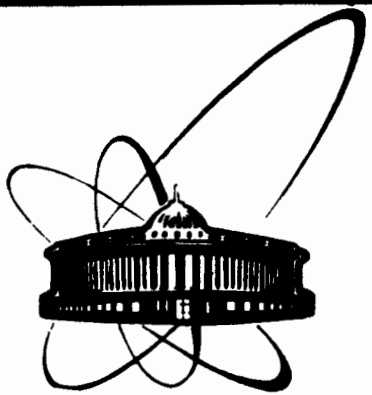


объединенный
институт
ядерных
исследований
дубна

Р10-89-163

А.А.Луцкий*, Ю.В.Столярский

СИСТЕМА БАЗ ДАННЫХ ФИЗИКИ ЧАСТИЦ НА ЕС ЭВМ

Направлено в Оргкомитет XIV Международной школы "Программирование-89", НРБ, май 1989 г.

*Институт теоретической и экспериментальной физики, Москва

1989

ВВЕДЕНИЕ

Идея создания системы баз данных физики частиц (Particle Physics Data System, далее PPDS) для сбора, оценки экспериментальных данных и обеспечения их доступности пользователям была выдвинута в начале 70-х годов американской группой (PDG(US))[1].

В настоящее время базы данных физики частиц поддерживаются в основном совместными усилиями следующих групп: американской PDG(US), английской PDG(UK) и группой физиков ИФВЭ /Протвино/. В 1983 году в это сотрудничество включился Институт теоретической и экспериментальной физики, а в 1988 году - Объединенный институт ядерных исследований.

В ИФВЭ на основе PPDS, начиная с 1979 года, развивается проект КОМПАС[8] /компиляция, анализ и систематизация экспериментальных данных физики частиц/. В результате реализации этого проекта должны быть достигнуты следующие цели:

- организация работ по сопровождению основных теоретических схем описания и интерпретации экспериментальных данных физики частиц;
- организация работ по системному поиску эмпирических закономерностей на базах данных PPDS;
- формирование наборов актуальных целеуказаний для текущих и планируемых экспериментов;
- организация своевременного информационного обеспечения прогнозных исследований фактическими данными на машинных носителях;
- автоматизация рутинных процедур теоретических и феноменологических исследований.

Основу PPDS составляет система документальных и фактографических баз данных. Документальные базы данных[3] предоставляют физикам возможность оперативного ретроспективного поиска ссылок на документы, содержащие экспериментальные данные физики частиц. Они также являются базовым источником ссылок, необходимых при формировании фактографических баз данных. Специфика проведения экспериментов в физике частиц и представления результатов обуславливают необходимость сопровождения двух документальных баз данных с перекрестными связями. Одна из них /DOCUMENTS[4]/ содержит описание публикаций оригинального и обзорного характера по экспериментальной физике частиц. Дру-

гая /EXPERIMENTS[5]/ содержит данные о текущих и законченных экспериментах в физике частиц. На основе этой базы данных подготавливается компиляция текущих экспериментов LBL-91[5]. Основное назначение базы данных EXPERIMENTS и компиляции LBL-91 - дать агрегированную информацию о современной ситуации в экспериментальных исследованиях на каждом ускорителе.

Фактографическая база данных REACTIONS[7]/ содержит фактический числовой материал о результатах измерений и оценки различных характеристик реакций с частицами, опубликованных в периодической печати и каталогизированных в базе данных DOCUMENTS.

Для работы с базами данных физики частиц используется система управления базами данных Berkley Database Management System /далее BDMS/, разработанная в Беркли /США/ и развиваемая группой КОМПАС ИФВЭ.

Версия BDMS 2.2 адаптирована для ЕС ЭВМ. В ИТЭФ проведена работа по адаптации ядра системы, в ОИЯИ - работа по адаптации процессоров баз данных DOCUMENTS, EXPERIMENTS и REACTIONS.

1. КРАТКОЕ ОПИСАНИЕ BDMS

BDMS представляет собой систему управления базами данных, поддерживающую иерархическую модель данных[2].

Наименьшей логической единицей информации в BDMS является элемент данных /DATA ELEMENT или DE/. Совокупность значений связанных DE образует запись данных. Например, в библиографической базе данных запись представляет собой описание отдельного документа. При определении базы данных каждому элементу данных присваивается уникальное имя и, возможно, один или несколько синонимов. Обращение к DE осуществляется по имени или синониму. Ограничение на количество DE, определяемых для базы данных, отсутствует.

Значением DE может быть строка символов, число или вектор, представляющих собой последовательность чисел /целых или действительных с одинарной или двойной точностью/. Строки символов могут быть любой длины в пределах ограничений, накладываемых размером доступного рабочего пространства в оперативной памяти ЭВМ во время работы BDMS.

Любой элемент данных может быть объявлен ключевым /KEY DATA ELEMENT, или KDE/. Система поддерживает инвертированные списки для всех KDE, что позволяет осуществлять эффективный поиск в базах данных.

Физически база данных, поддерживаемая BDMS, реализуется в виде трех файлов прямого доступа:

DATA FILE - содержит таблицу определения файлов и записи данных;

DIRECTORY FILE - содержит указатели адресов записей данных в DATA FILE;

INVERT FILE - содержит инвертированные списки для всех KDE.

Работа с базами данных под управлением BDMS возможна как в пакетном, так и в интерактивном режиме.

BDMS имеет выходы, позволяющие подключать программы пользователя /процессоры/. Процессоры осуществляют проверку входных данных, необходимые преобразования данных при вводе, выводе, инвертировании KDE, поиске, а также автоматическую генерацию элементов данных. Каждый процессор предназначен для работы с конкретной базой данных.

Большая часть программ, составляющих BDMS, написана на алгоритмическом языке FORTRAN-IV. Интерфейс с операционной системой и машинозависимые программы выделены в отдельные модули.

2. СОЗДАНИЕ И ПОДДЕРЖАНИЕ БАЗ ДАННЫХ

2.1. Общие сведения

Для того, чтобы создать базу данных под управлением BDMS, нужно описать данные, вводимые в базу, то есть определить имена и типы элементов данных, их иерархические отношения, определить, для каких DE будут поддерживаться инвертированные списки, и так далее. Нужно также сообщить системе информацию обо всех подключаемых процессорах при обработке того или иного элемента данных.

Вся эта информация сообщается системе с помощью языка описания данных. Описание базы, составленное на этом языке, обрабатывается специальным транслятором. Выходом транслятора является двоичная таблица определения файлов /FILE DEFINITION TABLE или FDT/, которая становится нулевой записью вновь создаваемой базы данных и используется системой при дальнейшей работе с базой.

2.2. Язык описания данных

Язык описания данных имеет простой синтаксис. Он состоит из последовательности выражений вида:

< название атрибута > = < значение >;

Допускаются следующие названия атрибутов и их значения:

DE = < имя элемента данных >;

Определенное таким образом имя элемента данных должно быть уникальным для данной базы. Оно не может содержать пробелов, знаков равенства или точек с запятой.

SYN = < синоним имени элемента данных >;

Синонимы могут использоваться наряду с именами элементов данных при работе с базой данных. Правила составления синонимов те же, что для имен элементов данных.

PAR = < имя элемента данных - родителя > ;

Таким образом определяется имя исходного по отношению к данному элементу данных в иерархической структуре. Сам исходный элемент данных должен быть определен ранее.

TY = < тип элемента данных >;

Допускаются следующие значения типа элемента данных:

INTEGER - целое /вектор/;
REAL - действительное /вектор/;
DOUBLE - действительное с двойной точностью /вектор/;
CHAR - строка символов;
NODE - узел.

VIRTUAL;

- Может следовать за определением DE. В этом случае значение элемента данных не запоминается в базе данных, однако может использоваться при вводе информации в базу, а элемент данных может быть даже определен как ключ. При желании он может воссоздаваться, когда запись читается из базы данных.

KEY;

- Может следовать за определением DE. В этом случае для элемента данных будет поддерживаться инвертированный список.

LENGTH = < длина ключа >;

- Указывает количество слов памяти, используемых для хранения значения ключа в INVERT файле, если следует за KEY атрибутом. В случае отсутствия атрибута длина ключа по умолчанию равна одному слову.

IPROC = < формат значения DE >;
OPROC = < формат значения DE >;
KPROC = < формат значения DE >;
QPROC = < формат значения DE >;

Эти атрибуты могут следовать за определением DE. Они указывают на то, будет ли вызываться тот или иной процессор и каким образом будет при этом обрабатываться значение элемента данных.

В качестве примера можно привести определение гипотетической базы данных, записи которой содержат данные из отчетов об экспериментах. Структура записей представлена на рисунке.

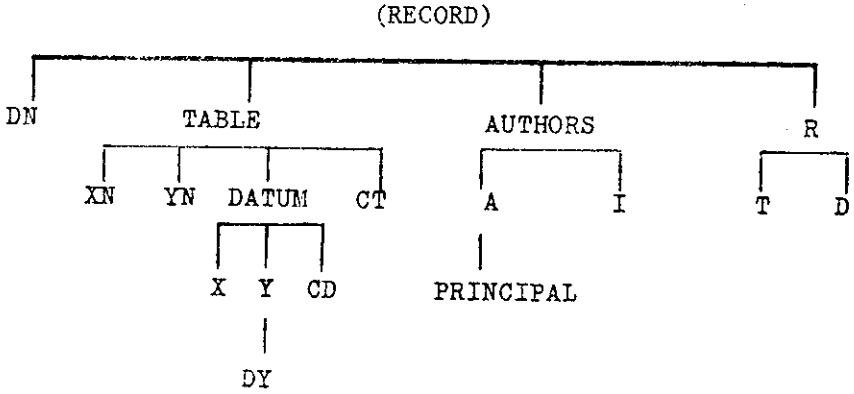


Рис.

Смысловое содержание элементов данных в записи следующее:

DN - номер документа, TABLE - имя коллекции данных, связывающее воедино имена переменных X и Y - XN и YN соответственно, возможные комментарии к таблице, CT, и собственно данные, DATUM. DATUM - узел, который связывает значения X и Y и возможные комментарии к данным, CD·DY - ошибка в значении Y. В каждом экземпляре записи может быть несколько экземпляров TABLE. В каждом экземпляре TABLE может быть несколько экземпляров DATUM и CT. В каждом экземпляре DATUM может быть несколько экземпляров CD. Остальные элементы данных, XN, YN, X, Y и DY присутствуют только один раз в каждом из экземпляров исходного элемента данных.

AUTHORS - узел, связывающий одного или нескольких авторов, A, с институтом I, в котором они работают. Элемент данных PRINCIPAL типа "узел" используется, чтобы пометить какого-либо из авторов как главного.

R - ссылка на документ /например, номер отчета/, а T и D - его тип и дата выпуска.

База данных, содержащая записи с такой структурой,может быть описана на языке описания данных BDMS следующим образом:

```
FILE=EXPT-REPORTS;
DE.=DN; SYN=DOCUMENT-NO; TY=INTEGER; KEY; LENGTH=1;
DE.=TABLE; TY=CHAR;
DE.=DATUM; PAR=TABLE; TY=NODE;
DE.=X; SYN=X-VALUE; PAR=DATUM; TY=REAL;
DE.=Y; SYN=Y-VALUE; PAR=DATUM; TY=REAL;
DE.=DY; SYN=Y-ERROR; PAR=Y; TY=REAL;
DE.=CD; SYN=COMMENT-DATUM; PAR=DATUM; TY=CHAR;
DE.=XN; SYN=X-NAME; PAR=TABLE; TY=CHAR;
```

```
DE.=YN; SYN=Y-NAME; PAR=TABLE; TY=CHAR; KEY;
LENGTH=1;
DE.=CT; SYN=COMMENT-TABLE; PAR=TABLE; TY=CHAR;
DE.=AUTHORS; TY=NODE;
DE.=A; SYN=AUTHOR-NAME; PAR=AUTHORS; TY=CHAR; KEY;
LENGTH=3;
DE.=I; SYN=INSTITUTION; PAR=AUTHORS; TY=CHAR;
DE;=PRINCIPAL; PAR=A; TY=NODE;
DE.=R; SYN=REFERENCE; TY=CHAR;
DE.=T; SYN=TITLE; PAR=R; TY=CHAR;
DE.=D; SYN=DATE; PAR=R; TY=CHAR;
```

2.3. Загрузка базы данных

Для загрузки вновь создаваемой базы данных используется утилита LOAD, работающая в пакетном режиме. Она не может использоваться для добавления записей в существующую базу данных. Это можно сделать в интерактивном или пакетном режиме с помощью команды редактора ADD. Загрузка с помощью утилиты LOAD гораздо более эффективна, так как при этом в INVERT файле создаются сбалансированные деревья после запоминания всех записей в базе данных.

Ввод данных осуществляется в свободном формате как при загрузке базы с помощью утилиты LOAD, так и при добавлении записей с помощью команды ADD. Поток входных данных содержит команды и операторы присваивания значений элементам данных вида:

< имя элемента данных > = < значение >;

- Может использоваться как имя элемента данных, так и его синоним. Значение в соответствии с типом элемента данных может быть числовым, символьным, либо пустым.

В случае пустого значения оператор присваивания имеет вид:

< имя элемента данных > = ;

Это может быть полезным, если нужно ввести значение для порожденного элемента данных, а значение исходного при этом неизвестно.

Порядок, в котором вводятся операторы присваивания и имена узловых элементов данных, определяет порядок, в котором создаются экземпляры элементов данных и их связь с экземплярами исходных элементов данных. В общем, появление каждого оператора присвоения или имени узлового элемента данных создает новый экземпляр элемента данных, который будет добавлен в конец списка экземпляров элементов данных, связанных с последним созданным экземпляром исходного элемента данных.

Например, если данные загружаются в гипотетическую базу данных, описанную в разделе 2.2, то следующий поток входных данных

```
AUTHORS.;
A.=JONES;
A.=SMITH;
I.=LBL;
AUTHORS.;
A.=BAKER;
I.=UCB;
```

создает две группы авторов, первую - из двух авторов, работающих в LBL, вторую - из одного автора, работающего в UCB.

Точка после имени элемента данных означает "Следующий". Имя элемента данных без последующей точки означает "Первый", или экземпляр номер 1. Имена исходных элементов данных типа узел можно опускать при вводе. Экземпляры этих элементов данных будут автоматически сгенерированы, если:

1/ во входном потоке встречается имя порожденного элемента данных, а экземпляр исходного еще не создан;

2/ встречается экземпляр номер 1 порожденного элемента данных, то есть имя элемента данных вводится без последующей точки, а последний из созданных экземпляров исходного имеет хотя бы один экземпляр порожденного элемента данных.

Таким образом, для получения того же результата, что и в предыдущем примере, можно входные данные кодировать одним из следующих способов:

```
A=JONES;          I=LBL;
A.=SMITH;         A=JONES;
I=LBL;      или  A.=SMITH;
A=BAKER;          I=UCB;
I=UCB;            A=BAKER;
```

Каждое появление A в первом случае, или I во втором вызывает автоматическую генерацию экземпляра узла AUTHORS.

Несколько последовательных экземпляров элемента данных одного типа могут быть созданы с помощью оператора множественного присваивания, имеющего вид:

```
имя DE = значение 1; значение 2; ...
```

С помощью оператора множественного присваивания тот же самый результат, что и в предыдущем примере, можно получить одним из следующих способов:

```
A=JONES; SMITH; I=LBL;
A=BAKER; I=UCB;
```

или

```
I=LBL; A=JONES; SMITH;
I=UCB; A=BAKER;
```

Создание текущей записи завершается при появлении во входном потоке команды "*E", либо при появлении имени элемента данных уровня записи более одного раза подряд без последующей точки.

Сигналом прекращения входного потока данных служит команда "**".

2.4. Редактирование информации в базе данных

Изменить содержимое введенной в базу данных записи можно с помощью редактора BDMS. В режиме редактирования используются следующие команды:

*R - замена значения элемента данных;
*S - замена символьной цепочки в значении элемента данных;
*D - удаление экземпляра элемента данных;
*I - вставка экземпляра элемента данных.

Более подробную информацию об использовании команд редактора можно найти в руководстве[2].

2.5. Создание транспортабельной копии базы данных

Разгрузка базы данных в формате, пригодном для последующей загрузки, производится утилитой DUMP. Разгруженная таким образом база данных может быть легко восстановлена с помощью утилиты LOAD. Если необходимо разгрузить не всю базу данных, а только определенную ее часть, можно использовать команду FIND и последующую команду DUMP.

3. РАБОТА С БАЗАМИ ДАННЫХ ПОД УПРАВЛЕНИЕМ BDMS на ЕС ЭВМ

3.1. Особенности использования BDMS в операционной среде ОС/ЕС и СВМ/ЕС

Версия BDMS, адаптированная для работы в операционной среде ОС/ЕС, использует в интерактивном режиме в качестве диалогового монитора диалоговую систему TERM[6].

Версия BDMS, работающая под управлением операционной системы СВМ/ЕС, использует диалоговый монитор ПДО. В этой версии имеется возможность использовать системную функцию HELP для получения справочной информации по системе во время работы BDMS.

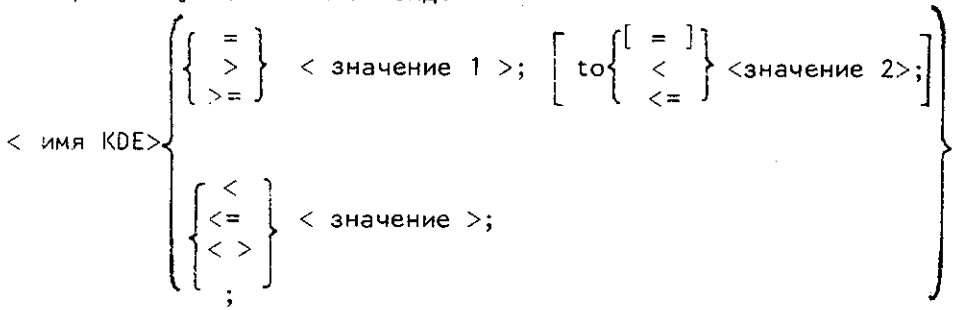
3.2. Поисковые возможности BDMS

Поиск информации в базе данных под управлением BDMS производится по команде:

FIND < условие >** .

Результатом выполнения команды FIND является набор записей, содержащий значения KDE, удовлетворяющие условию. Условие может быть простым, либо сложным.

Простое условие имеет вид:



где фигурные скобки содержат набор опций, одна из которых должна быть выбрана, а квадратные скобки содержат не обязательную опцию. Операторы отношения имеют следующие значения:

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

Примеры правильных простых условий:

A = JONES;
A <>SMITH;
X <=7;
X=5; to 8; означает X∈[5, 8]
X>5; to < 8; означает X∈]5, 8[

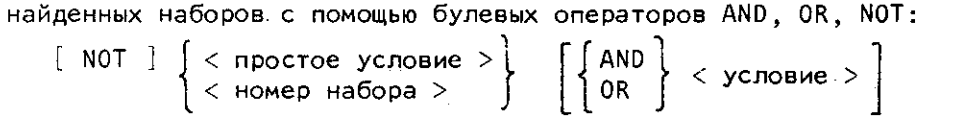
Для того, чтобы найти все записи, содержащие пустое значение KDE, нужно использовать простое условие:

FIND < имя KDE > = ; * *.

Для того, чтобы найти все записи, содержащие любое значение KDE, нужно использовать простое условие:

FIND < имя KDE >; * *.

Сложное условие строится из простых условий и номеров ранее найденных наборов с помощью булевых операторов AND, OR, NOT:



При необходимости можно использовать вложенные скобки. Команда FIND (1) AND (2)** создаст новый текущий набор записей, который будет являться пересечением наборов /1/ и /2/, созданных ранее.

3.3. Просмотр и выдача данных на печать

После выполнения поиска по команде FIND система создает текущий набор записей, который можно просмотреть на экране тер-

минала с помощью команды LIST и/или распечатать с помощью команд PRINT и DOCUMENT.

Количество выводимой на экран информации может регулироваться параметрами команды LIST:

LIST, < N >, < имя DE >,..., < имя DE >.

- вывести на экран указанные элементы данных записи номер N из текущего набора.

Команда PRINT имеет те же параметры, что и команда LIST, и позволяет выдать записи текущего набора на печатающее устройство в формате LIST.

Количество выводимой информации может также регулироваться с помощью команд SUPP и DISP:

SUPP, < имя DE >,..., < имя DE >. - подавить вывод указанных элементов данных;

SUPP, ALL-DE. - подавить вывод всех элементов данных;

DISP, < имя DE >,..., < имя DE >. - разрешить вывод указанных элементов данных;

DISP, ALL-DE. - разрешить вывод всех элементов данных.

С помощью команды DOCUMENT можно распечатать записи текущего набора в формате, определяемом соответствующим процессором для той или иной базы данных. Команда DOCUMENT имеет несколько режимов работы, предварительно устанавливаемых с помощью команды

MODD, < N >,

где N - номер режима. Работа команды DOCUMENT различна для различных баз данных и описывается подробно в руководствах по конкретной базе данных.

4. ЗАКЛЮЧЕНИЕ

В настоящее время для работы на ЕС ЭВМ адаптирована BDMS, а также процессоры баз данных EXPERIMENTS, DOCUMENTS и REACTIONS. Кроме того, в систему введена возможность спасения страниц INVERT файла, изменяемых при работе команды ADD. Это позволяет восстанавливать INVERT файл после сбоя ЭВМ, что обеспечивает возможность продолжать загрузку базы. Таким образом, практически реализована возможность загрузки баз данных под управлением BDMS в условиях неустойчивой работы ЭВМ.

В ИТЭФ BDMS используется для поддержания базы данных ВИНИТИ по физике частиц /раздел "Частицы и поля"/ и базы данных, содержащей сведения о препринтах, имеющихся в ИТЭФ. В ОИЯИ поддерживается архивный банк данных PPDS, включающий базы EXPERIMENTS, DOCUMENTS и REACTIONS.

Адаптация BDMS и названных процессоров позволяет сделать более доступной систему баз данных физики частиц для специалистов, работающих в организациях, оснащенных ЭВМ серии ЕС.

Авторы выражают благодарность сотрудникам группы КОМПАС ИФВЭ за консультации и помощь в адаптации BDMS.

ЛИТЕРАТУРА

1. Rosenfeld A.H. - Ann.Rev.Nucl.Sci., 1975, v.25, p.555.
2. Richards D.R. BDMS User's Manual, - LBL-4683, 1978.
3. Алехин С.И. и др. - Препринт ИФВЭ 87-178, Серпухов, 1987.
4. Yost G.D. et al. A Guide to Data in Elementary Particle Physics, - LBL-90, 1986.
5. Wohl C.G. et al. Current Experiments in Elementary Particle Physics, - LBL-91, 1987.
6. Кореньков В.В., Гончаков В.С. - ОИЯИ, Б-11-84-393, Дубна, 1984.
7. Fox G., Read B., Rittenberg A. Reaction-Data File Encoding Manual, - Particle Data Group (LBL, Tech., Durham, RAL), 1978.
8. Ezhela V.V. Particle Physics Data Systematization of IHEP, - CPC, v.33, p.225, 1984.

Рукопись поступила в издательский отдел
13 марта 1989 года.