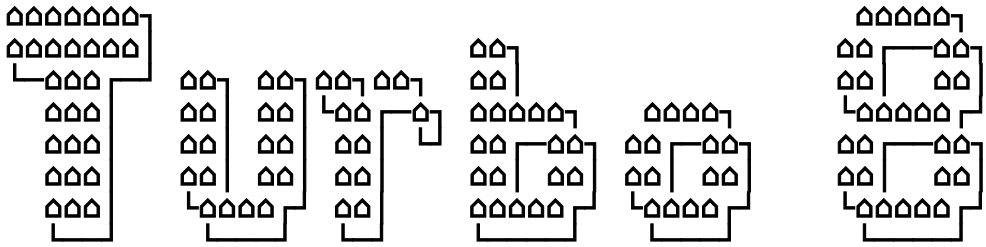


Москва

1996

Для ANDOS

Комплект:
1. Turbo8 - основная программа.
2. Turbo8. 10 - для БК10 с доп. ОЗУ 16К
3. ARIFM.OBJ - модуль выполнения команд расширенной арифметики.
4. TURBO8.VXT - документация.

Максимальная длина исходного текста - **74000**
Максимальная длина получаемой программы - **77000**

Система программирования на языке Ассемблера Turbo8 является дальнейшим развитием ассемблеров серии TURBO (MicroWS, TURBO4H, TURBO5M, TURBO6M). Система работает только с ANDOS.

Основной задачей, при создании Turbo8, было удобство создания и отладки коротких программ.

Отличия от предыдущих версий ассемблера:

1. В редакторе и компиляторе введен контроль строк.
2. В арифметике над метками введено деление на 2, что удобно для работы с массивами слов.
3. Убраны псевдокоманды .TTYIN, .TTYOUT, .ENABL, .DSABL, присвоение имен регистрам и добавлена псевдокоманда .ADDR.
4. Введено динамическое перераспределение памяти между транслируемой программой, текстом и таблицей меток.
5. При трансляции проверяется четность адреса команды и величина аргумента .BLKB и .BLKW.
6. Программа записывается в каталог по адресу, указанному командой LA или псевдокомандой .LA.
7. Время трансляции по команде CO значительно уменьшено и лишь незначительно больше, чем по команде CL.
8. Во всех случаях правильно работает арифметика над метками.
9. Полностью изменена работа с принтером. Можно печатать любые фрагменты текста, правильно работает печать меток.
10. В редакторе введены переход по номеру строки и смыкание строк, значительно ускорена работа редактора.
11. На БК11 использована такая система загрузки программы, которая не портит ни монитор, ни DOS, а та часть оболочки,

на место которой загружается система, восстанавливается при выходе. После выхода в монитор командой MO можно вернуться в систему или выйти в DOS. Если программа короткая, то можно одновременно работать с отладчиком.

12. Запрещен вход в редактор, если текст поврежден при трансляции.
13. Добавлена команда трансляции до первой ошибки с установкой метки на ошибочной строке.
14. После запуска программы на исполнение и возврата в Turbo восстанавливаются драйверы экрана и клавиатуры.

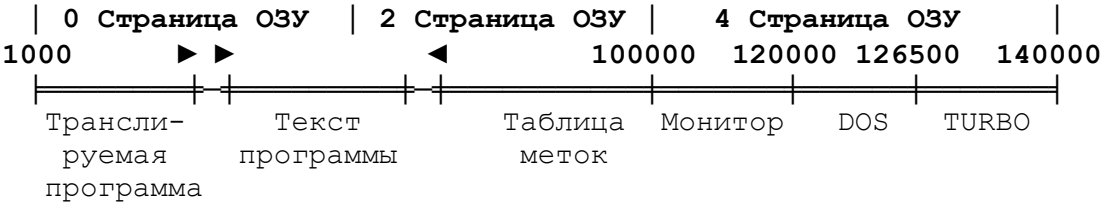
Недостатки:

Если первый модуль программы объектный и в нем есть начальные присвоения, то адрес трансляции, установленный командой LA, должен соответствовать адресу, установленному псевдокомандой .LA в этом модуле. Это не касается случаев, когда первый модуль текстовый и связано с изменением трансляции по CL, что позволило правильно обрабатывать арифметику над метками во всех модулях, что более важно.

Автор надеется, что работа с системой будет для Вас не только трудом, но и удовольствием.

РАБОТА Turbo8

Основная часть программы располагается за ANDOSом с адреса 126500. В 3 странице ОЗУ находятся подгружаемые модули, а также сохраняется та часть DiskMASTER, на место которой загружена основная часть TURBO. 0 и 2 страницы занимает ОЗУ пользователя. VDISK уничтожается.



Текст программы перед началом трансляции расположен в памяти с адреса, установленного командой TA (по умолчанию 7776). Транслируемая программа располагается с адреса 1000. При трансляции обрабатываемый адрес программы сдвигается вправо, обрабатываемый адрес текста тоже вправо, но так как текст команды длиннее, чем сама команда, то программа отстает от текста.

Исключение - псевдокоманды .BLKW и .BLKB, поэтому введен контроль за тем, чтобы при обработке этих псевдокоманд транслируемая программа не нарушила текст. Если аргумент этих псевдокоманд слишком велик, то выдается сообщение об ошибке, и если нужно обнулить большой участок программы, то лучше это сделать в конце программы, когда расстояние между программой и текстом больше, если текст не слишком велик и не сдвигался, чтобы освободить место для меток.

Таблица меток сдвигается влево, поэтому контролируется расстояние между концом текста и таблицей меток, и если места для меток не хватает, то текст сдвигается в сторону программы, оставляя запас для .BLKW и .BLKB 1000 байт. Таким образом можно транслировать тексты максимальной длины.

Если транслируемая программа короче 2776 байт, то текст программы при трансляции не нарушается, поэтому короткие прог-

раммы можно запускать на выполнение командой RU и вводить исправления в текст, не перезагружая его.

На БК10 система загружается в доп. ОЗУ с адреса 140000.

СОСТАВ СИСТЕМЫ Turbo8

1. Монитор.
2. Ассемблер.
3. Редактор связей.
4. Редактор текста.

МОНИТОР

LO - Загрузка исходного текста.
LF - Подстыковка текста.
TA - Установка адреса загрузки текста (по умолчанию 7776). Чем больше адрес, тем больше места для отлаживаемой программы, и для буфера текста в редакторе. Пределы - 3776-20000.
ST - Запись всего текста на диск.
SF - Запись текста от начала до курсора.
SS - Запись текста от курсора до конца.
SC - Вход в редактор текста. Если перед этим производилась трансляция и текст был поврежден, то вход в редактор запрещен. Надо заново загрузить текст.
SA - Запись оттранслированной программы на диск.
SL - Запись объектного модуля, предназначенного для компоновки с другими модулями.
CO - Трансляция программы в абсолютном формате, с выдачей номера ошибки, номера строки, в которой обнаружена ошибка, и самой ошибочной строки, причем трансляция не прекращается, по окончании трансляции выдается длина программы, а также список меток и их адреса. При наличии ошибок список меток не выдается.
CE - То же, что CO, но до первой ошибки и с установкой метки на ошибочной строке.
CL - Трансляция программы в относительном (перемещаемом) формате, с возможностью затем связать модули в единую программу командой LI, таблица меток не выдается.
LA - Установка адреса последующей загрузки программы. По умолчанию = 1000. Отдают перед командой CO, при трансляции в абсолютном формате, или перед LI, при трансляции в перемещаемом формате. ПРОБЕЛ - отказ.
LI - Компановка программы из модулей. Если первый модуль объектный, то адрес трансляции, установленный командой LA должен соответствовать адресу, установленному псевдокомандой .LA в этом модуле.
LL - Линковка метки в конец программы, может пригодится для установки буферов в конце модуля.
LM - Линковка любого массива данных как объектного модуля. Загружается массив, затем спрашивается имя метки, которое нужно ему присвоить. Удобно для линковки к программам текстов, графики и пр.
LN - Просмотр длины текста.
TT - Просмотр таблицы меток.
PT - Печать таблицы меток на принтере. Неопределенные метки печатаются подчеркнутыми.
OP - Оптимизация ассемблерного текста, т.е. удаление из текста пустых строк и комментариев.

RS - Перезапуск системы.
MO - Выход в монитор БК11 БК10
Возврат в систему - S126502 S140000
Возврат с сохранением текста - S126504 S140002
Выход в DOS - S126500
EX - Выход в DOS.
RU - Запуск программы на исполнение.
VE - Восстановление экрана (если экран сдвинут или поврежден программой пользователя).
СТОП-Прерывание выполнения программы.

АССЕМБЛЕР

Исходная программа записывается в виде последовательности команд, причем в одной строке может быть несколько команд или псевдокоманд, разделенных пробелами. Между меткой и командой, между командой и операндами, а также между операндами и комментарием может быть произвольное количество пробелов (не менее одного). Операнды разделяются запятыми, пробелов между ними не должно быть. Метка не обязательно должна быть в одной строке с командой или псевдокомандой, она может быть в предыдущей строке.

МЕТКА: КОМАНДА(ПСЕВДОКОМАНДА) ОПЕРАНДЫ ;КОММЕНТАРИЙ

МЕТКА: - Имя, определяемое пользователем, которому соответствует адрес трансляции. Длина метки - до 6 символов, метка ограничивается двоеточием (:). Метка может содержать заглавные ЛАТ буквы, цифры, а также знаки ($ и .). Метка начинающаяся с буквы - глобальная, с цифры - локальная, ее действие распространяется до определения глобальной метки. Знаки ($ и .) не могут быть в начале метки. В командах из двух или трех слов метки должны быть глобальными. В поле меток может быть несколько меток, имеющих одно и тоже значение. Если при трансляции встречается ранее определенная метка, то выдается сообщение об ошибке. Метку можно поставить перед любой командой или псевдокомандой. Если ее поставить перед псевдокомандой .END, то ее адрес будет следующим после последнего адреса программы (как и при установке метки командой LL монитора).

Пример:
```
START:STOP: BIC  #177770,R0
            CALL RE
            . . . . .
RE:         CMP  R0,#7
```

Здесь метки START и STOP имеют одно и тоже значение и обращение может производиться по любой из них. Метка RE глобальная, так как команда CALL ... состоит из двух слов.

ПРЯМОЕ ПРИСВАИВАНИЕ

Присваивает имени определенное значение. Имя принимает абсолютное значение. Присваивание допустимо до начала текста программы.

Пример:
```
START=1000
STOP=START+2000
```

ИМЕНА РЕГИСТРОВ

Обращение к регистру по его имени:

R0 R1 R2 R3 R4 R5 SP PC

ВЫРАЖЕНИЯ

В поле операнда допустима запись выражений, что существенно облегчает программирование и сокращает время на разработку программы.

1. Запись восьмеричных и десятичных констант.
   Точка - признак, что константа десятичная (64.=100).
   MOV #10.,R1 аналогично MOV #12,R1

2. Запись положительных и отрицательных чисел.
   MOV #16,R1 или MOV #-2,R1, MOV -2(R4),R3

3. Запись одного кода ASCII, как число.
   MOV #'A,R1 то же, что MOV #101,R1
   Можно использовать в любом случае вместо числа,
   например: ***TRAP '&*** вместо ***TRAP 46***

4. Запись двух кодов ASCII, как число.
   MOV #"AB,R1 (два байта)

5. Точка - значение адреса первого слова команды.
   Удобно для задания смещения для перехода.

ПРИМЕЧАНИЕ: точка может быть использована в качестве любого аргумента команды, но не рекомендуется использовать точку с именами меток при трансляции объектного модуля (CL).

Примеры:
```
START:  MOV R2,R1
        BCS .-2     - переход на метку START.
        HALT
```

Можно транслировать в перемещаемом формате (CL) (имени метки нет).

```
ST:     MOV RC,R1
        ADD (PC)+,R1
        .WORD END-.
END:    .END
```

Можно транслировать только в абсолютном формате (CO). В перемещаемом надо записать .WORD @END+2 или в любом случае использовать псевдокоманду .ADDR (.ADDR R1,END).

6. Использование арифметики над метками:

Данная версия ассемблера допускает использование арифметических выражений в качестве аргумента команды, причем длина выражения не ограничена.

Пример:
```
SUMA=101
SUMB=102
.WORD SUMB-SUMA+3
.END
```

В этом случае по директиве .WORD запишется число '4'.

Пример:
```
MET=100
SUM=40
MOV #MET-SUM+MET,R1
.END
```

В этом случае в регистр R1 записывается число '140'.

Пример:
```
        .ADDR R0,START
        MOV  #140000,R1
        MOV  #/END-/START,R2
1$:     MOV  (R0)+,(R1)+
        SOB  R2,1$
START:  . . . . .
        . . . . .
END:    .END
```

Здесь массив от метки START до метки END пересылается на адрес 140000. Знак деления (/) означает, что адрес метки делится на 2, что в результате дает в R2 длину массива в словах. Пересылка словами в два раза быстрее, чем байтами. Использование деления возможно только с именами меток при непосредственной адресации (27), в псевдокоманде .WORD и в начальных присваиваниях. По директиве .ADDR мы получаем в R0 абсолютный адрес метки START, даже если программа перемещаемая.

7. Запись байта числа в строку букв.
   .ASCII /ABC/<12><15> - в строку дописать коды ПС,ВК

8. Получение смещения к метке.
   .WORD @START

МЕТОДЫ АДРЕСАЦИИ, КОМАНДЫ

Аналогичны системе команд ЭЛЕКТРОНИКИ-60.

CLR MET, MOV #MET,R1, CLR @#MET, CLR @MET
CLR MET(R1), CLR @MET(R1), CLR R1, CLR (R1)
CLR -(R1) , CLR @-(R1), CLR (R1)+, CLR @(R1)+

Двухоперандные:

MOV[B] S,D; CMP[B] S,D; BIT[B] S,D; BIC[B] S,D;
BIS[B] S,D; XOR Rn,D; ADD S,D; SUB S,D;

Однооперандные:

CLR[B] D; COM[B] D; INC[B] D; DEC[B] D; NEG[B] D;
ADC[B] D; SBC[B] D; TST[B] D; ROR[B] D; ROL[B] D;
ASR[B] D; ASL[B] D; SWAB D; SXT D;
MTPS S; MFPS D

Ветвления:

BR MET; BNE MET; BEQ MET; BPL MET; BMI MET; BCC MET;
BCS MET; BVS MET; BVC MET; BHI MET; BLOS MET;
BHIS MET; BLO MET; BGE MET; BLT MET; BGT MET; BLE MET;

Управления:

JMP D; SOB Rn,MET; JSR Rn,D; RTS Rn; MARK n; TRAP n;
EMT n; NOP; BPT; IOT; WAIT; RESET; HALT; RTI; RTT;
CALL D; RETURN;

Условия:

CLC; CLV; CLN; CLZ; CCC; SEC; SEV; SEN; SEZ; SCC;

Расширенной арифметики:

MUL S,Rn; DIV S,Rn; FMUL Rn; FDIV Rn; FADD Rn;
FSUB Rn; ASH S,Rn; ASHC S,Rn;

ПРИМЕЧАНИЕ: Нельзя использовать относительную адресацию (67,77) при обращении к внешним, по отношению к транслируемой программе, адресам при трансляции в перемещаемом формате. Нужно использовать абсолютную адресацию.

Примеры: ***MOV R0,@#177714*** Ограничений нет.

MOV R0,177714 Можно транслировать только в абсолютном формате (CO).

Если нужно транслировать по CL, то можно сделать таким образом:

```
PORT=177714     ;Перед началом программы.
 . . . .
 . . . .
MOV  R0,PORT
```

ПСЕВДОКОМАНДЫ

.LA адрес - псевдокоманда, равносильная команде "LA" монитора системы, подавать до начальных присвоений.

.BLKB N - резервирование N байт.

.BLKW N - резервирование N слов.

ПРИМЕЧАНИЕ: Если аргумент этих псевдокоманд слишком велик, то возможно сообщение об ошибке (см. в начале описания).

.WORD N,N,...,N - запись слова (число или имя). Если нужно записать смещение к метке, то это делается (MET-.) или (@MET), что означает из адреса метки вычесть текущее значение счетчика. Операнды разделяются запятыми, пробелов между ними не должно быть.
(.WORD 1,-1,@START,START-.,START+STOP-END,'A,"BC).

.BYTE N,N,...,N - запись байта (число).
(***.BYTE 1,-177776,'A,'B***). При записи числа надо учитывать, что, например, -1=177777 и в байт это число

записать нельзя, а -177776=2 и в байт записать можно.

.EVEN - если содержание счетчика команд нечетное, то добавляется нуль, иначе игнорируется.

.ASCII/.../ - запись строки символов в память. Строка может ограничиваться знаками (/ ' "). Знаки в начале и конце строки должны быть одинаковыми. Символ в числовой форме заключается в угловые скобки. Нежелательно в строку записывать более одного пробела. Если это необходимо, то лучше сделать в виде: <40><40> и т.д.

Пример:

.ASCII /ВЫХОД В МОНИТОР ?/<40><40>"Да/Нет:"<12>

В первом случае для ограничения строки использована дробная черта, во втором, поскольку она есть в строке, использованы кавычки.

.ASCIZ/.../ - то же с добавлением в конце строки нуля.

.RAD50/.../ - запись строки в коде RAD50 в память, причем в поле операнда допустимо наличие пробелов. (.RAD50/HALT/).

.PRINT #TXT - расшифровывается как:
```
MOV  #TXT,R1
CLR  R2
EMT  20
```

.ADDR Rn,MET - получение в регистре Rn абсолютного адреса метки MET, расшифровывается как:
```
MOV  PC,Rn
ADD  (PC)+,Rn
.WORD @MET+2
```

Пример:
```
START:  .ADDR R1,TEXT
        CLR  R2
        EMT  20
        HALT
TEXT:   .ASCIZ<234>/ERROR !/<234>
        .EVEN
```

Аналогично ***.PRINT #TEXT***, но ***.PRINT #TEXT*** нельзя использовать в перемещаемой программе.

.END - оператор завершения программы.

РАБОТА С БЛОКОМ РАСШИРЕННОЙ АРИФМЕТИКИ

Для организации работы с блоком расширенной арифметики необходимо перед его использованием в программе инициализировать ее.

Это делается внесением в текст команды CALL ARIFM, после чего возможно написание программ с использованием мнемоники команд расширенной арифметики.

После трансляции такой программы ее необходимо связать линкером с объектным модулем ARIFM.OBJ командой:

LI ИМЯ ? ARIFM.OBJ <ВК> и записать полученный файл командой **SA ИМЯ ? NAME <ВК>**, полученная программа работает автоном-

но. При поступлении кода команды расширенной арифметики происходит прерывание по вектору 10 и соответствующая команда обрабатывается, т. о. на данной машине возможно использования программного обеспечения, написанного для процессора 1801ВМ2.

Можно в начале программы сделать:

MOV #160016,@#10, но эмулятор расширенной арифметики в в БК11 не обрабатывает чисел с плавающей запятой.

DIV - деление 32 разрядного слова RnRn+1 на число. Регистр в команде должен быть четным.

```
MOV #75.,R1
CLR R0
DIV #10.,R0 ;Деление числа 75. на 10.,
HALT        ;в R0 - результат, в R1 - остаток.
```

MUL - умножение регистра на число. Причем если номер регистра нечетный, то сохраняется младшая часть результата.

```
MOV #7,R1
MUL #10.,R1 ;умножение 7*10=70 в регистре R1.
HALT
```

ASH - арифметический сдвиг регистра вправо, влево на (-32+32) позиции в зависимости от значения 5 бита аргумента сдвига. При 1 в 5 бите - сдвиг вправо, при нуле - влево. ASH #5,R1; сдвиг регистра R1 на 5 позиций влево.

ASHC - арифметический сдвиг двойного слова, причем регистр с нечетным номером содержит младшую часть слова, а с четным старшую, остальное аналогично команде ASH.

FMUL - умножение чисел с плавающей запятой. BA*B - результат на место аргумента B.

FDIV - деление чисел с плавающей запятой. BA/B - результат на место аргумента B. Если делитель (B) равен нулю, то результат в стек не записывается.

FADD - сложение чисел с плавающей запятой, регистр указывает на адрес нахождения аргументов. BA+B - результат помещается на место аргумента B.

FSUB - вычитание чисел с плавающей запятой. BA-B - результат в B.

```
        FSUB R5          ;R5 указывает на адрес MET
        HALT

MET:    .WORD A1,A2      ;два слова аргумента A
        .WORD B1,B2      ;два слова аргумента B
```

ФОРМАТ ЧИСЕЛ С ПЛАВАЮЩЕЙ ЗАПЯТОЙ

младшая часть мантисы
S - знак числа.

Использование команд расширенной арифметики позволяет упростить процедуру программирования, особенно те фрагменты, где используются арифметические операции.

РЕДАКТОР ТЕКСТА

КОМАНДЫ РЕДАКТОРА

При работе редактора текста в служебной строке показываются: номер строки, в которой находится курсор, количество свободной памяти и имя файла, с которым Вы работаете. При нажатии префикса (ВС) заголовок инвертируется.

ВС - префикс редактора.

◄	- курсор влево.
►	- курсор вправо.
▲	- курсор вверх.
▼	- курсор вниз.
|<==	- сдвижка строки.
|==>	- раздвижка строки.
<==|	- забой, смыкание строк.
СУ/Т и ТАБ	- табулятор.
СУ/Э	- поиск по модели.
СУ/ъ	- установить метку.
СУ/Щ	- удалить строку.
СУ/Ч	- вставить строку.
ВС ◄	- курсор в начало строки.
ВС ►	- курсор в конец строки.
ВС ▲	- на страницу вверх.
ВС ▼	- на страницу вниз.
ВС |<==	- стереть до начала строки.
ВС |==>	- стереть до конца строки.
ВС <==|	- восстановить строку.
ВС М	- ввод макропоследовательности, можно описывать макрокоманды для редактора и многое другое, запоминается последовательность нажатий клавиш, причем клавишу "ВВОД" заменяет "_". При перезапуске текст ключа сохраняется.
ВС Э	- установить модель поиска.
ВС ъ	- уйти к метке, установленой командой СУ/ъ или при трансляции по СЕ.
ВС Щ	- текст в буфер.
ВС Ч	- текст из буфера.
ВС N	- в начало текста.
ВС К	- в конец текста.

ВС L - переход по номеру строки. ПРОБЕЛ - отказ.
ВС U - удаление текста (буфер сохраняется).
ВВОД - ввести строку (разбить на две).
ВС ВВОД - напечатать фрагмент текста, начиная с метки, до строки, после которой находится курсор.
КТ и ВС КТ - выход из редактора.
АР2+1 - вывод макропоследовательности.
СТОП - остановка печати с выходом из редактора.

Нежелательно использовать клавишу СТОП для выхода из редактора, так как при этом не сохраняется строка, в которой находится курсор.

ФУНКЦИОНИРОВАНИЕ КЛАВИАТУРЫ

1. **◄** - перемещение курсора по строке влево, при достижении начала строки курсор автоматически переносится в конец предыдущей строки.

2. **►** - перемещение курсора по строке вправо, при достижении конца строки курсор переносится в начало следующей строки.

3. **<==|** - удаление символа перед курсором, причем если курсор стоит на печатном символе, то автоматически происходит сдвижка конца строки вслед за курсором, а если символ не печатный, исключая <ВК>, то сдвижка строки не производится. При достижении начала строки строка переносится вверх и присоединяется к концу предыдущей строки. Если общая длина двух строк слишком велика, то команда не выполняется.

4. **▲▼** - занесение текущей строки из буфера строки в основную память и переход к следующей строке.

5. **/ <==|** - восстановление предыдущего состояния строки, которое было до момента редактирования текущей строки (действует до момента нажатия на клавишы '').

ПОИСК ПО МОДЕЛИ

В редакторе предусмотрен поиск по модели. Для этого необходимо определить модель. Это делается командой ВС+Э, при этом выдается запрос модели поиска, по которому нужно ввести модель и нажать ВВОД. Поиск осуществляется командой СУ/Э, начиная с текущей позиции, при обнаружении модели курсор останавливается за найденной моделью.

Если модель не была обнаружена, то выдается звуковой сигнал и курсор остается в той же позиции.

Если текст модели состоит из нескольких слов, то надо следить за количеством пробелов.

РАБОТА С БУФЕРОМ

Буфер используется для:

1. Удаления фрагмента текста.
2. Переноса фрагмента текста.
3. Размножения фрагмента текста.

Размер буфера 2500 байт, что соответствует примерно двум страницам текста. Для определения фрагмента его нужно пометить. Это делается установкой метки на начало фрагмента, командой СУ/ъ. На конец фрагмента указывает курсор (не включительно).

Занесение фрагмента текста в буфер осуществляется командой ВС+Щ, при этом фрагмент удаляется и заносится в буфер. Если фрагмент текста не помещается в буфере, то выдается звуковой сигнал и фрагмент не удаляется.

Для вызова фрагмента текста из буфера нужно установить курсор в нужную позицию и отдать команду ВС+Ч. При этом, начиная с текущей позиции, вставляется фрагмент из буфера. Вставку можно повторять несколько раз.

МАКРОПОСЛЕДОВАТЕЛЬНОСТЬ

Удобно при печати большого количества повторяющихся команд. Например: при печати текстов достаточно сделать: ВС М ТАБ . ASCIZ/ _ Ввод. Тогда достаточно нажать АР2+1, и при каждом нажатии будет печататься:

.ASCIZ/
.ASCIZ/ и т.д.

В тексте ключа могут быть управляющие коды: СДВИЖКА, РАЗДВИЖКА, команды управления курсором, команды редактора.

УСТАНОВКА МЕТКИ

Необходима для печати на принтере, работы с буфером и для быстрого возврата к нужному участку текста. Запоминается номер строки и адрес строки в памяти. При трансляции по СЕ метка на ошибочной строке ставится автоматически.

РАБОТА С ПРИНТЕРОМ

Программа позволяет напечатать фрагмент текста любой величины, от одной строки до всего текста. Для этого необходимо по команде SC войти в редактор текста, поставить курсор на первую строку нужного фрагмента и поставить метку командой СУ/ъ. После этого поставить курсор на следующую после последней строки фрагмента строку и дать команду ВС+ВВОД. При этом экран очищается. После вывода всего фрагмента на принтер экран снова заполняется текстом, начиная с помеченой строки.

Если команда дана ошибочно или необходимо прекратить печать, нажать СТОП. Система выйдет в монитор.

После начального запуска или перезапуска командой RS метка стоит на первой строке текста, то есть, если метку не ставить, то текст будет печататься с первой строки.

Если при команде ВС+ВВОД курсор стоит на помеченой строке, или перед ней, то подается звуковой сигнал ошибки и команда не выполняется, так же, как и при удалении фрагмента в бу-

фер.

При печати таблицы меток по команде PT неопределенные метки печатаются подчеркнутыми.

СХЕМА ПОДКЛЮЧЕНИЯ ПРИНТЕРА МС6313

ПОЛОЖЕНИЕ ПЕРЕКЛЮЧАТЕЛЕЙ ПРИНТЕРА МС6313

КОИ8	01101001	11100111	00000101

КОМПИЛЯЦИЯ

Компиляция исходного текста может производиться в двух форматах выдачи:

1. Трансляция с получением объектного модуля.

2. Трансляция с получением загрузочного модуля.

Объектный модуль позиционно независим и может быть затем связан с другими модулями или привязан сам по любому адресу, установленному ранее командой LA.

Трансляция объектного модуля производится командой CL, полученный модуль должен быть записан командой SL, с целью дальнейшей связки в виде загрузочного модуля командой LI.

Загрузочный модуль транслируется по адресам, установленным командой LA и может быть записан командой SA и загружаться, как готовая программа.

После окончания трансляции выдается звуковой сигнал.

ЗАПУСК ПРОГРАММ

Для запуска программы введена команда RU. При отдаче этой команды система производит следующие действия:

1. Система выставляет адрес трансляции 1000.

2. Система выставляет режим трансляции в формате получения загрузочного модуля.

3. Производится трансляция загруженной в память программы, и, если при трансляции не было обнаружено ошибок, то прог-

рамма запускается на исполнение, в противном случае выдается список обнаруженных ошибок и запуска не происходит.

Однако, следует иметь в виду, что для возврата в систему программа должна заканчиваться либо RTS PC, либо RETURN, либо HALT. Также возврат в систему производится клавишей СТОП.

Также можно после трансляции командой CO выйти в монитор и поработать с полученной программой. Она находится с адреса 1000. Нажав СТОП вы обратно попадете в TURBO. Текст при этом сохраняется, если программа короче 6776.

ОШИБКИ ПРИ ТРАНСЛЯЦИИ

~~~~~~~~~~~~~~~~~~~~

100 - Нет места для меток.
101 - Псевдокоманда .LA должна быть первой в тексте.
102 - Ошибка длины или направления перехода в команде SOB.
103 - Недопустимый символ в строке.
104 - Начальное присваивание не перед началом программы.
105 - Ошибка или отсутствие числового аргумента.
106 - Неправильная псевдокоманда.
107 - Неправильная команда.
110 - Ошибка длины перехода по оператору ветвления.
111 - Недопустимое использование имени метки.
112 - Ошибка аргумента MARK, TRAP, EMT.
113 - Ошибка в имени регистра.
114 - Ошибка в псевдокоманде.
115 - Ошибка в команде.
116 - Метка уже определена ранее.
117 - Аргумент .BLKW или .BLKB слишком велик.
120 - Нечетный адрес команды.

Появление ошибки 100 практически невозможно из-за динамического перераспределения памяти между программой и текстом.

При трансляции текста по командам CO и CL выдается номер ошибки, адрес программы, в котором допущена ошибка, номер строки текста, в которой обнаружена ошибка и сама строка, при этом список меток не выдается. Его можно посмотреть, дав команду TT.

При трансляции по команде CE после появления ошибки трансляция прекращается и на ошибочную строку ставится метка. Можно войти в редактор и по команде BC+ъ перейти к ошибочной строке.

ПРИМЕЧАНИЕ: при появлении ошибки 110 номер строки может не соответствовать адресу, так как ошибка может быть обнаружена не сразу. Правильным является адрес.

При связке объектных модулей командой LI проверяется только длина перехода по командам ветвления, (кроме SOB), и следить за тем, чтобы метки не повторялись, надо программисту. При этом выдается номер ошибки и адрес в полученной программе, так как текста в этом случае нет.

Кроме того, при выводе по SL файлы снабжаются контрольным кодом (52525), что исключает ввод иных файлов, при этом, если система при считывании файла по команде LI не обнаружила контрольный код, то выдается сообщение - 'Файл не OBJ'.
~~~~~~~~~~~~~~~~~~~~

Автор приносит свою благодарность:

А. Надежину: за ANDOS, TURBO4 и еще много прекрасных программ.

А. Надежину: за терпение, с которым он давал советы начинающему программисту-любителю.

А. Надежину: за подарок - TURBO6M.

С. Камневу: за великолепную сервисную оболочку DiskMASTER и ряд полезных советов.

В. Ретуновскому & А. Суханову: за TRACER.

С. Клименкову: за PARADISE.

М. Королеву: за DESS, READER14 и MKDOS, с которой и начал автор работу с БК.

Д. Бутырскому: за TURBO7MK, который хоть и не был в числе предшественников, но из которого была заимствована одна идея.

Д. Романову: за систему VorteX, в которой было написано данное описание, и с которой вообще автор провел много приятных минут.

В. Савину: за учебную программу DEMO к MicroSW.

П. Суходольскому: за АОН вообще и РУСЬ14 в частности.

В. Тукову, благодаря которому автор познакомился с БК.

И, конечно, В. Коренкову, "отцу-основателю" серии ассемблеров TURBO, и авторам TURBO4, TURBO5, TURBO6, без которых этой работы не было бы вообще.

Хотелось бы также поблагодарить своего кота Василия за моральную поддержку (не очень мешал).

Крылов Дмитрий Константинович 398-83-12
Москва, сентябрь 1995г - март 1996г.

Т.к. TURBO8 и TURBO8.10 работают только с ANDOS, было логично использовать для печати системный драйвер принтера ANDOS. Что я и сделал. На принтер в обеих программах передается строка инициализации принтера.

С TURBO8.10 все понятно, а по поводу TURBO8 поясню:- TURBO8 загружается на место, в том числе занимаемое драйвером принтера. Поэтому при запуске программы, драйвер печати ANDOS вместе с нужными и не нужными ему подпрограммами перегоняется в 6ю страницу ОЗУ, на свой родной адрес. Где и работает.

С наилучшими пожеланиями, Вадим Смирнов.

Верстка в PDF MAXIOL 2008