩЕМ ДИСКЕ.

ОДНАКО ЕСЛИ ПЕРЕД ЭТИМ ОСУЩЕСТВЛЯЕТСЯ ОБРАЩЕНИЕ К ДРУГИМ ДИСКАМ (ЧТЕНИЕ, ЗАПИСЬ И ПР.)

ТО ВЫВОДИТСЯ ОБ'ЕМ НЕИСПОЛЬЗОВАННОГО ПРОСТРАНСТВА ДЛЯ ВСЕХ ЭТИХ ДИСКОВ.

ДЛЯ ВЫВОДА ОБ'ЕМА НЕИСПОЛЬЗОВАННОГО ПРОСТРАНСТВА НА КОНКРЕТНОМ ДИСКЕ, ПОСЛЕ КОМАНДЫ

STAT МОЖНО УКАЗАТЬ ЕГО ИМЯ (A:, B:, И Т.Д.)

STAT FILENAME.EXT ВЫВЕСТИ ОБ'ЕМ ФАЙЛА 'FILENAME.EXT' В КИЛОБАЙТАХ

STAT FILENAME.* ВЫВЕСТИ РАЗМЕРЫ ВСЕХ ФАЙЛОВ С ИМЕНЕМ 'FILENAME' В КИЛОБАЙТАХ

STAT *.EXT ВЫВЕСТИ РАЗМЕРЫ ВСЕХ ФАЙЛОВ С РАСШИРЕНИЕМ '.EXT' В КИЛОБАЙТАХ

STAT *.* ВЫВЕСТИ РАЗМЕРЫ ВСЕХ ФАЙЛОВ НА ДИСКЕ

ПРИМЕЧАНИЕ: ДЛЯ ПОЛУЧЕНИЯ БОЛЕЕ ДЕТАЛЬНОЙ ИНФОРМАЦИИ О РАЗМЕРАХ ФАЙЛОВ В ПЕРЕЧИСЛЕННЫЕ ВЫШЕ КОМАНДНЫЕ СТРОКИ МОЖНО ДОПОЛНИТЕЛЬНО ВКЛЮЧИТЬ ПАРАМЕТР '\$S'.

STAT FILENAME.EXT \$R/O УСТАНОВИТЬ СТАТУС ЗАЩИТЫ ФАЙЛА (ФАЙЛОВ) 'R/O' ('ТОЛЬКО ЧТЕНИЕ')

STAT FILENAME.* \$R/O

STAT *.EXT \$R/O

STAT *.* \$R/O

STAT FILENAME.EXT \$R/W УСТАНОВИТЬ СТАТУС ЗАЩИТЫ ФАЙЛА (ФАЙЛОВ) 'R/W' ('ЧТЕНИЕ/ЗАПИСЬ')

STAT FILENAME.# \$R/W

STAT *.EXT \$R/W

STAT *.* \$R/W

STAT X:=R/Ø ВРЕМЕННО УСТАНОВИТЬ СТАТУС ЗАЩИТЫ ДИСКА С ИМЕНЕМ 'X' КАК R/Ø ('ТОЛЬКО ЧТЕНИЕ')

STAT FILENAME.EXT \$SYS УСТАНОВИТЬ ДЛЯ ФАЙЛА (ФАЙЛОВ) АТРИБУТ 'СИСТЕМНЫЙ' С ТЕМ ЧТОБЫ

ЕГО (ИХ) ИМЯ НЕЛЬЗЯ БЫЛО ВЫВЕСТИ НА КОНСОЛЬ КОМАНДОЙ DIR.

STAT FILENAME.* \$SYS

STAT *.EXT \$SYS

STAT *.* \$SYS

STAT FILENAME.EXT \$DIR УСТАОВИТЬ ФАЙЛУ (ФАЙЛАМ) АТРИБУТ 'DIR' С ТЕМ, ЧТОБЫ ЕГО (ИХ)

МОЖНО БЫЛО ВЫВЕСТИ НА КОНСОЛЬ С ПОМОЩЬЮ КОМАНДЫ DIR.

STAT FILENAME.* \$DIR

STAT *.EXT \$DIR

STAT *.* \$DIR

STAT USR: ВЫВЕСТИ НОМЕР ТЕКУЩЕЙ ОБЛАСТИ ПОЛЬЗОВАТЕЛЯ И НОМЕРА ВСЕХ НЕПУСТЫХ
 (СОДЕРЖАЩИХ ФАЙЛЫ) ОБЛАСТЕЙ ПОЛЬЗОВАТЕЛЯ.
 STAT DSK: ВЫВЕСТИ ХАРАКТЕРИСТИКИ ВСЕХ АКТИВНЫХ ФАЙЛОВ.
 STAT X:DSK: ВЫВЕСТИ ХАРАКТЕРИСТИКИ ДИСКА С ИМЕНЕМ 'X'
 STAT DEV: ВЫВЕСТИ СПИСОК ТЕКУЩИХ НАЗНАЧЕНИЙ ФИЗИЧЕСКИХ УСТРОЙСТВ ЛОГИЧЕСКИМ.
 STAT VAL: ВЫВЕСТИ СПИСОК ДОПУСТИМЫХ НАЗНАЧЕНИЙ И ДРУГИХ ДОПУСТИМЫХ ФОРМАТОВ
 КОМАНДЫ STAT.
 STAT LOG:=PHU: НАЗНАЧИТЬ ФИЗИЧЕСКОЕ УСТРОЙСТВО 'PHU:' ЛОГИЧЕСКОМУ УСТРОЙСТВУ 'LOG:',
 ГДЕ ДОПУСТИМЫМИ НАЗНАЧЕНИЯМИ ЯВЛЯЮТСЯ:
 ЛОГИЧЕСКОЕ УСТРОЙСТВО ФИЗИЧЕСКОЕ УСТРОЙСТВО
 CON:= TTY:, CRT:, BAT: ИЛИ VC1:
 RDR:= TTY:, PTR:, VR1: ИЛИ VR2:
 PUN:= TTY:, PTR:, VP1: ИЛИ VP2:
 LST:= TTY:, CRT:, LPT: ИЛИ VL1:

PIP (ПРОГРАММА ОБМЕНА С ПЕРИФЕРИЙНЫМИ УСТРОЙСТВАМИ)

PIP ЗАГРУЗИТЬ ПРОГРАММУ 'PIP' В ОПЕРАТИВНУЮ ПАМЯТЬ И ПЕРЕЙТИ В РЕЖИМ ДИАЛОГА.
 PIP Y:=X: FILENAME.EXT КОПИРОВАТЬ ФАЙЛ (ФАЙЛЫ) С ДИСКА 'X:' НА ДИСК 'Y:'
 PIP Y:=X: FILENAME.*
 PIP Y:=X: *.EXT
 PIP Y:=X: *.*

 PIP Y:NEWFILE.EXT=X:OLDFILE.EXT КОПИРОВАТЬ ФАЙЛ 'OLDFILE.EXT', РАСПОЛОЖЕННЫЙ НА ДИСКЕ 'X:',
 НА ДИСК 'Y:' И ИЗМЕНИТЬ ИМЯ ФАЙЛА-КОПИИ НА 'NEWFILE.EXT'

 PIP Y:NEWFILE.EXT=X:OLDFILE1.EXT, OLDFILE2.EXT, OLDFILE3.EXT
 КОПИРОВАТЬ В УКАЗАННОМ ВЫШЕ ПОРЯДКЕ ФАЙЛЫ, РАСПОЛОЖЕННЫЕ НА
 ДИСКЕ 'X:' И ПОМЕСТИТЬ КОПИИ В ТОМ ЖЕ САМОМ ПОРЯДКЕ В ФАЙЛ
 'NEWFILE.EXT', РАСПОЛОЖЕННЫЙ НА ДИСКЕ 'Y:' (ОБЪЕДИНИТЬ ФАЙЛЫ)

 PIP CON:=X:FILENAME.EXT КОПИРОВАТЬ ФАЙЛ 'FILENAME.EXT', РАСПОЛОЖЕННЫЙ НА ДИСКЕ
 PIP TTY:=X:FILENAME.EXT 'X:' НА УКАЗАННОЕ КОНСОЛЬНОЕ УСТРОЙСТВО
 PIP VC1:=X:FILENAME.EXT
 PIP CRT:=X:FILENAME.EXT

 PIP PUN:=X:FILENAME.EXT КОПИРОВАТЬ ФАЙЛ 'FILENAME.EXT', РАСПОЛОЖЕННЫЙ НА ДИСКЕ
 PIP PTR:=X:FILENAME.EXT 'X:' НА УКАЗАННОЕ УСТРОЙСТВО ВЫВОДА
 PIP VP1:=X:FILENAME.EXT
 PIP VP2:=X:FILENAME.EXT
 PIP OUT:=X:FILENAME.EXT

 PIP LST:=X:FILENAME.EXT КОПИРОВАТЬ ФАЙЛ 'FILENAME.EXT', РАСПОЛОЖЕННЫЙ НА ДИСКЕ
 PIP LPT:=X:FILENAME.EXT 'X:', НА УКАЗАННОЕ УСТРОЙСТВО ПЕЧАТИ
 PIP UL1:=X:FILENAME.EXT
 PIP PRN:=X:FILENAME.EXT

 PIP X:FILENAME.EXT=CON: ЗАПИСЫВАТЬ ДАННЫЕ, ВВОДИМЫЕ С УКАЗАННОГО КОНСОЛЬНОГО
 PIP X:FILENAME.EXT=TTY: УСТРОЙСТВА, В ФАЙЛ 'FILENAME.EXT', РАСПОЛОЖЕННЫЙ
 PIP X:FILENAME.EXT=CRT: НА ДИСКЕ "X:" ДО ВВОДА КОМАНДЫ CTRL/Z
 PIP X:FILENAME.EXT=UC1:

 PIP X:FILENAME.EXT=RDR: ЗАПИСЫВАТЬ ДАННЫЕ, ВВОДИМЫЕ С УКАЗАННОГО УСТРОЙСТВА,
 PIP X:FILENAME.EXT=PTR: РАБОТАЮЩЕГО В РЕЖИМЕ "ТОЛЬКО ВВОД", В ФАЙЛ "FILENAME.EXT",
 PIP X:FILENAME.EXT=UR1: РАСПОЛОЖЕННЫЙ НА ДИСКЕ "X:", ДО ВВОДА КОМАНДЫ CTRL/Z
 PIP X:FILENAME.EXT=UR2:
 PIP X:FILENAME.EXT=INP:

PIP PUN:=NUL: Вывести начальные и хвостовые промежутки при копировании
 PIP PUN:=NUL:, X:FILENAME.EXT файла 'FILENAME.EXT', расположенного на диске 'X:',
 PIP PUN:=NUL:, X:FILENAME.EXT, NUL: на устройство, работающее в режиме 'только вывод'
 PIP PUN:=X:FILENAME.EXT, NUL:

PIP PUN:=X:FILENAME.EXT, EOR Вывести маркер конца файла при копировании файла
 'FILENAME.EXT', расположенного на диске 'X:',
 на устройство, работающее в режиме 'только вывод'.

ДОПОЛНИТЕЛЬНЫЕ КОМАНДЫ PIP

[V] ПРОВЕРИТЬ ПРАВИЛЬНОСТЬ КОПИРОВАНИЯ ПУТЕМ СРАВНЕНИЯ РЕЗУЛЬТИРУЮЩЕГО И ИСХОДНОГО
 ФАЙЛОВ.
 [E] ОТОБРАЖАТЬ КОПИРУЕМЫЙ ТЕКСТ НА КОНСОЛИ.
 [SSSTRING^Z] НАЧАТЬ КОПИРОВАНИЕ ПОСЛЕ ТОГО, КАК В ИСХОДНОМ ФАЙЛЕ БУДЕТ ОБНАРУЖЕНА СТРОКА
 'STRING' (для завершения строки используется команда CTRL/Z)
 [QSTRING^Z] ЗАВЕРШИТЬ КОПИРОВАНИЕ ПОСЛЕ ТОГО, КАК В ИСХОДНОМ ФАЙЛЕ БУДЕТ ОБНАРУЖЕНА
 СТРОКА 'STRING' (для завершения строки используется команда CTRL/Z)
 [L] В ПРОЦЕССЕ КОПИРОВАНИЯ ПРЕОБРАЗОВЫВАТЬ ВСЕ СИМВОЛЫ ВЕРХНЕГО РЕГИСТРА В
 СИМВОЛЫ НИЖНЕГО РЕГИСТРА.
 [U] В ПРОЦЕССЕ КОПИРОВАНИЯ ПРЕОБРАЗОВЫВАТЬ ВСЕ СИМВОЛЫ НИЖНЕГО РЕГИСТРА В
 СИМВОЛЫ ВЕРХНЕГО РЕГИСТРА.
 [DN] В ПРОЦЕССЕ КОПИРОВАНИЯ УДАЛЯТЬ В КАЖДОЙ СТРОКЕ ВСЕ СИМВОЛЫ, РАСПОЛОЖЕННЫЕ
 ПОСЛЕ ПОЗИЦИИ С НОМЕРОМ "N".
 [TN] В ПРОЦЕССЕ КОПИРОВАНИЯ ПРЕОБРАЗОВЫВАТЬ ВСЕ СИМВОЛЫ ТАБУЛЯЦИИ КОДА
 ASCII В 'N'ПРОБЕЛОВ.
 [F] В ПРОЦЕССЕ КОПИРОВАНИЯ УДАЛИТЬ ВСЕ СИМВОЛЫ ПЕРЕВОДА ФОРМАТА.
 [PN] В ПРОЦЕССЕ КОПИРОВАНИЯ ЧЕРЕЗ КАЖДЫЕ 'N' СТРОК ВСТАВЛЯТЬ СИМВОЛЫ ПЕРЕВОДА ФОРМАТА.
 [N] В ПРОЦЕССЕ КОПИРОВАНИЯ ДОПОЛНИТЬ СТРОКИ СООТВЕТСТВУЮЩИМИ ПОРЯДКОВЫМИ НОМЕРАМИ БЕЗ
 ВЕДУЩИХ НУЛЕЙ.
 [N2] В ПРОЦЕССЕ КОПИРОВАНИЯ ДОПОЛНИТЬ СТРОКИ СООТВЕТСТВУЮЩИМИ ПОРЯДКОВЫМИ НОМЕРАМИ С
 ЛИДИРУЮЩИМИ НУЛЯМИ.
 [R] КОПИРОВАТЬ 'СИСТЕМНЫЙ' ФАЙЛ (УСТАНОВЛИВАЕТСЯ С ПОМОЩЬЮ КОМАНДЫ STAT).
 [W] УДАЛИТЬ ФАЙЛ СО СТАТУСОМ ЗАЩИТЫ 'R/O' (УСТАНОВЛИВАЕТСЯ С ПОМОЩЬЮ КОМАНДЫ STAT),
 ИМЕЮЩИМИ ТОЖЕ ИМЯ ЧТО И РЕЗУЛЬТИРУЮЩИЙ ФАЙЛ.
 [GN] КОПИРОВАТЬ ФАЙЛ (ФАЙЛЫ) ИЗ ОБЛАСТИ ПОЛЬЗОВАТЕЛЯ С НОМЕРОМ 'N' (ОТ 0 ДО 15) В ФАЙЛ,
 РАСПОЛОЖЕННЫЙ В ТЕКУЩЕЙ ОБЛАСТИ ПОЛЬЗОВАТЕЛЯ.
 [O] РАССМАТРИВАТЬ СОДЕРЖИМОЕ КОПИРУЕМОГО ФАЙЛА (ФАЙЛОВ) КАК ОБ'ЕКТНЫЙ/МАШИННЫЙ КОД.
 [B] УСТАНОВИТЬ 'БЛОЧНЫЙ' РЕЖИМ КОПИРОВАНИЯ.
 [H] ИСПОЛЬЗОВАТЬ ШЕСТНАДЦАТИРИЧНЫЙ ФОРМАТ ФИРМЫ INTEL ПРИ ПЕРЕДАЧЕ/ПРИЕМЕ ДАННЫХ НА/С
 УСТРОЙСТВО(A), РАБОТАЮЩЕЕ В РЕЖИМЕ 'ТОЛЬКО ВЫВОД'/'ТОЛЬКО ВВОД'.
 [I] ПРИ ПЕРЕДАЧЕ ДАННЫХ В ШЕСТНАДЦАТИРИЧНОМ ФОРМАТЕ ФИРМЫ INTEL ИГНОРИРОВАТЬ ВСЕ НУЛЕВЫЕ
 (NULL) ЗАПИСИ.
 [Z] В ПРОЦЕССЕ КОПИРОВАНИЯ СИМВОЛОВ В КОДЕ ASCII УСТАНОВЛИВАТЬ БИТ ЧЕТНОСТИ В 0.

ED (ТЕКСТОВЫЙ РЕДАКТОР)

ED FILENAME.EXT ЗАГРУЗИТЬ ПРОГРАММУ 'ED' В ОПЕРАТИВНУЮ ПАМЯТЬ И ОТКРЫТЬ ФАЙЛ 'FILENAME.EXT' ДЛЯ СОЗДАНИЯ И/ИЛИ РЕДАКТИРОВАНИЯ.

КОМАНДЫ ПРОГРАММЫ 'ED':

NA ДОБАВИТЬ 'N' СТРОК ТЕКСТА ИЗ ИСХОДНОГО ФАЙЛА В БУФЕР РЕДАКТИРОВАНИЯ ('NA') ДО ТЕХ ПОР, ПОКА ОА БУФЕР НЕ БУДЕТ ЗАПОЛНЕН НА ПОЛОВИНУ ('OA') ИЛИ ПОЛНОСТЬЮ ('#A').
#A

I ВСТАВЛЯТЬ ТЕКСТ ДО ТЕХ ПОР, ПОКА НЕ БУДЕТ ПОЛУЧЕНА КОМАНДА CTRL/Z
I TEXT ВСТАВИТЬ УКАЗАННЫЙ ТЕКСТ И ОСТАТЬСЯ В РЕЖИМЕ ВСТАВКИ
I TEXT^Z ВСТАВИТЬ УКАЗАННЫЙ ТЕКСТ И ВЫЙТИ ИЗ РЕЖИМА ВСТАВКИ

U УСТАНОВИТЬ ('U') ИЛИ ОТМЕНИТЬ ('-U') ПРЕОБРАЗОВАНИЕ СИМВОЛОВ В КОДЫ ВЕРХНЕГО РЕГИСТРА
NC ПЕРЕМЕСТИТЬ УКАЗАТЕЛЬ СР НА 'N' СИМВОЛОВ ВПРАВО ('NC') ИЛИ ВЛЕВО ('-NC').
-NC

NL ПЕРЕМЕСТИТЬ УКАЗАТЕЛЬ СР НА 'M' СТРОК ВПЕРЕД ('NL') ИЛИ НАЗАД ('-NL').
-NL

V ВЫВЕСТИ ('V') ИЛИ ПРЕКРАТИТЬ ВЫВОД ('-V') НОМЕРОВ СТРОК
-V

N: ПЕРЕМЕСТИТЬ УКАЗАТЕЛЬ СР НА СТРОКУ С НОМЕРОМ 'N':
B ПЕРЕМЕСТИТЬ УКАЗАТЕЛЬСРВ НАЧАЛО (B) ИЛИ В КОНЕЦ БУФЕРА
-B

T ВЫВЕСТИ ТЕКСТ, РАСПОЛОЖЕННЫЙ В ТЕКУЩЕЙ СТРОКЕ ДО <T> ИЛИ ПОСЛЕ (ОТ) УКАЗАТЕЛЯ СР, ОТ ИЛИ ВСЮ СТРОКУ ПОЛНОСТЬЮ (ОТТ).
ОТТ

NT ВЫВЕСТИ 'N' СТРОК, РАСПОЛОЖЕННЫХ ПОСЛЕ ('NT') ИЛИ ДО ('-NT') УКАЗАТЕЛЯ СР,
-NT ИЛИ ВСЕ СТРОКИ, РАСПОЛОЖЕННЫЕ ПОСЛЕ ('#T') ИЛИ ДО ('-#T') УКАЗАТЕЛЯ СР.
#T
-#T

NP ПЕРЕМЕСТИТЬ УКАЗАТЕЛЬ СР НА 'N' СТРАНИЦ ВПЕРЕД ('NP') ИЛИ НАЗАД ('-NP') И ВЫВЕСТИ ТЕКСТ.
-NP

ND УДАЛИТЬ 'N' СИМВОЛОВ, РАСПОЛОЖЕННЫХ В ТЕКУЩЕЙ СТРОКЕ ПОСЛЕ ('ND') ИЛИ ДО ('-ND') УКАЗАТЕЛЯ.
-ND

NK УДАЛИТЬ 'N' СТРОК, РАСПОЛОЖЕННЫХ ПОСЛЕ ('NK') ИЛИ ДО ('-NK') ПОЗИЦИИ УКАЗАТЕЛЯ СР.
-NK

F TEXT^Z НАЙТИ В БУФЕРЕ СТРОКУ 'TEXT' ('FTEXT^Z')

NF TEXT^Z ИЛИ НАЙТИ В БУФЕРЕ 'N'-ОЕ ВХОЖДЕНИЕ СТРОКИ 'TEXT' ('NFTEXT^Z'). ДЛЯ ЗАВЕРШЕНИЯ СТРОКИ 'TEXT' НЕОБХОДИМО ИСПОЛЬЗОВАТЬ КОМАНДУ CTRL/Z.

N TEXT^Z АНАЛОГИЧНО ПРЕДЫДУЩЕЙ ЗА ИСКЛЮЧЕНИЕМ ТОГО, ЧТО ЗДЕСЬ ПРИ НЕОБХОДИМОСТИ ВЫПОЛНЯЕТСЯ

NN TEXT^Z АВТОМАТИЧЕСКАЯ ЗАГРУЗКА ТЕКСТА ИЗ ИСХОДНОГО ФАЙЛА.

S FIND^ZREPLACE^Z НАЙТИ В БУФЕРЕ СТРОКУ 'FIND' И ЗАМЕНИТЬ ЕЕ СТРОКОЙ 'REPLACE'.

NS FIND^ZREPLACE^Z ВЫПОЛНИТЬ КОМАНДУ ПОИСКА-ЗАМЕЩЕНИЯ 'N' РАЗ. СИМВОЛЫ '^Z' ОБОЗНАЧАЮТ КОМАНДУ CTRL/Z.

J FIND^ZINSERT^ZEND^Z НАЙТИ В БУФЕРЕ СТРОКУ 'FIND' .ВСТАВИТЬ ПОСЛЕ НЕЕ СТРОКУ 'INSERT' И УДАЛИТЬ РАСПОЛОЖЕННЫЙ ДАЛЕЕ ТЕКСТ ВПЛОТЬ ДО СТРОКИ 'END'.

NJ FIND^ZINSERT^ZEND^Z ВЫПОЛНИТЬ ОПЕРАЦИЮ ПОИСКА-СЦЕПЛЕНИЯ 'N' РАЗ. СИМВОЛЫ '^Z' ОБОЗНАЧАЮТ КОМАНДУ CTRL/Z .

NM COMMANDS ВЫПОЛНИТЬ 'N'РАЗ ПОСЛЕДОВАТЕЛЬНОСТЬ КОМАНД 'COMMANDS' .

OV ОПРЕДЕЛИТЬ РАЗМЕРЫ СВОБОДНОГО И ЗАНЯТОГО ПРОСТРАНСТВА В БУФЕРЕ.

W ЗАПИСЬ ИЗ БУФЕРА РЕДАКТИРОВАНИЯ ВО ВРЕМЕННЫЙ ФАЙЛ ПЕРВУЮ ('W')

NW ИЛИ 'N' ПЕРВЫХ ('NW') СТРОК.

OW ЗАПИСАТЬ ИЗ БУФЕРА РЕДАКТИРОВАНИЯ ВО ВРЕМЕННЫЙ ФАЙЛ ПОЛОВИНУ ('OW')

#W ИЛИ ВСЕ ('#W') СТРОКИ.

NX ЗАПИСАТЬ 'N' СТРОК (НАЧИНАЯ С ТЕКУЩЕЙ) ИЗ БУФЕРА ВО ВРЕМЕННЫЙ БИБЛИОТЕЧНЫЙ ФАЙЛ.

R ЗАПИСАТЬ В БУФЕР РЕДАКТИРОВАНИЯ ПОЛНОЕ СОДЕРЖИМОЕ ВРЕМЕННОГО БИБЛИОТЕЧНОГО ФАЙЛА, (СОЗДАННОГО ПО КОМАНДЕ 'NX').

R FILENAME ЗАПИСАТЬ В БУФЕР РЕДАКТИРОВАНИЯ ПОЛНОЕ СОДЕРЖИМОЕ БИБЛИОТЕЧНОГО ФАЙЛА 'FILENAME' (С РАСШИРЕНИЕМ 'LIB').

H ЗАПИСАТЬ РЕЗУЛЬТАТЫ РЕДАКТИРОВАНИЯ В ВЫХОДНОЙ ФАЙЛ БЕЗ ЗАВЕРШЕНИЯ РАБОТЫ ПРОГРАММЫ 'ED'.

E ЗАВЕРШИТЬ РАБОТУ ПРОГРАММЫ 'ED' (ВЕРНУТЬ УПРАВЛЕНИЕ ОС CP/M) И СОХРАНИТЬ РЕЗУЛЬТАТЫ РЕДАКТИРОВАНИЯ В ВЫХОДНОМ ФАЙЛЕ.

O ОЧИСТИТЬ БУФЕР РЕДАКТИРОВАНИЯ И ВЫХОДНОЙ ФАЙЛ БЕЗ ЗАВЕРШЕНИЯ РАБОТЫ ПРОГРАММЫ 'ED'.

Q ЗАВЕРШИТЬ РАБОТУ ПРОГРАММЫ 'ED' (ВЕРНУТЬ УПРАВЛЕНИЕ ОС CP/M) И ОЧИСТИТЬ БУФЕР И ВЫХОДНОЙ ФАЙЛ.

DDT (ИНСТРУМЕНГАЛЬНОЕ СРЕДСТВО ДИНАМИЧЕСКОЙ ОТЛАДКИ ПРОГРАММ)

DDT FILENAME.EXT ЗАГРУЗИТЬ ПРОГРАММУ 'DDT' И ФАЙЛ С ОТЛАЖИВАЕМОЙ ПРОГРАММОЙ 'FILENAME' В ОПЕРАТИВНУЮ ПАМЯТЬ.

КОМАНДЫ ПРОГРАММЫ 'DDT'

D ВЫВЕСТИ В ШЕСТНАДЦАТИРИЧНОМ ФОРМАТЕ СОДЕРЖИМОЕ СЛЕДУЮЩИХ 192 ЯЧЕЕК ПАМЯТИ, НАЧИНАЯ С ТЕКУЩЕЙ ('D') ИЛИ С ЯЧЕЙКИ, РАСПОЛОЖЕННОЙ ПО ЗАДАННОМУ АДРЕСУ ('DSTART').

DSTART,END ВЫВЕСТИ СОДЕРЖИМОЕ ЯЧЕЕК ПАМЯТИ, РАСПОЛОЖЕННЫХ НАЧИНАЯ С АДРЕСА 'START' И КОНЧАЯ АДРЕСОМ 'END'.

ESTART,END,DATA ЗАПИСАТЬ ВО ВСЕ ЯЧЕЙКИ ПАМЯТИ С АДРЕСА 'START' ПО АДРЕС 'END' ШЕСТНАДЦАТЕРИЧНОЕ ЗНАЧЕНИЕ 'DATA'.

IFILENAME.EXT СОЗДАТЬ БЛОК УПРАВЛЕНИЯ ФАЙЛОМ И ЗАПИСАТЬ В НЕГО ИМЯ 'FILENAME.EXT' С ТЕМ, ЧТОБЫ ЗАГРУЖАТЬ ФАЙЛ С ТАКИМ ИМЕНЕМ В ОПЕРАТИВНУЮ ПАМЯТЬ ПО КОМАНДЕ 'R'.

R ЗАГРУЗИТЬ СОДЕРЖИМОЕ ФАЙЛА, ИМЯ КОТОРОГО УКАЗАНО В КОМАНДЕ I, В ОПЕРАТИВНУЮ НАЧИНАЯ С ТЕКУЩЕЙ ЯЧЕЙКИ ('R') ИЛИ С ЯЧЕЙКИ, РАСПОЛОЖЕННОЙ ПО АДРЕСУ 'START' ('RSTART').

L РЕАССЕМБЛИРОВАТЬ И ВЫВЕСТИ НА КОНСОЛЬ СОДЕРЖИМОЕ СЛЕДУЮЩИХ 192 ЯЧЕЕК ПАМЯТИ, НАЧИНАЯ С ТЕКУЩЕЙ ЯЧЕЙКИ('L'>ИЛИ С ЯЧЕЙКИ, РАСПОЛОЖЕННОЙ ПО АДРЕСУ 'START' ('LSTART').

LSTART,END РЕАССЕМБЛИРОВАТЬ И ВЫВЕСТИ НА КОНСОЛЬ СОДЕРЖИМОЕ ЯЧЕЕК ПАМЯТИ,НАЧИНАЯ С АДРЕСА 'START'И КОНЧАЯ АДРЕСОМ 'END'.

SSTART ВЫВЕСТИ В ШЕСТНАДЦАТЕРИЧНОМ ФОРМАТЕ И, ВОЗМОЖНО, ИЗМЕНИТЬ СОДЕРЖИМОЕ ЯЧЕЙКИ ПАМЯТИ, РАСПОЛОЖЕННОЙ ПО АДРЕСУ 'START'. ПРОДОЛЖИТЬ ОБРАБОТКУ СЛЕДУЮЩИХ ЯЧЕЕК ДО ВВОДА СИМВОЛА ./.

ASTART ВСТАВИТЬ, НАЧИНАЯ С АДРЕСА 'START'ОПЕРАТОР ЯЗЫКА АССЕМБЛЕР, ПРОДОЛЖАТЬ ОБРАБОТКУ ПОСЛЕДУЮЩИХ ЯЧЕЕК ПАМЯТИ ДО ВВОДА СИМВОЛА ./.

MSTART1,END1,START2 КОПИРОВАТЬ СОДЕРЖИМОЕ ОБЛАСТИ ПАМЯТИ, РАСПОЛОЖЕННОЙ С АДРЕСА 'START1' ПО АДРЕС 'END1', В ДРУГУЮ ОБЛАСТЬ ПАМЯТИ, РАСПОЛОЖЕННУЮ НАЧИНАЯ С АДРЕСА 'START2'.

G ВЫПОЛНИТЬ ВСЮ ПРОГРАММУ БЕЗ ПРЕРЫВАНИЙ.

GSTART ВЫПОЛНИТЬ ПРОГРАММУ НАЧИНАЯ С АДРЕСА 'START'.

GSTART,END НАЧАТЬ ВЫПОЛНЕНИЕ В ТОЧКЕ С АДРЕСОМ 'START' И ПРЕРВАТЬ ВЫПОЛНЕНИЕ В ТОЧКЕ С АДРЕСОМ 'END'.

GSTART,END1,END2 НАЧАТЬ ВЫПОЛНЕНИЕ ПРОГРАММЫ В ТОЧКЕ С АДРЕСОМ 'START' И ПРЕРВАТЬ ВЫПОЛНЕНИЕ ЛИБО В ТОЧКЕ С АДРЕСОМ 'END1', ЛИБО В ТОЧКЕ С АДРЕСОМ 'END2'.

G,END НАЧАТЬ ВЫПОЛНЕНИЕ ПРОГРАММЫ С ТЕКУЩЕЙ ЯЧЕЙКИ И ПРЕРВАТЬ ВЫПОЛНЕНИЕ В ТОЧКЕ С АДРЕСОМ 'END'.

G, END1, END2	НАЧАТЬ ВЫПОЛНЕНИЕ ПРОГРАММЫ С ТЕКУЩЕЙ ЯЧЕЙКИ И ПРЕРВАТЬ ВЫПОЛНЕНИЕ ЛИБО В ТОЧКЕ С АДРЕСОМ 'END1', ЛИБО В ТОЧКЕ С АДРЕСОМ 'END2'.
U NUMBER	ВЫПОЛНИТЬ 'NUMBER' МАШИННЫХ ИНСТРУКЦИЙ И ПО ЗАВЕРШЕНИИ ВЫВЕСТИ СОДЕРЖИМОЕ РЕГИСТРОВ ЦЕНТРАЛЬНОГО ПРОЦЕССОРА.
T NUMBER	ТРАССИРОВАТЬ ВЫПОЛНЕНИЕ 'NUMBER' МАШИННЫХ ИНСТРУКЦИЙ, РАСПЕЧАТЫВАЮЩИХ СОДЕРЖИМОЕ РЕГИСТРОВ ЦЕНТРАЛЬНОГО ПРОЦЕССОРА ПОСЛЕ ВЫПОЛНЕНИЯ КАЖДОЙ ИНСТРУКЦИИ.
X	ВЫВЕСТИ ТЕКУЩЕЕ СОДЕРЖИМОЕ ВСЕХ РЕГИСТРОВ ЦЕНТРАЛЬНОГО ПРОЦЕССОРА.
XREGISTER	РАСПЕЧАТАТЬ СОДЕРЖИМОЕ РЕГИСТРА 'REGISTER'.

ОПИСАНИЕ КОМАНД ЭКРАННОГО РЕДАКТОРА WORLD MASTER

ВИДЕО РЕЖИМ

^O, ^M	ВКЛЮЧИТЬ, ВЫКЛЮЧИТЬ ВСТАВКУ	^I	ВВЕСТИ ТАБУЛЯЦИЮ (ТАВ)
^S	МАРКЕР ВЛЕВО НА ОДИН СИМВОЛ	^N	ВВЕСТИ LF И CR
^D	МАРКЕР ВПРАВО НА ОДИН СИМВОЛ	^PX	ВВЕСТИ В ФАЙЛ СИМВОЛ X
^A	МАРКЕР ВЛЕВО НА ОДНО СЛОВО	^J, LF	HELP MACRO
^F	МАРКЕР ВПРАВО НА ОДНО СЛОВО	^W	ОПУСТИТЬ ЭКРАН НА ОДНУ СТРОКУ
^Q	МАРКЕР НА ОДНУ ТАБУЛЯЦИЮ	^Z	ПОЛНЯТЬ ЭКРАН НА ОДНУ СТРОКУ
^E	МАРКЕР ВВЕРХ НА ОДНУ СТРОКУ	^R	ПОКАЗАТЬ ПРЕДЫДУЩУЮ СТРАНИЦУ
^X	МАРКЕР ВНИЗ НА ОДНУ СТРОКУ	^C	ПОКАЗАТЬ СЛЕДУЮЩУЮ СТРАНИЦУ
^^	МАРКЕР В НАЧАЛО/КОНЕЦ ЭКРАНА	DEL	ОТМЕНИТЬ ПРЕДЫДУЩУЮ КОМАНДУ

^G	УНИЧТОЖИТЬ СИМВОЛ в ПОЗИЦИИ МАРКЕРА
^\ ^T ^U ^K ^Y	УДАЛИТЬ ЛЕВУЮ ЧАСТЬ ТЕКУЩЕГО СЛОВА УДАЛИТЬ ПРАВУЮ ЧАСТЬ ТЕКУЩЕГО СЛОВА УДАЛИТЬ ЛЕВУЮ ЧАСТЬ ТЕКУЩЕЙ СТРОКИ УДАЛИТЬ ПРАВУЮ ЧАСТЬ ТЕКУЩЕЙ СТРОКИ УДАЛИТЬ СТРОКУ И ПРИЗНАК ЕЕ КОНЦА

КОМАНДНЫЙ РЕЖИМ

УСЛОВНЫЕ ОБОЗНАЧЕНИЯ:

+-	ВЫПОЛНЕНИЕ ОПЕРАЦИЙ ВПЕРЕД/ИАЗАД
@	ОБОЗНАЧЕНИЕ CR ИЛИ LF
\$	ESC ИЛИ ^Z ИЛИ CR
N	ВЫПОЛНИТЬ N РАЗ. ПО УМОЛЧАНИЮ - 1 РАЗ, N=# - ТО 65535 РАЗ

КОМАНДЫ:

+-NC	ПЕРЕДВИНУТЬ МАРКЕР НА N ЗНАКОВ
+-NL	ПЕРЕДВИНУТЬ МАРКЕР НА N СТРОК
+-NT	ОТПЕЧАТАТЬ N СТРОК
+-NP	ПЕРЕДВИНУТЬ НА 23 СТРОКИ И РАСПЕЧАТАТЬ
+-ND	УДАЛИТЬ N СИМВОЛОВ
+-NK	УДАЛИТЬ N СТРОК
NZ	ЗАДЕРЖКА N СЕКУНД
NITEXT\$	ВВЕСТИ "ТЕХТ" N РАЗ
I@	ВКЛЮЧИТЬ РЕЖИМ ВВОДА (ВЫХОД ESC ИЛИ ^Z)
A@	ДОБАВИТЬ ИЛИ ВСТАВИТЬ ОДНУ СТРОКУ
NATEXT\$	
N<...>	ЦИКЛ: ПОВТОРИТЬ ... N РАЗ, ПО УМОЛЧАНИЮ #

КОМАНДНЫЙ РЕЖИМ

+-NFKлюч\$	КОРОТКИЙ ПОИСК КЛЮЧА N РАЗ
+-NNKлюч\$	ДЛИННЫЙ ПОИСК КЛЮЧА N РАЗ
+-NSKлюч\$ТЕХТ\$	ПОДСТАНОВКА "ТЕХТ" ПОСЛЕ КОРОТКОГО ПОИСКА КЛЮЧА
+-NRKлюч\$ТЕХТ\$	ЗАМЕНА "ТЕХТ" ПОСЛЕ ДЛИННОГО ПОИСКА КЛЮЧА
/F, /N, /S, /R,	ТО ЖЕ В МАКРООПРЕДЕЛЕНИЯХ (СКОБКАХ), ГДЕ НЕЛЬЗЯ СТАВИТЬ \$
Y[D:] ИМЯ.ТИП	ЧИТАТЬ ФАЙЛ "ИМЯ.ТИП" С ДИСКА D В ПОЗИЦИЮ УКАЗАТЕЛЯ
NW[D:] ИМЯ.ТИП	ЗАПИСАТЬ В ФАЙЛ "ИМЯ.ТИП" N СТРОК С ПОЗИЦИИ УКАЗАТЕЛЯ

^N	ВСТАВИТЬ CR И LF В ТЕХТ
^Y	ВСТАВИТЬ ESC
^A	НЕОПРЕДЕЛЕННЫЙ СИМВОЛ В КЛЮЧЕ ПОИСКА
^OX	СИМВОЛ, ОТЛИЧНЫЙ ОТ 'X' В КЛЮЧЕ ПОИСКА

КОМАНДНЫЙ РЕЖИМ

NQP	ПОМЕСТИТЬ N ЛИНИЙ В Q БУФЕР
N/QP	ДОБАВИТЬ В Q БУФЕР N СТРОК И УНИЧТОЖИТЬ ИХ В ФАЙЛЕ
NQG	ПОЛУЧИТЬ N КОПИЙ Q БУФЕРА В ПОЗИЦИЮ МАРКЕРА
QT	РАСПЕЧАТАТЬ Q БУФЕР
QK	УНИЧТОЖИТЬ Q БУФЕР
QX	ВЫПОЛНИТЬ КОМАНДНУЮ СТРОКУ В Q БУФЕРЕ
QLTEXT\$	ЗАНЕСТИ ТЕХТ В Q БУФЕР
N/QLTEXT\$	ДОБАВИТЬ N РАЗ "ТЕХТ" В Q БУФЕР
V	ВХОД В ВИДЕО РЕЖИМ
N!	ВСТАВИТЬ КОД N В ПОЗИЦИЮ МАРКЕРА
;	ВСЕ ПОСЛЕДУЮЩЕЕ СЧИТАТЬ КОММЕНТАРИЯМИ
E	ВЫХОД ИЗ РЕДАКТОРА
H	КОНЕЦ РЕДАКТИРОВАНИЯ И НАЧАТЬ СНОВА
Q и ^C	ПРЕРЫВАНИЕ РЕДАКТИРОВАНИЯ
O	ВОЗВРАТ К ИСХОДНОМУ ФАЙЛУ
^Q	HELP MACRO

СТАНДАРТНЫЕ ТИПЫ ФАЙЛОВ

.ASM -ФАЙЛ СОДЕРЖИТ ИСХОДНЫЙ ТЕКСТ ПРОГРАММЫ НА АССЕМБЛЕРЕ	DUMP.ASM PRINTER.ASM
.BAK -ФАЙЛ ЯВЛЯЕТСЯ РЕЗЕРВНОЙ КОПИЕЙ ДРУГОГО ФАЙЛА, ПРОЖДАЮТСЯ ТЕКСТОВЫМИ РЕДАКТОРАМИ.	TEXT.BAK LETTER1.BAK
.BAS -ФАЙЛ СОДЕРЖИТ ИСХОДНЫЙ ТЕКСТ ПРОГРАММЫ НА ЯЗЫКЕ БЕЙСИК	ACCOUNT.BAS GAME1.BAS
.BSC -ФАЙЛ СОДЕРЖИТ ИСХОДНЫЙ ТЕКСТ ПРОГРАММЫ НА ЯЗЫКЕ БЕЙСИК ИНТЕРПРЕТАТОРОВ BSC.COM, LBASIC.COM И OBASIC.COM (ДЛЯ ПЭВМ "ОКЕАН")	PROG.BSC
.CRF -ФАЙЛ СОДЕРЖИТ ТАБЛИЦУ ПЕРЕКРЕСТНЫХ ССЫЛОК ПРОГРАММЫ НА АССЕМБЛЕРЕ	COMP.CRF MAR.CRF
.COB -ФАЙЛ СОДЕРЖИТ ИСХОДНЫЙ ТЕКСТ ПРОГРАММЫ НА ЯЗЫКЕ КОБОЛ	ACCOUNT.COB SEC.COB
.COM -ФАЙЛ СОДЕРЖИТ ТРАНЗИТНУЮ ПРОГРАММУ ОС СР/М. ЭТО РАСШИРЕНИЕ ДОЛЖНЫ ИМЕТЬ ИМЕНА ВСЕХ ФАЙЛОВ С ТРАНЗИТНЫМИ ПРОГРАММАМИ ОС СР/М С ТЕМ, ЧТОБЫ ИХ РАСПОЗНАТЬ КАК КОМАНДНЫЕ.	STAT.COM PIP.COM
.DAT -ФАЙЛ СОДЕРЖИТ ДАННЫЕ ОПРЕДЕЛЕННОГО ТИПА (ОБЫЧНО СИМВОЛЫ В КОДЕ ASCII)..	PROG1.DAT TEXT.DAT
.DOC -НЕОБЯЗАТЕЛЬНОЕ РАСШИРЕНИЕ ДЛЯ ОБОЗНАЧЕНИЯ ТЕКСТОВЫХ ФАЙЛОВ (ИЛИ ДОКУМЕНТОВ)	INFO.DOC HELP.DOC
.FOR -ФАЙЛ СОДЕРЖИТ ИСХОДНЫЙ ТЕКСТ ПРОГРАММЫ НА ЯЗЫКЕ ФОРТРАН	DISPLAY.FOR HEATH.FOR
.HEX -ФАЙЛ СОЗДАН В РЕЗУЛЬТАТЕ ВЫПОЛНЕНИЯ КОМАНДЫ ASM (СОДЕРЖИТ ШЕСНАДЦАТИРИЧИЙ МАШИННЫЙ КОД ПРОГРАММЫ. НЕОБХОДИМО ПРИ ОБРАЩЕНИИ К КОМАНДЕ LOAD С ЦЕЛЬЮ СОЗДАНИЯ КОМАНД. ("СОМ")ФАЙЛА.	DUMP.HEX PRINTER.HEX
.INT -ФАЙЛ СОДЕРЖИТ ПРОМЕЖУТОЧНЫЙ ОБ'ЕКТНЫЙ КОД ПРОГРАММЫ НА ЯЗЫКЕ БЕЙСИК. НЕОБХОДИМО ДЛЯ БОЛЬШИНСТВА ТРАНСЛЯТОРОВ С ЯЗЫКА БЕЙСИК, РАБОТАЮЩИХ В РЕЖИМЕ КОМПИЛЯЦИИ	ACCOUNT.INT GAME1.INT
.LIB -ФАЙЛ ЯВЛЯЕТСЯ БИБЛИОТЕЧНЫМ И СОДЕРЖИТ ТЕКСТ. ПРИ РАБОТЕ С ПРОГРАММОЙ EV - ДЛЯ ОПРЕДЕЛЕНИЯ МЕСТОНАХОЖДЕНИЯ ЗАМЕЩЕНИЯ ИЛИ ДОБАВЛЕНИЯ ТЕКСТА В ДРУГОМ ТЕКСТОВОМ ФАЙЛЕ, ИМЯ ФАЙЛА ДОЛЖНО ИМЕТЬ РАСШИРЕНИЕ ".LIB"	BUF.LIB TEXT1.LIB
.LST -ФАЙЛ СОДЕРЖИТ ИСХОДНЫЙ ТЕКСТ ПРОГРАММЫ НА ЯЗЫКЕ БЕЙСИК И СООБЩЕНИЯ ОБ ОШИБКАХ, ЧТО ОБЛЕГЧАЕТ ОТЛАДКУ	SPACE.LST ERRS.LST
.MAC -ФАЙЛ СОДЕРЖИТ ИСХОДНЫЙ ТЕКСТ ПРОГРАММЫ НА АССЕМБЛЕРЕ	PRINTER.MAC
.MSG -ИСПОЛЬЗУЕТСЯ КАК И ".DOC"	ERR.MSG
.OVR -ФАЙЛ СОДЕРЖИТ ПРОГРАММУ ОВЕРЛЕЙНОЙ СТРУКТУРЫ	MSG1.OVR
.PAS -ФАЙЛ СОДЕРЖИТ ИСХОДНЫЙ ТЕКСТ ПРОГРОГРАММЫ НА ЯЗЫКЕ ПАСКАЛЬ. НЕОБХОДИМО ДЛЯ НЕКОТОРЫХ КОМПИЛЯТОРОВ С ЯЗЫКА ПАСКАЛЬ	LEDVERS.PAS SCREEN.PAS

.PRN -ФАЙЛ СОДЕРЖИТ ТЕКСТ ПОДГОТОВЛЕННЫЙ К ВЫВОДУ НА ПЕЧАТЬ И СОЗДАН ТЕКСТОВЫМИ ФОРМАТЕРАМИ, ТРАНСЛЯТОРАМИ С ЯЗЫКОВ ВЫСОКОГО УРОВНЯ, АССЕМБЛЕРАМИ И МАКРОАССЕМБЛЕРАМИ	LOAD.PRN
.REL -ФАЙЛ СОДЕРЖИТ ПЕРЕМЕЩАЕМЫЙ ОБ'ЕКТНЫЙ КОД И СОЗДАН АССЕМБЛЕРАМИ И ДРУГИМИ ТРАНСЛЯТОРАМИ С ЯЗЫКОВ ПРОГРАММИРОВАНИЯ ВЫСОКОГО УРОВНЯ	CIRCLE.REL
.SUB -ФАЙЛ СОДЕРЖИТ КОМАНДЫ ОС СР/М, ПОДГОТОВЛЕННЫЕ ДЛЯ АВТОМАТИЧЕСКОГО ЗАПУСКА С ПОМОЩЬЮ КОМАНДЫ SUBMIT	INIT.SUB
.SYM -ТЕКСТОВЫЙ ФАЙЛ СОДЕРЖИТ ТАБЛИЦУ СИМВОЛОВ, СОЗДАН НЕКОТОРЫМИ ТИПАМИ МАКРОАССЕМБЛЕРОВ	BOOT.SYM
.TEX -ФАЙЛ С ИСХОДНЫМ ТЕКСТОМ	MANUAL.TEX
.TXT -ИСПОЛЬЗУЕТСЯ КАК И ".DOC"	LETTER.TXT
.\$\$\$ -УКАЗЫВАЕТ ВРЕМЕННЫЙ ФАЙЛ, СОЗДАННЫЙ ПРОГРАММАМИ РІР ED И ДР.	MANUAL.\$\$\$
.G -НЕОБЯЗАТЕЛЬНОЕ РАСШИРЕНИЕ ФАЙЛОВ С ESC-ПОСЛЕДОВАТЕЛЬНОСТЯМИ, ИСПОЛЬЗУЮТСЯ С КОМАНДОЙ ТУРЕ, (ДЛЯ ПЭВМ "ОКЕАН")	RS240.G
.HLP -ФАЙЛ С ТЕКСТОВОЙ ИНФОРМАЦИЕЙ, КАК ПРАВИЛО, СПРАВОЧНОЙ	VBOS.HLP
.BIN -ФАЙЛ С ДВОИЧНОЙ ИНФОРМАЦИЕЙ, ПРИМЕНЯЕТСЯ ВСПОМОГАТЕЛЬНЫМИ ПРОГРАММАМИ, НАПРИМЕР ПРОГРАММАТОРОМ, (ДЛЯ ПЭВМ "ОКЕАН")	OKTEST.BIN
.ADA -ИСХОДНЫЙ ТЕКСТ ПРОГРАММЫ НА ЯЗЫКЕ АДА	SORT.ADA
.C -ИСХОДНЫЙ ТЕКСТ ПРОГРАММЫ НА ЯЗЫКЕ СИ	KK.C
.CRL -БИБЛИОТЕКА ДЛЯ ТРАНСЛЯТОРА СИ	DEFF.CRL
.CCC -ВСПОМОГАТЕЛЬНЫЙ ФАЙЛ ДЛЯ ТРАНСЛЯТОРА СИ	C.CCC

***** THE END *****

B A S I C M I C R O S O F T

ЗАГРУЗКА: BASIC..***** МЕМТОР(АДРЕС)
*****: ИМЯ ФАЙЛА
МЕМТОР: ВЕРХНЯЯ ГРАНИНА ИСПОЛЬЗУЕМОЙ ОБЛАСТИ

ДАННЫЕ: A\$ - СТРОЧНАЯ ПЕРЕМЕННАЯ (КОЛ-ВО БАЙТ = КОЛ-ВО СИМВОЛОВ)
A% - ЦЕЛЫЕ (ДВА БАЙТА)
A! - ОБЫЧНАЯ ТОЧНОСТЬ (4 БАЙТА), ИЛИ ПО УМОЛЧАНИЮ (*****E**)
A# - ДВОЙНАЯ ТОЧНОСТЬ (8 БАЙТ) (*****D**)

КОНСТАНТЫ:Н : ШЕСТНАДЦАТИРИЧНЫЕ
.....Q : ВОСЬМЕРИЧНЫЕ

ОПЕРАЦИИ: (), +, ^ (СТЕПЕНЬ), -, *, /, \ (ЦЕЛОЧИСЛЕННОЕ ДЕЛЕНИЕ),
MOD(МОДУЛЬ)-ОСТАТОК ОТ ДЕЛЕНИЯ ..A MOD..

ОТНОШЕНИЯ: =, >, <, ><, <>, >=, =>, <=, =>

ЛОГИЧЕСКИЕ: NOT, AND, OR, XOR, IMP, EQU.

NOT - ОТРИЦАНИЕ

AND - ЛОГ. 'И'

OR - ЛОГ. 'ИЛИ'

XOR - ЛОГ. 'ИСКЛЮЧАЮЩЕЕ ИЛИ'

IMP - ИМПЛИКАЦИЯ

EQU - ЭКВИВАЛЕНТНОСТЬ

РЕДАКТОР:

- ВХОД: 1. КОМАНДНЫЙ - EDIT НОМЕР СТРОКИ
2. ПРИ ВВОДЕ ОЧЕРЕДНОЙ СТРОКИ ПРОГРАММЫ: СУ+А
3. ПРИ ВЫПОЛНЕНИИ ПРОГРАММЫ ПОСЛЕ ОБНАРУЖЕНИЯ СИНТАКСИЧЕСКОЙ ОШИБКИ

КОМАНДЫ:

- (ЦЕЛОЕ) D - СТЕРЕТЬ (ЦЕЛОЕ) СИМВОЛОВ СПРАВА ОТ КУРСОРА
L - ВЫВЕСТИ ПЕРВОНАЧАЛЬНУЮ СТРОКУ. НАЧАЛО РЕДАКТИРУЮЩЕГО РЕЖИМА
I СИМВОЛ... - ВСТАВИТЬ СИМВОЛ(Ы) ПОСЛЕ ПОСЛЕДНЕЙ НАПЕЧАТАННОЙ СТРОКИ.
ВЫХОД ИЗ РЕЖИМА ВСТАВКИ - ESC.
H СИМВОЛ... - СТЕРЕТЬ ВСЕ СИМВОЛЫ СПРАВА ОТ КУРСОРА, ВСТАВИТЬ НОВЫЕ. ОКОНЧАНИЕ - ESC
X СИМВОЛ... - ПЕЧАТЬ СТРОКИ, КУРСОР В КОНЕЦ СТРОКИ, ВСТАВКА СИМВОЛОВ
(ЦЕЛОЕ) S СИМВОЛ - ПОИСК (ЦЕЛОЕ) ВХОЖДЕНИЕ СИМВОЛА СПРАВА ОТ КУРСОРА
(ЦЕЛОЕ) K СИМВОЛ - АНАЛОГИЧНО S, НО СТИРАНИЕ ВСЕХ СИМВОЛОВ ДО ВХОЖДЕНИЯ
(ЦЕЛОЕ) C СИМВОЛ... - ИЗМЕНЕНИЕ СЛЕДУЮЩИХ (ЦЕЛОЕ) СИМВОЛОВ НА УКАЗАННЫЕ
Q - ВЫХОД ИЗ РЕДАКТОРА БЕЗ СОХРАНЕНИЯ РЕЗУЛЬТАТОВ РЕДАКТИРОВАНИЯ
A - ВОССТАНОВЛЕНИЕ ОТРЕДАКТИРОВАННОЙ СТРОКИ К ПЕРВОНАЧАЛЬНОМУ ВИДУ
E - ВЫХОД С СОХРАНЕНИЕМ
<BK> -ВЫХОД С СОХРАНЕНИЕМ, ПЕЧАТЬ ОСТАТКА РЕДАКТИРУЕМОЙ СТРОКИ

О Ш И Б К И :

1. NEXT WITHOUT FOR
2. SYNTAX
3. RETURN WITHOUT GOSUB
4. OUT OF DATA - ЧИТАЕТСЯ READ, А ВСЕ DATA ПРОЧИТАНЫ
5. ILLEGAL FUNCTION CALL - НЕСООТВЕТВИЕ ПЕРЕДАВАЕМЫХ ПАРАМЕТРОВ ПРЕДЕЛАМ:
 1. ОТРИЦАТЕЛЬНЫЙ ИНДЕКС МАССИВА
 2. ИНДЕКС МАССИВА > 32767
 3. LOG ОТ 0 ИЛИ ОТРИЦАТЕЛЬНОГО
 4. SQR ОТ ОТРИЦАТЕЛЬНОГО
 5. A^B, И `A` - ОТРИЦАТЕЛЬНОЕ A `B` - НЕ ЦЕЛОЕ
 6. USR БЕЗ СООТВЕТСТВУЮЩЕГО DEFUSR

7. НЕПРАВИЛЬНЫЕ АРГУМЕНТЫ ДЛЯ: MID\$, LEFT\$, RIGHT\$, INP, OUT, WAIT, PEEK, POKE, TAB, SPC, STRING\$, SPACE\$, INSTR ON...GOTO
6. OVERFLOW - ПРЕВЫШЕНИЕ ЗНАЧЕНИЯ ПЕРЕМЕННОЙ ДОПУСТИМЫХ ПРЕДЕЛОВ ТИПА
7. OUT OF MEMORY - БОЛЬШАЯ ПРОГРАММА: МНОГО ЦИКЛОВ, ПОДПРОГРАММ, ПЕРЕМЕННЫХ ИЛИ СЛОЖНЫХ ВЫРАЖЕНИЙ
8. UNDERFINED LINE NUMBER - ССЫЛКА НА НЕСУЩЕСТВУЮЩУЮ СТРОКУ
9. SUBSCRIPT OF RANGE - ИНДЕКС МАССИВА БОЛЬШЕ ОПИСАННОЙ РАЗМЕРНОСТИ ИЛИ НЕПРАВИЛЬНАЯ РАЗМЕРНОСТЬ МАССИВА
10. DUPLICATE DEFINITION - ПОВТОРНОЕ ОПИСАНИЕ МАССИВА
11. DIVISION BY ZERO - ДЕЛЕНИЕ НА 0
12. ILLIGAL DIRECT - НЕПРАВИЛЬНАЯ ИНСТРУКЦИЯ В КОМАНДНОМ СПОСОБЕ
13. TYPE MISMATCH - НЕПРАВИЛЬНЫЙ ТИП ДАННЫХ ДЛЯ ФУНКЦИИ
ЛЮБОЕ НЕПРАВИЛЬНОЕ СМЕШИВАНИЕ ТИПОВ ДАННЫХ
14. DOT OF STRING SPACE - НЕ ХВАТАЕТ ПАМЯТИ ДЛЯ СТРОЧНЫХ ДАННЫХ: ВОСПОЛЬЗОВАТЬСЯ CLEAR
15. STRING TOO LONG - ПРЕВЫШЕНИЕ ДЛИНЫ СТРОКИ
16. STRING FORMULA TOO COMPLEX - ОЧЕНЬ СЛОЖНАЯ СТРОКА: РАЗДЕЛИТЬ НА 2 ИЛИ БОЛЕЕ
17. CAN'T CONTINUE - НЕПРАВИЛЬНАЯ ПОПЫТКА ПРОДОЛЖИТЬ ПРОГРАММУ
18. UNDERFINED USER - ССЫЛКА НА НЕОПРЕДЕЛЕННУЮ ФУНКЦИЮ USR
19. NO RESUME - ПОДПРОГРАММА ОБРАБОТКИ ОШИБКИ БЕЗ RESUME
20. RESUME WITHOUT ERROR
21. UNPRINTABLE ERROR - ДЛЯ ДАННОЙ ОШИБКИ НЕ СУЩЕСТВУЕТ СООБЩЕНИЯ ОБ ОШИБКЕ:
ПРОВЕРИТЬ ОПЕРАТОРЫ ERROR ДЛЯ НЕОПРЕДЕЛЕННЫХ ОШИБОК
22. MISSING OPERAND - ОПЕРАТОР БЕЗ ОПЕРАНДОВ
23. LINE BUFFER OVERFLOW - ДЛИННАЯ СТРОКА, НЕ ВХОДИТ В БУФЕР СТРОКИ: РАЗБИТЬ НА 2 И БОЛЬШЕ
24. FOR WITHOUT NEXT -
50. FIELD OVERFLOW - БОЛЕЕ 128 СИМВОЛОВ БЫЛО НАЗНАЧЕНО В ОПЕРАТОРЕ FILED
51. INTERNAL ERROR - ОШИБКА ПРИ ВЫПОЛНЕНИИ БЗЙСИК-80
52. BAD FILE NUMBER - НЕ ОТКРЫТ ФАЙЛ, К КОТОРОМУ ПРОИСХОДИТ ОБРАЩЕНИЕ
53. FILE NOT FOUND - ОПЕРАТОРЫ LOAD, KILL, OPEN НЕ НАШЛИ ФАЙЛ НА ДИСКЕ
54. BAD FILE MODE - 1: ТИП ФАЙЛА НЕ 'I', 'O', 'R'
2: ОБРАЩЕНИЕ С PUT ИЛИ GET К ПОСЛЕДОВАТЕЛЬНОМУ ФАЙЛУ
3: PRINT ДЛЯ ФАЙЛА ПРЯМОГО ДОСТУПА
4: ОТКРЫТИЕ ФАЙЛА ПРЯМОГО ДОСТУПА ДЛЯ ПОСЛЕДОВАТЕЛЬНОГО
ВВОДА/ВЫВОДА ИЛИ НАОБОРОТ
55. FILE ALREADY OPEN - ФАЙЛ УЖЕ ОТКРЫТ
57. DISK I/O ERROR - ОШИБКА ВВОДА/ВЫВОДА НА ДИСКЕ
58. FILE ALREADY EXISTS - ФАЙЛ УЖЕ СУЩЕСТВУЕТ
61. DISK FULL - ЗАПОЛНЕН ДИСК
62. INPUT PAST END - БЫЛ ВЫДАН ОПЕРАТОР INPUT ПОСЛЕ КОНЦА ФАЙЛА
63. BAD RECORD NUMBER - НОМЕР ЗАПИСИ В PUT ИЛИ GET БОЛЬШЕ 2048 ИЛИ МЕНЬШЕ 0
64. BAD FILE NAME - В ИМЕНИ ФАЙЛА 0 ИЛИ БОЛЕЕ 8 СИМВОЛОВ, ИЛИ ПЕРВЫЙ БАЙТ 0
66. DIRECT STATEMENT IN FILE - ВО ВРЕМЯ ЗАГРУЗКИ НАЙДЕН НЕПОСРЕДСТВЕННЫЙ
ОПЕРАТОР - ЗАГРУЗКА ПРЕКРАЩАЕТСЯ
67. TOO MANY FILES - ПОПЫТКА ОТКРЫТЬ БОЛЕЕ 255 ФАЙЛОВ

О П Е Р А Т О Р Ы О Б Р А Б О Т К И О Ш И Б О К

- ON ERROR GOTO - ВСТРЕЧАЕТСЯ ДО ПОЯВЛЕНИЯ ПЕРВОЙ ОШИБКИ, В СЛУЧАЕ ВОЗНИКНОВЕНИЯ ЛЮБОЙ ОШИБКИ ПЕРЕДАЕТ УПРАВЛЕНИЕ ПОДПРОГРАММЕ СО СТРОКИ ЗА GOTO
- RESUME - ПОСЛЕДНИЙ ОПЕРАТОР ПОДПРОГРАММЫ ОБРАБОТКИ ОШИБКИ ВОЗБНОВЛЯЕТ ВЫПОЛНЕНИЕ ПРОГРАММЫ (NEXT; НОМЕР СТРОКИ - ВОЗМОЖНЫЙ ВИД)
- ON ERROR GOTO 0 - ВЫКЛЮЧАЕТ ДЕЙСТВИЕ ЛЮБОГО РАНЕЕ ВСТРЕЧЕННОГО ON ERROR GOTO
- ERROR N - ОПЕРАТОР ИММИТАЦИИ ОШИБКИ С КОДОМ N
- TRON - ВКЛЮЧЕНИЕ РЕЖИМА ТРАССИРОВКИ
- TROFF - ВЫКЛЮЧЕНИЕ РЕЖИМА ТРАССИРОВКИ

ДИРЕКТИВЫ И ОПЕРАТОРЫ :

1. ATTRIB - НЕТ
2. AUTO - AUTO N1 N2 - N1 - ПЕРВАЯ СТРОКА, N2 - ПРИРАЩЕНИЕ; * - УЖЕ ЕСТЬ СТРОКА, СОХРАНЕНИЕ ПОСЛЕ * - СУ+С, СТИРАНИЕ - <ВК>
3. CLEAR - ОБНУЛЯЕТ ПЕРЕМЕННЫЕ, ФУНКЦИИ, ЗАКРЫВАЕТ ФАЙЛЫ
CLEAR ***, **** *** - ПРОСТРАНСТВО ДЛЯ СТРОК В БАЙТАХ
**** - МАКСИМАЛЬНЫЙ ВЕРХНИЙ АДРЕС (HEX)
ДЛЯ BASIC (АНАЛОГИЧНО МЕМТОР)
4. CLOSE - ЗАКРЫВАЕТ ВСЕ ФАЙЛЫ, ПРЕРЫВАЕТ СВЯЗЬ ИМЕНИ ФАЙЛА С НОМЕРОМ
CLOSE (#) I ИМЯ ФАЙЛА, (#) - НОМЕР ФАЙЛА ...
5. CONT - (ДИРЕКТИВА) ПРОДОЛЖАЕТ ВЫПОЛНЕНИЕ ПОСЛЕ STOP, СУ+С, ОШИБКИ. ПОДОЛЖЕНИЕ НЕВОЗМОЖНО ПОСЛЕ МОДИФИКАЦИИ ПРОГРАММЫ, ПЕРЕМЕННЫЕ МОГУТ БЫТЬ ИЗМЕНЕНЫ.
6. DATA - СТРОКА ДАННЫХ, РАЗДЕЛЯЕТСЯ ЗАПЯТЫМИ. СТРОКИ МОГУТ БЫТЬ В КАВЫЧКАХ И БЕЗ
7. DEF FN (X) - ОПРЕДЕЛЕНИЕ ФУНКЦИИ. ИМЯ: FN ** (АРГУМЕНТЫ)=ВЫРАЖЕНИЯ
8. DEF*** - ОПРЕДЕЛЕНИЕ ТИПА ПЕРЕМЕННЫХ
ФОРМАТ: DEFINT M (-N), K (-L) ...
*** = INT - ЦЕЛЫЕ
SEG - ОБЫЧНЫЕ
DBL - ДВОЙНАЯ ТОЧНОСТЬ
STR - СТРОКИ
9. DFUSR N - ОПРЕДЕЛЕНИЕ НАЧАЛЬНОГО АДРЕСА ПОДПРОГРАММЫ НА ДРУГОМ ЯЗЫКЕ.
N НОМЕР ПОДПРОГРАММЫ = 0...24.
ФОРМАТ: DFUSR N = **** , **** - АДРЕС В HEX, ИЛИ % - ЦЕЛЫХ ДЕСЯТИЧНЫХ
10. DELETE - (ДИРЕКТИВА) - СТИРАНИЕ: ** - НОМЕР СТРОКИ
- ** С НАЧАЛА ПО НОМЕР СТРОКИ
- ИНТЕРВАЛ СТРОК
11. DIM - ОПИСАНИЕ МАССИВА. ПЕРВЫЙ НОМЕР ПО УМОЛЧАНИЮ - 0
12. DIR - НЕТ, ЕСТЬ:
FILES - (ДИРЕКТИВА) - КАТАЛОГ ДИСКА
13. EDIT - (ДИРЕКТИВА) - ПЕРЕХОД В РЕДАКТОР. ФОРМАТ: EDIT НОМЕР СТРОКИ
14. END - КОНЕЦ (ЗАКРЫВАЕТ ФАЙЛЫ, ВЫДАЕТ 'OK')
15. ERROR
16. EXIT - НЕТ, ЕСТЬ:
SYSTEM - ВЫХОД В CP/M
17. FILED - ОПИСАНИЕ ПАМЯТИ В ОДНОМ ИЗ 6-ТИ 128-БАЙТНЫХ БУФЕРОВ ПРЯМОГО ДОСТУПА,
СВЯЗЫВАЕТ ДАННЫЕ С ИХ ПОЗИЦИЕЙ В БУФЕРЕ
ФОРМАТ: FILED # НОМЕР ФАЙЛА, ЧИСЛО СИМВОЛОВ AS, СТРОЧНАЯ ПЕРЕМЕННАЯ, ...
18. FOR - NEXT - STEP -
19. GET - ЧТЕНИЕ ЗАПИСИ ИЗ ФАЙЛА ПРЯМОГО ВВОДА/ВЫВОДА В БУФЕР
ФОРМАТ: GET # НОМЕР ФАЙЛА (, НОМЕР ЗАПИСИ)
20. GOSUB
21. GOTO
22. IF - THEN - ELSE
23. INPUT
24. KILL - (ДИРЕКТИВА) - СТИРАНИЕ ФАЙЛА ИЗ КАТАЛОГА
25. LET
26. LINE INPUT - ЧТЕНИЕ СТРОКИ ДАННЫХ И НАЗНАЧЕНИЕ УКАЗАННОЙ ПЕРЕМЕННОЙ С ТЕРМИНАЛА ИЛИ ИЗ ФАЙЛА.
ВВОД С ТЕРМИНАЛА: LINE INPUT (СТРОЧНОЕ ВЫРАЖЕНИЕ;) СТРОЧНАЯ ПЕРЕМЕННАЯ
ИЗ ФАЙЛА: LINE INPUT # НОМЕР ФАЙЛА, СТРОЧНАЯ ПЕРЕМЕННАЯ (СТРОЧНОЕ ВЫРАЖЕНИЕ)
- ПЕЧАТАЕТСЯ ПРИ ВЫПОЛНЕНИИ ПРОГРАММЫ
27. LIST - ЛИСТИНГ: ** - НОМЕР СТРОКИ
- ** - С ПЕРВОЙ СТРОКИ ПО **
**- - СО СТРОКИ ** ДО КОНЦА
- - ИНТЕРВАЛ СТРОК
28. LOAD
29. L SET, R SET - ОПЕРАТОРЫ ПРИСВОЕНИЯ ДЛЯ СТРОЧНЫХ ПЕРЕМЕННЫХ, ДЛЯ КОТОРЫХ ОПРЕДЕЛЕНА ПОЛЯ В БУФЕРЕ

ПРЯМОГО ВВОДА/ВЫВОДА ОПЕРАТОРА FILED.

ФОРМАТ: L/R SET СТРОЧНАЯ ПЕРЕМЕННАЯ = СТРОЧНОЕ ВЫРАЖЕНИЕ.

В СЛУЧАЕ ИСПОЛЬЗОВАНИЯ СО СТРОЧНЫМИ ПЕРЕМЕННЫМИ ДЛЯ КОТОРЫХ НЕ ВЫДЕЛЯЛОСЬ ПОЛЯ ОПЕРАТОРОМ FILED, СТРОЧНАЯ ПЕРЕМЕННАЯ СОХРАНЯЕТ СВОЮ ПРЕЖНЮЮ ДЛИНУ ПОСЛЕ ПЕРЕНАЗНАЧЕНИЯ ПО L SET ИЛИ R SET.

ЕСЛИ НОВОЕ СТРОЧНОЕ ВЫРАЖЕНИЕ КОРОЧЕ ПРОШЛОГО ЗНАЧЕНИЯ, ДОПОЛНИТЕЛЬНЫЕ ПОЗИЦИИ СТАНОВЯТСЯ ПРОБЕЛАМИ.

30. MERGE (ДИРЕКТИВА) - ЧИТАЕТ С ДИСКА ПРОГРАММУ И ПРИСОЕДИНЯЕТ ЕЕ К УЖЕ НАХОДЯЩЕЙСЯ В BASIC.
СТРОКИ С ОДИНАКОВЫМИ НОМЕРАМИ ТЕРЯЮТСЯ.
31. NEW - СТИРАНИЕ ПРОГРАММЫ И ПЕРЕМЕННЫХ, ЗАКРЫВАЕТ ВСЕ ФАЙЛЫ.
32. NEXT - КОНЕЦ ЦИКЛА ИЛИ ЦИКЛОВ NEXT I, J, K, L.
33. NULL (ДИРЕКТИВА) - УКАЗЫВАЕТ ЧИСЛО НУЛЕВЫХ СИМВОЛОВ, ДОБАВЛЯЕМЫХ К СТРОКЕ.
34. ON ERROR GOTO
35. ON ... GOSUB
36. ON ... GOTO
37. OPEN - ОТКРЫТЬ ФАЙЛ, ФОРМАТ :
OPEN ТИП, (#) НОМЕР ФАЙЛА, ИМЯ ФАЙЛА.
ТИП: I - ПОСЛЕДОВАТЕЛЬНЫЙ ДЛЯ ВВОДА
O - ПОСЛЕДОВАТЕЛЬНЫЙ ДЛЯ ВЫВОДА
R - ПРЯМОГО ВВОДА/ВЫВОДА
НОМЕР ФАЙЛА: 1 ... 6 (ЧТОБЫ ВЫПОЛНЯЛИСЬ ДИРЕКТИВЫ РАБОТЫ С ФАЙЛАМИ
МОЖЕТ БЫТЬ ОТКРЫТО ЛИШЬ 6 ФАЙЛОВ).
38. OPTION BASE (ДИРЕКТИВА) - УКАЗЫВАЕТ НАЧАЛЬНЫЕ ЭЛЕМЕНТЫ МАССИВОВ КАК 0 ИЛИ 1,
ФОРМАТ: OPTION BASE 0/1
39. OUT - ВЫВОД БАЙТА В ПОРТ,
ФОРМАТ: OUT НОМЕР ПОРТА, ВЫРАЖЕНИЕ
НОМЕР ПОРТА - В HEX ИЛИ OCT ИЛИ DEC
40. POKE - ЗАПИСЬ БАЙТА В ПАМЯТЬ: РОКЕ ЯЧЕЙКА, ВЫРАЖЕНИЕ
41. PRINT - ВЫВОД НА КОНСОЛЬ ИЛИ В ФАЙЛ (ПОСЛЕДОВАТЕЛЬНЫЙ). НА ТЕРМИНАЛ:
РАЗДЕЛИТЕЛИ: ',' - В ОЧЕРЕДНОЙ ПОЗИЦИИ ТАБУЛЯТОРА
';' - В СЛЕДУЮЩЕЙ ПОЗИЦИИ
В ФАЙЛ: PRINT # НОМЕР ФАЙЛА, ...
42. PRINT USING - ОПРЕДЕЛИТЕЛЬ ФОРМАТА ПРИ ВЫВОДЕ. ФОРМАТ:
PRINT USING (# НОМЕР ФАЙЛА,) СТРОКА ФОРМАТА; ВЫРАЖЕНИЯ...
СТРОКОВЫЕ ПОЛЯ: '!' - ПЕЧАТАЕТСЯ ПЕРВЫЙ СИМВОЛ ВЫРАЖЕНИЯ
N ПРОБЕЛОВ - ПЕЧАТАЮТСЯ ПЕРВЫЕ 2+N СИМВОЛОВ
\ - ПЕЧАТАЮТСЯ 2 СИМВОЛА
ЧИСЛОВЫЕ ПОЛЯ: '#' - ПОЗИЦИИ ЦИФР
'.' - ДЕСЯТИЧНАЯ ТОЧКА
'+' - В НАЧАЛЕ ИЛИ КОНЦЕ ПОЛЯ ФОРМАТА: ПЕЧАТЬ ЗНАКА ЧИСЛА
'-' - В КОНЦЕ ПОЛЯ ФОРМАТА: ПЕЧАТЬ '-' ИЛИ ПРОБЕЛ ВМЕСТО '+'
'**' - ЗАПОЛНЯЮТСЯ ЛИДИРУЮЩИЕ ПРОБЕЛЫ ЗВЕЗДОЧКАМИ
 '\$\$' - ДОБАВЛЯЮТ ЗНАК \$ СЛЕВА ОТ ЧИСЛА
 '**\$' - СОВМЕСТНОЕ ИСПОЛЬЗОВАНИЕ ** И \$\$
' ,' - СЛЕВА ОТ ДЕСЯТИЧНОЙ ТОЧКИ: ПЕЧАТЬ ЗАПЯТОЙ ЧЕРЕЗ КАЖДЫЕ 3
ЦИФРЫ СЛЕВА ОТ ДЕСЯТИЧНОЙ ТОЧКИ, СПРАВА: ПЕЧАТЬ В
УКАЗАННОЙ ПОЗИЦИИ ЗАПЯТОЙ
'^^^^' - УКАЗЫВАТ ЭКСПОНЕНТУ
'%' - ЧИСЛО БОЛЬШЕ, ЧЕМ ДОПУСКАЕТ ФОРМАТ
43. PRUN - НЕТ
44. PUT - ЗАНЕСЕНИЕ ЗАПИСИ ИЗ БУФЕРА В ФАЙЛ ПРЯМОГО ВВОДА/ВЫВОДА
ФОРМАТ: PUT НОМЕР ФАЙЛА, (НОМЕР ЗАПИСИ). ЕСЛИ НОМЕРА ЗАПИСИ НЕТ, ПОИЗВОДИТСЯ
ОЧЕРЕДНАЯ ЗАПИСЬ
45. READ
46. REM
47. RENAME - (ДИРЕКТИВА) - ПЕРЕИМЕНОВАНИЕ ФАЙЛА НА ДИСКЕ
ФОРМАТ: RENAME СТАР.ИМ. ТО НОВ.ИМ.
48. RENUM - (ДИРЕКТИВА) ИЗМЕНЕНИЕ НУМЕРАЦИИ СТРОК
ФОРМАТ: RENUM N1, N2, N3, ПО УМОЛЧАНИЮ ВСЕ N=10

N1 - НОВЫЙ НОМЕР ПЕРВОЙ СТРОКИ
N2 - СТАРЫЙ НОМЕР ПЕРВОЙ ПЕРЕНУМЕРУЕМОЙ СТРОКИ
N3 - ПРИРАЩЕНИЕ НОМЕРОВ

- 49. RESTORE - ОБНОВЛЕНИЕ УКАЗАТЕЛЯ В DATA
- 50. RESURN
- 51. RETURN
- 52. RUN - ВЫПОЛНЕНИЕ (С НОМЕРА СТРОКИ)
RUN НОМЕР СТРОКИ
- 53. SAVE - (ДИРЕКТИВА) - СОХРАНЕНИЕ НА ДИСКЕ ПРОГРАММЫ
ДОБАВЛЕНИЕ, А - СОХРАНЕНИЕ В КОДЕ К0И8
- 54. STOP - ВЫЗЫВАЕТ СООБЩЕНИЕ BREAK

Ф У Н К Ц И И :

- 1. ABS - АБСОЛЮТНОЕ ЗНАЧЕНИЕ ВЫРАЖЕНИЯ. ABS (ВЫРАЖЕНИЕ)
- 2. ASC - КОД К0И8 ПЕРВОГО СИМВОЛА СТРОКИ. ASC (СТРОЧНОЕ ВЫРАЖЕНИЕ)
- 3. ATN - АРКТАНГЕНС
- 4. CDBL - ИЗМЕНЯЕТ ТИП ВЫРАЖЕНИЯ НА ДВОЙНУЮ ТОЧНОСТЬ. CDBL (ВЫРАЖЕНИЕ)
(БОЛЕЕ ПРАВИЛЬНО ПРЕДСТАВЛЯЕТ РЕЗУЛЬТАТ ОПЕРАЦИИ:
A# = 1/3 ; ЛУЧШЕ - В# = CDBL (1/3))
- 5. CHR\$ - СТРОЧНАЯ. ПРИВОДИТ ВЫРАЖЕНИЕ К ЦЕЛОМУ 0...255, ВОЗВРАЩАЕТ
СООТВЕТСТВУЮЩИЙ СИМВОЛ К0И8. CHR\$ (ВЫРАЖЕНИЕ)
- 6. CINT - ОКРУГЛЯЕТ К ЦЕЛОМУ ЗНАЧЕНИЮ. CINT (ВЫРАЖЕНИЕ)
- 7. COS - КОСИНУС ВЫРАЖЕНИЯ В РАДИАНАХ
- 8. CSNG - ИЗМЕНЯЕТ ТИП ВЫРАЖЕНИЯ НА ОБЫЧНУЮ ТОЧНОСТЬ
- 9. CVI - ПРЕОБРАЗУЕТ 2-Х СИМВОЛЬНУЮ СТРОКУ В ЦЕЛОЕ ЧИСЛО
CVS - ПРЕОБРАЗУЕТ 4-Х СИМВОЛЬНУЮ СТРОКУ В ОБЫЧНОЙ ТОЧНОСТИ ЧИСЛО
CVD - ПРЕОБРАЗУЕТ 8-И СИМВОЛЬНУЮ СТРОКУ В ДВОЙНОЙ ТОЧНОСТИ ЧИСЛО
CV* (СТРОЧНОЕ ВЫРАЖЕНИЕ)
- 10. DSKF - ПОКАЗЫВАЕТ КОЛИЧЕСТВО СВОБОДНЫХ 128-И БАЙТНЫХ БЛОКОВ НА ДИСКЕ
DSKF (ДИСК)
- 11. EOF - ПОКАЗЫВАЕТ, ДОСТИГНУТ ЛИ КОНЕЦ ФАЙЛА ПРИ ЧТЕНИИ ПОСЛЕДОВАТЕЛЬНОГО ФАЙЛА.
ПРИ ДОСТИЖЕНИИ КОНЦА EOF = 1. EOF (НОМЕР ФАЙЛА)
- 12. ERL - ПОКАЗЫВАЕТ НОМЕР СТРОКИ ПОСЛЕДНЕЙ ОШИБКИ. ERL
- 13. ERR - ПОКАЗЫВАЕТ ТИП ПОСЛЕДНЕЙ ОШИБКИ. ERR
- 14. EXP - РЕЗУЛЬТАТ - ВОЗВЕДЕНИЕ ЧИСЛА Е В УКАЗАННУЮ В ВЫРАЖЕНИИ СТЕПЕНЬ
EXP (ВЫРАЖЕНИЕ)
- 15. FIX - УСЕКАЕТ ЧИСЛА ДО ЦЕЛОГО (ОТБРАСЫВАЕТ ДРОБНУЮ ЧАСТЬ).
FIX (ЧИСЛО)
- 16. FRE - ВОЗВРАЩАЕТ ОСТАВШЕЕСЯ КОЛ-ВО БАЙТ ПАМЯТИ, ЕСЛИ АРГУМЕНТ (X) ИЛИ
БАЙТ В ОБЛАСТИ ПАМЯТИ СТРОК, ЕСЛИ АРГУМЕНТ (X\$)
- 17. HEX\$ - ВОЗВРАЩАЕТ СТРОКУ 16-РИЧНЫХ ЦИФР, СООТВЕТСТВУЮЩИХ ЦЕЛОМУ АРГУМЕНТУ.
HEX\$ (ВЫРАЖЕНИЕ)
- 18. INP - ПРИНИМАЕТ БАЙТ С ПОРТА 0...255. INP (ВЫРАЖЕНИЕ)
- 19. INPUT\$ - ЧИТАЕТ УКАЗАННОЕ КОЛ-ВО СИМВОЛОВ ИЗ ПОСЛЕДОВАТЕЛЬНОГО ФАЙЛА
INPUT\$ (ВЫРАЖЕНИЕ, НОМЕР ФАЙЛА). ВЫРАЖЕНИЕ = ЧИСЛО СИМВОЛОВ
- 20. INSTR - I-ОЕ ВХОЖДЕНИЕ СТРОКИ2 В СТРОКЕ1. АРГУМЕНТ I - НЕ ОБЯЗАТЕЛЬНЫЙ.
INSTR (I, СТРОКА1, СТРОКА2), ВОЗВРАЩАЕТ ЦЕЛОЕ = ПОЗИЦИИ ВХОЖДЕНИЯ
- 21. INT - ВОЗВРАЩАЕТ НАИБОЛЬШЕЕ ЦЕЛОЕ, МЕНЬШЕЕ ИЛИ РАВНОЕ АРГУМЕНТУ. СОХРАНЯЕТ ЗНАК.
INT (ВЫРАЖЕНИЕ)
- 22. LEFT\$ - ВОЗВРАЩАЕТ I НАИБОЛЕЕ ЛЕВЫХ СИМВОЛОВ В СТРОКЕ X\$
LEFT\$ (X\$, I), I - ЦЕЛОЕ 0...255
- 23. LEN - ВОЗВРАЩАЕТ КОЛ-ВО СИМВОЛОВ В СТРОКЕ X\$. LEN (X\$)
- 24. LOG - НАТУРАЛЬНЫЙ ЛОГАРИФМ
- 25. LOC - ВОЗВРАЩАЕТ НОМЕР ПОСЛЕДНЕЙ ПРОЧИТАННОЙ ИЛИ ЗАПИСАННОЙ ЗАПИСИ В ФАЙЛЕ ПРЯМОГО ВВОДА/ВЫВОДА,
ИЛИ КОЛ-ВО ПРОЧИТАННЫХ ИЛИ ЗАПИСАННЫХ СЕКТОРОВ В ПОСЛЕДОВАТЕЛЬНОМ ФАЙЛЕ.
LOC (НОМЕР ФАЙЛА)

26. LOF - ВОЗВРАЩАЕТ ЧИСЛО ЗАПИСЕЙ В ФАЙЛЕ ПРЯМОГО ДОСТУПА ИЛИ ЧИСЛО СЕКТОРОВ С ДАННЫМИ В ПОСЛЕДОВАТЕЛЬНОМ ФАЙЛЕ. LOF (НОМЕР ФАЙЛА)
27. MID - ВОЗВРАЩАЕТ НАИБОЛЕЕ ПРАВЫЕ, НАЧИНАЯ С I-ОЙ ПОЗИЦИИ, J СИМВОЛОВ В СТРОКЕ X\$. I И J В ПРЕДЕЛАХ 0..255. ЕСЛИ I ОПУЩЕНО, ВОЗВРАЩАЮТСЯ J СИМВОЛОВ НАЧИНАЯ С ПЕРВОГО. MID (X\$, I, J)
ЕСЛИ J ОПУЩЕНО ИЛИ БОЛЬШЕ КОЛИЧЕСТВА СИМВОЛОВ В СТРОКЕ, ВОЗВРАЩАЕТСЯ ОСТАТОК СТРОКИ.
МОЖЕТ БЫТЬ ЛЕВОЙ ЧАСТЬЮ ОПЕРАТОРА ПРИСВАИВАНИЯ: MID (X\$, I, J) = Y\$
ЗАМЕНЯЕТ ОПИСЫВАЕМЫЕ В MID СИМВОЛЫ СТРОКИ X\$ НА ПЕРВЫЕ J ИЗ Y\$
28. MKI\$ - ПРЕОБРАЗУЕТ ЦЕЛЫЕ ЧИСЛА В ДВУХБАЙТОВЫЕ СТРОКИ
MKS\$ - ПРЕОБРАЗУЕТ ОБЫЧНОЙ ТОЧНОСТИ ЧИСЛА В 4-Х БАЙТОВЫЕ СТРОКИ
MKD\$ - ПРЕОБРАЗУЕТ ДВОЙНОЙ ТОЧНОСТИ ЧИСЛА В 8-И БАЙТОВЫЕ СТРОКИ
MK*\$ (ЗНАЧЕНИЕ СООТВЕТСТВУЮЩЕГО ТИПА)
29. OST\$ - ВОЗВРАЩАЕТ СТРОКУ 8-РИЧНЫХ ЦИФР, СООТВЕТСТВУЮЩУЮ ЦЕЛОМУ АРГУМЕНТУ.
OST\$ (ВЫРАЖЕНИЕ)
30. PEEK - ЧИТАЕТ БАЙТ ИЗ АДРЕСА. PEEK (ВЫРАЖЕНИЕ)
31. POS - ВОЗВРАЩАЕТ НОМЕР ПОЗИЦИИ КУРСОРА ПОСЛЕ ПОСЛЕДНЕГО PRINT.
POS (ЦЕЛОЕ). (ЦЕЛОЕ) - ФИКТИВНЫЙ АРГУМЕНТ
32. RANDOMIZE - ВВОД СЛУЧАЙНОГО ЧИСЛА ДЛЯ ГЕНЕРАЦИИ ПОСЛЕДОВАТЕЛЬНОСТИ RND.
МОЖЕТ СОДЕРЖАТЬ ВЫРАЖЕНИЕ, ТОГДА НЕТ ЗАПРОСА НА ВВОД
33. RIGHTS - ВОЗВРАЩАЕТ I НАИБОЛЕЕ ПРАВЫХ СИМВОЛОВ СТРОКИ X\$. RIGHT (X\$, I)
34. RND - СЛУЧАЙНОЕ ЧИСЛО ОБЫЧНОЙ ТОЧНОСТИ МЕЖДУ 0 И 1.
35. SGN - ЗНАК ВЫРАЖЕНИЯ: >0 = 1; <0 = -1; 0 = 0. SGN (ВЫРАЖЕНИЕ)
36. SIN - СИНУС АРГУМЕНТА В РАДИАНАХ. SIN (ВЫРАЖЕНИЕ)
37. SPACE\$ - ВОЗВРАЩАЕТ СТРОКУ ПРОБЕЛОВ, ЧИСЛО ИХ РАВНО ВЫРАЖЕНИЮ.
SPACE\$ (ЦЕЛОЕ ВЫРАЖЕНИЕ)
38. SPC - АНАЛОГИЧНО SPACE\$, НО ТОЛЬКО ДЛЯ ОПЕРАТОРА PRINT И АРГУМЕНТ ЦЕЛАЯ КОНСТАНТА
SPC (I)
39. SQR - КОРЕНЬ
40. STRING\$ - ВОЗВРАЩАЕТ СТРОКУ В КОТОРОЙ УКАЗАННЫЙ СИМВОЛ ПОВТОРЯЕТСЯ N РАЗ.
STRING\$ (N, I) ИЛИ STRING\$ (N, X\$)
I - ЦЕЛОЕ - КОД КОИ8 СИМВОЛА
X\$ - ЛИТЕРА ИЛИ СТРОКА (ПОВТОРЯЕТСЯ ПЕРВЫЙ СИМВОЛ СТРОКИ)
41. STR\$ - ВОЗВРАЩАЕТ СТРОКУ ДЕСЯТИЧНЫХ ЗНАКОВ, ПРЕДСТАВЛЯЮЩУЮ ЦЕЛОЕ ВЫРАЖЕНИЕ.
STR\$ (ЦЕЛОЕ ВЫРАЖЕНИЕ)
42. TAB - ИСПОЛЬЗУЕТСЯ С ОПЕРАТОРОМ PRINT. ПЕРЕВОДИТ ПОЗИЦИЮ ДЛЯ ПЕЧАТИ В УКАЗАННЫЙ СТОЛБЕЦ.
TAB (I)
43. TAN - ТАНГЕНС АРГУМЕНТА В РАДИАНАХ.
44. USR N - ОБРАЩЕНИЕ К ОПРЕДЕЛЕННЫМ ПОЛЬЗОВАТЕЛЕМ ПОДПРОГРАММАМ, N - ЦЕЛОЕ 0 ... 24,
ПО УМОЛЧАНИЮ - 0. СИМВОЛ ТИПА ОПРЕДЕЛЯЕТ ТИП РЕЗУЛЬТАТА.
ЕСЛИ НЕТ ОБОЗНАЧЕНИЯ ТИПА - РЕЗУЛЬТАТ ВОЗВРАЩАЕТСЯ В ПАРЕ HL.
USR[!/#/%/\$] N (СПИСОК ПАРАМЕТРОВ - ЦЕЛОЕ ИЛИ ФУНКЦИЯ VARPTR)
45. VAL - ВОЗВРАЩАЕТ ЧИСЛОВОЕ ЗНАЧЕНИЕ (ЦЕЛОЕ) СТРОКИ СИМВОЛОВ ЦИФР.
VAL (СТРОКА)
46. VARPTR - ВОЗВРАЩАЕТ АДРЕС ПЕРЕМЕННОЙ В ПАМЯТИ ИЛИ БУФЕРЕ ВВОДА/ВЫВОДА.
VARPTR (ПЕРЕМЕННАЯ ИЛИ # НОМЕР ФАЙЛА).

ПОСЛЕДОВАТЕЛЬНЫЙ ВВОД / ВЫВОД

ФАЙЛЫ, СОДЕРЖАЩИЕ СТРОКИ И ЧИСЛА ДЛИНОЙ ДО 255 СИМВОЛОВ

ОПЕРАЦИИ: 1. ОТКРЫТЬ ФАЙЛ ДЛЯ ВВОДА ИЛИ ВЫВОДА

2. ОПЕРАТОРЫ ВВОДА ИЛИ ВЫВОДА

3. ЗАКРЫТЬ ФАЙЛ

ЛЮБАЯ ОПЕРАЦИЯ ВВОДА/ВЫВОДА, СЛЕДУЮЩАЯ НЕПОСРЕДСТВЕННО ЗА ОПЕН НАЧИНАЕТ РАБОТУ С НАЧАЛА ФАЙЛА. ОДНОВРЕМЕННО МОЖЕТ БЫТЬ ОТКРЫТО 6 ФАЙЛОВ. ПРИ ОТКРЫТИИ ФАЙЛА УКАЗЫВАЕТСЯ ОПЕРАЦИЯ : ВВОД ИЛИ ВЫВОД И СТАВИТСЯ В СООТВЕТСТВИЕ С ИМЕНЕМ ФАЙЛА - НОМЕР <# N>. ЕСЛИ НЕ СУЩЕСТВУЕТ ФАЙЛА НА ДИСКЕ, ОБОЗНАЧЕННОГО В ОПЕРАТОРЕ ОПЕН, ФАЙЛ СОЗДАЕТСЯ. В ОПЕРАТОРЕ ОПЕН ЛЮБОЙ ИЗ АРГУМЕНТОВ МОЖЕТ ПРЕДСТАВЛЯТЬСЯ ПЕРЕМЕННОЙ.

ЗАПИСЬ В ФАЙЛ: ОПЕРАТОРОМ PRINT ЗА КОТОРЫМ СЛЕДУЕТ <# N, >.

ЕСЛИ ОТКРЫТЬ СУЩЕСТВУЮЩИЙ ФАЙЛ ДЛЯ ВЫВОДА ЕГО СОДЕРЖИМОЕ ТЕРЯЕТСЯ. МОЖНО ИСПОЛЬЗОВАТЬ ОПЕРАТОР:

PRINT USING # N, '###, #' . . .

ЧТЕНИЕ ИЗ ФАЙЛА: ОПЕРАТОРАМИ < INPUT # N, > И < LINE INPUT # N, >. В ФАЙЛЕ ЧИСЛОВЫЕ И СТРОЧНЫЕ ВЕЛИЧИНЫ МОГУТ ОТДЕЛЯТЬСЯ ПРОБЕЛАМИ, ЗАПЯТЫМИ, ВК, ПС. СТРОЧНЫЕ ВЕЛИЧИНЫ МОГУТ БЫТЬ В КАВЫЧКАХ И БЕЗ НИХ. ЧИСЛОВЫЕ И СТРОЧНЫЕ ДАННЫЕ ПОСЛЕДОВАТЕЛЬНО НАЗНАЧАЮТСЯ ПЕРЕМЕННЫМИ В СПИСКЕ ОПЕРАТОРА INPUT. ПОВТОРНОЕ ИСПОЛЬЗОВАНИЕ INPUT ПРОДОЛЖАЕТ ЧИТАТЬ ДАННЫЕ ИЗ ФАЙЛА.

LINE INPUT ИГНОРИРУЕТ КАВЫЧКИ, ЗАПЯТЫЕ. РАЗДЕЛИТЕЛЯМИ МОГУТ БЫТЬ ТОЛЬКО ВК-ПС. ЕСЛИ INPUT НЕ СЧИТАЛ ВСЕ ДАННЫЕ ДО КОНЦА СТРОКИ (ВК-ПС), ТО СЛЕДУЮЩИЙ LINE INPUT СНАЧАЛА СЧИТАЕТ ОСТАТОК СТРОКИ.

ФУНКЦИЯ INPUT\$ СЧИТЫВАЕТ УКАЗАННОЕ ЧИСЛО СИМВОЛОВ ИЗ ПОСЛЕДОВАТЕЛЬНОГО ФАЙЛА. ПОПЫТКА СЧИТАТЬ ДАННЫЕ ПОСЛЕ УСТАНОВКИ ФУНКЦИИ EOF В (+1) ВЫЗЫВАЕТ ОШИБКУ. МОЖНО ПРОВЕРЯТЬ ДОСТИЖЕНИЕ КОНЦА ФАЙЛА:

IF EOF (1) THEN ***

ОПЕРАТОР CLOSE # N - ЗАКРЫВАЕТ ФАЙЛ.

ПРЯМОЙ ВВОД / ВЫВОД

ВВОД/ВЫВОД ВЕДЕТСЯ ЧЕРЕЗ БУФЕР ПРЯМОГО ВВОДА/ВЫВОДА НЕПОСРЕДСТВЕННО 128-МИ БАЙТНЫМИ ЗАПИСЯМИ. ПОСЛЕ ОТКРЫТИЯ ФАЙЛА МОЖНО ПРОИЗВОЛЬНО ЧИТАТЬ ИЛИ ПИСАТЬ ЛЮБУЮ ИЗ СУЩЕСТВУЮЩИХ ЗАПИСЕЙ. ЗАПИСИ В ФАЙЛЕ НУМЕРУЮТСЯ. СПЕЦИАЛЬНЫМ ОПЕРАТОРОМ FILED ЗАПИСЬ В 128 БАЙТ В БУФЕРЕ ОПРЕДЕЛЕННЫМ ОБРАЗОМ НАЗНАЧАЕТСЯ СТРОКОВЫМ ПЕРЕМЕННЫМ. ВСЕ ОПЕРАЦИИ ПРОИЗВОДЯТСЯ ТОЛЬКО СО СТРОЧНЫМИ ДАННЫМИ, ДЛЯ ЧЕГО СУЩЕСТВУЮТ ФУНКЦИИ ПРЕОБРАЗОВАНИЯ ЧИСЛОВЫХ ВЕЛИЧИН В СТРОКИ И НАОБОРОТ.

ОДНОВРЕМЕННО МОЖНО ОТКРЫТЬ ТОЛЬКО 6 ФАЙЛОВ. ОПЕРАТОРЫ ЧТЕНИЯ И ЗАПИСИ PUT И GET ПЕРЕНОСЯТ ОПРЕДЕЛЕННУЮ ЗАПИСЬ ФАЙЛА В БУФЕР ИЛИ НАОБОРОТ. СПЕЦИАЛЬНЫЕ СИМВОЛЫ (, 'ВК' 'ПС') НЕ ИМЕЮТ ОСОБОГО ЗНАЧЕНИЯ ПРИ ИСПОЛЬЗОВАНИИ ДЛЯ ЗАПИСИ/ЧТЕНИЯ ДАННЫХ МЕНЬШИХ, ЧЕМ 128 БАЙТ, ОСТАВШИЕСЯ ПОЗИЦИИ В БУФЕРЕ ЗАПОЛНЯЮТСЯ ПРОБЕЛАМИ.

ПОРЯДОК ОПЕРАЦИЙ: 1. ОТКРЫТЬ ФАЙЛ

2. ПРИ ЧТЕНИИ ЗАПИСИ ИЗ ФАЙЛА МОЖНО ПРОИЗВОЛЬНО ИСПОЛЬЗОВАТЬ ОПЕРАТОРЫ FILED И GET

3. ПРИ ЗАПИСИ В ФАЙЛ: СНАЧАЛА FILED, ЗАТЕМ PUT

4. ЗАКРЫТЬ ФАЙЛ.

ФОРМАТ ЗАПИСИ (СОГЛАШЕНИЕ): ПЕРВЫЕ 20 БАЙТ - ИМЕНА, ЗАТЕМ 9 БАЙТ - СТРАХОВОЙ НОМЕР, ЗАТЕМ - ИНФО (ИЛИ ПРОБЕЛЫ).

ОПЕРАТОР ОПЕН С ТИПОМ 'R' ОТКРЫВАЕТ ФАЙЛ ДЛЯ ПРЯМОГО ВВОДА/ВЫВОДА И СТАВИТ ЕМУ В СООТВЕТСТВИЕ НОМЕР. ОПЕРАТОРЫ PUT И GET СОДЕРЖАТ НОМЕР ФАЙЛА И, ВОЗМОЖНО, НОМЕР ЗАПИСИ. ЕСЛИ НОМЕР ЗАПИСИ ОПУЩЕН, ИСПОЛЬЗУЕТСЯ ОЧЕРЕДНАЯ ПОСЛЕ ПРОШЛОГО ИСПОЛЬЗОВАНИЯ ОПЕРАТОРОВ.

ПРИ ПРЯМОМ И ОБРАТНОМ ПРЕОБРАЗОВАНИИ ЧИСЛОВЫХ ДАННЫХ В СТРОЧНЫЕ И ПРИ НАЗНАЧЕНИИ СТРОЧНЫМ ПЕРЕМЕННЫМ ПОЛЯ В ОПЕРАТОРЕ FILED, НЕОБХОДИМО СЛЕДИТЬ ЗА СООТВЕТСТВИЕМ ТИПА ЧИСЛОВЫХ ДАННЫХ И КОЛИЧЕСТВОМ БАЙТ В СТРОЧНЫХ ПЕРЕМЕННЫХ: ЦЕЛОЕ - 2 БАЙТА

ОБЫЧНАЯ ТОЧНОСТЬ - 4 БАЙТА

ДВОЙНАЯ ТОЧНОСТЬ - 8 БАЙТ.

ДЛЯ ПРИСВОЕНИЯ ОПРЕДЕЛЕННЫХ ЗНАЧЕНИЙ ПЕРЕМЕННЫМ, РАЗМЕЩЕННЫМ В ПОЛЯХ БУФЕРА В ОПЕРАТОРЕ FILED, СУЩЕСТВУЮТ ОПЕРАТОРЫ ПРИСВОЕНИЯ L SET, R SET. ОПЕРАТОРОМ L SET ПРИСВАИВАЕТСЯ ТО КОЛ-ВО НАИБОЛЕЕ ЛЕВЫХ СИМВОЛОВ СТРОЧНОЙ ПЕРЕМЕННОЙ СПРАВА ОТ ЗНАКА РАВЕНСТВА, КОТОРОЕ СООТВЕТСТВУЕТ КОЛИЧЕСТВУ БАЙТ В ПОЛЕ ПЕРЕМЕННОЙ СЛЕВА ОТ ЗНАКА РАВЕНСТВА - ЭТО ЛЕВОСТОРОННЕЕ ВЫРАВНИВАНИЕ. ОПЕРАТОР RSET ПРИСВАИВАЕТ С ПРАВОВОСТОРОННИМ ВЫРАВНИВАНИЕМ. ЕСЛИ ПРИСВАИВАЕМАЯ ПЕРЕМЕННАЯ КОРОЧЕ ПОЛЯ, ОСТАТОК ДОПОЛНЯЕТСЯ ПРОБЕЛАМИ.

ДЛЯ ОПРЕДЕЛЕНИЯ ТЕКУЩЕЙ ЗАПИСИ И ВСЕГО КОЛ-ВА ЗАПИСЕЙ ИСПОЛЬЗУЮТСЯ ФУНКЦИИ LOF И LOC.

**ДОРАБОТКА ПЭВМ "ОКЕАН-240"
РАСШИРЕНИЕ ЭЛЕКТРОННОГО ДИСКА ДО 192 КБАЙТ**

1. ОТКЛЮЧИТЬ ВЫВОДЫ (КРОМЕ ПИТАНИЯ) МС D32 ОТ ПРЕЖНИХ ЦЕПЕЙ.
2. D32: ВЫВОДЫ 4,5,6,10,12,13 ПОДКЛЮЧИТЬ К 8.D16 (АНАЛОГИЧНО МС D33).
3. D32: ВЫВОДЫ 3,11 ПОДКЛЮЧИТЬ К ЦЕНИ '+В' (ВЫВОДЫ 3 ИЛИ 11 МС D33).
4. D32: ВЫВОД 2 ПОДКЛЮЧИТЬ К ЦЕПИ "А8" (ВЫВОД 2 МС D33).
5. D32: ВЫВОД 14 ПОДКЛЮЧИТЬ К ЦЕПИ "SR" (ВЫВОД 14 МС D33).
6. СВЕРХУ МС D22 УСТАНОВИТЬ ДОПОЛНИТЕЛЬНУЮ МС K555ЛАЗ.
7. ДОП. МС: ВЫВОД 13 ПОДКЛЮЧИТЬ К ЦЕПИ "А17" (ВЫВОД 20 МС D17).
8. ДОП. МС: ВЫВОД 12 ПОДКЛЮЧИТЬ К ЦЕПИ "S1" (ВЫВОД 1 МС D13).
9. ДОП. МС: СОЕДИНИТЬ ВЫВОДЫ 10 И 11, ПОДКЛЮЧИТЬ ИХ К ВЫВОДАМ 1 И 15 МС D32.
10. ДОП. МС: ВЫВОД 9 ПОДКЛЮЧИТЬ К ЦЕПИ "+В".
11. ДОП. МС: ВЫВОД 8 ПОДКЛЮЧИТЬ К ВЫВОДАМ 1 И 15 МС D33.
12. ВЫВОДЫ 7 И 9 МС D32, 7 И 9 МС D33: КАЖДЫЙ ИЗ НИХ ПОДКЛЮЧИТЬ ЧЕРЕЗ РЕЗИСТОРЫ 3...5 КОМ К ЦЕПИ "+5В" (ОБЯЗАТЕЛЬНО).
13. УСТАНОВИТЬ ДОПОЛНИТЕЛЬНЫЕ МС ПАМЯТИ (565РУ5) СВЕРХУ ИМЕЮЩИХСЯ, ОТГИБАЯ ВЫВОДЫ 15 (CAS).
14. ВЫВОДЫ 15 ДОП. МС ПАМЯТИ, УСТАНОВЛЕННЫХ СВЕРХУ D48...D55 ОБЪЕДИНИТЬ И ПОДКЛЮЧИТЬ К ВЫВОДУ 7 МС D32 (НОВАЯ ЦЕПЬ CAS3).
15. ВЫВОДЫ 15 ДОП. МС ПАМЯТИ, УСТАНОВЛЕННЫХ СВЕРХУ D56...D63 ОБЪЕДИНИТЬ И ПОДКЛЮЧИТЬ К ВЫВОДУ 9 МС D32 (НОВАЯ ЦЕПЬ CAS4).
16. ПРОИЗВЕСТИ ИЗМЕНЕНИЯ В СИСТЕМНОМ ПЗУ. ДЛЯ REL 5:

F62A	И	F654	:	E6	D796	:	BF			
F62B	И	F655	:	07	DB30	:	BF			
F62C	И	F656	:	C6						
F62D	И	F657	:	02						
---				ПЗУ1 (MON)	----				ПЗУ2 (CP/M)	----

ДЛЯ REL 7: F62A СООТВЕТСТВУЕТ F9DB

F654 --- FA05

ДЛЯ REL 8: F62A --- FB46

F654 --- FB70

ДЛЯ REL 7 И REL 8: D796 СООТВЕТСТВУЕТ D9B4,

А ВМЕСТО DB30 – ПО АДРЕСУ D645 СЛЕДУЕТ ЗАПИСАТЬ C0.

17. КРОМЕ ТОГО, ПОСЛЕ ТАКОЙ ПЕРЕДЕЛКИ ПРИ ЧТЕНИИ ФАЙЛОВ С МАГНИТОФОНА ДИСК БУДЕТ ЗАПОЛНЯТЬСЯ НЕ ПОЛНОСТЬЮ (ПРИМЕРНО 110 КБАЙТ), А ДАЛЬНЕЙШЕЕ ЧТЕНИЕ БУДЕТ ПРИВОДИТЬ К ОШИБКЕ (SQ 2). ДЛЯ УСТРАНЕНИЯ ЭТОГО СЛЕДУЕТ ИЗМЕНИТЬ ЯЧЕЙКУ СИСТЕМНОГО ПЗУ:

DE9D : 0B (ДЛЯ REL 5)

DFDE : 0B (ДЛЯ REL 7 И REL 8)

ПОСЛЕ ЭТОГО ВНОВЬ ЗАПИСАННЫЕ ФАЙЛЫ БУДУТ ЧИТАТЬСЯ ДО ПОЛНОГО ЗАПОЛНЕНИЯ ДИСКА, СТАРЫЕ ЗАПИСИ ЧИТАЮТСЯ ПОПРЕЖНЕМУ, ИХ ЖЕЛАТЕЛЬНО ПЕРЕПИСАТЬ ЗАНОВО.

18. КРОМЕ ТОГО, ПРАКТИКА ПОКАЗАЛА, ЧТО ИЗ-ЗА ПЕРЕГРУЗКИ ШИНЫ ДАННЫХ ПРОЦЕССОРА ПОВЫШАЕТСЯ ВЕРОЯТНОСТЬ СБОЕВ ОЗУ СИСТЕМЫ. УСТРАНЯЕТСЯ ЭТОТ НЕДОСТАТОК ВВЕДЕНИЕМ БУФЕРА ШИНЫ 555АП5 МЕЖДУ ВЫВОДАМИ ПРОЦЕССОРА И СЛЕДУЮЩИМИ ПОТРЕБИТЕЛЯМИ, РАБОТАЮЩИМИ ТОЛЬКО НА ВВОД:

- СИСТЕМНЫЙ ПОРТ D17

- СИСТЕМНЫЙ ПОРТ D67

- ВХОДЫ ДВУХ СТАРЫХ И ДВУХ ДОПОЛНИТЕЛЬНЫХ ЛИНЕЕК ПАМЯТИ.

ДАННЫЙ УЧАСТОК ШИНЫ ПЕРЕРЕЗАЕТСЯ НА ПЕЧАТНОЙ ПЛАТЕ МЕЖДУ ТОРЦАМИ МС 580ВМ80 И 580ВВ55 (D17).

БУФЕР 555АП5 УСТАНОВЛИВАЕТСЯ СВЕРХУ ПОРТА D17 И ПРОВОДАМИ РАСПАИВАЕТСЯ ШИНА:

С ПРОЦЕССОРА НА ВХОДЫ БУФЕРА, НА ВХОДЫ ШИНЫ ДАННЫХ ПОРТА – С ВЫХОДОВ БУФЕРА.

ВЫВОД 20 (+5В) – К ШИНЕ +5В.

ВЫВОДЫ 10, 1, 19 – К ОБЩЕМУ ПРОВОДУ.

(ВХОДЫ БУФЕРА: 2, 4, 6, 8, 11, 13, 15, 17; ВЫХОДЫ: 18, 16, 14, 12, 9, 7, 5, 3)

ТЕПЕРЬ МОЖЕТЕ СЕБЯ ПОЗДРАВИТЬ: **ВЫ СЧАТЛИВЫЙ ОБЛАДАТЕЛЬ 192 К.**

<<<<<<< THE END >>>>>>>

ПРИМЕРНЫЙ СОСТАВ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ
 (ПО СОСТОЯНИЮ НА ФЕВРАЛЬ 1990 Г.)
 (* - ПРОИЗВОДСТВО СЕВЕРОМОРСКА)

1. ТОМ "БЭЙСИК И ДЕМОСТРАЦИОННЫЕ ПРОГРАММЫ"

1. 1	LBASIC	.COM	(8K)	-	ИНТЕРПРЕТАТОР 1977Г, ПРОТОТИП.
1. 2	BSC	.COM	(8K)	-	АДАптированный для "ОКЕАНА" LBASIC. РАБОТАЕТ С ФАЙЛАМИ «****.BSC». ГРАФИКА СООТВЕТСТВУЕТ ТОЛЬКО REL5.
1. 3	ARB	.BSC	(2K)	-	ГРАФИКА \
1. 4	ARGRAF	.BSC*	(2K)	-	ГРАФИКА \
1. 5	MUSA	.BSC	(2K)	-	МУЗЫКА > - ДЕМОСТРАЦИОННЫЕ ДЛЯ
1. 6	PKOG	.BSC	(2K)	-	ПРИМЕР / BSC.COM
1. 7	SL	.BSC	(2K)	-	"ШЛЯПА" /
1. 8	MBAS	.COM	(24K)	-	ИНТЕРПРЕТАТОР MICROSOFT, ПРОТОТИП. РАБОТАЕТ С ФАЙЛАМИ "****.BAS". ДОПУСТИМА ГРАФИКА ЧЕРЕЗ CHR\$(). АДАптированный для "ОКЕАНА" MBAS. ГРАФИКА - ДЛЯ REL5.
1. 9	BASIC	.COM	(24K)	-	
1.10	ELLIP	.BAS	(2K)	-	\
1.11	NIKE	.BAS	(2K)	-	\
1.12	PR	.BAS	(2R)	-	\ ДЕМОСТРАЦИОННЫЕ ДЛЯ BASIC.COM
1.13	RITM	.BAS	(4K)	-	/ < ДЛЯ REL5 >
1.14	BIORITM	.BAS	(BK)	-	/
1.15	GRAFIC	.BAS	(2K)	-	/
1.16	ANDREJ	.BAS*	(10K)	-	СИМВОЛЬНАЯ ИГРА ДЛЯ MBAS.COM
1.17	STAR5	.BAS*	(4K)	-	ДЕМОСТРАЦИОННАЯ ДЛЯ REL5
1.18	STAR8	.BAS*	(4K)	-	-"- ДЛЯ REL8
1.19	BASCOM	.COM	(32K)	-	КОМПИЛЯТОР ПРОГРАММ ****.BAS
1.20	BASLIB	.REL	(26K)	-	БИБЛИОТЕКА ПЕРИОДА КОМПОНОВКИ
1.21	OBSLIB	.REL	(48K)	-	ДОПОЛНИТЕЛЬНАЯ БИБЛИОТЕКА
1.22	BRUN	.COM	(16K)	-	ИСПОЛНИТЕЛЬ ПРОГРАММ ****.INT (К ПАКЕТУ "СBASIC")
1.23	RS240	.G	(18K)	-	КАРТИНКА ИЗ <ESC> ДЛЯ REL5 ВЫВОДИТСЯ ПО КОМАНДЕ TYPE
1.24	SHIFT	.COM	(2K)	-	СДВИГАТЕЛЬ ЭКРАНА, ПЕРЕКЛЮЧАТЕЛЬ ПАЛИТР
1.25	ARB	.COM	(2K)	-	ДЕМОСТРАЦИЯ ГРАФИКИ ДЛЯ REL5
1.26	LAD	.COM	(40K)	-	\
1.27	LAD002	.DAT	(2K)	-	\
1.28	CAT	.COM	(30K)	-	\
1.29	CATCHUM	.DAT	(2K)	-	\ СИМВОЛЬНЫЕ ИГРЫ
1.30	REVERSI	.COM	(26K)	-	"РЕВЕРСИ" /
1.31	SN	.COM	(8K)	-	"ПИТОН"(REL5) /
1.32	KK8	.COM*	(8K)	-	"ПИТОН"(REL8) /
1.33	CHES	.COM	(26K)	-	ШАХМАТЫ /
1.34	CHILD	.COM*	(10K)	-	"ПРОГНОЗ РОЖДАЕМОСТИ"

2. ТОМ "ПАКЕТЫ ЯЗЫКОВ ВЫСОКОГО УРОВНЯ"

2. 1	ADA	.COM	(6K)	-	\
2. 2	ADA2	.COM	(20K)	-	\ - ПАКЕТ
2. 3	ADA3	.COM	(10K)	-	/ "АДА"
2. 4	KAPSE	.	(6K)	-	/
2. 5	SQRT	.ADA	(2K)	-	ПРИМЕР
2. 6	GETDIR	.ADA	(2K)	-	-"-
2. 7	C1	.COM	(16K)	-	КОМПИЛЯТОР, 1 ПРОХОД \
2. 8	CC2	.COM	(16K)	-	КОМПИЛЯТОР, 2 ПРОХОД \
2. 9	C	.CCC	(2K)	-	ВСПОМОГАТЕЛЬНЫЙ ФАЙЛ \
2.10	CL	.COM	(6K)	-	КОМПОНОВЩИК > ПАКЕТ
2.11	DEFF	.CRL	(10K)	-	БИБЛИОТЕКА КОМПОНОВЩИКА / "СИ"
2.12	DEFF2	.CRL	(8K)	-	-"- /
2.13	CLIB	.COM	(6K)	-	БИБЛИОТЕКАРЬ /
2.14	KK8	.C *	(4K)	-	ПРИМЕР, "ПИТОН" ДЛЯ REL8

2.15	F80	.COM	(23K)	-	КОМПИЛЯТОР	\	
2.16	FORLIB	.REL	(26K)	-	БИБЛИОТЕКА	>	ПАКЕТ "ФОРТРАН'1
2.17	L80	.COM	(10K)	-	КОМПОНОВЩИК	/	
2.18	CHILD	.FOR*	(4K)	-	ПРИМЕР, ПРОГРАММА		"ПРОГНОЗ"

3. ТОМ "СИСТЕМНЫЕ ПРОГРАММЫ"

(СИМВОЛОМ "'''" ОТМЕЧЕНЫ ПРОГРАММЫ, ОПИСАННЫЕ В КНИГЕ:

М.УЭИТ, ДЖ. АНГЕРМЕЙЕР "ОПЕРАЦИОННАЯ СИСТЕМА CP/M", М."РАДИО И СВЯЗЬ", 1986г.)

3. 1	POWER	.COM	(16K)	-	СИСТЕМНЫЙ РАСШИРИТЕЛЬ		
3. 2	PW	.COM	(16K)	-	МОДИФИЦИРОВАННЫЙ POWER		
3. 3	PWRTOOL	.COM	(2K)	-	МОДИФИКАТОР КОМАНД POWERA		
3. 4	M80	.COM	(18K)'	-	МАКРОАССЕМБЛЕР MICROSOFT		
3. 5	MAC	.COM	(16K)'	-	МАКРОАССЕМБЛЕР DIGITAL RESEARCH		
3. 6	L80	.COM	(10K)'	-	КОМПОНОВЩИК (СБОРЩИК) MICROSOFT		
3. 7	LIB	.COM	(6K)'	-	ПРОГРАММА ОБСЛУЖИВАНИЯ БИБЛИОТЕК		
3. 8	W32	.COM	(10K)	-	\		
3. 9	W64	.COM	(10K)	-	\		
3.10	W80	.COM	(10K)	-	\ ЭКРАННЫЙ РЕДАКТОР WORLD MASTER		
3.11	W407	.COM	(10K)	-	/ ДЛЯ РАЗЛИЧНЫХ РЕЖИМОВ ЭКРАНА И		
3.12	W647	.COM	(10K)	-	/ РЕАЛИЗАЦИИ СИСТЕМЫ (EELS И REL7)		
3.13	W807	.COM	(10K)	-	/		
3.14	WM	.HLP	(4K)	-	СПРАВОЧНИК ДЛЯ WORLD MASTER		
3.15	ED	.COM	(10K)'	-	СТРОКОВЫЙ РЕДАКТОР		
3.16	TEXTOS	.COM	(16K)	-	\		
3.17	TPRINT	.OVR	(8K)	-	\ - ТЕКСТОВЫЙ ПРОЦЕССОР		
3.18	TSMSG5	.OVR	(23K)	-	/ WORLD STAR		
3.19	TSOVLY1	.OVR	(34K)	-	/		
3.20	DDT	.COM	(8K)'	-	СИМВОЛИЧЕСКИЙ ОТЛАДЧИК ПРОГРАММ		
3.21	SID	.COM	(8K)'	-	УЛУЧШЕННЫЙ ВАРИАНТ ОТЛАДЧИКА		
3.22	SID4	.COM	(8K)	-	\ - МОДИФИЦИРОВАННЫЕ ВЕРСИИ		
3.23	SID64	.COM	(8K)	-	/ ОТЛАДЧИКА SID		
3.24	CREF80	.COM	(4K)'	-	ПОСТРОИТЕЛЬ ТАБЛИЦ ПЕРЕКРЕСТНЫХ ССЫЛОК		
3.25	XLT	.COM	(22K)'	-	\ ПАКЕТ КОНВЕРТЕРА ПРОГРАММ НА		
3.26	XLT00	.OVL	(8K)'	-	> - АССЕМБЛЕРЕ 8080 В ПРОГРАММЫ НА		
3.27	XLT01	.OVL	(10K)'	-	/ АССЕМБЛЕРЕ 8086(8088)		
3.28	RESOURS	.COM	(6K)	-	ДИЗАССЕМБЛЕР		
3.29	DIS	.COM	(8K)	-	ДИЗАССЕМБЛЕР		
3.30	PIP	.COM	(8K)	-	ПРОГРАММА ПЕРЕМЕЩЕНИЯ ФАЙЛОВ		
3.31	ARPIP	.COM	(6K)	-	МОДИФИЦИРОВАННЫЙ ВАРИАНТ PIP		
3.32	COPY	.COM	(2K)	-	ПРОГРАММА КОПИРОВАНИЯ ФАЙЛОВ		
3.33	SW	.COM	(8K)	-	УНИВЕРСАЛЬНАЯ ПРОГРАММА ОБСЛУЖИВАНИЯ ФАЙЛОВ		
3.34	XDIR	.COM	(4K)	-	ПРОГРАММА ВЫВОДА КАТАЛОГА ДИСКА		
3.35	DESPOOL	.COM	(4K)'	-	ПРОГРАММА ФОНОВОЙ ПЕЧАТИ		
3.36	FMAP	.COM	(2K)	-	ПРОГРАММА ВЫВОДА ИНФОРМАЦИИ О РАЗМЕЩЕНИИ ФАЙЛА НА		
					ДИСКЕ		
3.37	STAT	.COM	(6K)'	-	ПРОГРАММА ВЫВОДА СТАТУСА ФАЙЛОВ И ДИСКОВ		
3.38	DOC	.COM	(36K)	-	УНИВЕРСАЛЬНАЯ ПРОГРАММА РЕДАКТИРОВАНИЯ И		
					ДИАГНОСТИКИ ДИСКОВ		
3.39	DOCTOR	.COM	(8K)	-	ПРОГРАММА ВОССТАНОВЛЕНИЯ ИНФОРМАЦИИ С ПОРЧЕННЫХ		
					ДИСКОВ		
3.40	DTEST	.COM	(4K)	-	ПРОГРАММА ТЕСТИРОВАНИЯ, ПРОВЕРКИ И ФОРМАТИРОВАНИЯ		
					ДИСКОВ		
3.41	CHECK	.COM	(4K)	-	ПРОГРАММА ПОДСЧЕТА КОНТРОЛЬНОЙ СУММЫ ФАЙЛА		
3.42	CONVRT	.COM	(2K)	-	ПРОГРАММА КОНВЕРТИРОВАНИЯ ТЕКСТОВЫХ ФАЙЛОВ В		
					РАЗЛИЧНЫХ КОДОВЫХ ТАБЛИЦАХ		
3.43	IBM	.COM	(2K)	-	ПРОГРАММА УСТАНОВКИ НА ДИСКЕ C: ФОРМАТА IBM PC		
3.44	ASM48	.COM	(14K)	-	КРОССТРАНСЛЯТОР ПРОГРАММ НА АССЕМБЛЕРЕ 1816BE48		
3.45	DDT48	.COM	(12K)	-	КРОССОТЛАДЧИК 1816BE48		
3.46	INST51	.COM	(2K)	-	\		
3.47	BEGIN	.COM	(2K)	-	\		
3.48	CORRECT	.COM	(2K)	-	\ - ПАКЕТ ПРОГРАММ ДЛЯ ОБМЕНА		
3.49	BYTE	.COM	(10K)	-	/ ИНФОРМАЦИЕЙ ПО RS-232		
3.50	KERMIT	.COM	(14K)	-	/		
3.51	D	.COM	(4K)	-	/		

ТОМ 4. "ПАКЕТЫ ТАБЛИЦ И БАЗ ДАННЫХ"

4. 1	WT	.COM	(24K)	-	\	
4. 2	WT	.OVL	(16K)	-	>	- ЭЛЕКТРОННАЯ ТАБЛИЦА "ВАРИТАБ"
4. 3	WT	.HLP	(6K)	-	/	
4. 4	TFW	.COM	(10K)	-	\	- ПРЕОБРАЗОВАТЕЛЬ ФАЙЛОВ
4. 5	TFW	.OVL	(16K)	-	/	ДЛЯ "ВАРИТАБ"
4. 6	MINFOR	.COM	(20K)	-	\	
4. 7	MINERRES	.MES	(6K)	-	\	
4. 8	MINHELP	.TXT	(6K)	-	\	- ПАКЕТ БАЗЫ ДАННЫХ СИСТЕМЫ
4. 9	MINFOR	.003	(14K)	-	/	"МИКРО-ИНФО"
4.10	MIMFOR	.004	(20K)	-	/	
4.11	MINFOR	.005	(4k)	-	/	
4.12	CREATDB	.COM	(22K)	-		СОЗДАНИЕ БАЗЫ ДЛЯ "MINFOR"
4.13	DBDXCT	.DAT	(2K)	-		ПРИМЕР
4.14	DBASER	.COM	(20K)	-	\	
4.15	REBUSOVR	.COM	(40K)	-	\	
4.16	REBUSMSG	.TXT	(38K)	-	\	- ПАКЕТ БАЗЫ ДАННЫХ "РЕБУС"
4.17	RSORT	.CMD	(6K)	-	/	
4.18	SETS	.CMD	(2K)	-	/	
4.19	INSTALL	.COM	(16K)	-	/	
4.20	DBASE	.COM	(20K)	-	\	
4.21	DBASEOVR	.COM	(40K)	-	\	- ПАКЕТ БАЗЫ ДАННЫХ "DBASE-II"
4.22	DBASEMSG	.TXT	(52K)	-	/	
4.23	DBINST	.COM	(14K)	-	/	

ТОМ 5. "МАКРОБИБЛИОТЕКА"

5. 1	README	.LIB	(2K)	-	\	
5. 2	COMPARE	.LIB	(2K)	-		
5. 3	DOWHILE	.LIB	(2K)	-		
5. 4	DSTACK	.LIB	(10K)	-		
5. 5	INTER	.LIB	(4K)	-		
5. 6	DISKDEF	.LIB	(8K)	-		
5. 7	MACRO	.LIB	(18K)	-		
5. 8	NCOMP	.LIB	(2K)	-	>	- ФАЙЛЫ МАКРОБИБЛИОТЕКИ
5. 9	SELECT	.LIB	(2K)	-		
5.10	SIMP10	.LIB	(2K)	-		
5.11	STACK	.LIB	(2K)	-		
5.12	TREADLE	.LIB	(2K)	-		
5.13	WHEN	.LIB	(2K)	-		
5.14	SEQ10	.LIB	(12K)	-		
5.15	MCN	.LIB	(8K)	-	/	

ТОМ 6. "ИГРЫ"

!!! "''" - ИГРЫ ТРЕБУЮТ НЕСТРОБИРУЕМОГО ОБМЕНА С КЛАВИАТУРОЙ (ДОРАБОТКА СХЕМЫ)

6. 1	GR	.COM	(8K)	-		ГРАФИЧЕСКИЙ РЕДАКТОР (ДЛЯ REL5)
6. 2	DEM	.COM	(8K)	-		ДЕМОНСТРАЦИОННАЯ
6. 3	DIVER	.COM	(14K)	'		
6. 4	HUNTER	.COM	(10K)	'		
6. 5	JACK	.COM	(10K)	'		
6. 6	KARATE	.COM	(16K)	'		
6. 7	POINTER	.COM	(6K)	'		
6. 8	SAFARY	.COM	(20K)	'		
6. 9	SHERIF	.COM	(30K)	'		
6.10	VEGA	.COM	(20K)	'		
6.11	CHESS	.COM	(38K)	'		
6.12	RALLY	.COM	(8K)	'		
6.13	WAR	.COM	(12K)	'		
6.14	TETRI	.COM	(10K)	'		
6.15	TETRIS	.COM	(10K)	'		
6.16	BATTY	.COM	(10K)	'		
6.17	BOMBER	.COM	(4K)	'		

6.18	LIFE8	.COM*	(4K)	
6.19	LTN8	.COM*	(6K)	
6.20	MUR8	.COM*	(8K)	- МУЗЫКАЛЬНЫЙ РЕДАКТОР
6.21	ARGO	.COM*	(6K)	

ТОМ 7. "ОПИСАНИЯ ПРОГРАММНЫХ СРЕДСТВ"

7. 1	CPM	.DOC	(42K)	- ОПИСАНИЕ ОПЕРАЦИОННОЙ СИСТЕМЫ
7. 2	DIR	.PW	(6K)	\
7. 3	COMAND	.PW	(122K)	\ - ОПИСАНИЕ СИСТЕМОГО РАСШИРИТЕЛЯ
7. 4	INTRO	.PW	(36K)	> "POWER .COM"
7. 5	ADAPT	.PW	(24K)	/
7. 6	WHAT-DO	.PW	(22K)	/
7. 7	M80	.DOC	(34K)	- ОПИСАНИЕ ТРАНСЛЯТОРА С МАКРО
7. 8	M80	.DOK	(44K)	- РАСШИРЕННОЕ ОПИСАНИЕ МАКРО
7. 9	L80	.DOC	(8K)	- ОПИСАНИЕ КОМПОНОВЩИКА "L80 .COM"
7.10	LIB	.DOC	(6K)	- ОПИСАНИЕ БИБЛИОТЕКАРЯ "LIB .COM"
7.11	RESOURS	.HLP	(32K)	- ОПИСАНИЕ ДИЗАССЕМЛЕРА "RESOURS.COM"
7.12	MBASIC	.HLP	(22K)	- ОПИСАНИЕ ИНТЕРПРЕТАТОРА "MBAS.COM" (НЕПЕРЕВЕДЕННОЕ)
7.13	TEXTOS	.DOC	(44K)	- ОПИСАНИЕ ТЕКСТ-ПРОЦЕССОРА "TEXTOS"
7.14	WM	.HLP	(4K)	- КОМАНДЫ РЕДАКТОРА "W*** .COM"
7.15	READ48	.ME	(4K)	- ОПИСАНИЕ КРОСС-СРЕДСТВ 1816BE48
7.16	XLT	.HLP	(22K)	- ОПИСАНИЕ КОНВЕРТЕРА "XLT .COM"
7.17	WT	.HLP	(8K)	- ОПИСАНИЕ ПАКЕТА "ВАРИТАБ"
7.18	MINHELP	.TXT	(6K)	- ОПИСАНИЕ ПАКЕТА "MINFOR"
7.19-.22	FORT**	.DOK	(250K)	- ОПИСАНИЕ ТРАНСЛЯТОРА С FORTRAN
7.23-.32	PL**	.DOC	(300K)	- ОПИСАНИЕ ТРАНСЛЯТОРА С PASKAL MT+
7.33-.41	DB**	.DOC	(166K)	- ОПИСАНИЕ ДАЗЫ ДАННЫХ "DBASE-2"
7.42-.56	DOS**	.DOC	(600K)	- КНИГА П.НОРТОНА ОБ ОРГАНИЗАЦИИ IBM PC
7.57-.60	BOOK*	.TXT	(106K)*	- СПРАВОЧНОЕ ПОСОБИЕ "ОКЕАН-240"
7.61	UEL1	.DOC	(6K)*	- ТАБЛИЧКА КОДОВ КР580BM80A
7.62	UEL3	.DOC	(8K)*	- ТАБЛИЧКА СИСТЕМЫ КОМАНД КР580BM80A

ТОМ 8. "ПАКЕТ ПАСКАЛЬ МТ+"

8. 1	MTPLUS	.COM		- ТРАНСЛЯТОР
8. 2-.8	MTPLUS	.00 -	MTPLUS .06	- ОВЕРЛЕЙНЫЕ МОДУЛИ
8. 9	LINKMT	.COM		- КОМПОНОВЩИК
8.10	DEBUGGER	.ERL	\	
8.11	PASLIB	.ERL	\	
8.12	RANDOMIO	.ERL	>	- БИБЛИОТЕКИ
8.13	TEANCEND	.ERL	/	
8.14	FPREALS	.ERL	/	
8.15	LIBMT	.COM		- БИБЛИОТЕКАРЬ
8.16	DIS8080	.COM		- ДИЗАССЕМЛЕР
