

ИНСТИТУТ МАТЕМАТИКИ И МЕХАНИКИ
УРАЛЬСКОГО НАУЧНОГО ЦЕНТРА АН СССР

Корякин В.К.

СИМВОЛЬНЫЙ ОТЛАДЧИК ОС ДИАПАК
Руководство пользователя

г. Свердловск
1979 г.

Символьный Отладчик - это система, предназначенная для отладки программ с терминала в режиме диалога с ЭВМ. Она разработана применительно к операционной системе ДИАШАК/БЭСМ-6 и соответствующей версии мониторной системы ДУБНА [1]. Для программ, получаемых с помощью ассемблера MADLEN [2] и компиляторов с Фортрана [3] (ФОРТРАН-ДУБНА и BESM-FORTRAN) система обеспечивает отладку с использованием символических средств этих языков. Кроме того возможна отладка в терминах внутреннего языка БЭСМ-6. По сравнению с Отладчиком, существовавшим ранее [4], в новом значительно расширен набор операций, предоставляющих дополнительные возможности для отладки.

Отладчик рассчитан на одновременное обслуживание семи пользователей, использующих в качестве терминалов дисплеи типа VIDEOTON-340.

ОГЛАВЛЕНИЕ

I. Введение.....	4
2. Общая характеристика и структура системы.....	4
3. Установка связи с Отладчиком.....	7
4. Основные элементы языка отладки. Обозначения.....	7
5. Установка/снятие контрольных точек.....	13
5.1. Основные операции.....	13
5.2. Операции трассировки.....	16
5.3. Операции NAME и MC.....	18
5.4. Сообщения о прерывании, выводимые на терминал....	19
6. Возобновление исполнения программы после прерывания...	20
7. Команда STOP.....	21
8. Команды вывода и присваивания.....	21
8.1. Вывод.....	22
8.2. Команды присваивания.....	23
9. Группа команд для работы с выходным файлом задачи.....	25
10. Выполнение нестандартных действий.....	26
11. Завершение сеанса отладки.....	27
12. Вспомогательные команды.....	27
13. Дополнительные возможности языка отладки.....	28
13.1. Вызов подпрограммы.....	28
13.2. Макрокоманды.....	29
14. Диагностические сообщения.....	32
15. Доступ к файлам системы управления данными.....	34
16. Заключение.....	34
17. Литература.....	35
Приложение. Список команд Отладчика с кратким пояснением.....	36

I. ВВЕДЕНИЕ

В общем развитии вычислительной техники и программирования можно выделить два явления, оказавших наибольшее влияние на процесс отладки программ. Это, с одной стороны, широкое распространение языков программирования, применение которых сократило как число ошибок, содержащихся в программе к моменту начала её отладки, так и время самой отладки. С другой стороны, развитие систем разделения времени позволило организовать общение человека с ЭЕМ в режиме диалога. Взаимодействие с машиной в процессе отладки, возможность оперативного вмешательства в работу программы значительно повысили эффективность труда программиста. Естественное сочетание этих двух направлений привело к появлению интерактивных систем для отладки в терминах исходного языка программирования, систем, объединяющих в себе преимущества непосредственного доступа к ЭЕМ через терминал и удобства работы на входном языке отлаживаемой программы. Символьный Отладчик, о котором пойдет речь в дальнейшем, представляет собой попытку создания такого средства отладки в ОС ДИАПАК. Его предшественником является эксплуатировавшийся более трех лет кодовый отладчик, обеспечивающий работу на внутреннем языке БЭСМ-6 [4].

2. ОБЩАЯ ХАРАКТЕРИСТИКА И СТРУКТУРА СИСТЕМЫ

Описываемая версия Символьного Отладчика предусматривает работу с программами, написанными в рамках мониторной системы ДУБА [1] на Фортране и автокоде MADLEN; допускается также отладка на уровне внутреннего языка БЭСМ-6. Взаимодействие пользователя с системой ведется в режиме диалога (для этого обычно используется дисплей типа VIDEOTON-340). Его применение не требует никаких изменений в отлаживаемой программе и в процессе отладки не нарушается обычное для задачи распределение памяти.

Чтобы обеспечить символьную отладку, оба входящих в состав мониторной системы компилятора с Фортрана (ФОРТРАН-ДУБА, BESM-FORTRAN) и ассемблер MADLEN были дополнены блоками, которые на основе информации, имеющейся к моменту окончания трансляции, формируют таблицу символьных имен данной подпрограммы. Эта таблица содержит относительные адреса всех внутренних объектов подпрограмм и их полное описание (тип переменной, число индексов массива и их максимальные значения и т.д.).

Собственно Отладчик, являющийся совокупностью блоков операционной системы, можно представить состоящим из трех компонент:

- анализатор прерываний,
- блок связи,
- блоки выполнения операций отладки.

Анализатор прерываний, работающий в режиме супервизора, кроме действий, непосредственно следующих из его названия, обеспечивает выполнение команд в позициях прерываний, которые реализуются в системе аппаратными средствами. Блок связи, обеспечивающий обмен информацией с терминалом, и блоки выполнения операций объединены в один модуль. Этот модуль работает как программа математика, имеющая высший приоритет, а его взаимодействие с задачами пользователя и анализатором прерываний обеспечивается непосредственным доступом к некоторым управляющим таблицам операционной системы и специально разработанными экстракодами. На рис. 1 приведена структура системы и движение данных во время отладки.

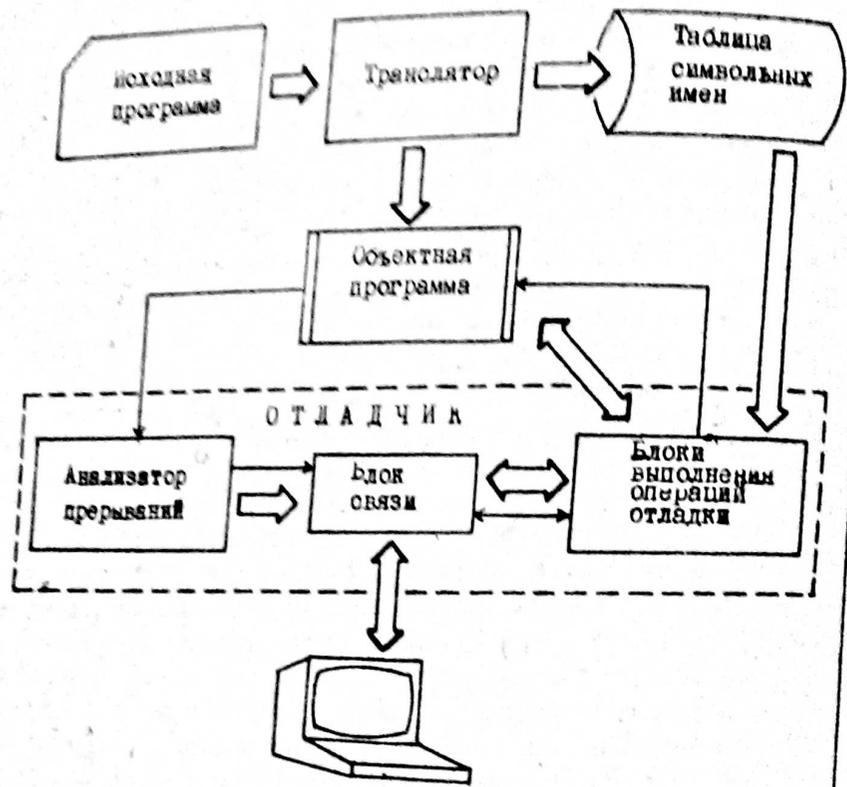


Рис. 1. Структура системы.

3. УСТАНОВКА СВЯЗИ С ОТЛАДЧИКОМ

Чтобы воспользоваться услугами Отладчика достаточно ответить в оперативной памяти задачи два соседних слова; адрес первого из них А указывается в специальном разделе паспорта

ТЭЛС А

Таблицы, необходимые для символической отладки в мониторингной системе ДУБА, формируются при наличии среди управляющих карт задания условий трансляции карты *DEBUG. Действие этой карты распространяется на все транслируемые за ней программные единицы до карты *NODEBUG или *EXECUTE; число карт *DEBUG и *NODEBUG в пакете не ограничено. Для каждой подпрограммы организуется своя таблица и все она размещается на магнитных барабанах 7-го логического направления (задача поэтому должна располагать небольшим резервом трактов).

Непосредственная связь с Отладчиком устанавливается при включении задачи в решение после выполнения на терминале директивы

ШИР XX...X,

где XX...X - цифры соответствующего раздела паспорта задачи; если последние 6 цифр нули, то их можно не набирать. С готовности системы к работе свидетельствует появление на экране дисплея текста:

```

ОТЛАДЧИК ОС ДИАПАК
дата                время
фамилия, шифр задачи пользователя

```

4. ОСНОВНЫЕ ЭЛЕМЕНТЫ ЯЗЫКА ОТЛАДКИ. ОБОЗНАЧЕНИЯ

Сообщение, содержащее команды языка отладки, можно вводить только после получения разрешения от системы, которое выражается в выдаче на терминал знака равенства. В одном сообщении

может содержаться несколько команд, разделенных символом "LINE FEED", - они будут выполнены последовательно. Совокупность команд отладки мы будем называть отладочным действием. Сообщение заканчивается нажатием клавиши **ESC** или одновременным нажатием клавиш **CTRL** и **T**; в последнем случае курсор останется в той же строке. Длина сообщения не более 78 символов.

Команды отладки в общем случае состоят из названия операции и её атрибутов. Последние указывают объекты программ, к которым применяется операция, и/или содержат информацию, отражающую особенности её исполнения. Название можно сокращать, оставляя только необходимые для однозначного определения операции первые буквы.

Основными элементами языка отладки являются адрес объекта и его имя, которые определяют слово в оперативной памяти.

Адрес объекта может быть задан четырьмя способами.

1. В простейшем случае обращение к объекту происходит непосредственно по его абсолютному адресу, представляющему собой целое восьмеричное число a , $0 \leq a \leq 77777_8$.

2. Специальной командой может быть определен некоторый базовый адрес b (см. п. 12), после чего все адреса, записанные как абсолютные, считаются смещениями относительно b , т.е.

$$\text{адрес объекта} = (b + a) \text{ mod } 2^{15}$$

3. Запись вида $a(i)$ означает, что

$$\text{адрес объекта} = (a + \langle i \rangle) \text{ mod } 2^{15}$$

где $\langle i \rangle$ - содержимое индекс-регистра с восьмеричным номером i (суммирование производится без учета базы b).

4. И, наконец, последний способ задания адресов, предназначенный для отладки в мониторной системе:

$\langle \text{полный адрес} \rangle ::= \langle a \rangle . \langle \text{имя базы} \rangle | \langle a \rangle . | . \langle \text{имя базы} \rangle$
 $\langle \text{имя базы} \rangle ::= \langle \text{имя подпрограммы} \rangle |$

$\star \langle \text{идентификатор COMMON-блока} \rangle \star$

где a - смещение, определяющее объект относительно начала подпрограммы или общего блока $\langle \text{имя базы} \rangle$.

Адрес объекта есть сумма смещения (относительного адреса) a и базы, роль которой в данном случае играет адрес загрузки

соответствующей подпрограммы или общего блока. Имена дополнительных входов, определенных оператором ENTRY, можно использовать наравне с именами подпрограмм. Если имя базы отсутствует, то базирование осуществляется последним применявшимся для этого адресом. В конструкции, содержащей только имя базы, смещение a полагается равным нулю.

Примеры:

- I25 (?) - если $\langle \text{ИР } \# 7 \rangle = 7777_6$, то определен адрес I23;
 0 (I6) - слово с адресом, равным содержимому ИР $\# 14$;
 7.PROGRAM - 7-е слово головной программы;
 .RES - адрес загрузки COMMON-блока RES;
 43. - элемент того же блока с относительным адресом 43.

$\langle \text{имя объекта} \rangle ::= \langle \text{простое имя} \rangle | \langle \text{полное имя} \rangle$
 $\langle \text{простое имя} \rangle ::= \# \langle n \rangle | \star \langle l \rangle | \langle \text{идентификатор} \rangle$

где n - номер оператора (целое десятичное);

l - метка оператора;

символы $\#$ и \star - признаки номера и метки оператора соответственно;

идентификатор - имя переменной или метка в смысле

MADLEN-a; если переменная является элементом массива, то индексы указываются после имени массива в квадратных скобках; индексами могут быть целые без знака и определенные в программе простые целые переменные; использование выражений запрещается; в остальных правила обращения к элементам массива совпадают с принятыми в Фортране [3].

$\langle \text{полное имя} \rangle ::= \langle \text{простое имя} \rangle . \langle \text{имя подпрограммы} \rangle$

Если имя подпрограммы отсутствует, то считается, что объект находится в текущей подпрограмме, т.е. в подпрограмме, имя которой было указано последним или на которой произошло последнее прерывание (останов).

Примеры:

- $\#48.SUBR$ - оператор номер 48 из подпрограммы SUBR;
 $\star I0$ - оператор с меткой I0 из текущей подпрограммы;
 $AR[3,N].SUMM$ - элемент массива AR из подпрограммы SUMM.

Обращение к объекту программы по его имени возможно только в режиме МС - специальном режиме работы Отладчика, обеспечивающем символьную отладку в мониторной системе ДУБНА. Этот режим устанавливается операциями МС и NAME (см. п. 5.3), но автоматически отменяется, когда обнаруживается, что задача не располагает таблицами символьных имен. Определение объектов программ с помощью их адресов разрешено всегда.

Для более лаконичного описания формата команд Отладчика, введем переменную α , которую будем называть обобщенным адресом:

$\langle \alpha \rangle ::= \langle \text{адрес объекта} \rangle | \langle \text{имя объекта} \rangle | \langle \text{имя объекта} \rangle \pm \langle \delta \rangle$
 Последняя конструкция означает, что позиция в программе может быть определена положительным или отрицательным смещением δ относительно объекта, заданного своим именем.
 $\langle \delta \rangle ::= \langle \text{целое десятичное} \rangle | \langle \text{целое восьмеричное} \rangle \cdot (I)$
 $\delta \in \{2767\}$.

Пример:

пусть переменная T5 имеет в подпрограмме ТЕСТ относительный адрес I27; тогда запись T5.ТЕСТ+2 определяет адрес I31;
 T5-25I - 76.

Правила записи констант в Отладчике довольно простые:

$\langle \text{константа} \rangle ::= S \langle \text{тело константы} \rangle | \langle \text{тело константы} \rangle$
 где S - спецификация типа.

Допустимы пять типов констант (в скобках указаны соответствующие спецификации):

- восьмеричные (C, O, B);
- целые (I);
- вещественные (E, D, F);
- текстовые (B, H, T);
- командные (K).

Восьмеричные константы состоят не более чем из 16 цифр, которые размещаются в машинном слове справа, а свободное место заполняется нулями. Перед отрицательной константой ставится знак минус, и она представляется в дополнительном коде. Можно использовать любую спецификацию или вообще обходиться без нее.

Примеры (справа дается полная запись константы):

C377770	0000 0000 0037 7770
B-52	7777 7777, 7777 7726
I00	0000 0000 0000 0100

Константы типа I являются целыми десятичными числами, состоящими не более чем из 11 цифр. Спецификация I может быть опущена, если в теле константы присутствуют невосемьзначные цифры (сказанное относится и к записи смещения δ).

Примеры: I526, I-40, 7948.

Форма представления целых констант в машинных словах совпадает с принятой в мониторной системе.

Тело констант вещественного типа имеет в общем случае следующий вид: $\pm m.n E \pm p$, где m - целая часть, n - дробная часть, p - показатель степени числа 10. Спецификацию типа можно не указывать, но обязательной является десятичная точка или буква E.

Примеры: 20.4, .3597, -2.9E-3, I2E-I.

Спецификации B, H, T определяют один из трех кодов, применяемых при образовании текстовых констант: ASCII-128 (GOST), ISO, TEXT соответственно. Два первых кода являются восьмиразрядными, поэтому слово содержит 6 символов, константы типа TEXT состоят из 8 символов в шестиразрядной кодировке (таблицы кодов можно найти в [3]). Символы констант типа H и T размещаются в машинном слове начиная со старших разрядов (кадрируются влево), и если тело константы содержит менее 6(8) символов, свободное место слева заполняется пробелами в соответствующем коде. Константы типа GOST кадрируются вправо и дополняются нулями. При записи текстовые константы ограничиваются слева спецификацией типа, справа - символом EIX или "E" (последний генерируется при одновременном нажатии клавиши CTRL и E).

Примеры:

константа B.ABC будет представлена в виде 00.ABC, H.DUBNA - DUBNA, T.MADJEN - MADJEN.

Командные константы состоят из 18 или 9 цифр, образующих команды БЭСМ-6. Если задано 9 цифр, то они помещаются в левую половину слова, а в правую записывается команда 00 22 00000;

между цифрами константы допустимы пробелы.

Примеры:

K 012577777 01 010 2000 ;

константе K00 0360110 соответствует команда

00 036 0110 00 22 00000.

За названиями аппаратных регистров БЭСМ-6 закреплены следующие обозначения:

СМ - сумматор ;

РМР - регистр младших разрядов ;

РК - регистр признаков и режимов работы арифметического устройства ;

ИРi - индекс-регистр i (i - восьмеричное).

В описании формата команд Отладчика используются следующие общепринятые соглашения :

- элементы или текст, заключенные в квадратные скобки, необязательны ;
- если элементы записаны друг под другом в фигурных скобках, то один из них (любой) должен быть обязательно указан ; если используются квадратные скобки, то можно ничего не указывать или указать один ;
- слова, напечатанные малыми буквами, обозначают переменные и при наборе операции заменяются конкретными значениями ;
- большие буквы и специальные символы (разделители) должны использоваться так, как указано в описании формата команды ; чаще всего разделителем служит пробел, необходимость которого в том или ином месте текста будет подчеркиваться его изображением (_) ; когда не оговорено противное, допускается использование пробелов между отдельными частями команды - они в этом случае игнорируются ;
- многоточие означает допустимость нескольких однородных элементов.

5. УСТАНОВКА/СНЯТИЕ КОНТРОЛЬНЫХ ТОЧЕК

Контрольная точка есть определенное слово в оперативной памяти, при обращении к которому прерывается исполнение программы. Для установки контрольных точек в Отладчике используются аппаратные средства, а обращение к слову означает: выполнение находящейся в нем команды, запись в это слово или выборку (считывание) из него. Обращению за командой соответствует контрольная точка, называемая узлом. Узлы служат для явного определения позиций в программе, в которых необходимо прерывание (останов) - оно возникает непосредственно перед выполнением расположенной в узле команды. Прерывание по записи/выборке происходит при попытке изменить или использовать в качестве аргумента значение некоторой переменной, объявленной контрольной точкой, - последняя, таким образом, определяет позицию прерывания неявно.

5.1. Основные операции

Узлы могут быть установлены несколькими операциями, но в этом разделе описываются только две из них, использующие разные способы получения прерываний. Прерывание в контрольной точке, установленной операцией УЗЕЛ, обеспечивается специальным преобразованием находящейся в узле команды. Поэтому прерывание не произойдет, если содержимое узла изменилось после его установки (например, в задачах, работающих в МС ДУБНА и использующих динамическую сегментацию программ, автоматически снимаются узлы, расположенные в области памяти, подверженной изменениям при загрузке новых разделов). Этот недостаток отсутствует у контрольных точек, устанавливаемых операцией КРА, т.к. в этом случае прерывание происходит при совпадении адреса очередной команды с адресом узла, который запоминается в специальном регистре. К сожалению, аппаратура позволяет определить только один такой узел. Число контрольных точек, устанавливаемых операцией УЗЕЛ, не может превышать 61.

Прерывания при обращении к контрольным точкам по записи/выборке - их установка осуществляется операциями ЗП и СЧ - обеспечиваются другим специальным регистром, общим для обеих операций. Поэтому допускается существование лишь одной контрольной точки этого вида - как и в операции КРА, вновь определяемая контрольная точка заменяет установленную ранее.

Форматы основных команд установки/снятия контрольных точек (в качестве обобщенного адреса α здесь нельзя использовать конструкцию $a(i)$ (см. п. 2)):

Установить узел в α $\left\{ \begin{array}{l} U[ZEI] \\ BR[EAK] \end{array} \right\} _ \alpha [(\text{описание})] _ \dots$

$K[PA] _ \alpha [(\text{описание})]$

Прервать исполнение программы при записи в α $\left\{ \begin{array}{l} Z[P] \\ W[RITE] \end{array} \right\} _ \alpha [(\text{описание})]$

Прервать исполнение программы при выборке из α $\left\{ \begin{array}{l} CY \\ R[EAD] \end{array} \right\} _ \alpha [(\text{описание})]$

Снять узлы (кроме установленного операцией КРА) $\left\{ \begin{array}{l} SN[ЯТЬ] \\ KI[LL] \end{array} \right\} _ \alpha _ \dots$

Отсутствие операнда в командах КРА, ЗП/СЧ означает снятие соответствующих контрольных точек, в команде СНЯТЬ - всех узлов, установленных операциями УЗЕЛ.

Описание контрольной точки состоит из 2-х частей: первая определяет тип контрольной точки, во второй могут быть перечислены отладочные действия, которые необходимо выполнить в случае прерывания, соответствующего данной контрольной точке (см. п. 13.2). Здесь рассматривается только первая часть - описание типа.

Существуют контрольные точки трех типов: простые, кратные, условные.

Если описание типа не задано, то устанавливается простая контрольная точка, первое же обращение к которой вызывает прерывание и её автоматическое снятие.

Целое положительное число K , указанное в описании определя-

ет кратную контрольную точку. Обычно кратность K задает число прерываний, которые происходят при каждом обращении к данной контрольной точке; последняя автоматически снимается после K -го прерывания. Но если кратность сопровождается знаком равенства ($K=$), то прерывания возникают через каждые K обращений к контрольной точке, снять которую можно только соответствующей операцией. Записывается кратность K аналогично смещению δ (см. (1)).

Обращение к условной контрольной точке вызывает прерывание только при выполнении указанного в её описании условия, т.е. только в том случае, когда истинно выражение отношения:

$$\left\{ \begin{array}{l} \text{имя объекта } [i\delta] \\ \text{/адрес объекта} \\ \text{/аппаратный регистр} \end{array} \right\} \text{ } \nu \left\{ \begin{array}{l} \text{константа} \\ \text{/}d \end{array} \right\}$$

где ν - знак отношения:

$$\nu ::= 'EQ' | 'NE' | 'GT' | 'GE' | 'LT' | 'LE' | '=' | '\neq' | '>' | '<'$$

(в кавчки заключаются знаки отношения, принятые в Фортране [3]). Условная контрольная точка автоматически снимается после первого вызванного ею прерывания; но и здесь, если описание типа заканчивается знаком равенства, автоматическое снятие контрольной точки не происходит и до её отмены с терминала может возникнуть несколько прерываний (но каждый раз при выполнении соответствующего условия).

П р и м е р ы :

УЗЕЛ $\#27.SUMM(64I) \#29 \#45$

установить узлы на операторах 27 (с кратностью 64), 29,45 (простые) подпрограммы *SUMM*;

$K \#702(24=)$

через каждые 20 проходов через оператор с меткой 702 прерывать исполнение программы ;

WRITE A.SUMM(B.SUBR 'GT'.5)

когда значение переменной *B* из подпрограммы *SUBR* станет больше 0.5, остановиться при изменении значения переменной *A* из *SUMM*.

BR #65.SUBR(L<K=)

прерывание на операторе 65 подпрограммы **SUBR** необходимо всякий раз, когда значение переменной **L** меньше значения переменной **K** (обе переменные из **SUBR**);

У 12.(/CM=0)

прервать исполнение в 12-ом слове текущей подпрограммы, если сумматор нулевой;

СНЯТЬ T5.ТЕСТ T5-5 T5+7

снимаются узлы в написанной на **MADLENe** подпрограмме ТЕСТ, последние узлы определены отрицательным и положительным смещением относительно метки T5.

5.2. Операции трассировки

Трассировкой обычно называют пошаговое прослеживание работы программы, выполняемое автоматически с помощью специальных отладочных средств. В Отладчике трассировка обеспечивается установкой контрольных точек (узлов) в те позиции программы, которые соответствуют некоторому очередному шагу её исполнения.

а) трассировка программ, написанных на Фортране, может выполняться двумя операциями:

- TR[ACE]** - устанавливает узлы на всех операторах, чем достигается пооператорное исполнение;
- TR[ACE], LABEL** - узлы устанавливаются только на имеющих метку операторах.

Параметры, определяющие область трассировки, не следуют, как обычно, за названием операции, а вводятся пользователем после появления на экране дисплея приглашений, указывающих какой именно параметр должен быть введен. Тексты приглашений заканчиваются знаком равенства.

NAME = [subname]

subname - имя подпрограммы; если оно не указано, то операция применяется к текущей подпрограмме; если подпрограмма

написана не на Фортране, выдается сообщение об ошибке.

**BGN = [n₁
N]**

**END = [n₂
N]**

n₁, n₂ - номера операторов, определяющие начало и конец диапазона трассировки; если они отсутствуют, то полагается **n₁ = 1, n₂ = N** - номер последнего оператора подпрограммы; непосредственное выполнение операции, т.е. установка узлов, начинается после ввода последнего параметра; трассировка не осуществляется, если вместо номера оператора указать символ "N"; при обнаружении ошибки в задании параметра, он запрашивается вновь.

Отмена трассировки в некотором диапазоне выполняется аналогично работающей операцией

NO[TRACE][LABEL]

Выборочное снятие узлов можно осуществлять операцией СНЯТЬ.

П р и м е ч а н и е .

Следует учитывать, что последовательность команд, реализующих оператор Фортрана, может начинаться с правой команды слова, но Отладчик считает началом такого оператора левую команду следующего слова.

б) Для трассировки программ на уровне внутреннего языка БЭСМ-6 служит команда

**{ L[INH]
L[INE] } [α]**

которая возобновляет исполнение задачи с позиции **α** или, если операнда нет, с прерванного места, и организует прерывание на первом же команде передачи управления (прерывание обеспечивается установкой узла в слово, содержащее эту команду).

Поиск очередной команды передачи управления осуществляется с учетом следующих признаков:

ПВ - не устанавливать узлы на командах З1 (безусловный переход с запоминанием адреса возврата) ;
 ЦКЛ - не устанавливать узлы на командах З7 (конец цикла).
 Эти признаки формируются одноименными операциями - ПВ и Ц[ИКС] сложением с I по модулю 2. При вызове Отладчика они полагаются равными нулю и операция ЛИН устанавливает узлы на всех без исключения командах передачи управления.

5.3. Операции NAME и MC

$N[AME] \rightarrow subname$

Эта команда продолжает задачу с прерванного места и с помощью специальным образом установленных контрольных точек обеспечивает прерывание перед обращением к подпрограмме *subname*.
 В случае, когда используется динамическая сегментация и подпрограмма *subname* загружается в составе некоторого раздела оператором *CALL LOADGO (... "overlay")*, для выхода на *subname* необходимо сначала выполнить команду *NAME overlay* и только после прерывания на *overlay* выполнить команду *NAME subname*. Если подпрограмма уже располагается в оперативной памяти, то в команде могут быть указаны имена её дополнительных входов. Кроме того, если режим MC не установлен, он устанавливается описываемой операцией, т.е. в дальнейшем допускается использование символьных имен программных объектов.

Операция *NAME* имеет следующие особенности:

- в процессе её выполнения отменяются контрольные точки, ранее установленные операциями КРА и ЗП/СЧ ;
- и наоборот этими операциями не следует пользоваться до завершения выполнения *NAME*, т.е. до получения прерывания на нужной подпрограмме - в противном случае оно не гарантируется ;
- новая команда *NAME*, введенная до завершения предыдущей, отменяет её - прерывание возникнет при обращении к подпрограмме, указанной последней.

Операцией *NAME* удобно пользоваться в начале сеанса отладки, когда бывает необходимо дождаться окончания этапов

трансляции и начальной загрузки и после выхода на счет получить прерывание на некоторой подпрограмме. Если требуется остановиться на головной программе, определенной оператором *PROGRAM*, можно использовать операцию MC, которая интерпретируется как *NAME PROGRAM*, если выполняется до начала исполнения задачи пользователя. В остальных случаях эта операция лишь устанавливает или снимает режим MC (соответствующий признак, как и все остальные, формируется сложением с единицей по модулю 2); при вызове Отладчика режим MC не установлен.

5.4. Сообщения о прерывании, выводимые на терминал

Ниже приводятся варианты сообщений о прерывании, вызванном обращением к контрольной точке или попыткой аварийного снятия задачи пользователя. Используются следующие обозначения:

аааа - абсолютный адрес прерывания ;

сссс - относительный адрес (смещение) ;

имя - имя подпрограммы, на которой произошло прерывание.

1. Обычный режим: аааа причина прерывания

2. Обычный режим в мониторной системе:
 аааа имя+сссс причина прерывания

3. Режим MC: аааа имя.№ оператора±сссс причина прерывания
 или метка

(если подпрограмма не имеет информационной таблицы, сообщение выдается в виде 2)

Команда $V[T]$ традиционным для Отладчика образом устанавливает или отменяет режим, в котором в сообщении о прерывании включается дополнительная информация:

1-я строка: команда, содержащаяся в аааа и содержимое SM в виде вещественной и восьмеричной констант ;

2-я строка: содержимое ИР1+ИР7 и РМР ;

3-я строка: содержимое ИР10+ИР17 и РК.

После выдачи сообщения о прерывании система переходит в состояние ожидания ввода команд отладки.

П р и м е ч а н и е.

Задача, ведущая обмен с терминалом, не может быть прервана, пока этот обмен не завершится (см. [5]); в противном случае задача аварийно завершается и связь с Отладчиком прекращается.

6. ВОЗОБНОВЛЕНИЕ ИСПОЛНЕНИЯ ПРОГРАММЫ ПОСЛЕ ПРЕРЫВАНИЯ

$$\left\{ \begin{array}{l} И[ДИ] \\ Г[ОТО] \end{array} \right\} \left[\begin{array}{l} \alpha[П] \\ ОП \end{array} \right]$$

Исполнение программы начинается с позиции α , а при отсутствии операнда - с прерванного места. Признак "П" указывается для передачи управления на правую половину командного слова; если это то слово, на котором произошло прерывание, признаку "П" должен предшествовать ноль.

После возникновения следующих аварийных ситуаций, носящих фатальный характер, дальнейшее исполнение программы невозможно:

ошибка МБ БЭСМ-6,
ошибка МД БЭСМ-6,
ошибка МД-МБ БЭСМ-6,
ошибка МД,
дай метры АЦПУ,
истекло время,
истекло время по экстракоду,
сбой; введи по инструкции.

Все сказанное здесь относится и к операции ЛИИ, рассмотренной в п.5.2.

П р и м е ч а н и е.

Если пользователь корректировал команду в слове α , на котором произошло прерывание, то ИДИ(ЕТХ) не всегда обеспечивает исполнение команд из α в новом виде; для этого лучше использовать ИЛИ α .

7. КОМАНДА СТОП

Во время исполнения задачи пользователя не может быть введена ни одна команда отладки. Единственным исключением из этого правила является команда СТО[П], которая используется при необходимости немедленно прервать исполнение задачи, не дожидаясь обращения к контрольной точке или авоста.

Реализация команды СТОП такова, что сам акт прерывания и его обработка разделены небольшим промежутком времени (обычно не более 5 секунд). Поэтому при успешном выполнении команды пользователь получает сначала сообщение "ИДИ" и лишь через несколько секунд после этого выдается информация о месте прерывания.

Следующие ответы системы указывают на причину, по которой выполнение операции СТОП в данный момент невозможно:

ОБМЕН - задача производит обмен данными с внешними запоминающими устройствами;

АРХИВ - идет обработка и выполнение процедур системы управления данными;

В ПУЛ - сформирована, но еще не завершилась задача вталкивания файла пользователя в оперативный пул системы АРХИВ;

ПР.МЛ (проверь МЛ) - не проходит обмен с магнитной лентой, но аварийное снятие задачи преждевременно, т.к. возможно, что причина сбоя легко устраняется; система ожидает соответствующих действий обслуживающего персонала, после чего повторит попытку обмена.

8. КОМАНДЫ ВЫВОДА И ПРИСВАИВАНИЯ

Прежде всего заметим, что описываемые здесь команды позволяют обращаться не только к отдельным словам, но и к массивам. Последние определяются начальным адресом и длиной ($\alpha + l$) или диапазоном адресов ($\alpha_1 - \alpha_2$). Введем теперь переменную β , которую будем использовать в этом разделе вместо конструкции

$$\begin{cases} d \\ d+l \\ d_1-d_2 \end{cases}$$

Длина l задается аналогично смещению δ из определения обобщенного адреса (I), поэтому δ не может использоваться в записи аргумента команд вывода/присваивания. Максимальная длина массива - 1024 слова, а вычисление его границ производится по следующим правилам:

$$d+l: [d, d+l'], \text{ где } l' = l \bmod 1024;$$

$$d_1-d_2: [d_1', d_2'], \text{ где } d_1' = \min(d_1, d_2);$$

$$d_2' = d_1' + |d_2 - d_1| \bmod 1024$$

(соответствующие примеры см. в п.8.1).

8.1. Вывод

Формат команд вывода содержимого памяти на терминал

$$[I]\beta = [S]$$

и АЦПУ

$$\left\{ \begin{array}{l} П [СЧЕТЬ] \\ Д [УМР] \end{array} \right\} \leftarrow \beta [I, S]$$

где S - спецификация формата, определяющая форму представления выводимой информации; спецификациями формата служат спецификации типа констант, рассмотренные в п.2. Если S не указана, то, в случае использования символьных имен, формат вывода определяется описанием переменной в программе. При выдаче состояния аппаратных регистров (см. ниже) и памяти, заданной адресами, вся информация по умолчанию считается восьмеричной. Косая черта (слэш) набирается в тех случаях, когда может возникнуть путаница из-за совпадения имени переменной с названием одной из операций Отладчика.

Данные на терминал выводятся кадрами; очередной кадр может быть получен нажатием клавиши $\langle \text{F1} \rangle$; если же массив выдан полностью, нажатие этой клавиши приводит к выдаче следующего слова в том же формате.

Вывод состояния аппаратных регистров на терминал осуществляется командами

CM = [S]; Pa[r] = [S]; PK = ; IR[i] = ;
если номер l не указан, выдается содержимое всех 15-ти индекс-регистров.

Примеры:

DUMP A[K].SUMM+5

распечатать значения шести элементов массива A подпрограммы SUMM, начиная с K-го;

B+1025I=N

в коде ISO будет выдано содержимое двух слов - B и B+I из текущей подпрограммы;

J.SUBR - J[N]=

выдать на экран дисплея значения N+1 первых элементов целочисленного массива J из SUBR (если машинное представление целой переменной не соответствует принятому в MC способу (см. §40 [3]), то при выводе ей предшествует предупреждение

"NOINT");

ПЕЧ I1077-1000,F

выдать на АЦПУ десятичные числа, расположенные в словах 1000+1077; по спецификации F действительные числа с порядком $|p| \leq 7$ выдаются в их обычной записи - с десятичной точкой без порядка, при $|p| > 7$ вывод по спецификации F совпадает с выводом по спецификациям E и D, когда числа представляются на экране в виде $\pm m \pm p$, где m - мантисса, p - порядок.

8.2. Команды присваивания

После выполнения команды

$$[I]\beta := \left\{ \begin{array}{l} \text{константа} \\ \alpha \end{array} \right\} \leftarrow \dots [I] \dots$$

переменной или каждому элементу массива β будет присвоено значение константы или переменной α . Многоточие указывает

на допустимость нескольких операндов - в этом случае указанные величины последовательно записываются в слова: β , $\beta+1$, $\beta+2$ и т.д.; запятая означает, что содержимое очередного слова остается без изменений. Если β - массив с границами β_1 , β_2 , то первая после знака присваивания величина записывается в слова $\beta_1 \div \beta_2$, а остальные размещаются, начиная с β_2+1 .

При записи в память длина тела текстовых констант может превышать длину одного слова БЭСМ-6 - информация вводится в последовательно расположенные слова и кадрирование в последнем занимаемом слове происходит влево с заполнением свободного места пробелами (для всех спецификаций).

Введены еще две спецификации (КП и КЛ) для командных констант, заменяющих соответственно правую и левую половину слова; тело константы состоит из 9 цифр.

изменение состояния аппаратных регистров

$$\left\{ \begin{array}{l} \text{CM} \\ \text{PM}[P] \end{array} \right\} := \left\{ \begin{array}{l} \text{константа} \\ \alpha \end{array} \right\}$$

PK := восьмеричная константа

MPi := восьмеричная константа ... [i] ...

(здесь возможно последовательное присваивание, аналогичное описанному выше).

Примеры:

2057. TEXT := 426

в 2057 слово подпрограммы TEXT записать восьмеричную константу 426;

A. SUMM := C. SUBR

переменной A из подпрограммы SUMM присвоить значение переменной C из SUBR.

T := H... SUPER-DEBUGGER

текстовая константа состоит из 16 символов; после выполнения команды она займет 3 слова COMMON-блока T, в младших разрядах последнего будет стоять два пробела в коде ISO;

100+17:=0

в слова с абсолютными адресами 100+17 записывается нуль;

200:=435 _ TEXAMP"E", 527, 36

в слово с адресом 200 записывается восьмеричная константа 435, в 201 слово - текстовая константа, изменяется также содержимое 204 и 206 слов, а содержимое 202, 203, 205 слов остается прежним.

9. ГРУППА КОМАНД ДЛЯ РАБОТЫ С ВЫХОДНЫМ ФАЙЛОМ ЗАДАЧИ

Операционная система ДИАПАК осуществляет буферизацию всей выводимой информации. Это означает, в частности, что данные, выводимые на печать в процессе решения задачи, сначала накапливаются в специально организованном выходном файле и лишь после окончания задачи выдаются на реальное устройство (АЦПУ).

С помощью описываемых ниже команд пользователь может в любое время в течение сеанса отладки ознакомиться с содержимым выходного файла своей задачи, выдать его на свободное устройство или, наоборот, уничтожить накопленную в нем информацию.

o[UT][n] просмотр содержимого выходного файла.

Здесь целое десятичное n - порядковый номер строки (данные в выходном файле сгруппированы в строки так, как они будут расположены на бумаге; строки нумеруются с 1). Параметр n определяет номер строки, начиная с которой информация выдается на терминал; в начале каждой строки указывается её десятичный номер. Выдача происходит кадрами по 6+11 строк в зависимости от их длины. Продолжение просмотра (получение очередного кадра) осуществляется нажатием клавиши **STX**, если последней выполнялась команда **OUT**, или командой **OUT** без параметра в противном случае; продолжения не последует, если выдан весь файл. Если $n=0$, то на терминал выдается число реально существующих строк.

A[ЦПУ] начать выдачу содержимого выходного файла на реальное устройство.

PRO[СЧИТАТЬ][n₁[-n₂]] уничтожить выдачу на АЦПУ.
 n_1, n_2 - десятичные номера строк ($n_2 \geq n_1$), определяющие

уничтожаемый фрагмент выходного файла ; при отсутствии N_2 уничтожаются строки начиная с N_1 до конца файла. Если операндов нет (ПРО), то уничтожается вся накопленная для печати информация, но во время дальнейшего выполнения задачи выходной файл организуется вновь. Этого не произойдет, если в команде была указана звездочка (ПРО*); кроме того, в этом случае на АЦПУ не будет выдана даже стандартная информация о завершении задачи.

{ ПРА
PR[INT] }

параллельный вывод на АЦПУ

Команда ПРА устанавливает (и снимает) признак, обеспечивающий запись в выходной файл задачи протокола работы с Отладчиком: вводимых пользователем команд, ответов и сообщений системы. Полезность такого режима доказывается следующими соображениями: во-первых, полученная на АЦПУ копия протокола связи может быть использована для "домашнего" анализа; во-вторых, не прерывая работы, пользователь с помощью операции OUT может "вернуться назад" и просмотреть выданную ранее информацию о предыдущих этапах отладки. Это в какой-то мере компенсирует недостатки дисплея, связанные с ограниченностью размеров экрана и отсутствием печатного документа о работе.

10. ВЫПОЛНЕНИЕ НЕСТАНДАРТНЫХ ДЕЙСТВИЙ

Нестандартными действиями в Отладчике называются машинные команды, которые вводятся (только на момент исполнения) с термinals и исполняются в рамках задачи пользователя, изменяя тем образом её состояние.

Исполнение нестандартных действий осуществляется командой

• [ИСПОЛНИТЬ] к,

где к - тело командной константы. После выполнения нестандартного действия задача может быть продолжена с прерванного места.

Примеры:

В 00221000 00.70 0100 ;

ИЛИ 00 012 0354.

Отметим, что на нестандартное действие и последующее испол-

нение программы не влияют предшествовавшие команде ВЫПОЛНИТЬ 22 и 23 операции БЭСМ-6.

II. ЗАВЕРШЕНИЕ СЕАНСА ОТЛАДКИ

После выполнения команды

{ KH[4]
E[ND] } [Lnd]

связь пользователя с Отладчиком прекращается и

- исполнение задачи продолжается с позиции α (предварительно автоматически отменяются контрольные точки, установленные операциями КРА, ЭП/СЧ, УЗД);
- при отсутствии операнда задача снимается; если последнее прерывание носило характер авоста, то его обработка, предусмотренная в задаче, будет выполняться.

Команда

{ H[OB]
RE[START] } [*]

также заканчивает отладку снятием задачи, но в этом случае сохраняется информация об её исходном состоянии. Поэтому сеанс отладки данной задачи может быть повторен:

- если название операции сопровождается звездочкой, для этого необходимо выполнить директиву НИР (см. п.1);
- если звездочки нет, считается, что директива НИР выполнена и задача будет включена в решение, как только представится такая возможность.

12. СПОМОГАТЕЛЬНЫЕ КОМАНДЫ

БА[ЗА] [адрес]

- значением операнда осуществлять базирование во всех последующих командах отладки, содержащих абсолютные восьмеричные адреса;
- отсутствие операнда означает отмену базирования.

Пример:

БАЗА IOOO

У I7(7) установить кратный узел в слове IOI7;

20+5=K выдать содержимое IO20+IO25 слов.

С помощью перечисленных ниже команд на терминал можно выдать:

- шифр задачи пользователя и его фамилию - КТ[0];
- информацию о последнем происшедшем в задаче прерывании - Г[ДЕ];
- астрономическое время и использованное в этом сеансе отладки время центрального процессора - БР[ЕМЯ].

13. ДОПОЛНИТЕЛЬНЫЕ ВОЗМОЖНОСТИ ЯЗЫКА ОТЛАДКИ

В этом разделе описываются средства, с помощью которых могут быть расширены операционные возможности системы и получены некоторые дополнительные удобства при отладке.

13.1. Вызов подпрограмм

`CALL subname(p1, ..., p4)`

где *subname* - имя подпрограммы,

p1, ..., p4 - список параметров (не более четырех).

Команда `CALL` предназначена для вызова подпрограмм, расположенных во временной библиотеке задачи пользователя или в общих библиотеках типа `LIBRARY`. Исполняются подпрограммы в рамках самой задачи пользователя, Отладчик лишь организует обращение к ним, имитируя для этого оператор

`CALL LOADGO(p1, ..., p4, 'subname')`

см. § 58 [3]).

Параметрами могут быть обобщенные адреса (см. п.2) или литеральные величины, имеющие вид

= <константа>.

Для размещения последних необходимо присутствие в памяти

программы `COMMON`-блока `/DEBUG/`, длина которого зависит от предполагаемого числа и типа литеральных параметров. Если задана текстовая константа, то справа ее ограничителем является символ "E", соответствующий одновременному нажатию клавиш `CTRL` и `E`.

Состояние задачи (содержимое аппаратных регистров) в точке выполнения `CALL` сохраняется и автоматически восстанавливается, если исполнение *subname* завершилось успешно. Если же произошел аворт и подпрограмма не закончила свою работу, то состояние задачи, предшествовавшее выполнению `CALL`, можно восстановить специальной командой `RET[URN]`. После окончания процедуры восстановления Отладчик переходит в режим ожидания очередной команды пользователя.

С помощью команды `CALL` пользователь может легко расширить основной набор операции Отладчика своими собственными. Например, могут быть написаны подпрограммы, реализующие отсутствующие в системе форматы ввода/вывода данных, выполняющие какие-либо вспомогательные (сервисные) функции и т.д.

13.2. Макрокоманды

Опыт показывает, что в течение сеанса отладки пользователю часто приходится выполнять одни и те же действия. Поэтому целесообразно заранее создавать набор наиболее употребимых на данном этапе команд и хранить их. Этим наборам удобно присваивать краткие обозначения (номера), с помощью которых иницируются соответствующие отладочные действия. Команду или последовательность команд, получившую такое обозначение, будем называть флэшем (*flash* - кусочек). Флэш, как это будет видно из дальнейшего, является, по-существу, макрокомандой языка отладки.

Определение (и переопределение) флэша осуществляется так

$$F[LASH]f := \langle \text{текст команды} \rangle \& \dots \& \langle \text{текст команды} \rangle,$$

текст флэша

где *f* - номер флэша, восьмеричное число ($1 \leq f \leq I7_8$).

Общая длина текста флэша не может превышать 63 символа. Корректность команд, образующих флэш, на этапе его определения

не проверяется, допущенные ошибки будут выявлены лишь при его исполнении.

Для исполнения флэшей достаточно набрать

$$F[LASH] f_1 \dots f_n$$

где f_1, \dots, f_n ($n \leq 8$) - список последовательно исполняющихся флэшей.

Примеры:

$F5 := AREA[K, L]$

$F15 := PBA \& RES. SUMM - RES[4, 4] = \& PBA$

$F1 := GOTO$

- $F5$ - на терминал выдается элемент массива $AREA$ из текущей подпрограммы;
- $F15_1$ - устанавливается режим PBA (см. п.7), массив RES из $SUMM$ выводится на терминал и АЦПУ, режим PBA отменяется и исполнение программы продолжается с прерванного места.

Список флэшей может быть включен в описание (не простой) контрольной точки:

(<описание типа> : $f_1[A] \dots f_n[A]$) , $n \leq 8$.

После прерывания, вызванного контрольной точкой, имеющей такое описание, флэши, перечисленные в списке, будут последовательно исполняться при очередном нажатии клавиши $[STX]$. Кроме того, если два или более соседних номера флэша указаны с буквой "А", то они исполняются автоматически, не требуя нажатия $[STX]$.

Примеры:

$F1 := GOTO$

$F2 := DUMP K.SUBR \& DUMP L \& DUMP AREA - AREA[K, L]$
 $BREAK \#78.SUMM(5 = :2A 1A)$

на каждом пятом проходе через оператор $\%8$ подпрограммы $SUMM$ на АЦПУ будут выданы значения переменных K и L из $SUBR$ и определяемый ими фрагмент массива $AREA$, затем исполнение задачи продолжится; все действия выполняются без вмешательства пользователя.

$F31 := CM = \& IP4 =$

$F101 := И$

КРА 127(77:3A 10)

Перед выполнением команд из слова 127 на терминал выводится содержимое сумматора и 4-го индекс-регистра; после нажатия клавиши $[STX]$ исполнение задачи возобновляется с прерванного места.

Команды, образующие флэш, могут содержать в качестве операндов (адресов) позиционные формальные параметры - обозначаются символом $\$$; - фактические значения которым присваиваются на этапе исполнения флэша:

$$F[LASH] f_1 \dots f_n (p_1, \dots, p_n), n \leq 8.$$

Здесь p_1, \dots, p_n - список фактических значений, указанных в порядке, соответствующем последовательности формальных параметров в тексте флэшей; задаются фактические значения так же как параметры подпрограммы в команде $CALL$ (п. II.I). Флэш или исполняемый список флэшей не должен содержать более четырех формальных параметров; кроме того нельзя использовать последние во флэшах, включенных в описание контрольной точки.

Примеры:

$F10 := /\$:= /\$$

$F10 10(A.SUBR, =1., B, C.SUMM)$

- исполняется список флэшей, состоящий из двух команд присваивания: переменные A и B из $SUBR$ полагаются равными 1,0 и значению переменной C из $SUMM$ соответственно.

В Отладчике отсутствует возможность ввода/вывода данных в виде шестнадцатеричных констант. Пусть у пользователя существует подпрограмма, осуществляющая преобразование данных из кода ISO в шестнадцатеричные константы и обратно.

Введем флэши $F11 := CALL TO16(\$, \$)$

$F12 := CALL FROM16(\$, \$) \& /\$ +1 = H$

Тогда команда $F11 (=H5573E4FD85F7'E', DATA)$ записывает в слово $DATA$ шестнадцатеричную константу; заданную литераль

ним параметром ; а команда **F12(DATA, RES, RES)** выполняет обратное преобразование и выдает его результат на терминал (напомним, что для использования литеральных параметров необходим **COMMON**-блок **/DEBUG/**).

Исполнение списка флэшей прерывается, если до его исчерпания выполняется какая-то другая ("посторонняя") команда или обнаруживается ошибка в одном из флэшей списка. Перед исполнением флэша на терминал выдается его текст, который можно выдать и специальной командой

F[LASH][f] =

(при отсутствии номера **f** выдается весь список флэшей).

14. ДИАГНОСТИЧЕСКИЕ СООБЩЕНИЯ

ОШ. В ОПИСАНИИ КТ	ошибка в описании контрольной точки; сообщению предшествует фрагмент текста команды, содержащий ошибочный символ ;
ОШ. В ЗАПИСИ ИР	ошибка в записи индекса-регистра ;
ОШ. В ЗАПИСИ КОМАНДЫ	командная константа задана неправильно ;
ОШ. В ЗАПИСИ В-ИНФОРМАЦИИ	ошибка в восьмеричной информации (адрес, константа) ;
ЦВР > MAX	цвр в константе или адресе больше допустимого ;
IO->2: ПЕРЕКЛЮЧЕНИЕ АУ	запись вещественной константы не допускает её перевод в машинное представление ;
НЕДОПУСТИМЫЙ ФОРМАТ	задана несуществующая в системе спецификация типа константы или формата вывода ;
АДРЕС В ИНФ. ПОЛЕ	в команде присваивания указан адрес A или A+I, где A из раздела ТЕМЕ А (см. п.1) ;

УЗЛОВ > 61

НЕТ <МС>

НЕТ ИНФ. ТАБЛИЦ

ОШ. МБ/МД

НЕТ ИМЕНИ ПРОГРАММЫ

ОШ. В ЗАПИСИ ИДЕНТИФИКАТОРА

iden - НЕ ОПРЕДЕЛЕН

ОШ. В ИНДЕКСАЦИИ

N ОПЕРАТОРА > MAX

name - ОТСУТСТВУЕТ В ОП

ОШ. ПАРАМЕТРЫ

/DEBUG/ IS ABSENT

FLASH ERROR

СБОИ АС, ПОВТОРИТЕ

число узлов, установленных операциями УЗМ не должно превышать 61 ;

не установлен режим МС ;

нет информационных таблиц для работы в режиме МС ; возможная причина - отсутствие управляющей карты *DEBUG ; не проходит чтение

- информационной таблицы из вспомогательной памяти - сеанс отладки придется начать с начала ;

- зоны буфера вывода в операции OUT ; неизвестна подпрограмма, и которой относится объект ;

идентификатор *iden* не определен в данной подпрограмме ;

ошибка в индексации: число индексов больше допустимого или их значения больше максимального ;

номер оператора фортрановской подпрограммы больше числа содержащихся в ней ;

подпрограммы или **COMMON**-блока *name* нет в оперативной памяти ;

работать можно только с загруженными в ОП разделами ;

ошибка в задании параметров флэша или подпрограммы в команде **CALL** ;

отсутствует **COMMON**-блок **/DEBUG/**, необходимый для размещения литеральных параметров (см. п. II) ;

номер флэша > I7₈ или предпринята попытка выполнить несуществующий флэш ;

последнее сообщение пользователя не принято по вине аппаратуры ;

ДУ АНА 5 МИНУТ

17

НЕ ПОЛБМАД

в течение 15 минут от пользователя не поступило ни одного сообщения ; через 5 минут задача снимается ; требуется вмешательство системного программиста ; наиболее типичный ответ системы на действия пользователя.

15. ДОСТУП К ФАЙЛАМ СИСТЕМЫ УПРАВЛЕНИЯ ДАННЫМИ

Во время отладки осуществляется контроль за доступом к файлам системы управления данными. При попытке открыть файл, защищенный паролем, исполнение задачи прерывается и на терминал выводится приглашение: Pxx (xx- код файла), в ответ на которое пользователь должен указать пароль.

16. ЗАКЛЮЧЕНИЕ

Символьный Отладчик выходится в эксплуатацию около полугода. В настоящее время он широко используется как программистами, работающими с проблемными программами, так и системными программистами для отладки базового математического обеспечения. По оценке пользователей применение системы сокращает время отладки минимум в 1,5 раза.

Л И Т Е Р А Т У Р А

1. Г.Л.Мазный. Программирование на БЭСМ-6 в системе "Дубна". М., "Наука", 1978.
2. А.И.Волков. Автокод MADLEN СИМ, Б4-11-4634, Дубна, 1969.
3. А.И.Салтыков, Г.И.Макаренко. Программирование на языке ФОРТРАН, М., "Наука", 1976.
4. Материалы по математическому обеспечению вычислительных машин БЭСМ-6 ВД/ИПМ, Информатор, вып. 2, ИММ АН СССР, М., 1976.
5. Материалы по математическому обеспечению вычислительных машин БЭСМ-6 ВД/ИПМ, информатор, вып. 1, ИММ АН СССР, М., 1976.

Приложение

СПИСОК КОМАНД ОТЛАДЧИКА С КРАТКИМ ПОЯСНЕНИЕМ

<u>КОМАНДА</u>		<u>СТРАНИЦА В ОПИСАНИИ</u>
УЗЕМ	установить узлы	13-16
СНЯТЬ	снять узлы	14
КРА	установить(снять) узел	13-16
ЗП	установить(снять) контрольную точку по записи	13-16
СЧ	установить(снять) контрольную точку по выборке	13-16
TRACE	трассировка Фортрановских подпрограмм	16
NOTRACE	отмена трассировки Фортрановских подпрограмм	17
ЛИИ	прохождение линейных участков	17
ИВ	в операции ЛИИ игнорировать команды ЗИ	18
ЦУКА	в операции ЛИИ игнорировать команды З7	
NAME	выход на подпрограмму	18
МС	установка(отмена) режима МС	19
VT	(не) выдавать полную информацию о прерывании	19
ИЛИИ	возобновить исполнение задачи	20
СТОИ	прервать исполнение задачи	21
В=	вывод данных на терминал	21-23
ПЕЧАТЬ	вывод данных на АЦПУ	21-23
В:=	изменение содержимого памяти	23
СМ ИР РМР РК	вывод или изменение состояния аппаратных регистров	23,24

КОМАНДА

		<u>СТРАНИЦА В ОПИСАНИИ</u>
АЦПУ	просмотр содержимого выходного файла	25
	выдача содержимого выходного файла на реальное устройство	25
ПРОПУСТИТЬ	уничтожить содержимое выходного файла	25
ПВА	выдавать протокол работы с системой на АЦПУ	26
ВЫПОЛНИТЬ	выполнять команду БЭСМ-6 с терминала	26
КНЦ	завершение сеанса отладки	27
НОВ	восстановить исходное состояние задачи	27
БАЗА	установка(снятие) базового адреса	27
КТО	выдать шифр задачи пользователя	28
ГДЕ	выдать данные о последнем прерывании	28
ВРЕМЯ	астрономическое время и время ЦД	28
CALL	вызов подпрограммы	28
FLASH \ddagger =	определение макрокоманды	29
FLASH \ddagger =	выдача текста макрокоманды	32
FLASH \ddagger	исполнение макрокоманды	30-32

Корякин В. К.

Символьный отладчик ОС ДИАПАК

Руководство пользователя.

Рекомендовано к изданию ученым советом Института
математики и механики УНН АН СССР.

Отр. за выпуск Закурдаев Н. В.

РИСО УНН № 6 (80). Подписано к печати 9.01.80.

ИС № 25097. Усл.- печ. л. 2,5. Уч.- изд. л. 2,0

Формат 60 x 84/16. Тираж 1000. Цена 20 к. Заказ № 130

ЦЕХ № 4 ОБЪЕДИНЕНИЯ "ПОЛИГРАФИСТ",
СВЕРДЛОВСК, ТУРГЕНЕВА, 20