

**ГОСУДАРСТВЕННАЯ КОРПОРАЦИЯ ПО АТОМНОЙ ЭНЕРГИИ «РОСАТОМ»
АКЦИОНЕРНОЕ ОБЩЕСТВО «ИНЖЕНЕРНО-ТЕХНИЧЕСКИЙ ЦЕНТР «ДЖЭТ»
(АО ИТЦ «ДЖЭТ»)**



**ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ
ESUSDS**

РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ

б/н

Номер редакции 1.0

На 36 листах

Москва, 2019

АО «ИТЦ «ДЖЭТ»	Программное обеспечение ESUSDS Руководство пользователя	Номер редакции 1
----------------	--	------------------

ПРАВА НА СОДЕРЖАНИЕ

Настоящий документ является собственностью АО ИТЦ «ДЖЭТ» и защищен законодательством Российской Федерации и международными соглашениями об авторских правах и интеллектуальной собственности

Копирование документа либо его фрагментов в любой форме, распространение, в том числе в переводе, воспроизводство, изменение в любой форме или частично, а также передача во временное или постоянное пользование третьим лицам, разглашение или использование сведений в коммерческих интересах третьих лиц возможны только с письменного разрешения АО ИТЦ «ДЖЭТ».

Документ и связанные с ним графические изображения могут быть использованы только в информационных, некоммерческих или личных целях.

АО ИТЦ «ДЖЭТ» оставляет за собой право на изменение или обновление настоящего документа без предварительного уведомления.

Все названия компаний и продуктов, которые являются товарными знаками или зарегистрированными товарными знаками, являются собственностью соответствующих владельцев.

За содержание, качество, актуальность и достоверность используемых в документе материалов, права на которые принадлежат другим правообладателям, а также за возможный ущерб, связанный с использованием этих материалов, ответственности не несет.

АО ИТЦ «ДЖЭТ»

117335, г. Москва, Нахимовский проспект, дом 58

Сайт компании: <https://get-sim.ru/>

Тел.: +7 495 788 04 06

Электронный адрес службы поддержки: itcget@rosatom.ru

СОДЕРЖАНИЕ

ПЕРЕЧЕНЬ ПРИНЯТЫХ СОКРАЩЕНИЙ.....	5
1 ВВЕДЕНИЕ.....	6
1.1 Цель данного руководства.....	6
1.2 Соглашения, принятые при написании данного руководства	6
2 ОБЗОР СИСТЕМЫ.....	7
2.1 Введение.....	7
2.2 Уровни программного обеспечения.....	7
2.3 Заполнение Базы Данных	7
2.4 Структура программ тренажера	8
3 СИСТЕМА УПРАВЛЕНИЯ РАЗРАБОТКОЙ И БАЗОЙ ДАННЫХ (DBM)	9
3.1 Система управления разработкой	9
3.2 Основные команды системы управления разработкой	9
3.2.1 /close – закрыть файл, в который записана информация.....	9
3.2.2 /exit – выход.....	9
3.2.3 /extract – копировать MSF данные и текст с ODS уровня	9
3.2.4 /incl – определить новый модуль	10
3.2.5 /lmsf – просмотр списка модулей MSF.....	11
3.2.6 /open – открыть файл для записи данных.....	12
3.2.7 /remo – удалить информацию из MSF	12
3.3 Обслуживание Базы Данных (DBM).....	13
3.4 Командные директивы и директивы данных.....	13
3.5 Знаки подстановки.....	13
3.6 Директивы DBM	13
3.6.1 /add – добавить новую переменную в Базу Данных	14
3.6.2 /close – закрыть файл для записи данных.....	17
3.6.3 /del – удалить точку.....	17
3.6.4 /exit – выход из DBM	17
3.6.5 /lmdd – вывод информации из MDD	17
3.6.6 /map - генерация карты распределения памяти для прямых "последователей"	19
3.6.7 /mod - модификация точек на уровне пользователя	20
3.6.8 /open - открыть файл для записи.....	20
3.6.9 /root - создание раздела памяти ("глобала")	20
3.6.10 /sel - обработка командного файла	21
4 КОМПИЛЯЦИЯ ПРОГРАММНЫХ МОДУЛЕЙ (FSCAN).....	22
4.1 Функции FSCAN.....	22
4.2 Запуск FSCAN.....	22
5 Компиляция программных модулей (CSCAN).....	24
5.1 Запуск CSCAN	24
6 СОЗДАНИЕ ИСПОЛНЯЕМЫХ МОДУЛЕЙ (LLD)	25
6.1 Введение.....	25
6.2 Инициализация общих разделов памяти (const and dfglob)	25

АО «ИТЦ «ДЖЭТ»	Программное обеспечение ESUSDS Руководство пользователя	Номер редакции 1
----------------	--	------------------

6.3 Команды LLD.....	26
6.3.1 close - закрыть предварительно открытый файл.....	26
6.3.2 end - завершить список команд и построить EXEC.....	26
6.3.3 exec - построить исполняемую задачу (экзекутиву).....	26
6.3.4 exit или X - выход из LLD.....	27
6.3.5 global - задать дополнительные глобалы.....	27
6.3.6 go - активизировать интерактивную экзекутиву и выйти	27
6.3.7 initialize - инициализировать разделы общей памяти	27
6.3.8 iok - разрешает загрузку неконсистентных модулей	27
6.3.9 load - загружает контрольный модуль и требуемые сегменты.....	27
6.3.10 local - объявляет раздел памяти локальным	28
6.3.11 open - открыть выходной файл	28
6.3.12 sel - определить командный файл для обработки	28
6.3.13 share - определить как используемые другими экзекутивами дополнительные общие разделы памяти	28
6.3.14 user - указывает имя пользователя.....	28
6.4 Активизация интерактивного исполнения (IEXEC).....	29
7 ИНТЕРАКТИВНЫЙ СИМВОЛЬНЫЙ ОТЛАДЧИК (ISD)	30
7.1 Введение.....	30
7.2 Режим Монитора.....	30
7.3 Директивы ISD.....	30
8 СОГЛАШЕНИЯ ОБ ИМЕНАХ	32
8.1 Введение.....	32
8.2 Идентификаторы систем.....	32
8.3 Типы переменных	32
8.4 Имена программных символов	33
8.5 Тип переменной (третий символ).....	33
8.6 Функции дистанционного управления.....	34
8.7 Отказы	34
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	35
Лист регистрации изменений.....	36

АО «ИТЦ «ДЖЭТ»	Программное обеспечение ESUSDS Руководство пользователя	Номер редакции 1
----------------	--	------------------

ПЕРЕЧЕНЬ ПРИНЯТЫХ СОКРАЩЕНИЙ

ГОСТ Р	Государственный стандарт Российской Федерации
ПО	Программное обеспечение
ПК	Персональный компьютер
РП	Руководство пользователя
РЭ	Руководство по эксплуатации
ОС	Операционная система
САПФИР	Система Автоматического Проектирования Физических Инженерных Расчетов – интегрированная программная система для создания и проведения различных физических расчетов и их отладки.
СУБД	Система управления базами данных
ESUSDS	англ. executive system of Universal Software Development System (досл. исполнительная система Универсальной Системы Разработки Программного Обеспечения) – интегрированная программная система, которая поддерживает документирование, разработку, выполнение в режиме реального времени и тестирование всего комплекса программного обеспечения тренажера.
P&ID	англ. Piping and instrumentation diagram (досл. Схема трубопроводов и контрольно-измерительных приборов)

1 ВВЕДЕНИЕ

1.1 Цель данного руководства

Руководство по Универсальной Системе Разработки Программ USDS (Universal Software Development System) предназначено для инженеров, применяющих ее для моделирования различных сложных объектов, в том числе используемых в процессе проектирования, эксплуатации и обучения персонала атомных станций.

1.2 Соглашения, принятые при написании данного руководства

- Лексические символы, которые должны быть замены их фактическими аргументами в командной линии, представлены строкой строчных символов, заключенных в левые и правые скобки вида < и >. Например, <name> команды /lmsf означает, что в этом месте должно быть вставлено имя.
- Повторение обозначается многоточием (...). Например, /lmsf <arg> означает, что переменное число аргументов может быть использовано в командной строке. Обычно физический предел длины строки определяется размерами терминальной строки.
- Дополнительные необязательные аргументы включены в левые и правые квадратные скобки. Например, /lmdd [<arg>] означает, что можно не указывать никаких аргументов.
- Обязательные альтернативные аргументы включены в фигурные скобки. Например:

```
/mmsf { <name> }  
      { <type>  }  
}
```

означает, что использование одного из аргументов имени или типа обязательно.

2 ОБЗОР СИСТЕМЫ

2.1 Введение

Универсальная система разработки программ - это интегрированная программная система, которая поддерживает разработку, документирование, исполнение и отладку программного обеспечения.

Система идеально подходит для разработки моделей тренажеров, в которых используется большая база данных, все программирование ведется на языке FORTRAN и используется многопроцессорная среда.

Исполнительная система состоит из следующих программных модулей:

- Обслуживание Базы Данных (DBM);
- Система управления разработкой (DSD);
- FORTRAN прекомпилятор и постпроцессор (FSCAN);
- C прекомпилятор и постпроцессор (CSCAN);
- Создание исполняемых модулей (LLD);
- Интерактивный символьный отладчик (ISD).

2.2 Уровни программного обеспечения

В системе могут существовать два самостоятельных уровня. Эти уровни определены следующим образом:

- Официальный уровень разработки (ODS)

Полный набор программного обеспечения тренажера, который представляет последнюю официальную версию программного обеспечения тренажера, которое может находиться в процессе разработки или модернизации.

- Пользовательский уровень разработки (UDS)

Множественные наборы программного обеспечения тренажера (по одному на каждую систему). Программное обеспечение на этом уровне представляет собой отдельные порции программного обеспечения тренажера, которые в настоящее время подвергаются модификации.

Программное обеспечение тренажера может быть модифицировано только на пользовательском уровне.

2.3 Заполнение Базы Данных

Создание и модификация Базы Данных осуществляется с помощью системы обслуживания базы данных (DBM). Все данные тренажера (переменные и константы) определены в словаре базы данных (MDD). Каждая запись содержит:

- Имя длиной в 32 символов.
- Короткое описание длиной до 72 символов.
- Полное описание (до 5 строк по 72 символа каждая).
- Тип и точность (integer, real, and logical types).
- Информация о размерности массивов и матриц (до 7 размерностей).

АО «ИТЦ «ДЖЭТ»	Программное обеспечение ESUSDS Руководство пользователя	Номер редакции 1
----------------	--	------------------

- Информация о размере и размещении.
- Единицы измерения для целей документирования.
- Формат вывода для ФОРТРАНА.
- Начальное значение переменной или константы.
- Информация о дате и уровне.
- Идентификатор системы.

2.4 Структура программ тренажера

Универсальная система разработки программ предлагает программную архитектуру, которая позволяет иметь высокую степень модульности программ.

Каждая крупная моделируемая система (парогенератор, чистый конденсат и питательная вода, ЕТС) обычно разбивается на две и более программ, с одной или более частотой вычисления.

Каждая программа может быть построена с использованием следующих типов модулей:

- Контрольный модуль (Control Module)
 - Вызывается исполняемой программой (эксекутивой) с требуемой частотой.
 - Выполняет верхний уровень управляющей логики и вызывает соответствующие сегменты в нужном порядке.
- Сегменты (Segment Modules)
 - Части модельной программы, содержащие группу зависимых уравнений.
 - Обычно сегмент должен содержать порядка ста или менее операторов ФОРТРАНА.
- Компоненты (Component Modules)
 - Контрольный модуль или модуль сегмента может вызывать один или более модулей компонента.
 - Компоненты - это стандартно построенные блоки программ, используемые для расчета часто используемых станционных компонент (задвижки, насосы и т.п.)
- Подпрограммы (Subroutines)
 - Все программные модули (контрольные, сегменты, и другие подпрограммы) могут вызывать подпрограммы.
 - Подпрограммы могут быть из стандартной библиотеки ФОРТРАНА или могут быть специально разработаны для часто используемых функций.

Пользователь должен помнить, что если модуль встроен в иерархию системы, то он будет автоматически загружен и исполнен.

3 СИСТЕМА УПРАВЛЕНИЯ РАЗРАБОТКОЙ И БАЗОЙ ДАННЫХ (DBM)

3.1 Система управления разработкой

Система управления разработкой используется для создания Файла Спецификации Модулей (MSF). Вызов осуществляется в интерактивном режиме набором команды 'dbm' в консольном окне.

3.2 Основные команды системы управления разработкой

Ниже приведен список основных команд, описание которых приведено в следующих разделах.

/close	Закрывает файл, который был определен для приема вывода на терминал (см. /open)
/exit	Выход
/extract	Копирует данные о модуле (MSF данные) и текст программы с ODS уровня на уровень пользователя
/incl	Определяет новый модуль в списке (MSF).
/msf	Выводит информацию о программных модулях
/open	Открывает файл для записи информации
/remo	Удаляет пользовательские копии выбранных программных модулей с уровня пользователя
/sel	Выбор командного файла для обработки

3.2.1 /close – закрыть файл, в который записана информация

Команда /close закрывает файл, предварительно открытый с помощью команды /open, в который записывается выводимая информация.

Синтаксис: /close

3.2.2 /exit – выход

Команда /exit выполняет выход из программы.

Синтаксис: /exit
или
/x
или
x

3.2.3 /extract – копировать MSF данные и текст с ODS уровня

Команда /extract используется для получения копии ODS модуля для модификации.

Синтаксис: /ext[ract] <name>

Где: <name> – имя ODS модуля.

АО «ИТЦ «ДЖЭТ»	Программное обеспечение ESUSDS Руководство пользователя	Номер редакции 1
----------------	--	------------------

Если модуль уже определен на уровне пользователя, команда не будет выполнена. Если команда /extract обнаружит файл на уровне пользователя, вы получите дополнительное предупреждение о перезаписи файла.

3.2.4 /incl – определить новый модуль

Команда /incl определяет новый модуль в MSF. В команде должны быть указаны имя (до 32 символов), тип модуля, идентификатор моделируемой системы (ID). Для подпрограммы-функции также должно быть указано описание типа возвращаемого значения.

Синтаксис: /incl <name>,<type>,[<return>],<sys>,[<desc>]

Где: <name> – имя модуля (до 32 символов). Имя файла, содержащего исходный код на языке ФОРТРАН, состоит из "name" и префикса "s."

<type> – тип модуля. Это поле может быть задано либо цифрой (2-9), либо ключевым словом, как указано ниже:

- 2 или control – контрольный модуль
- 3 или segment – сегмент
- 4 или component – компонент
- 5 или subroutine – подпрограмма или подпрограмма-функция
- 8 или library – внешняя подпрограмма библиотеки ФОРТРАНА
- 9 или intrinsic – внешняя встроенная подпрограмма ФОРТРАНА

Для правильной загрузки все модули должны быть отнесены к соответствующей категории, приведенной выше. Различие категорий заключается в следующем:

- Контрольные модули - это основные программы для каждой системы. Они загружаются отдельно и вызываются напрямую исполнителями реального времени.
- Сегменты загружаются автоматически вместе с контрольным модулем, из которого они вызываются. Сегменты должны вызываться только одним контрольным модулем.
- Компоненты вызываются из сегментов или контрольных модулей. Компоненты не должны вызываться более чем одним сегментом.
- Подпрограммы могут вызываться несколькими модулями моделируемой системы. Подпрограммы-функции определены как подпрограммы с дополнительным двухсимвольным описанием типа и точности возвращаемой величины.
- Внешние подпрограммы содержатся в библиотеке ФОРТРАНА.

<return> – задание типа и точности величины, возвращаемой подпрограммой-функцией. Подпрограмма-функция может возвращать:

- i1, i2, или i4 - целые функции.
- r4 или r8 - действительные функции.
- l1, l2, или l4 - логические функции.

<sys> – сокращенное наименование системы (до 3-х букв), используемое для идентификации каждого модуля моделируемой системы.

<desc> – информация о модуле (до 32 символов).

Однажды определив модуль с помощью команды /incl, вы можете модифицировать любые его поля только с помощью последовательного использования команд /remo и /incl, за исключением полей система и описание, которые могут модифицироваться формой 4 команды /lmsf.

АО «ИТЦ «ДЖЭТ»	Программное обеспечение ESUSDS Руководство пользователя	Номер редакции 1
----------------	--	------------------

Примеры:

```
/incl fw4cpsxrv,control,,fw,fw control 4 cps
```

Определен контрольный модуль с именем fw4cpsxrv (соответствующее имя файла исходных кодов будет иметь вид s. fw4cpsxrv) для системы Питательной Воды (FW).

```
/incl scwl01md2,3,,cw
```

Определен сегмент системы Цирководы (CW) с именем scwl01md2 (имя файла кодов - s. scwl01md2). Поле описания не заполнено, так как не является обязательным.

```
/incl yssqrt2,subroutine,r4,ys
```

Определена подпрограмма-функция общего назначения с именем yssqrt2 (имя файла кодов - s.yssqrt2), возвращающая значение действительной переменной обычной точности.

3.2.5 /lmsf – просмотр списка модулей MSF

Команда /lmsf выводит список модулей из MSF. Модули могут выбираться по длинному имени, короткому имени, уровню, типу модулей или по статусу консистентности. Кроме основного формата пользователь может выбрать формат вывода из списка.

Синтаксис: /lmsf [<name>][,<type>][,<flag>][,<level>][,<form>]

Где: <name> – указывает модуль(ли), который нужно вывести. Допускается указывать до 32 символов имени. Знаки "?" и "*" могут быть использованы. Если аргументы опущены, то будут выведены все модули UDS и ODS уровней.

<type> – указывает тип модуля(лей), который нужно вывести. Допустимы следующие типы:

- control - выводит список контрольных модулей.
- segment - выводит список сегментов.
- component - выводит список компонентов.
- subroutine/function - выводит список подпрограмм.
- library - выводит список библиотечных подпрограмм ФОРТРАНа.
- intrinsic
- По умолчанию выводятся модули любого типа.

<flag> – выводит модули, помеченные как консистентные или неконсистентные. Допустимые аргументы:

- incons[istent] - неконсистентные.
- cons[istent] - консистентные.

По умолчанию выводятся консистентные модули.

<level> – выводит модули заданного уровня:

- u - UDS уровень.
- o - ODS уровень.

По умолчанию выводятся модули обоих уровней.

<form> – указывает формат вывода. Допустимы следующие форматы:

- 1 - "короткий формат". Выводятся уровень, имя модуля, тип, система и статус консистентности. По умолчанию используется этот формат.
- 2 - или L[ong] - "длинный формат". Выводится вся информация из MSF. Она включает уровень, имя модуля, тип, систему, статус консистентности, описание, номер ревизии, дату включения и дату компиляции.

АО «ИТЦ «ДЖЭТ»	Программное обеспечение ESUSDS Руководство пользователя	Номер редакции 1
----------------	--	------------------

- 3 или X или 23 или sequ - "Формат перекрестных ссылок вверх". Выводятся уровень, имя модуля, тип, система и статус консистентности всех модулей, которые ссылаются на заданный для вывода модуль. Опция 'sequ' выводит так же исходное уравнение для каждого вызова заданного модуля.
- 4 - "Интерактивный формат". После выбора модуля 'dbm' переходит в интерактивный режим вывода каждый модуль в коротком формате и запрашивает диспозицию. Допустимые ответы:
 - 12 - выводит список в формате 2 и запрашивает новый ввод.
 - 13 - выводит список в формате 3 и запрашивает новый ввод.
 - 15 или 18 - выводит список в формате 5 и запрашивает новый ввод.
 - 16 - выводит список в формате 6 (/incl команда) и запрашивает новый ввод.
 - 17 - выводит список в формате 7 и запрашивает новый ввод.
 - x или q - выход и возврат в DBM.
 - любой другой символ - переход к следующей позиции.
- 5 или 8 = "Формат перекрестных ссылок по символам". Выводит список символов из MDD, на которые ссылается модуль (на момент последней компиляции). Символы из левых частей уравнений будут помечены звездочкой (*). Символы уровня пользователя и ODS уровня группируются отдельно.
- 9 или "FULL" - "Полный формат". Комбинация форматов 2,3,7 и 8.
- 6 или 'incl' - "Формат ввода". Выводит список модулей как серию команд /incl.
- 7 - "Формат перекрестных ссылок модулей вниз". Выводит имя и идентификатор системы модулей, вызываемых выбранным модулем.

Примеры:

/lmsf

Выводится список всех модулей UDS и ODS уровней в "коротком" формате.

/lmsf *,o,7

Выводится список всех модулей ODS уровня с именами модулей, которые они вызывают.

/lmsf incons,subroutine

Выводится список всех неконсистентных модулей, определенных как подпрограммы, в "коротком" формате.

3.2.6 /open – открыть файл для записи данных

Команда /open позволяет пользователю открыть файл, в который будут записываться результаты выполнения всех команд, исполняемых в текущей сессии работы с 'DBM'. Файл будет оставаться открытым и записывать все данные до тех пор, пока пользователь не закроет его командой /close или не выйдет из DBM.

Синтаксис: /open <filename>

Где: <filename> – имя файла для записи данных. Если файл не существует, он будет создан DBM. Полный путь (альтернативная директория) должен быть включен в одинарные кавычки.

3.2.7 /remo – удалить информацию из MSF

Команда /remo удаляет заданный программный модуль и всю ссылочную информацию как из MSF так и из MDD.

АО «ИТЦ «ДЖЭТ»	Программное обеспечение ESUSDS Руководство пользователя	Номер редакции 1
----------------	--	------------------

Синтаксис: `/remo <module1>,<module2>,...<modulen>`

Где: `<module>` – имя модуля , которое необходимо удалить из MSF.

DBM будет запрашивать подтверждение на удаление каждого файла исходного текста. Объектные файлы и вся перекрестная ссылочная информация будут удалены с пользовательского уровня.

Пример:

`/remo fw4cps`

Информация о модуле с коротким именем fw4cps будет удалена из MSF, и файлы с исходным текстом и объектный - будут стерты.

3.3 Обслуживание Базы Данных (DBM)

Программа обслуживания Базы Данных (DBM) используется для работы с файлами Базы Данных (MDD) на пользовательском уровне, в том числе для обновления и вывода информации. DBM может быть использована как в интерактивном, так и в командном режиме (см. команду `/sel`). Интерактивный режим запускается вводом команды `'dbm'` в консольном окне. DBM выдаст приглашение `'dbm>'`. Набрать одну из команд, перечисленных ниже. Командный режим запускается специальным командным файлом. Синтаксис вызова DBM в командном режиме:

Синтаксис: `dbm -s <pathname1> ... <pathnamen> [<redirection>]`

Где: `<pathname>` – полный путь командного файла, содержащего перечень команд.

`<redirection>` – стандартная команда для перенаправления вывода из базы данных. Если перенаправление не задано, весь вывод будет сделан на терминал пользователя.

3.4 Командные директивы и директивы данных

Существует два уровня директив. Командные директивы предписывают необходимые действия и начинаются знаком `"/` (знак `"/` может быть опущен). Директивы данных вводят значения в Базу Данных и должны начинаться знаком `"."`.

3.5 Знаки подстановки

Команда вывода списка переменных (`/lmd`) допускает в имени переменной использование знаков подстановки `"?"` и `"*"`. Знак `"?"` заменяет один любой символ, а знак `"*"` заменяет последовательность символов любой длины.

3.6 Директивы DBM

Основные команды доступные в DBM и их описания приведены ниже:

<code>/add</code>	Добавить переменную в пользовательский MDD
<code>/close</code>	Закрыть ранее открытый для записи файл
<code>/del</code>	Стереть переменную или всю древовидную структуру, относящуюся к выбранному имени переменной
<code>/exit</code>	Выход из DBM

АО «ИТЦ «ДЖЭТ»	Программное обеспечение ESUSDS Руководство пользователя	Номер редакции 1
----------------	--	------------------

/lmdd	Вывести информацию по указанному входу в MDD
/map	Вывести карту распределения памяти
/mod	Модифицировать значения полей переменной в MDD
/open	Открыть файл для записи данных
/root	Создать новый раздел общей памяти
/sel	Выбрать командный файл для обработки

3.6.1 /add – добавить новую переменную в Базу Данных

Директива /add добавляет новую переменную (точку) в MDD пользователя.

Синтаксис: /add <ptid>

Где: <ptid> имя переменной (не более 32 символов).

Примечание: Нельзя добавить новую точку в Базу Данных, если она уже существует на ODS уровне или на текущем уровне пользователя.

Для каждой добавляемой точки необходимо ввести информацию, которая в интерактивном режиме запрашивается DBM.

.desc короткое и полное описание точки.

.dim размерность точки (для массива).

.form формат выдачи значения точки на экран (опционально).

.pred имя раздела общей памяти (предecessор), порядок размещения и смещение в разделе.

.sys идентификатор моделируемой системы.

.type тип и точность.

.unit единицы измерения.

.valu начальное значение.

Если пользователь не хочет вводить значения в некоторые поля, то он может просто нажать клавишу "возврат каретки" в ответ на приглашение из DBM. Если точка добавляется с использованием команды /sel, то директивы данных должны быть включены в командный файл.

Ниже приведено более подробное описание директив данных.

.desc – описание точки

Директива .desc обязательна. Она используется для ввода короткого и полного описания точки.

Синтаксис: .desc <brief>

<full>

Где: <brief> – короткое описание (до 72 символов).

<full> – максимум 5 строк по 72 символа каждая, для детального описания.

Эта часть описания опциональна.

Любой символ (включая пробел) можно использовать в строках описания, за исключением того, что первым символом в строках полного описания не может быть точка (.) или наклонная черта (/), чтобы избежать конфликта с директивами. Лидирующие пробелы в коротком описании игнорируются. В интерактивном режиме ввод пробела для полного описания прерывает ввод.

АО «ИТЦ «ДЖЭТ»	Программное обеспечение ESUSDS Руководство пользователя	Номер редакции 1
----------------	--	------------------

Полное описание может содержать информацию для Инструкторской Станции и использует ключевые слова с префиксом (@).

.dim – размерность массива

Директива **.dim** опциональна. Она используется для задания размерности массива точки. Заметим, что если переменная объявлена как массив, ее начальные значения задаются в файле 'block data', а не директивой '**.valu**'.

Синтаксис: **.dim** <n>

или

.dim (<n₁>,<n₂>,...,<n₇>)

Где: <n> – число элементов в каждой размерности. Любое отсутствующее поле полагается равным 1, т.е. не существующим. Максимально допускается использование 7-мерных массивов.

.form – формат выдачи значения точки

Директива **.form** опциональна. Она используется для указания формата выдачи значения точки.

Синтаксис: **.form** <format>

Где: <format> – любой формат в соответствии с руководством пользователя языка FORTRAN-77. По умолчанию используется формат i19, e19.2 или l19 в зависимости от типа переменной.

.pred – определение местоположения точки

Директива **./pred** обязательна за исключением случая, когда 'p' указано как тип переменной в директиве **.valu**. Структурная информация о точке включает в себя имя раздела общей памяти (предecessора), категорию порядка размещения и смещение относительно начала предecessора.

Синтаксис: **.pred** <ptid>[,<type>][,<offset>]

Где: <ptid> – имя предecessора для размещения точки (не более 16 символов).
<type> – либо "f" при свободном размещении, либо "b" при размещении в заданных границах выбранного раздела общей памяти. По умолчанию используется свободное размещение (f).

<offset> – смещение от начала выбранного раздела общей памяти. Задаётся десятичным числом "n" или числом с буквой "nb", "nh", "nw" или "nd" для удобства задания смещения в байтах, полусловах, словах или словах с двойной точностью. При использовании свободного размещения, смещение опционально. По умолчанию используется смещение 0.

При использовании свободного размещения точки DBM автоматически определяет свободное место в выбранном разделе общей памяти. При размещении в заданных границах выбранного раздела общей памяти, если место уже занято другой точкой, операция не будет завершена.

.sys – идентификатор моделируемой системы

Директива **.sys** обязательна. Она указывает моделируемую систему, которая ответственна за данную точку.

Синтаксис: **.sys** <sys>

Где: <sys> – двух или трех символьный идентификатор системы.

.type – тип данных и точность

Директива **.type** обязательна. Она определяет тип и точность переменной.

АО «ИТЦ «ДЖЭТ»	Программное обеспечение ESUSDS Руководство пользователя	Номер редакции 1
----------------	--	------------------

Синтаксис: `.type <type>,<prec>`

Где: `<type>` – тип данных “i”, для целого “r”, для действительного или “l” для логического

`<prec>` – точность: 1 для байта, 2 для полуслова, 4 для слова или 8 для числа с двойной точностью. По соглашению целые числа могут быть 1, 2 или 4 байтные; действительные могут быть 2 и 4 байтные; логические могут быть 1, 2 и 4 байтные. (На практике i*1 не используется. Предпочтительно использовать i*4).

.unit – единицы измерения

Директива `.unit` опциональна. Используется для задания инженерных единиц измерения.

Синтаксис: `.unit <unit>`

Где: `<unit>` – инженерные единицы измерения (не более 6 символов).

Отсутствуют по умолчанию.

.valu – начальное значение переменной или константы

Директива `.valu` опциональна. Используется для задания начального значения переменной или константы и указание на вид переменной.

Синтаксис: `.valu <value>[,<vcopt>]`

Где: `<value>` – любое допустимое в ФОРТРАНЕ значение, соответствующее типу переменной. По умолчанию это значение равно 0 или `.FALSE`.

`<vcopt>` – “v” для переменной, “c” для константы, “p” для параметра или “t” для временной величины. По умолчанию используется значение “v”. “p” используется для определения параметров «a», так как параметры не имеют указания на общую область памяти.

Примеры использования директивы `/add`:

```
/add fwpressure21a
.desc 01010 Pressure in htr. 21a
.form f5.3
.sys fw
.type r,4
.pred global01fw
```

Переменная `fwpressure21a`, действительное 4-байтное число, будет добавлена в MDD в «глобал» `global01fw`.

Примечание: глобал - это принятое в системе наименование раздела общей памяти.

```
/add cwdegrees
.desc 05090 Temp. of compressor bearing.
.sys cw
.type r,8
.unit deg c
.valu 50.5
.pred global01cw
```

Переменная `cwdegrees`, действительное 8-байтное число с начальным значением 50.5, будет добавлена в MDD в глобал `global01cw`.

```
/add ioarray
.desc 03040 Status light box Panel 12.
.dim (3,29)
.form i4
```


АО «ИТЦ «ДЖЭТ»	Программное обеспечение ESUSDS Руководство пользователя	Номер редакции 1
----------------	--	------------------

<globalid> – задает имя глобала, все точки которого будут выведены. Раздел задают в форме “global<specific_id>”.

<ord> – указывает порядок вывода точек. Порядок может быть задан следующим образом:

- alpha – алфавитный порядок. Это порядок по умолчанию.
- rel – относительный, по адресам, порядок.
- forward – порядок выдачи точек по возрастанию адресов в MDD.
- reverse – порядок выдачи точек по убыванию адресов в MDD.

<form> – задает формат выдачи. Существуют следующие форматы:

- s – “короткий формат”. Состоит из двух строк информации о точке, включающих все, что указано в команде /add за исключением полного описания. Дополнительно, короткий формат содержит индикатор уровня (пользовательский/ODS), имя глобала и код, и адрес глобала. Это формат по умолчанию.
- l – “длинный формат”. Содержит все, что содержит короткий формат плюс полное описание и дату добавления/модификации.
- f – «полный формат». Все, что в длинном формате плюс информация об указателе, создаваемым внутри dbm.
- w – «широкий формат». Содержит ту же информацию, что и короткий формат, но выводится в одну строку шириной в 132 символа.
- i – «формат ввода». Содержит всю информацию о точке в формате командного файла с ключевым словом '/cmd' в качестве командной директивы. Директива '/cmd' может быть указана как одна из:

```
ADD      /add
MOD      /mod
DEL      /del
```

<xrf> – задание режима выдачи ссылочной информации. Допустимо использование следующих указателей:

- x – выводит имена модулей, которые ссылаются на выбранную точку.
- pr – выводит точки, не имеющие ссылочной информации.
- sequ – выводит имена модулей, которые ссылаются на выбранную точку и часть исходного текста, где эта точка упоминается.
- xo – выводит только те точки, которые ссылаются на модули, чьи идентификаторы систем отличаются от идентификаторов систем точек (т.е. интерфейсные точки).

<level> – указывает программный уровень, для поиска соответствующей информации о переменной. Допустимо использование следующих указателей уровня:

- u – выводит только те переменные, которые определены на пользовательском уровне MDD.
- o – выводит только те переменные, которые определены на ODS уровне MDD.
- uo или [u]+[o] – выводит точки из "исправленной" MDD (по умолчанию). "Исправленная" MDD состоит из всех точек уровня пользователя и точек уровня ODS, причем выводятся только точки уровня пользователя, если они присутствуют в обоих уровнях.
- u&o – выводит точки обоих уровней только в том случае, если они присутствуют в обоих уровнях.

АО «ИТЦ «ДЖЭТ»	Программное обеспечение ESUSDS Руководство пользователя	Номер редакции 1
----------------	--	------------------

- u-o - выводит точки, которые присутствуют на уровне пользователя, но отсутствуют на ODS.

<ic> – выводится информация о точке, содержащая значения из файла начальных условий.

Синтаксис: UIC,<nn>[,<nn>]
OIC,<nn>[,<nn>]
IC<nn>
<nn>

Где: UIC указывает, что будут использованы файлы начальных условий уровня пользователя.

OIC указывает, что будут использованы файлы начальных условий ODS уровня.

<nn> номер начальных условий или диапазон номеров начальных условий. Если отсутствует UIC, OIC или IC аргумент, то будет использоваться "исправленный" файл начальных условий.

map – указывает на то, что адреса последователей для карты памяти аналогичны тем, которые используются в команде DBM /map.

succ – выводит информацию только о прямых последователях указанной точки.

tree – выводит информацию о всех последователях и их последователях после указанной точки.

Примеры:

/lmdd

Будет выведена информация о всех точках "исправленной" MDD.

/lmdd *,parm

Будет выведена информация о всех точках "исправленной" MDD, определенных как параметры.

/lmdd *,sys,fw,l

Будет выведена информация о всех точках "исправленной" MDD, в длинном формате и принадлежащих системе "fw".

/lmdd fw:*,sequ,u

Будет выведена информация о всех точках уровня пользователя, которые начинаются с "fw:". Для всех выведенных точек так же будут выведены все модули, ссылающиеся на эти точки, и строки исходного текста программ, где они упоминаются.

3.6.6 /map - генерация карты распределения памяти для прямых "последователей"

Директива /map выдает карту памяти прямых последователей MDD точки уровня пользователя. Вывод осуществляется в виде последовательности 16-ричных цифр. Двоичное представление этих цифр указывает, что соответствующий байт точки занят некоторым прямым последователем этой точки, в то время как нуль указывает на то, что байт не занят никаким прямым последователем.

Синтаксис: /map <ptid>[,<level>]

Где: <ptid> – имя точки, для прямого последователя которой выдается карта памяти (не более 16 символов).

<level> – программный уровень точки. Допускаются следующие значения:

АО «ИТЦ «ДЖЭТ»	Программное обеспечение ESUSDS Руководство пользователя	Номер редакции 1
----------------	--	------------------

- u - уровень пользователя.
- o - ODS уровень.

По умолчанию просматривается уровень пользователя и продолжается на ODS, если точка отсутствует на уровне пользователя.

3.6.7 /mod - модификация точек на уровне пользователя

Директива /mod позволяет пользователю модифицировать информацию о точке на уровне пользовательской MDD. Вся информация о точке, которая была введена с помощью директивы /add, может быть модифицирована.

Синтаксис: /mod <ptid>[,h][,<level>]

Где: <ptid> – имя точки (до 16 символов), которая будет модифицирована.

h означает, что DBM будет выводить текущее значение каждого поля, а затем запрашивать ввод на изменение его. Если поле значение поля оставляется прежним, просто нажмите возврат каретки. Если необходимо - введите новое значение и нажмите возврат каретки. Если "h" не указано, пользователь сначала должен ввести соответствующую директиву данных, сопроводив ее новыми данными поля.

<level> – задается программный уровень точки. По умолчанию используется уровень пользователя - "u".

Заметим, что модификация предцессора точки, это то же самое, что и перемещение всего дерева структуры точки. Изменения размерности массива приводит к увеличению места, занимаемого точкой. Чтобы перевести точку из "глобала" в параметр, необходимо в директиве данных .valu добавить второй аргумент "p".

Если точка, которую пользователь хочет модифицировать, находится на ODS уровне и не присутствует на уровне пользователя, то она будет экстрагирована на пользовательский уровень MDD перед началом любой модификации. То же произойдет, если новый предцессор отсутствует на уровне пользователя.

3.6.8 /open - открыть файл для записи

Директива /open позволяет пользователю открыть файл, в который будут записаны результаты выполнения всех директив во время текущей сессии DBM.

Файл будет оставаться открытым до тех пор, пока не будет закрыт директивой /close или выходом из DBM.

Синтаксис: /open <filename>

Где: <filename> – имя файла для записи данных.

Если файл не существует, он будет создан DSD. Полный путь (альтернативная директория) должен быть включен в одинарные кавычки.

3.6.9 /root - создание раздела памяти ("глобала")

Директива /root создает глобальный раздел в памяти.

Синтаксис: /root global<nn>

Где: <nn> – идентификатор глобала. Может содержать до 5 цифробуквенных символов и/или символов подчеркивания (_).

Для каждого создаваемого глобала необходимо указать дополнительную информацию. В интерактивном режиме DBM будет запрашивать эту информацию; если информация для дополнительных полей не задается, то просто нажмите возврат каретки. Если глобал создается с

АО «ИТЦ «ДЖЭТ»	Программное обеспечение ESUSDS Руководство пользователя	Номер редакции 1
----------------	--	------------------

помощью директивы /sel , директивы данных должны быть включены пользователем в командный файл.

Заметим, что директива .dim не опциональна при создании раздела глобала.

Она используется в сочетании с директивой .type для определения размеров глобала. Число байт в глобале будет определяться умножением точности на размерность массива. Например, если тип глобала был указан как:

.type i,4

и размерность была задана как:

.dim (1,1024)

то размер глобала будет - 4096 байт.

Кроме того, заметим, что для глобала необходимо указать предцессор. При вводе этого поля укажите имя самого глобала.

Пример:

```
/root global01fw
.desc Global partition 01.
.dim (4,1024)
.valu 0,v
.sys fw
.type i,4
.pred global01fw
```

Будет создан раздел global01fw. Размер глобала - 4 килобита (16,384 bytes). Раздел идентифицируется как переменные (.valu 0,v).

3.6.10 /sel - обработка командного файла

Директива /sel запускает командный файл, который обрабатывается DBM.

Синтаксис: /sel <filename>

Где: <filename> – имя файла, содержащего директивы DBM. Полный путь (альтернативная директория) должен быть включен в одинарные кавычки.

4 КОМПИЛЯЦИЯ ПРОГРАММНЫХ МОДУЛЕЙ (FSCAN)

После того как пользователь написал тексты программ, включил их в MSF, заполнил Базу Данных (все используемые переменные добавлены в MDD), модули могут быть откомпилированы с использованием FSCAN.

4.1 Функции FSCAN

FSCAN выполняет следующее:

- Проверяет что все модули, вызываемые модулем, включены в MSF.
- Проверяет что все переменные, на которые ссылается модуль, находятся в MDD.
- Преобразует все ссылки на длинные имена модуля в короткие имена.
- Преобразует ключевые слова в стандартный ФОРТРАН синтаксис. (т.е. cseg, segment, component.)
- Строит таблицу символов и таблицу ссылок по номерам строк.
- Строит файл заголовков (header file) <name>.h , который будет включен в компиляцию. Этот файл содержит объявления всех переменных и общих блоков памяти, используемых модулем.
- Строит файл < name >.f для обработки компилятором ФОРТРАНА.
- Строит файл для документирования < name >.l, который можно распечатать.
- Передает для обработки компилятором ФОРТРАНА файлы < name>.f и < name>.h.
- Обновляет ссылочную информацию в MDD и MSF в случае успешного завершения компиляции.

4.2 Запуск FSCAN

Сначала смените директорию на S3_HOME/<sys>/src. FSCAN запускается на приглашение вводом следующей команды в консольном окне:

Синтаксис: fscan [-<options>]<filename>[,<filename>...]

Где: <options> – указание на опцию, передаваемую для FSCAN. Допустимы следующие опции:

- s - принуждает FSCAN оставить файлы <name>.h, <name>.f и <name>.l. Эта опция используется если пользователь просмотреть или распечатать эти рабочие файлы.
- c - подавляет компиляцию модулей. Используется в том случае, если пользователь хочет проверить, все ли переменные определены в MDD. Все рабочие файлы при этом сохраняются.
- a - создается файл <name>.MDD в котором содержится минимальная скелетная информация для определения любых неопределенных символов в тексте программы.
- p - выводит файл <name>.l на печать перед удалением рабочих файлов.
- q - подавляет создание файла <name>.l.

<filename> – имя файла для компиляции. Заметим, что этот файл должен находиться в пользовательской поддиректории "src". Кроме того <filename> может содержать символы замещения.

АО «ИТЦ «ДЖЭТ»	Программное обеспечение ESUSDS Руководство пользователя	Номер редакции 1
----------------	--	------------------

Если во время работа FSCAN обнаружится ошибка, она будет записана в файле <name>.l, который может быть просмотрен и распечатан.

Пример:

fscan s.scwl01

Будет откомпилирован файл исходного текста программы "s.scwl01", размещенный в поддиректории "src". Результирующий объектный код будет помещен в поддиректорию "obj" в файл с именем "scwl01.o"

5 Компиляция программных модулей (CSCAN)

Для создания программ наряду с языком FORTRAN пользователи могут использовать язык программирования C. Для этого следует воспользоваться прекомпилятором и постпроцессором CSCAN, имеющим функции, аналогичные FSCAN. При этом имя файла, содержащего исходный код на языке C должно соответствовать стандарту языка и состоять из "name" и расширения ".c", где "name" - имя модуля в MSF.

5.1 Запуск CSCAN

Сначала смените директорию на S3_HOME/<sys>/src. CSCAN запускается на приглашение вводом следующей команды в консольном окне:

Синтаксис: `cscan [-<options>]<filename>[,<filename>...]`

Где: <options> – указание на опцию, передаваемую для CSCAN. Допустимы следующие опции:

- c – подавляет компиляцию модулей. Используется в том случае, если пользователь хочет проверить, все ли переменные определены в MDD. Все рабочие файлы при этом сохраняются
- f – нумерация элементов массивов как в FORTRAN
- s – принуждает CSCAN оставить файлы <name>.h, и <name>.c. Эта опция используется если пользователь просмотреть или распечатать эти рабочие файлы.

Переменные и модули из MDD и MSF, используемые в модуле, должны быть в тексте программы в явном виде описаны как комментарии:

```
/*program <modulename> */
```

```
/*global <list> */      Для MDD переменных
```

```
/*module <list> */      Для MSF модулей
```

Кроме того, в исходный текст программы необходимо также добавить строку:

```
#include "<program name>.h"
```

6 СОЗДАНИЕ ИСПОЛНЯЕМЫХ МОДУЛЕЙ (LLD)

6.1 Введение

Загрузчик/компоновщик (LLD) предоставляет пользователю средство для загрузки модулей модели в исполняемую задачу для запуска и тестирования. Структурные характеристики LLD следующие:

- Многопользовательский – каждый пользователь имеет активную задачу LLD в процессе разработки модели.
- Работает интерактивно, но допускает использование командных файлов.

LLD активизируется набором команды “lld” в консольном окне. Обычно используется при разработке моделей для:

- Построения исполняемой задачи («эксекутивы»).
- Задания контрольных модулей, которые будут загружены в эксекутиву.

LLD затем будет выполнять для каждого заданного контрольного модуля следующие функции:

- Загружать указанные контрольные модули.
- Загружать все сегменты, указанные в контрольном модуле.
- Загружать все компоненты, указанные в модулях сегментов.

Когда все вышеперечисленные модули будут загружены, все требуемые модули подпрограмм так же будут загружены.

6.2 Инициализация общих разделов памяти (const and dfglob)

Файлы данных (d.global*), создаваемые программами **const** и **dfglob** используются эксекутивами EXEC и Главной Синхронизирующей Задачей (MST) для инициализации общих разделов памяти значениями, определенными в Базе Данных.

Прежде чем активизировать LLD, должны быть запущены в консольном окне программы **const** и **dfglob** для создания и инициализации файлов с образами общих разделов памяти. **const** активизируется набором команды “const”. Программа будет читать MDD и создаст файлы “d.global*”.

Синтаксис: const [o]

Где: [o] – указание на работу с уровня ODS.

Из-за того, что MDD способен работать только с одним значением данных на одну точку, необходимо использовать файлы “block data”, для присвоения значений массивам (обычно константам). Так как размеры таких массивов часто бывают очень большими, информацию “block data” можно делить на несколько файлов. Каждый файл блок даты должен содержать операторы Фортрана для задания значений массивам переменных и констант. Эти операторы интерпретируются программой **dfglob** и данные помещаются в файлы образов глобалов, созданных ранее программой **const**. Файлы блок даты могут так же содержать комментарии Фортрана, программные декларации блок даты и оператор end. Заметим, что значения точек, которые не являются размерными (не массивы) должны быть заданы в MDD, и не должны помещаться в блок даты. Программа **dfglob** активизируется из консольного окна:

АО «ИТЦ «ДЖЭТ»	Программное обеспечение ESUSDS Руководство пользователя	Номер редакции 1
----------------	--	------------------

Синтаксис: `dfglob [ { -u } ] <filename>`
`{`
`{`
`-o }`
`}`

Где: `<filename>` – имя файла блок даты.
`-u` по умолчанию – уровень пользователя.
`-o` ODS уровень (файлы блок даты также на ODS).

Заметим, что если пользователь модифицирует значения точек в MDD или блок даты, программы **const** и **dfglob** должны быть запущены, прежде чем активизировать EXEC или MST.

6.3 Команды LLD

Команды, доступные в LLD перечислены ниже и обсуждаются в следующих разделах.

close	Посылает выходные сообщения на терминал.
end	Завершает процесс загрузки, загружая все требуемые компоненты и подпрограммы
exec	Задаёт имя исполняемой задаче, которая будет построена LLD
exit	Выход из LLD
global	Определяет дополнительные общие разделы памяти и определяет их как не используемые другими экзекутивами
go	Активизирует последний загруженный экзекутив и выходит из LLD
initialize	Инициализация глобалов начальными значениями
iok	Разрешает загрузку неконсистентных модулей
load	Загружает указанный контрольный модуль и требуемые сегменты
local	Определяет дополнительные общие разделы памяти и определяет их как не используемые другими экзекутивами
open	Посылает выходные сообщения в файл
sel	Определяет командный файл для обработки
share	Определяет дополнительные общие разделы памяти и определяет их как используемые другими экзекутивами
user	Задаёт имя пользователя для текущей загрузки

6.3.1 close - закрыть предварительно открытый файл

Команда "close" закрывает предварительно открытый файл и перенаправляет вывод снова на терминал.

6.3.2 end - завершить список команд и построить EXEC

Команда "end" даёт сигнал прервать группу команд загрузки, что приведёт к загрузке всех требуемых компонентов и подпрограмм в EXEC.

6.3.3 exec - построить исполняемую задачу (экзекутиву)

Команда "exec" задаёт имя построенной исполняемой задачи.

Синтаксис: `exec <name>`

АО «ИТЦ «ДЖЭТ»	Программное обеспечение ESUSDS Руководство пользователя	Номер редакции 1
----------------	--	------------------

Где: <name> – имя , которое будет использоваться для идентификации EХЕС.

Пример:

ехес swехес

6.3.4 exit или X - выход из LLD

Команда "exit" выполняет действия команды "end" и выход из LLD.

6.3.5 global - задать дополнительные глобалы

Команда "global" задает разделы памяти, которые требуются для экзекутивы. По умолчанию включаются все разделы памяти уровня пользователя. Эта команда требуется чтобы включить разделы памяти ODS уровня. Пользователь должен помнить, что глобалы, включенные через команду "global", не могут использоваться другими интерактивными экзекутивами.

Синтаксис: global <global₁>[,<global₂>,...,<global_n>]

Где: <global> – имя глобала ODS уровня, который должен быть включен в экзекутиву.

Пример:

global globaldi,globalao,globallo

Глобалы globaldi,globalao,globallo ODS уровня будут включены в экзекутиву.

6.3.6 go - активизировать интерактивную экзекутиву и выйти

Команда "go" выполняет действия команды "end", активизирует последнюю загруженную интерактивную экзекутиву и выходит из LLD.

6.3.7 initialize - инициализировать разделы общей памяти

Команда "initialize" указывает разделы памяти, которые инициализируются их начальными значениями. Эта команда обычно используется в комбинации с командой "share", так как все другие глобалы инициализируются по умолчанию.

Синтаксис: initialize <global₁>[,<global₂>,...,<global_n>]

Где: <global> – имя глобала , который должен быть инициализирован.

Пример:

initialize globaldi,globalao,globallo

6.3.8 iok - разрешает загрузку неконсистентных модулей

Команда "iok" позволяет LLD загружать неконсистентные модули без прерывания загрузки. Все неконсистентные модули, которые будут загружены в этом режиме запрещенные, т.е. они загружены так, что никогда не будут выполняться.

6.3.9 load - загружает контрольный модуль и требуемые сегменты

Команда "load" загружает контрольные модули и все требуемые сегменты в EХЕС. В этой команде можно указывать только контрольные модули; все другие модули будут загружены автоматически. Все модули будут проверены на консистентность в Базе Данных перед загрузкой.

Синтаксис: load <cname>,<rate>,<group>[,F]

Где: <cname> – имя контрольного модуля для загрузки

АО «ИТЦ «ДЖЭТ»	Программное обеспечение ESUSDS Руководство пользователя	Номер редакции 1
----------------	--	------------------

<rate> – частота исполнения в Герцах. Диапазон частот от 1 до 20.

<group> – группа, в которой модули будут работать.

F – указывает на тот факт, что модуль будет выполняться в режиме пауза, как и в режиме пуск.

Пример:

load scwl01,4,2

Контрольный модуль scwl01 и связанные с ним сегменты, будут загружены в экзекутиву и будут запускаться с частотой 4 Гц во 2 группе.

6.3.10 local - объявляет раздел памяти локальным

Команда "local" используется для объявления, что глобал не будет использоваться различными экзекутивами и не будет записываться в файлы начальных условий (**snap**) или восстанавливаться из таковых (**reset**) при интерактивном режиме. (Для режима реального времени различий между "global" и "local" нет).

6.3.11 open - открыть выходной файл

Команда "open" перенаправляет терминальные сообщения в файл.

Синтаксис: /open <filename>

Где: <filename> – имя файла для записи данных. Если файл не существует, он будет создан. Полный путь не воспринимается, поэтому файл должен располагаться в текущей директории.

6.3.12 sel - определить командный файл для обработки

Команда "sel" выбирает командный файл для обработки.

Синтаксис: sel <filename>

Где: <filename> – имя файла, содержащего команды LLD.

Отметим, что файл может содержать только цифробуквенную информацию и должен располагаться только в текущей директории.

6.3.13 share - определить как используемые другими экзекутивами дополнительные общие разделы памяти

Команда "share" используется для объявления, что глобал будет использоваться различными экзекутивами. Эта директива игнорируется если пользователь использует интерактивную экзекутиву без опции "-s". Она необходима если пользователь использует режим реального времени.

6.3.14 user - указывает имя пользователя

Команда "user" позволяет пользователю загружать контрольные модули из системы другого пользователя, отличной от текущей системы. Заметим, что если в результате выполнения команды "user", один и тот же модуль используется разными системами, только первый из строки загрузки будет использоваться и исполняться.

АО «ИТЦ «ДЖЭТ»	Программное обеспечение ESUSDS Руководство пользователя	Номер редакции 1
----------------	--	------------------

6.4 Активизация интерактивного исполнения (IEXEC)

После того как IEXEC построен с использованием LLD, он должен быть активизирован в фоновом режиме, прежде чем можно будет использовать Интерактивный Символьный Отладчик (ISD) для проверки модели. Активизация может быть выполнена двумя способами: через команду "go" LLD, или выйти из LLD и выполнить команду в консольном окне:

Синтаксис: `<name> [<options>]&`

Где `<options>` – смотри ниже.

`<name>` – имя экзекутивы, созданной LLD.

IEXEC может быть активизирован без опций или с использованием одной из следующих опций:

-c `<mask>` указывает процессор, на котором будет исполняться EXEC.

-f запрещает диагностику ошибок при вычислении с плавающей запятой.

-F разрешает диагностику ошибок на переполнение при вычислении с плавающей запятой (только для компьютеров компании "Silicon Graphics").

-q уменьшает число сообщений при запуске.

-s разрешает IEXECам использовать "share globals".

-t разрешает временную статистику.

Где: `<mask>` – специфическая системная маска для указания номера процессора.

7 ИНТЕРАКТИВНЫЙ СИМВОЛЬНЫЙ ОТЛАДЧИК (ISD)

7.1 Введение

Интерактивный символьный отладчик (ISD) предоставляет возможность мониторинга тренажерных модулей. ISD запускается как самостоятельная задача и работает асинхронно с отлаживаемыми модулями. Он имеет разветвленный интерфейс с базами данных и экзекутивами для обеспечения эффективного управления модельной системы. Интерактивный символьный отладчик работает в режиме монитора, который позволяет наблюдать в динамике за 40 переменными одновременно на одном экране.

ISD активизируется набором в консольном окне команды "isd". После активизации, ISD запрашивает имя экзекутивы, к которой он должен присоединиться. После успешного присоединения, ISD переходит в режим монитора и выводит на экран все чистые линии.

7.2 Режим Монитора

Режим монитора ISD позволяет наблюдать в динамике за 40 переменными одновременно на одном экране, с возможностью иметь 10 переключаемых экранов. Данные отображаются с именами, определенными в MDD. Малый экран содержит так же: статус, модельное время, таймер, статус выбранного модуля и поле определения одной из динамических позиций.

В режиме монитора команды могут подаваться как с приглашением ISD так и без него.

Чтобы временно выйти из ISD для работы в консольном окне, пользователь должен нажать "esc" без приглашения ISD или "!" после приглашения ISD. Чтобы вернуться обратно в ISD, нажмите "<ctrl>d".

7.3 Директивы ISD

Список некоторых команд, доступных из ISD приведен ниже:

=	Сдвигает символьный дисплей вверх
-	Сдвигает символьный дисплей вниз
!	Запускает консольное окно для ввода системных команд
change	Изменяет различные поля символа
cle	Стирает выведенные символы
clock	Устанавливает проблемное (модельное) время
close	Закрывает файл печати
cop	Копирует символ с одной линии на другую
d20 или isd	Устанавливает табло монитора в режим вывода 20 символов
d40	Устанавливает табло монитора в режим вывода 40 символов
desc	Выводит короткое описание символов
display	Выводит символ на табло монитора
ds	Перерисовывает экран
dump	Выводит дамп информации
exec	Замещает текущую экзекутиву реального времени
fast	Укорачивает паузы между фреймами

АО «ИТЦ «ДЖЭТ»	Программное обеспечение ESUSDS Руководство пользователя	Номер редакции 1
----------------	--	------------------

flip	Перебрасывает символы между правой/левой колонками
frz	Перевод IEXEC в режим паузы (фриз)
lim	Выводит текущее, наименьшее и наибольшее значения выводимых символов
line	Выводит дополнительную информацию о символе
log	Печать данных о символе
look	Выводит исходный текст модуля
mod	Ввести/вывести модуль из обработки
mov	Перемещает символ с одной линии на другую
normal	Возвращает паузу между фреймами к нормальной величине
page	Выводит содержимое страницы из сохраненного файла страницы
print	Управляет печатью
q	Выход из отладчика
read или sel	Выбирает командный файл для обработки
reset	Устанавливает тренажер в начальное состояние
run	Переводит тренажер в режим пуска
save	Сохраняет текущий экран в файл
set	Присвоить символу заданное значение
sho	Выводит информацию о модуле, глобалах, времени выполнения или сообщение о причине остановки тренажера из за ошибки
slow	Удлинняет паузу между фреймами
snap	Записывает исходное состояние
step	Запускает IEXEC на указанное время
suspend	Установить интервал паузы между обновлениями экрана
sw	Переключение на другой экран монитора
user	Доступ к MDD другого пользователя для показа точки

8 СОГЛАШЕНИЯ ОБ ИМЕНАХ

8.1 Введение

Этот раздел посвящен соглашению об именах, используемому при разработке программного обеспечения тренажеров для АЭС.

Эти соглашения касаются:

- идентификаторов систем
- типа переменных
- функций дистанционного управления и отказов

8.2 Идентификаторы систем

- an Табло сигнализации
- cc КО СУЗ, ПК и СПиР
- ch Локализация аварий
- cr Активная зона
- cv Чистый конденсат
- cw Цирквода
- ed Главная схема и С.Н.
- eg Генератор
- fw Конденсатно-питательный тракт
- gc Разработка компонентов
- ms Острый пар
- ni Контроль реактора
- rc ГЦН и вспомогательные системы
- rd Система управления реактором
- rm Радиационная обстановка
- rx Регулирование реактора
- si Система аварийного охлаждения реактора
- sw Техвода
- tc САРЗ турбины
- tu Турбоустановка
- wd Газовый контур
- ys Подпрограммы общего пользования

8.3 Типы переменных

Все переменные и константы, генерируемые или используемые математическими моделями систем, должны располагаться в области общей памяти за исключением:

- индексные переменные в операторах DO.
- начальные, конечные и константы инкремента оператора DO.
- “DUMMY” аргументы в вызовах подпрограмм.

Каждое имя переменной должно быть определено в формате Базы Данных.

8.4 Имена программных символов

Имя программного символа должно состоять из следующей последовательности символов:

Первых два символа представлены идентификатором моделируемой системы.

Третий символ имени переменной определяет тип параметра, как это указано в следующем списке.

С четвертого по шестнадцатый символ – это цифробуквенная информация, которая должна по возможности наиболее близко отражать станционное обозначение задвижек, выключателей, трубопроводов, электрических шин и тому подобное.

8.5 Тип переменной (третий символ)

- a проводимость (гидравл), паросодержание
- c проводимость, теплоемкость
- d плотность, дифференциал
- e напряжение
- f частота, расход
- g угол
- h энтальпия, энергия, влажность
- i ток, целое
- j мнимая составляющая тока
- k константа
- l уровень
- m масса
- n нормализованное число оборотов насоса
- p давление
- q тепловой поток
- r радиация, активность
- s позиция
- t температура, время
- u момент, внутренняя энергия, коэффициент теплопередачи
- v объем, положение задвижки
- w ватт, мощность
- x паросодержание (масс)
- y управляющий сигнал
- z импеданс
- :_ логическая переменная

8.6 Функции дистанционного управления

Соглашение об именах функций дистанционного управления.

Значение символа	Символ (по счету)	Представление
Идент. Инструкторской Станции	1,2	ур
Логический байт	3	:
Целое число байт	3	i
Действительная переменная	3	x
Функция ДУ	4	r
Идент. моделируемой системы	5,6	Смотри "Идентификаторы систем "(приведен выше)
Номер функции ДУ	7,8	01-99
Например:		
ур:rcv01		
уріrcv02		
урхrcv03		

Если для функции дистанционного управления требуется сигнал обратной связи, то имя переменной сигнала обратной связи будет образовано из имени функции ДУ заменой символа "r" символом "f". Например: ур:fcv01 для ур:rcv01, уріfcv02 для уріrcv02, урхfcv03 для урхrcv03.

Исключением из этого правила являются сигналы обратной связи для функций ДУ логического типа (close/open) для задвижек, так как сигнал обратной связи в действительности возвращает положение задвижки, т.е. переменную типа "R*4".

8.7 Отказы

Соглашение об именах отказов.

Значение символа	Символ (по счету)	Представление
Идент. Инструкторской Станции	1,2	ур
Логический байт	3	:
Действительная переменная	3	x
Отказ	4	m
Идент. моделируемой системы	5,6	Смотри "Идентификаторы систем "(приведен выше)
Номер отказа	7,8	01-99
Множественный отказ	9+	a-z
Например:		
ур:mrc01a Останов ГЦН-11. В Базе Данных эквивалентен первому элементу массива ур:mrc01.		

АО «ИТЦ «ДЖЭТ»	Программное обеспечение ESUSDS Руководство пользователя	Номер редакции 1
----------------	--	------------------

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

- 1) ГОСТ 19.105–78 ЕСПД. Общие требования к программным документам (Раздел «Аннотация»).

Лист регистрации изменений

[illegible]