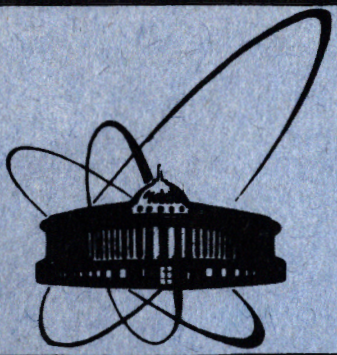

объединенный
институт
ядерных
исследований
дубна

10-84-746

А.М.Ершов

НЕКОТОРЫЕ ВОПРОСЫ ПРИКЛАДНОГО ПРОЕКТИРОВАНИЯ В СРЕДЕ СИСТЕМЫ УПРАВЛЕНИЯ БАЗАМИ ДАННЫХ ОКА

Направлено в журнал
"Управляющие системы и машины"

1984

ВВЕДЕНИЕ

Одной из наиболее распространенных в нашей стране систем управления базами данных /СУБД/ является система ОКА/1,2/. Средства этой системы предоставляют пользователю широкие возможности при создании информационных баз автоматизированных систем самого различного назначения. Как показывает практика, при разработке таких систем значительная часть времени и трудовых затрат приходится на этап прикладного проектирования /т.е. на написание и отладку прикладных программ/. Результаты проектирования во многом определяют эффективность функционирования всей конструируемой системы в целом и ставят задачу обеспечения разнообразной программной и методической поддержки процесса прикладного проектирования.

При эксплуатации СУБД ОКА постоянно возникает необходимость в выполнении определенного набора действий, требующих создания соответствующих прикладных программ. Примером такого рода операций могут служить первоначальная загрузка баз данных, их контрольная распечатка и т.п. Кроме того, при разработке прикладных программ исключительно полезными могут оказаться получение описаний данных, хранимых системой ОКА, предоставление некоторых средств отладки программ и т.д.

В настоящей работе на основе опыта эксплуатации в течение ряда лет системы управления базами данных ОКА рассмотрены некоторые вопросы прикладного проектирования в ее среде. Описаны универсальные модули, которые могут быть использованы в процессе разработки различных приложений, а также методы и средства прикладного проектирования, позволяющие в определенной степени облегчить и сократить сроки создания программного обеспечения информационных систем.

Разработка велась на ЭВМ ЕС-1060 под управлением ОС ЕС версии 6.1 с переменным числом задач.

ПРОЦЕДУРА ГЕНЕРАЦИИ ПРОГРАММ ЗАГРУЗКИ БАЗ ДАННЫХ

Непосредственной обработке баз данных в СУБД ОКА предшествует процесс определения всех их характеристик, заключающийся в генерации блоков описания баз данных (DBD) и блоков спецификации программ (PSB). Определение и создание баз данных, а также написание и отладка прикладных программ для первоначальной загрузки полностью возложены на пользователя. Описываемая процедура генерации программ загрузки баз данных/3/ имеет целью упрощение создания баз данных, управляемых системой ОКА. На практике разработчики не всегда могут воспользоваться существующими программными средствами подобного рода ввиду громоздкости, сложности в изучении, а также ввиду их отсутствия в математическом обеспечении ЭВМ. Кроме того, применение, например, генератора программ КОМПАКТ/4/, представляющего собой средство создания программ для загрузки баз данных и внесения в них всякого рода изменений, связано с освоением уникального входного языка. В то же время, когда требуется сгенерировать только программу загрузки, а исходная информация для нее не слишком сложна, желательно иметь средство, обеспечивающее перечисленные возможности с минимальными затратами труда пользователя.

Отличительной чертой рассматриваемой процедуры является отсутствие специального входного языка управляющих операторов. В качестве входной информации программы OKALOAD, выполняющейся на первом этапе процедуры, используется исходный текст блока PSB, предназначенного для загрузки базы данных. На последующих шагах процедуры производятся действия, необходимые для определения базы данных /генерация блока PSB для загрузки/, а также дополнительные действия для генерации программы первоначальной загрузки базы данных. Тем самым процедуру удается гармонично включить в процесс определения и создания баз данных в СУБД ОКА.

Программа OKALOAD выполняет синтаксический и семантический анализы исходного текста блока PSB. В случае обнаружения ошибок на печать выдаются соответствующие диагностические сообщения. В зависимости от серьезности ошибок могут быть отменены формирование блока PSB и генерация программы загрузки баз данных.

Для того, чтобы в процессе загрузки базы данных имелась возможность однозначно определить, к какому типу сегмента относится считываемая входная информация, используется понятие иерархического номера сегмента, т.е. числовое значение, которое ставится в соответствие каждому типу сегмента, указанному в блоке PSB. Программа OKALOAD допускает два вида задания иерархических номеров сегментов: неявное и явное. При неявном задании иерархический номер сегмента формируется в процессе анализа исходного текста блока PSB. Им будет порядковый номер сегмента в иерархии логической структуры базы данных, т.е. при рассмотрении всех сегментов базы данных сверху вниз и слева направо. Явное задание иерархических номеров сегментов имеет целью использование накопленных ранее исходных данных для загрузки при изменении состава чувствительных сегментов в блоке PSB, когда изменяются и их неявные иерархические номера. В этом случае есть возможность явно сопоставить каждому типу сегмента требуемый иерархический номер. Для этого в первых трех позициях каждой карты исходного текста PSB, содержащей предложение SENSEG, проставляется нужный иерархический номер. Он будет поставлен в со-

ответствие типу сегмента, определяемому в данном предложении SENSEG. Смешанное явное и неявное задание иерархических номеров не допускается. Если в исходном тексте блока PSB не обнаружено ошибок, препятствующих генерации программы загрузки базы данных, на печать выдается справочная таблица соответствия наименований сегментов и их иерархических номеров с указанием уровня сегмента в логической структуре базы данных.

Для выполнения генерации программы загрузки базы данных создана специальная каталогизированная процедура OKALOAD, состоящая из пяти шагов. Кроме вызова упоминавшейся программы OKALOAD, она обеспечивает компиляцию, редактирование связей и помещение в соответствующие библиотеки получаемые блок PSB и программу для загрузки базы данных. Присваиваемые им имена задаются в качестве параметров процедуры.

Исходные данные для загрузки базы данных организованы в виде последовательного набора данных с длиной блока, кратной 80, и могут находиться на любом выбранном машинном носителе: перфокартах, магнитных лентах и т.п. Эти данные могут быть подготовлены как обычным образом - с использованием соответствующих устройств подготовки, так и при помощи специальных программ. Первые три позиции каждой записи входного набора данных содержат числовое значение иерархического номера сегмента, либо пробелы, которые указывают на продолжение информации предыдущей записи, либо коды команд. Остальные позиции содержат данные, представляющие собой содержимое сегмента в формате, определенном в соответствующем блоке DBD. Есть возможность изменять общую длину информационной части записи посредством команды POS = nn, где nn - количество позиций записи /включая иерархический номер/, содержащих данные сегмента. По умолчанию это значение равно 72. При помощи команд ALL и ERR можно управлять формированием выходного протокола загрузки базы данных. Команда ALL /она используется по умолчанию/ вызывает передачу в протокол загрузки всех информационных сообщений, команда ERR - только сообщений об ошибках. Записи, содержащие команды POS, ALL и ERR, могут быть расположены в любом требуемом месте входного набора данных. Исходные данные предварительно должны быть отсортированы и расположены по возрастанию значений ключевых полей в иерархическом порядке. По окончании загрузки базы данных распечатывается статистическая информация о сегментах всех типов.

СРЕДСТВА ДОСТУПА К ОПИСАНИЯМ ДАННЫХ

Одним из элементов обеспечения независимости данных в СУБД является отдельное хранение описаний данных от прикладных программ. В случае использования системы ОКА информация о данных, заключенная в блоках DBD и PSB, недоступна прикладным программам, которым "известны" только рабочие области ввода-вывода и области связи, отражающие результаты выполнения запросов к базам данных. Вместе с тем целесообразность обеспечения программиста справками о структуре и других характеристиках данных, как это сделано в некоторых системах/5/, не вызывает сомнений. Кроме того, использование полных описаний данных, хранимых СУБД, открывает ряд принципиально новых возможностей для прикладного программирования/6/, так как при этом удовлетворяется потребность учета в каждом конкретном случае логической структуры базы данных, наименований и характеристик ее сегментов и полей и т.д. В качестве примера можно назвать автоматическую генерацию программ для обработки баз данных, контроля достоверности данных и т.п.

С целью обеспечения доступа к описаниям данных, хранимых СУБД ОКА, разработан программный модуль OKABLOCK, а также программа PRTBLOCK, формирующая на основе использования этого модуля подробную распечатку информации, содержащейся в заданных блоках DBD и/или PSB/7/.

Форматы таблиц выходной информации, формируемых программным модулем OKABLOCK, разработаны с учетом их наибольшего соответствия понятиям и элементам, используемым при определении баз данных в системе ОКА. Так, в информацию о каком-либо сегменте входят его имя, имя физически исходного сегмента, размер сегмента и другие подобные сведения. Такое соответствие позволяет без значительных усилий разобрать структуру и содержание выдаваемой модулем OKABLOCK информации и организовать ее эффективное использование при создании прикладных программ. Кроме того, при проектировании сделана попытка оптимального учета таких противоречивых факторов, как минимизация затрат памяти для хранения получаемых сведений о данных и удобство их программной обработки. С этой целью, с одной стороны, выделены в отдельные таблицы описания групп наборов данных, физических сегментов, логически порожденных сегментов, полей сегментов, блоков связи программ и чувствительных /доступных/ сегментов; с другой стороны, внутри таблиц непосредственно сочетаются логически взаимосвязанные элементы, например, физические сегменты и соответствующие им сегменты-источники (SOURCE), а также логически исходные сегменты.

Программный модуль OKABLOCK написан на языке Ассемблера ЕС ЭВМ, для его работы достаточно 3К оперативной памяти. Функционально программный модуль OKABLOCK состоит из двух частей, отвечающих за анализ блоков типа DBD и PSB. Каждой из этих частей соответствует своя точка входа, имя которой указывается в операторе вызова CALL при обращении к модулю. Форматы операторов вызова имеют следующий вид:

```
CALL 'LOOKDBD' USING NAME-DBD,RESULT,
                     DATABASE-CONTENT,
                     DATABASE-DMANS,
                     DATABASE-SEGMENTS,
                     DATABASE-LCHILDS,
                     DATABASE-FIELDS,
                     RANDOMIZING-MODULE.

CALL 'LOOKPSB' USING NAME-PSB,RESULT,LANG,
                     TP-PCB-TABLE,
                     DB-PCB-TABLE,
                     SENSEG-TABLE.
```

LOOKDBD - имя точки входа для анализа описания базы данных (DBD);
LOOKPSB - имя точки входа для анализа блока спецификации программы (PSB).

Переменные NAME-DBD и NAME-PSB содержат восьмисимвольные наименования соответствующих анализируемых блоков. Возвращаемое значение двоичного слова RESULT отражает результат анализа блока. Ввиду ограниченности размера статьи не приводятся структура и полное описание переменных в операторах обращения к модулю OKABLOCK.

Переменные DATABASE-DMANS, DATABASE-SEGMENTS, DATABASE-LCHILDS, DATABASE-FIELDS, TP-PCB-TABLE, DB-PCB-TABLE и SENSEG-TABLE являются таблицами в терминах языка кобол, т.е. представляют собой наборы повторяющихся структур групповых данных. Коэффициенты повторения в них, с одной стороны, должны обеспечивать разбор описаний баз данных с максимальными характеристиками составных элементов, допускаемыми СУБД ОКА, с другой стороны - они могут быть достаточными для анализа той или иной конкретной базы данных.

Программа PRTBLOCK, представляющая собой сервисное средство для системы ОКА, выполняет подробную распечатку информации о хранимых данных в виде отчета, при этом параллельно выдаются сведения о макрокомандах и их операндах, использованных при генерации анализируемого блока. Операнды макрокоманд приводятся в полном наборе, включая даже те элементы, значения которых соответствуют значениям, принимаемым по умолчанию. Программа может быть использована как для получения различных справок, так и в целях контроля за состоянием описаний хранимых данных. Кроме того, целесообразно ее применение для автоматического создания описаний блоков DBD и PSB при подготовке документации на информационное обеспечение проектируемой системы.

Программа PRTBLOCK выполняется как обычное задание ОС ЕС. Во входной информации указываются имена разделов библиотечных наборов данных, которые содержат подлежащие анализу блоки DBD и/или PSB. Указание производится в свободном формате, наименованиям блоков типа DBD предшествуют символы "DBD=", блоков типа PSB - символы "PSB=". Порядок задания блоков произвольный, для перечисления имен блоков одного типа можно использовать символ-разделитель "/". Программой обеспечивается распечатка информации блоков DBD и PSB даже в случае, если количество элементов баз данных имеет максимальные значения, допускаемые в СУБД ОКА. При этом под таблицы выходной информации зарезервировано около 73К. Общий объем памяти, необходимый для работы программы PRTBLOCK, составляет 125К без учета размеров блоков, динамически загружаемых для анализа описаний данных.

ДИНАМИЧЕСКОЕ ПОСТРОЕНИЕ АРГУМЕНТОВ ПОИСКА СЕГМЕНТОВ

При выполнении операций поиска в базах данных языком ВЕТА используются заданные аргументы поиска сегментов (SSA), определяющие критерии отбора требуемой информации. В обычной ситуации при создании прикладных программ известен весь набор запросов, связанных с обработкой баз данных. Поэтому, как правило, в программе резервируются рабочие поля, в которых размещаются все необходимые SSA. При использовании этих полей в SSA лишь подставляются нужные значения для сравнения. Однако в ряде приложений такая статичная форма задания критериев поиска информации оказывается неприемлемой ввиду отсутствия гибкости. В частности, при наличии большого количества поисковых признаков,т.е. различных полей сегментов, используемых для идентификации запрашиваемой информации, аргументы поиска сегментов заранее неизвестны, так как практически невозможно учесть все возможные комбинации квалификационных предложений, необходимых для обработки нерегламентированных сложных запросов к базам данных. Под сложным понимается запрос, состоящий из нескольких элементарных высказываний, соединенных между собой знаками конъюнкции и дизъюнкции. При этом потребовалось бы резервирование огромного количества рабочих полей для SSA, а прикладная программа стала бы чрезвычайно громоздкой и, что самое важное, чувствительной к изменениям в составе и форматах поисковых признаков. В этом случае для устранения перечисленных недостатков целесообразно реализовать в прикладной программе динамическое построение аргументов поиска сегментов на основании информации, получаемой из входного запроса/8/.

Кроме входного запроса, в котором явно или неявно задаются имена сегментов, полей, значения для сравнения и отношения, в качестве исходной информации для построения SSA могут быть использованы сведения из описаний баз данных - размеры полей, их размещение в сегментах и т.д., получаемые при помощи рассмотренных выше средств доступа к описаниям данных системы ОКА. Кроме того, возможна организация специальной справочной базы

данных с целью хранения различной дополнительной информации. К последней можно отнести номера или идентификаторы поисковых признаков, присвоенные им в информационной системе, фиксированные наборы значений поисковых признаков, их наименования, используемые при формировании выходных отчетов и справок, выдаваемых в ответ на запрос, и т.п. Формирование SSA выполняется в специально зарезервированных для этой цели рабочих областях прикладной программы в строгом соответствии со структурой SSA. При построении каждого квалификационного предложения особое внимание уделяется заполнению полей, содержащих значения для сравнения. При этом требуется учитывать выравнивание значений и тот факт, является ли соответствующий поисковый признак буквенно-цифровым или числовым, и если числовым, то каково его внутреннее представление. Сформированные SSA используются в вызове при обращении к языку BETA для поиска и извлечения информации из баз данных.

Рассмотренный метод динамического построения аргументов поиска сегментов позволяет реализовать в среде СУБД ОКА информационно-поисковую систему фактографического типа, предназначенную для поиска, отбора и извлечения информации из баз данных по запросам пользователей.

ОРГАНИЗАЦИЯ ПРОСМОТРА ДНЕВНОГО ФАЙЛА ЗАДАНИЙ

При эксплуатации информационной системы как в пакетном, так и оперативном режимах характерной является ситуация, когда требуется получить информацию о прохождении тех или иных заданий на ЭВМ. В частности, при инициализации очередного сеанса телеобработки, как правило, желательно проконтролировать результаты прохождения пакетных обрабатывающих программ, программ загрузки, корректировки баз данных и т.п. При использовании оперативного режима /режима передачи данных/ СУБД ОКА можно организовать выдачу необходимых сведений о прохождении соответствующих заданий в виде ответов на определенные входные сообщения /транзакции//9/. Такая организация не требует существенных затрат и не нарушает логики работы в СУБД ОКА. Выполняющая указанные функции прикладная программа HISTORY является, таким образом, программой обработки сообщений. При формировании ответов на запросы ею используется информация о прохождении на ЭВМ заданий и их шагов, об использовании центрального процессора, основной памяти, периферийных устройств и наборов данных. Эта информация, представляющая собой дневной файл заданий, накапливается в процессе функционирования ОС системной мониторной программой /СМП/, являющейся одним из дополнительных средств ОС ЕС. СМП собирает информацию о событиях в системе в виде записей стандартной формы, которые заносятся в специальные последовательные

наборы данных на томе прямого доступа или на магнитной ленте. Обычно наборы СМП располагаются на устройствах прямого доступа. При этом применяется два набора: основной - SYS1.MANX и альтернативный - SYS1.MANY, используемый в случае заполнения основного. Хранение на томе прямого доступа дает возможность в любой момент времени просмотреть содержимое наборов данных СМП. Для контроля результатов прохождения заданий и их шагов достаточно анализировать записи СМП только двух типов: запись окончания задания /тип 5/ и запись окончания шага задания /тип 4/.

При генерации требуемого варианта системы ОКА прикладной программе HISTORY приписываются два кода транзакции:

HISTORY - для получения сведений о выполнении задания;
HISTSTEP - для получения сведений о пошаговом выполнении задания.

Формат входных сообщений:

```
HISTORY
HISPSTEP    имя задания [,время задания [,дата задания]]
```

Время и дата задания в запросе конкретизируют точку отсчета, т.е. тот момент времени, позже которого начало выполняться задание, информацию о прохождении которого требуется получить. Время задания определяется четырехзначным числом в форме ЧЧММ, где ЧЧ - часы, ММ - минуты. Если параметр опущен или указан ошибочно, в качестве времени отсчета используется начало суток. Дата задается трехзначным числом и представляет собой порядковый номер дня в году. Если этот параметр опущен или указан ошибочно, используется порядковый номер текущего дня.

Непосредственное обращение к наборам данных СМП выполняется модулем READSMF, который обеспечивает своевременное открытие и закрытие наборов данных СМП и, анализируя их, отбирает нужную информацию согласно сформированным критериям. Скорость работы модуля оказывает определяющее влияние на продолжительность времени ответа на запрос. Модуль READSMF написан на языке Ассемблера.

Для того, чтобы наборы данных СМП были доступны прикладной программе HISTORY, описывающие их операторы DD языка управления заданиями должны быть включены в процедуру, содержащую управляющие операторы для запуска соответствующего раздела обработки сообщений. Стандартное наименование этой процедуры - OKAMSG:

```
//MESSAGE JOB...
//REGION EXEC PGM=DFSRRC00,REGION=100K,
//            PARM='MSG,001002003004'
//STEPLIB DD  DSN=OKA1.RESLIB,DISP=SHR
//        DD  DSN=OKA1.PGMLIB,DISP=SHR
//SYSMANX DD  DSN=SYS1.MANX,DISP=SHR
//SYSMANY DD  DSN=SYS1.MANY,DISP=SHR
```

Необходимо подчеркнуть, что операторы DD для наборов данных СМП включаются в процедуру OKAMSG, а не в каталогизированную процедуру ОКА запуска управляющей программы СУБД ОКА, как это делается для операторов DD, описывающих наборы данных обрабатываемых баз данных.

При эксплуатации СМП периодически выполняется процедура сброса /дампа/ наборов данных СМП, имеющая целью перенос накопленной информации в другой набор данных, например, на магнитной ленте, и освобождение наборов данных СМП. Полученная информация используется затем для формирования различных статистических отчетов. Скорость заполнения наборов данных СМП зависит от их размеров и от характеристик потока заданий. Таким образом, доступными для получения сведений о прохождении являются задания, выполнявшиеся в течение одного жизненного цикла наборов данных СМП, т.е. от одной процедуры дампа до другой.

РЕДАКТИРОВАНИЕ И ОТЛАДКА ПРИКЛАДНЫХ ПРОГРАММ В ОПЕРАТИВНОМ РЕЖИМЕ СУБД ОКА

В процессе развития эксплуатируемых информационных систем часто возникает необходимость внесения в программы некоторых изменений. Для выполнения в интерактивном режиме значительной по объему работы по редактированию исходных текстов программ целесообразно использовать специально ориентированные для этой цели диалоговые редакторы/10/. Вместе с тем, если объем работы по редактированию относительно невелик, желательно иметь средство, позволяющее произвести операции редактирования и отладки программ без останова и перезапуска СУБД ОКА, т.е. непосредственно в ее оперативном режиме. Останов системы потребовался бы ввиду того, что распределение внешних устройств в ОС ЕС является статическим и выполняется специальной SVC-программой распределения внешних устройств, вызываемой системной задачей инициатора-терминатора. Поэтому, в частности, терминал оказывается "принадлежащим" определенному заданию ОС ЕС, например, системе ОКА, и недоступным в то же самое время другому заданию, например, диалоговому редактору.

Интерактивное редактирование текстовой информации в режиме передачи данных СУБД ОКА целесообразно организовать при помощи специальной прикладной программы, выполняющей функции диалогового редактора и планирующейся в разделе обработки сообщений по мере необходимости. При этом входные сообщения, обрабатываемые такой программой, включают в себя непосредственно команды редактирования, а выходные сообщения информируют пользователя о результатах их выполнения/11/.

В качестве элемента - объекта редактирования выбран раздел библиотечного набора данных, содержащего текстовую информацию. С целью уменьшения размера оперативной памяти, необходимой для

работы прикладной программы - диалогового редактора, внутри нее накапливаются и содержатся только те сведения, которые отражают происходящие в редактируемом разделе изменения. Сам же раздел постоянно находится в библиотечном наборе данных в неизменном виде до выполнения операции его перезаписи, которая может быть инициирована пользователем.

В общем случае для оперативной обработки поступившего входного сообщения планируется соответствующая прикладная программа, причем при каждом очередном планировании загружается новая копия программы из исходного набора данных. Вследствие этого при редактировании текстов возникает проблема сохранения результатов работы прикладной программы, полученных при предшествующем планировании, для их учета при выполнении последующих операций редактирования. Решить эту задачу можно двумя путями: во-первых, сделать прикладную программу резидентной в разделе обработки сообщений, т.е. сохранять ее в этом разделе постоянно в течение всего периода редактирования; во-вторых, использовать предоставляемые системой ОКА средства написания диалоговых программ.В последнем случае программа обработки сообщений получает в свое распоряжение специальную рабочую область диалога. Это небольшая область основной памяти или небольшое пространство на устройстве прямого доступа, в котором хранятся промежуточные результаты ряда выполнений прикладной программы. Соответственно двум перечисленным возможностям организации интерактивного редактирования в оперативном режиме СУБД ОКА разработаны две прикладные программы: EDITOR, резидентно размещаемая в разделе обработки сообщений, и DIALPGM, представляющая собой программу, обрабатывающую диалоговые транзакции.

Формат входных сообщений, обрабатываемых прикладными программами EDITOR и DIALPGM, имеет следующий вид:

<КОД ТРАНЗАКЦИИ> <КОМАНДА РЕДАКТИРОВАНИЯ>

Код транзакции вызывает планирование соответствующей программы обработки сообщений. Это соответствие определяется при генерации системы ОКА, причем каждой прикладной программе может быть поставлено в соответствие несколько различных кодов транзакций. Команда редактирования конкретизирует те действия и операции, которые требуется выполнить для обработки входного сообщения. Всего имеется девять команд, ниже рассмотрены их форматы и выполняемые функции.

1. EDIT <имя редактируемого раздела>
   - просматривается справочник библиотечного набора данных; если раздел с заданным именем найден, он становится доступным для редактирования.

2. ISRT <номер строки> <символ "Конец строки"> <текст добавляемой строки>

- добавляется строка с заданным номером /как и в следующей команде REPL, входное сообщение состоит из двух сегментов, т.е. двух строк на экране терминала/.

3. REPL <номер строки 1> <символ "Конец строки"> <текст строки для замены>
   - заменяется строка с заданным номером.

4. DLET <номер строки 1> [<номер строки 2>]
   - уничтожаются строки с заданным диапазоном номеров; если параметр "Номер строки 2" опущен, удаляется одна строка с указанным номером.

5. LIST <номер строки> [<количество строк>]
   - на экране терминала высвечивается заданное количество строк, начиная с указанного номера; если параметр "Количество строк" опущен, высвечивается одна строка.

6. RESEQ [<начальное значение> [<приращение>]]
   - производится перенумерация строк с учетом задания начального номера строки и приращения номеров; если один или оба параметра опущены, их значения принимаются равными 10.

7. STORE <имя раздела>
   - отредактированный текст записывается в библиотечный набор данных в виде раздела с заданным именем; если это имя раздела совпадает с указанным в команде EDIT, то обновленная редакция заменяет собой старую.

8. BATCH <имя процедуры> <имя библиотеки> <имя раздела>
   - выполняется запуск процедуры с указанным именем аналогично команде START с консоли оператора ОС ЕС; заданные значения параметров "Имя библиотеки" и "Имя раздела" присваиваются параметрам процедуры, имеющим соответственно наименования LIB и MEM.

9. EXIT
   - завершается выполнение прикладной программы обработки сообщений.

Операнды и параметры команд отделяются друг от друга пробелами или запятыми. Номера строк редактируемого раздела библиотечного набора данных располагаются в последних восьми байтах каждой записи /позиции 73-80/. Если такая нумерация отсутствует, то после получения раздела по команде EDIT его записи могут быть пронумерованы при помощи команды RESEG.

При обработке команд ISRT, REPL и DLET прикладной программой /EDITOR или DIALPGM/ не выполняется немедленное отражение корректий непосредственно на внешнем запоминающем устройстве,а производится накопление этой информации в рабочей области программы. Физическое изменение редактируемого раздела происходит в процессе реализации команд RESEQ и STORE, вызывающих его перезапись в библиотечном наборе данных.

Команда BATCH обеспечивает запуск процедур для организации отладки редактируемых программ. При разработке комплекса требуемых процедур необходимо учитывать конкретные способы и виды этой отладки. Кроме того, считывание подготовленного задания и передачу его в зону пакетной обработки можно организовать при помощи запуска программы системного ввода ОС ЕС (RDR). Для получения информации о результатах выполнения задания может быть использован аппарат просмотра дневного файла заданий, рассмотренный выше.

Для того, чтобы библиотечный набор данных, содержащий редактируемые тексты программ, был доступен прикладным программам обработки сообщений EDITOR и DIALPGM, описывающий его оператор DD языка управления заданиями должен быть включен в процедуру, содержащую управляющие операторы для запуска соответствующего раздела обработки сообщений. Имя оператора DD - DEBUGLIB.

Программы EDITOR и DIALPGM могут быть использованы для редактирования в оперативном режиме системы ОКА различной текстовой информации. При этом, если редактируется и отлаживается прикладная программа обработки сообщений, то в том же сеансе телеобработки можно провести ее тестирование путем отправки соответствующих входных сообщений, обрабатываемых программой.

Кроме степени загруженности вычислительной системы, на величину времени ожидания ответа на запрос влияет и размер редактируемого раздела библиотечного набора данных. Причем степень этого влияния зависит от конкретной команды редактирования.Так, размер раздела совсем не отражается на времени выполнения команд EDIT, BATCH и EDIT, не требующих последовательных операций чтения и записи, в незначительной степени влияет на время выполнения команд ISRT, REPL, DLET и LIST, требующих частичного просмотра редактируемого раздела, и в большой степени определяет время выполнения команд RESEQ и STORE, связанных с полной перезаписью раздела в библиотечном наборе данных.

Прикладная программа EDITOR, остающаяся между обменами резидентной в разделе обработки сообщений, является более эффективной с точки зрения реакции на запросы пользователя по сравнению с прикладной программой диалогового режима DIALPGM. Это объясняется необходимостью выполнения дополнительных операций, связанных с планированием диалоговой программы в разделе обработки сообщений, а также с получением и сохранением рабочей области диалога. С другой стороны, в случае функционирования программы EDITOR нет возможности эффективно использовать занятый раздел для планирования прикладных программ, обрабатывающих поступающие в систему новые транзакции.

ЗАКЛЮЧИТЕЛЬНЫЕ ЗАМЕЧАНИЯ

В работе рассмотрены лишь некоторые вопросы программной и методической поддержки прикладного проектирования в среде системы управления базами данных ОКА. В этом плане можно упомянуть также о таких мероприятиях, как создание универсальных программ контрольной распечатки баз данных, удобных каталогизированных процедур для работы с системой ОКА и т.п. Как показывает практический опыт, разработка таких приложений позволяет затем рационально организовать процесс проектирования и отладки программного обеспечения информационных систем, исключить излишние затраты труда, времени, и во многом сократить сроки разработки.

ЛИТЕРАТУРА

1. Андон Ф.И. и др. "Управляющие системы и машины", 1977, №2, с.32-35.
2. Бойко В.В., Савинков В.М. Проектирование информационной базы автоматизированной системы на основе СУБД. "Финансы и статистика", М., 1982.
3. Ершов А.М. ОИЯИ, 10-82-685, Дубна, 1982.
4. Андон Ф.И. и др. "Управляющие системы и машины", 1978, №3, с.26-30.
5. Бабенко Л.П., Ющенко Е.Л. "Управляющие системы и машины", 1980, №1, с.106-109.
6. Бабенко Л.П., Ющенко Е.Л. "Управляющие системы и машины", 1982, №6, с.79-83.
7. Ершов А.М. ОИЯИ, 10-83-405, Дубна, 1983.
8. Ершов А.М. ОИЯИ, 10-82-686, Дубна, 1982.
9. Ершов А.М. ОИЯИ, 10-83-404, Дубна, 1983.
10. Любимский Э.З., Малинкин А.В. Современные диалоговые редакторы. Изд-во ИПМ АН СССР, М., 1979, с.51.
11. Ершов А.М. ОИЯИ, 10-83-889, Дубна, 1983.

Рукопись поступила в издательский отдел
22 ноября 1984 года.

Ершов А.М. 10-84-746

Некоторые вопросы прикладного проектирования в среде системы управления базами данных ОКА

На основе опыта эксплуатации системы управления базами данных ОКА рассмотрены некоторые вопросы прикладного проектирования в ее среде. Описаны процедура генерации программ загрузки баз данных, средства доступа к описаниям данных, хранимых системой ОКА, а также программа форматной распечатки этих описаний. Показан метод динамического построения аргументов поиска сегмента, обеспечивающий гибкое формирование критериев отбора информации в базах данных. Рассмотрены организация просмотра дневного файла заданий, редактирование и отладка прикладных программ в оперативном режиме СУБД ОКА.

Работа выполнена в Лаборатории вычислительной техники и автоматизации ОИЯИ.

Препринт Объединенного института ядерных исследований. Дубна 1984

Перевод О.С.Виноградовой

Ershov A.M. 10-84-746