

ГОССТРОЙ УССР

Научно-исследовательский институт автоматизированных систем планирования и управления в строительстве

НИИАСС



" Утверждаю "

Директор института, к.т.н.

Е.П. Дуброва

" \_\_\_\_\_ 1978г.

О Т Ч Е Т

О НАУЧНО-ИССЛЕДОВАТЕЛЬСКОЙ РАБОТЕ

ОПРЕДЕЛИТЬ ОСОБЕННОСТИ ПРОЕКТИРОВАНИЯ ИНФОРМАЦИОННОГО ОБЕСПЕЧЕНИЯ ПРИ ИСПОЛЬЗОВАНИИ СУБД, МОДУЛЬНЫХ СИСТЕМ ПРОГРАММИРОВАНИЯ, ГЕНЕРАТОРОВ ОТЧЕТОВ

( 0.80.08.01.Д7 )

" Согласовано "

Зав.сектором АСИ СОУ ЦНИИАСС  
руководитель работ по проблеме  
0.80.08.01.Д7

С.П. Никаноров

" 12 " 12 1978 г.

Зам. директора института  
по научной работе, к.т.н.

А.И. Кондратенко

" " \_\_\_\_\_ 1978г.

Научный руководитель темы, к.т.н.

В.А. Бисноватия

" 12 " XII 1978г.

Ответственный исполнитель,  
зав. лабораторией программирования

В.И. Танцур

" 12 " декабря 1978г.

38-7

**ПРОТОКОЛ**  
**сдачи отчета о научно-исследовательской**  
**работе**

г. Москва

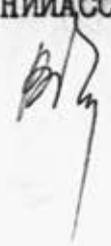
12.12.78

Харьковский отдел НИИАСС Госстроя УССР сдал, а ЦНИПИАСС Госстроя СССР принял отчет о НИР на тему "Выбор методов обеспечения надежности и достоверности выполнения информационных функций", выполненной в соответствии с краткой программой работ по проблеме 0.80.08 основное задание 01 пункт Д7.

Отчет выполнен в соответствии с согласованной рабочей программой и замечаний по нему не имеется.

Отчет сдал:

Руководитель работ по теме  
от НИИАСС



В. А. Бисноватый

Отчет принял:

Зав. сектором АСП СОУ ЦНИПИАСС,  
руководитель работ по проблеме  
0.80.08.01.Д7



С. П. Никаноров

## Список исполнителей:

- Бисноватый В.А., к.т.н., научный руководитель  
Колесник Т.Ю., зав. лабораторией  
Кондратенко А.И., зам. директора по научной работе,  
к.т.н., научный руководитель  
Танцур В.И., зав. лабораторией, ответственный  
исполнитель  
Штагер В.И., зав. лабораторией

## РЕФЕРАТ

Настоящий отчет является научно-исследовательской работой, выполненной по теме "Разработать руководство по проектированию машинной реализации информационной базы АСУС (6/4.00.)", имеющий выход по результату "Определить особенности проектирования информационного обеспечения при использовании СУБД, модульных систем программирования, генераторов отчетов (6/4.10)".

Отчет состоит из четырех разделов и введения. В первом разделе приводится обзор существующих систем формирования и управления базой данных: общая характеристика систем; состав банков данных (БД); возможности БД; классы БД; принципы организации баз данных. Дается характеристика конкретных систем, таких как: генератор программ ввода данных для ЕС ЭВМ (ГВВ); секторно-ориентировочная система программирования (НСИ-I-ДОС); общесистемное математическое обеспечение типовой АСУ на базе ЕС ЭВМ (ОСМО); система интегрированной обработки данных (СИОД); банк данных универсальной структуры (БАНК); математическое обеспечение банка данных (ОКА), а также приводится анализ и сравнительные характеристики рассматриваемых систем.

Второй раздел посвящен процессам проектирования внутримашинного информационного обеспечения: проектированию организации информационной базы; проектированию обработки данных в ЭВМ.

Третий раздел посвящен методам выбора систем математического обеспечения для создания, ведения и использования информационной базы: анализируются критерии оценки СУБД; характеристики информационной базы предприятия; приводится таблица характеристик конкретных систем.

В четвертом разделе приводятся выводы и предложения по выбору системы формирования и ведения базы данных для АСУ в строительстве (ДСК).

## СОДЕРЖАНИЕ

	Стр.
Введение .....	6
I. Обзор существующих систем формирования и управления базой данных .....	9
I.1. Общая характеристика систем .....	9
I.2. Генератор программ ввода данных для ЕС ЭВМ (ГВВ) .....	13
I.3. Секторно-ориентировочная система программирования (ИСИ-I-ДОС) .....	16
I.4. Общесистемное математическое обеспечение типовой АСУ на базе ЕС ЭВМ (ОСМО) .....	17
I.5. Система интегрированной обработки данных (СИОД) .....	21
I.6. Банк данных универсальной структуры (БАНК) .....	24
I.7. Математическое обеспечение банка данных (ОКА) .....	28
I.8. Анализ и сравнительные характеристики рассмотренных систем .....	35
2. Особенности проектирования внутримашинного информационного обеспечения .....	37
2.1. Характерные особенности информационного обеспечения в традиционных АСОД .....	37
2.2. Общие характеристики баз данных, требования по их организации и обработки в условиях БД .....	38
2.3. Особенности проектирования внутримашинного информационного обеспечения в условиях наличия БД .....	41
2.4. Уточнение входов в процессы проектирования .....	43

	Стр.
3. Методы выбора СМО для создания, ведения и использования информационной базы АСУ .....	45
3.1. Критерии выбора систем .....	45
3.2. Характеристика возможностей и ограничений СУБД .....	47
4. Выводы и предложения по выбору СМО для формирования, ведения и управления информационной базой АСУ ДСК .....	52
Литература .....	54

## В В Е Д Е Н И Е

В основных направлениях развития народного хозяйства СССР на 1976-80 г.г., принятых на XX съезде КПСС, указывается на необходимость "Обеспечить дальнейшее развитие и повышение эффективности автоматизированных систем управления . . .". Эффективность автоматизированных систем управления выражается в повышении технико-экономических показателей работы предприятия за счет улучшения организации производства, повышения обоснованности производственных планов, более рационального использования материальных и трудовых ресурсов, улучшения качества выпускаемой продукции. Однако практика показала, что многие автоматизированные системы управления в строительстве (АСУС), созданные в 9 пятилетке, не дали желаемого эффекта. Причин появления малоэффективных АСУС много. Одна из них - ограничение возможности ЭВМ второго поколения для обработки больших объемов информации и решения оперативных задач управления производством в реальном масштабе времени. В АСУС, созданных на базе ЭВМ 2-го поколения фонд ИСИ на машинных носителях представляет собой набор массивов данных, ориентированных на использование в заранее определенной группе задач. Для употреблений этих же данных другими задачами массивы обычно переупорядочивались или создавались новые.

Традиционная организация данных в виде специализированных массивов не могла обеспечить эффективное совместное использование общих данных, потому что она:

- не поддерживает между элементами данных динамические ассоциации, а поэтому затрудняет точное отображение сложного и изменяющегося объекта;
- требует частой реорганизации массивов и соответственно, изменений и перетрансляции программ в условиях непредска-

зубого характера использования данных и жестких требований к времени отклика;

- приводит к многократному дублированию данных в нескольких массивах, хранение которых требует большого расхода памяти, а логически целостная эксплуатация - большого труда и внимания.

С появлением машин третьего поколения, обладающих внешней памятью с прямым доступом и развитой операционной системой, появилась возможность организации эффективных информационных систем, получивших название банков данных (БД).

Потребность в банках данных может рассматриваться как закономерный результат развития автоматизированных систем обработки данных. Это развитие характеризуется переносом акцента в архитектуре вычислительных систем с процедур обработки на структуру и хранение данных. Такая переориентация затрагивает все программное обеспечение: операционные системы, системы управления данными и системы программирования.

Банк данных обеспечивает централизованное хранение всей необходимой для решения задач АСУС информации в виде взаимосвязанных массивов и выдачу пользователю той информации, которая необходима ему в данный момент.

Централизация информационного обеспечения АСУС имеет значительные преимущества перед созданием изолированных массивов для решения конкретных задач. Поэтому у нас в стране широким фронтом ведется работа по созданию информационных систем типа БД.

Для АСУ ДСК необходимо выбрать БД, в наибольшей степени удовлетворяющий требованиям предприятия (по объему информации, возможности внесения частных корректировок, технической оснащенности и т.д.).

Настоящий отчет содержит обзор известных в нашей стране систем информационного обеспечения АСУ и некоторые предложения по их

выбору.

В обзор включены системы по формированию и ведению баз данных генерирующего <sup>и</sup> интерпретирующего типа ("Генератор программ ввода данных (ГВВ)", "Общесистемное математическое обеспечение (ОСМО)", "Секторно-ориентировочная система программирования (НСИ-1-ДОС)"), и системы управления базами данных ("Система интегрированной обработки данных (СИОД)", "Банк данных универсальной структуры (БАНК)", "Математическое обеспечение банка данных (ОКА)").

Указанные системы имеют различные принципы организации и возможности, и работают под управлением различных операционных систем.

# І. ОБЗОР СУЩЕСТВУЮЩИХ СИСТЕМ ФОРМИРОВАНИЯ И УПРАВЛЕНИЯ БАЗОЙ ДАННЫХ

## І.І. Общая характеристика систем

Анализ АСУ, внедренных в девятой пятилетке на базе ЭВМ второго поколения показал, что структура информационного обеспечения в этих системах, по существу, повторяет структуру, сложившуюся в традиционных системах управления предприятием. Нормативно-справочная информация группируется под определенную группу задач, т.е. создается и ведется обособленно от решения функциональных задач. Оперативная информация вводится в память системы в виде изолированных массивов для решения конкретных задач.

При таком подходе к созданию информационного обеспечения структура системы обработки данных в АСУ приобретает вид ветвящегося дерева, количество формируемых и используемых в АСУ массивов резко возрастает. Это порождает многократное дублирование данных, и отсюда — значительный перерасход памяти и увеличение стоимости хранения и обслуживания данных. Такие системы не дают возможности обеспечить строгое соответствие хранимых данных реальным характеристикам отображаемого ими объекта при жестких ограничениях на время выдачи оперативной информации.

Состав БД. Требования новых применений привели к необходимости разработки систем, ориентированных на данные. В этих системах основной упор делается на непрерывное поддержание в системе динамической информационной модели сложного управляемого объекта. Такие системы получили название банк данных (БД). БД состоит из специальным образом организованной совокупности данных большого объема и сложной структуры, хранимых в памяти прямого доступа, и комплекса программных средств для создания, обновления, поддержки и использования содержимого БД, т.е. системы управления базой

данных (СУБД), а также комплекса технических средств, обеспечивающих хранение и доступ к данным.

Кроме программных средств, СУБД включает язык описания данных (ЯОД) и язык манипулирования данными (ЯМД). ЯОД, являясь независимым языком, представляет средства для задания физической и логической структуры базы данных, т.е. его назначение описывать отношения, которые существуют и должны соблюдаться между элементами базы данных.

При создании СУБД определяется некоторый базисный набор функций, реализация которого позволяет удовлетворить основные требования пользователей к БД.

Выполнение базисных функций БД обеспечивается с помощью ЯМД. К таким функциям относятся:

создание файлов базы данных;

поиск в базе данных на основе многочисленных критериев;

ведение базы данных;

реорганизация базы данных с целью лучшего использования памяти.

Рассмотрев функции СУБД следует отметить, что не каждая СУБД имеет хорошо развитый аппарат по ведению базы данных (ввод с перфоносителя, корректировка, контроль величин во-время ввода). Поэтому, часто наряду с использованием СУБД, следует использовать и системы по формированию баз данных (ввод, контроль, корректировка). Существует и ряд других причин выступающих за совместное использование данных систем, которые мы рассмотрим ниже.

Назначение БД. Ознакомившись с составными частями БД и основными требованиями к БД, которые обусловлены их главным назначением - служить центром накопления и поддержания в актуальном состоянии общих данных сложной структуры, используемых различными пользователями в различных задачах, а именно, они обеспечивают:

единство использования данных всеми существующими задачами

в АСУ;

независимость прикладных программ от структуры и состава данных;

сокращение избыточности элементов информации;

возможность поэтапного и непрерывного наращивания емкости базы данных, т.е. динамического способа формирования;

целостность данных;

выполнение базисного набора функций;

постоянную готовность к выдаче данных по запросам пользователей.

Классы БД. Различают два класса СУБД – системы с базовым языком и системы с замкнутой организацией.

Система с базовым языком – это система, которая строится на основе процедурного языка PL/I, КОБОЛ или языка АССЕМБЛЕРА.

Необходимо подчеркнуть, что пользователь системы с базовым языком осуществляет управление логической последовательностью выполнения своей программы и может употреблять любые операторы по своему усмотрению. Расширенная структура данных представляет пользователю более широкие возможности при работе с некоторыми структурами данных, чем в случае, если бы он пользовался только базовым языком. Естественно, что структура физической памяти остается скрытой от пользователя, хотя в своей программе он должен указывать тип носителей и определять структуру хранения базы данных на уровне файлов.

БД с замкнутой организацией предназначены для реализации определенного набора функций без использования традиционного процедурного программирования. Эти функции один раз программируются на процедурном языке с возможностью задания их параметров в языке высокого уровня.

В зависимости от условий и требуемых действий в системах с замкнутой организацией имеется возможность использовать предварительно запрограммированный или встроенный алгоритм, благодаря чему экономится время работы пользователя.

Недостаток систем, обладающих лишь замкнутыми возможностями, состоит в ограниченности набора их приложений. В рамках этих приложений такие системы осуществляют высокую степень автоматизации обработки данных и требуют лишь небольших усилий пользователя. Однако при этом неизбежно уменьшается универсальность и гибкость системы.

Функции банка данных. Создание базы. Создание начального экземпляра базы данных является одной из важных функций администратора данных. Она включает: определение данных, выбор типа носителей, создание набора записей для начальных экземпляров файлов и проверка достоверности данных.

Задание и распределение мест для базы данных на носителях ЭВМ оформляется внутри системы.

В процессе формирования файлов обычно имеется возможность установления достоверности данных путем проверки значений элементов, упорядоченности записей внутри файла, контрольных сумм и т.д. Некоторые системы предусматривают выдачу на печать уже сформированного файла базы данных для обеспечения возможности восстановления его в случаях неисправности оборудования, возникающих в ходе процесса создания базы данных.

Обновление базы данных. Обновление данных определяется как процесс изменения значений во всех или только выбранных записях, сегментах или элементах, хранящихся в базе данных. Обычно различают следующие виды обновления:

удаление записей или сегментов из файла;

вставка новых записей или сегментов в файл;

модификация записей или сегментов;

замена ключевых элементов записей или сегментов.

В большинстве замкнутых систем заложена возможность внесения изменений в файл базы данных, используя данные из нескольких сообщений, и возможность одновременного внесения изменений в два и более файла на основании одного сообщения.

ЯМД представляет пользователю возможность идентификации той части файла, которая должна быть изменена сообщением, т.е. правила отбора данных. Наиболее общей формой реализации отбора данных является задание логических условий, которые должны быть удовлетворены до того, как сообщение будет применено к файлу.

Второй метод реализации отбора данных состоит в сравнении сообщений с файлом по ключам записей. Если ключи совпали, то заданная обработка выполняется без дополнительного задания пользователем критериев отбора данных.

Согласно третьему методу обновление выполняется не над некоторыми записями, а над всеми записями файла.

Справочная функция. Справочная функция — это получение справки о состоянии БД, т.е. выдача ответов на запросы пользователя.

Получение справки из БД — это процесс, состоящий из операций отбора и извлечения некоторой части базы данных, промежуточной обработки отобранных данных и выдачи пользователю в виде отпечатанных документов или в форме машинных файлов для последующей обработки.

## 1.2. Генератор программ ввода данных для ЕС ЭВМ (ГВВ)

Генератор программ ввода данных для ЕС ЭВМ (ГВВ) создан для повышения эффективности программ ввода, контроля и компоновки

информации в АСУ. Он относится к классу компилирующих генераторов и предназначен для программ ввода информации с документов сложной структуры.

ГВВ рассчитан на использование с дисковой операционной системой (ДОС) ЕС ЭВМ.

Назначение. ГВВ обеспечивает в сгенерированных рабочих программах выполнение следующих функций:

ввод входной информации с любого из машинных носителей (перфокарты, перфоленты, МД, МЛ);

обработку разнообразных структур представления данных на носителях;

перекодировку вводимой информации во внутренний код представления данных в ЕС ЭВМ;

различные методы логического контроля данных (контроль строки, контроль пачки строк, контроль структуры файла);

выдачу на печать сообщений об ошибках во вводимой информации;

корректировку ошибочных входных данных с целью обеспечения достоверности вводимой информации;

редактирование вводимой информации во внутренние форматы представления данных в ЕС ЭВМ;

формирование любого числа выходных файлов.

Структура базы. В ГВВ принята иерархическая последовательная организация входного файла. Логическими единицами информации входного файла являются структура, строка, реквизит.

ГВВ допускает иерархическую зависимость реквизитов в строке, древовидную иерархию строк. Входной файл может включать различные типы структур.

ГВВ позволяет создавать рабочие программы двух типов. К первому типу относятся программы, в которых не предусматривается

система корректировки ошибочных данных. Обработка информации в этих программа производится за один этап, включающий ввод данных, их перекодировку, контроль и выдачу сообщений об ошибках, а также формирование выходных файлов.

Ко второму типу относятся рабочие программы, в которых предусматривается наличие системы корректировки. В процессе выполнения рабочей программы второго типа: также производится ввод входного файла, перекодировка и контроль ввода. Затем информация выводится в промежуточный файл, после чего выполняется процедура, логического контроля промежуточного файла. На основании сообщений об ошибках пользователь может подготовить файл корректур ошибочных данных. Затем осуществляется корректировка конкретного промежуточного файла и, в случае достоверности последнего, формирование выходных файлов.

В качестве носителей выходных файлов, получаемых с помощью ГВВ, могут использоваться МЛ, МД и ПК. Причем, на дисках допускаются выходные файлы с прямой и последовательной организацией. Вывод выходных файлов осуществляется с помощью логической системы ввода-вывода ДЭС ЕС. Поэтому выходные файлы удовлетворяют всем требованиям ДЭС ЕС.

На каждый образующийся новый файл (или группу файлов) ГВВ требует генерации, а при изменении структуры файла или его реквизитов, регенерации рабочей программы, что является недостатком существующей системы.

К недостаткам ГВВ следовало бы также отнести следующие:

отсутствие модулей печати основных файлов;

отсутствие модулей корректировки основных файлов;

использование логической СУВВ.

ГВВ разработан ЦНИИТУ.

Для его функционирования необходима оперативная память 64К,

2 накопителя на МД.

### 1.3. Секторно-ориентировочная система программирования (НСИ-1-ДОС)

НСИ-1-ДОС является системой с замкнутой организацией и, обеспечивает выполнение стандартных функций СУБД - загрузку базы, обновление данных, выдачи справок.

Система обслуживает изолированные файлы базы данных, располагающиеся на МЛ или МД, имеющие последовательную или индексно-последовательную организацию и однотипные записи. Обслуживаемый файл называется основным. В процессе обслуживания система организует и обрабатывает такие файлы:

- ТОД (таблица описания данных);
- входной;
- рабочий;
- выходные.

Таблица описания данных составляется пользователем с помощью инструкций языка описания данных для каждого файла и является связующим звеном между файлами базы данных и системой управления базой данных, обеспечивая независимость программ обработки от данных.

Входной файл может быть подготовлен на ПК, ПЛ, МЛ или МД содержать записи различных типов. Входной файл на МД может иметь последовательную или индексно-последовательную организацию.

При загрузке основного файла система последовательно выполняет следующие операции: вводит входной файл, проверяя его на формальную правильность, подготавливает входные данные и выводит их в основной файл. При этом выполняется сортировка по возрастанию значения ключевых полей входных записей. Во время загрузки данных

в основной файл может осуществляться контроль и преобразование данных, обеспечивающие подключение в определенных точках программ пользователя.

Система НСИ-I-ДОС представляет пользователю возможность выдачи основного файла на печать. При этом по оформлению печатного листа возможны: многострочная выдача записи, вывод пустых строк, строк заголовка, шапки, а также последней строки листа.

За один программный цикл система выполняет только одну из своих основных функций - определение данных, загрузка, поиск - для одного файла. Если нужно обслужить другой файл или выполнить другую функцию, программа вызывается вновь.

Система НСИ-I-ДОС работает под управлением ДОС/ЕС и состоит из взаимосвязанных программных модулей, которые являются секциями программ, реализующих отдельные подзадачи. Отдельные модули запрограммированы с возможностью самоперемещения, что позволяет использовать систему в режиме мультипрограммирования. Пользователь работает непосредственно только с двумя управляющими модулями, которые на основании функций, заданных пользователем, вызывают для выполнения секции программ.

Система НСИ-I-ДОС не требует специальной генерации.

Система разработана комбинатом Роботрон (ГДР) и адаптирована для ЕС ЭВМ ЦНИИТУ (г. Минск).

Для функционирования системы необходима оперативная память объемом 64К, 3 накопителя на МД (включая операционный пакет дисков).

#### I.4. Общесистемное математическое обеспечение типовой АСУ на базе ЕС ЭВМ (ОСМО)

Банк данных в ОСМО ЕС ЭВМ определяется как совокупность логически зависимых или (и) независимых баз данных.

База данных. База данных состоит из совокупности классов данных. Формально вся база данных делится на два класса:

логический класс данных (ЛКД);

физический класс данных (ФКД).

Логический класс данных включает информацию, которая адекватно отображает структуру и содержание внешних источников информации - документов.

Логический класс данных служит для целей контроля и коррекции базы данных при ее первоначальном создании. Связь пользователя с системой выполняется как через логический, так и через физический класс данных. Представление информации в этом классе в большинстве случаев текстовое. Метод организации - иерархически-последовательный.

Физический класс данных создается на базе логического с учетом минимизации объема памяти, занимаемой данными, и служит для целей внутримашинной обработки.

Каждый из файлов ФКД содержит однородную информацию об объекте. Сегмент файла представляет собой запись постоянной длины.

В ОСМО файлы логической базы данных называются архивными, физической - основными.

Логические элементы архивного файла это документ, строка, реквизит строки. Конкретный архивный файл содержит информацию о документах однородной структуры, т.е. каждый архивный файл содержит структуры только одного типа.

Архивные файлы в большинстве случаев хранятся на МЛ, так как являются вспомогательным звеном в процессе создания основных файлов. Все преобразования базы данных при решении конкретной задачи, выполняются над основными файлами, которые уже хранятся на МД.

Файлы логической и физической баз данных на МЛ и МД организуются в соответствии с требованиями логической СУБВ ДОС ЕС.

Доступ к файлу (архивному или основному) реализуется по идентификатору файла посредством использования комплекса программ обмена (КПО) ОСМО. По умолчанию КПО реализуется доступ к первому экстенду файла (архивного или основного) с дальнейшим переходом ко всем остальным его экстендам. Применением специальных средств можно добиться обработки файла с любого экстенда или даже любой физической записи.

Метаинформация. Для возможности достижения независимости прикладных программ от структуры и размещения элементов баз данных, в ОСМО ЕС ЭВМ описания данных создаются независимо от классов базы данных и программных средств прикладного характера.

Часть информации, необходимой для решения задач и представляющей собой данные описательного характера (описания записей, таблицу операций и т. д.), а также переменные параметры входов в программы (фазы), посредством которых решается задача, в ОСМО принято называть метаинформацией.

Доступ к системе метаинформации реализуется посредством задания операторов управления задачами (ОУЗ) и описания данных (ОД - предложений).

База метаданных единая для всех задач, решаемых в рамках АСУ.

Загрузка базы метаданных, доступ к классам в базе метаданных в ОСМО ЕС ЭВМ полностью автоматизированы и разрешаются на уровне уникальных имен классов метаданных.

Базе метаданных присущи два типа организации данных:

библиотечный;

файловый.

Программное обеспечение. Программное обеспечение системы управления базой данных в ОСМО ЕС ЭВМ построено по принципу адаптации программных средств. Принцип адаптации программных средств состоит в возможности авторегулирования программ в

зависимости от изменения входных параметров, определяемых пользователями.

Программы ОСМО, построенные по принципу адаптации, отличаются универсальностью и гибкостью возможностей и по быстродействию не уступают программам, построенным по принципу генерации.

Высокая степень автоматизации процессов создания и обработки баз данных в ОСМО ЕС ЭВМ в значительной мере возможна благодаря наличию комплекса программ обмена (КПО).

КПО представляет пользователю ряд средств, позволяющих создавать и обрабатывать файлы в соответствии с особой, иногда очень необычной спецификой каждой конкретной задачи. Однако, среди всех возможностей КПО в первую очередь важной является возможность работы с файлами без какого-либо дополнительного участия и вмешательства пользователя. Имеется в виду стандартный режим работы КПО, если не используются ни таблицы пользователя (ТП), ни блок пользователя (БП), ни блок обработки сбоев (БЭС). Работа с КПО в таком режиме довольно проста и требует от пользователя знакомства с методами доступа, описанием подпрограмм КПО и связью КПО с МОНИТОРОМ.

Принцип построения КПО модульный. Допускаются любые расширения и сужения его возможностей, при этом не требуется коррекции ядра КПО, а только изменение состава комплекса за счет введения или удаления некоторых программных единиц.

Высшая информационная единица обмена — файл. КПО обеспечивает параллельную обработку до 7 файлов, при этом запоминание информации о состоянии обработки каждого файла производится автоматически, с тем, чтобы при очередном обращении к файлу она могла быть восстановлена без вмешательства пользователя.

ОСМО разработано лабораторией системного программирования Научно-исследовательского института автоматизированных систем

управления (г.Киев). Для эксплуатации системы необходим полный комплект технических средств ЕС ЭВМ начиная с ЕС-1020 и выше.

### 1.5. Система интегрированной обработки данных (СИОД)

Семейство СИОД насчитывает четыре системы: СИОД1, СИОД2, СИОД3-работающих под управлением дисковой операционной системы и, СИОД ОС - работающей по управлению ОС. Система СИОД позволяет достаточно эффективно решать проблему организации нормативного хозяйства путем создания фонда НСИ на носители информации с прямым методом доступа.

Назначение системы. Структура базы данных СИОД, в основном, определяется тем, что она предназначена для хранения и обработки, главным образом, конструкторско-технологической информации и объединяет данные об изделиях, их структуре, материалах, технологии изготовления изделий и о рабочих местах в единую интегрированную систему.

Основными функциями СИОД являются:

- создание и ведение базы данных, состоящей из четырех файлов<sup>\*</sup>;
- решение задач технической подготовки производства;
- формирование на машинных носителях файлов НСИ, необходимых для решения функциональных задач АСУ.

База данных. Базы данных, создаваемые системами интеграции и обработки данных строятся из файлов двух типов - главных и связующих.

---

\* Одно из семейств СИОДа (СИОД3) позволяет создавать и вести базу данных состоящую из 20 файлов, но при этом пользователь должен самостоятельно переписать головной модуль системы СИОД.

В простейшей форме главный файл представляет собой набор информационных записей, имеющих общие средства идентификации при условии, что каждая из них снабжена уникальным, т.е. однозначно определяющим данную запись в этом файле, идентификатором. В свою очередь запись главного файла — это совокупность полей данных, логически связанных с определенным идентификатором. В качестве общего средства идентификации записей главного файла выступает ключ записи, который может состоять из одного или нескольких (обязательно смежных) управляющих полей.

Программы пользователя могут обращаться к записям главного файла одним из следующих способов:

- в логической последовательности, т.е. не указывая ни уникальный идентификатор (ключ) записи, ни ее адрес в ЗУПД;
- указывая программным модулям СИОД только ключ записи;
- указывая программным модулям СИОД только адрес записи;
- в порядке физического размещения записей в памяти ЗУПД (в физической последовательности).

Между записями одного или нескольких главных файлов могут быть установлены перекрестные связи, которые организуются с помощью связующих файлов.

Связующие файлы представляют второй тип файлов, образующих базу данных. Файлы этого типа, выполняя функции логического расширения главных файлов и являясь средством организации связей между данными, позволяют значительно уменьшить степень избыточности информации, хранимой в базе данных. Именно связующие файлы дают нам право определить базу данных как систему взаимосвязанных данных.

Связующие файлы имеют списковую структуру, т.е. представляют собой совокупность групп (списков) записей фиксированной длины, начало которых указывается в записях главных файлов.

Число записей в группах переменное. Каждая запись группы содержит адресный указатель на следующую запись этой группы. Таким образом, все записи одной группы связаны в цепочку.

Основная цель связующих файлов — установление двусторонней взаимосвязи между главными файлами или, другими словами, организация перекрестных ссылок между данными.

Системы интеграции и обработки данных позволяют с помощью одного связующего файла установить перекрестные взаимосвязи между пятью главными файлами.

СИОД позволяет создавать и обслуживать следующие массивы:

главный предметный массив (ГП);

связующий массив состава изделий (СИ);

связующий массив пооперационных трудовых нормативов (ПТН);

главный массив рабочих мест (РМ);

Массивы ГП и РМ являются по своему характеру главными, а массивы СИ и ПТН — массивами связей.

Связь между отдельными предметами осуществляется посредством адресных ссылок через записи массива СИ. Каждая такая запись соединяет пару записей в РМ и содержит данные, характерные только для этой связи.

Массив ПТН содержит по одной записи на каждую технологическую операцию обработки детали или сборки данного изделия. Записи массива ПТН имеют цепные связи с главными массивами ГП и РМ.

Поля записей массивов базы данных можно разделить на два вида: системные поля и поля пользователей. Наличие системных полей в записях является обязательным.

Каждая запись в массиве ГП содержит начальные адреса соответствующих цепочек в массиве СИ и ПТН. Аналогично, каждая запись в массиве РМ содержит начальные адреса цепочки в массиве ПТН. Каждая запись в массиве СИ содержит адреса двух записей

массива ПП:

адрес записи сборочной единицы непосредственно более высокого уровня;

адрес записи компонента непосредственно более низкого уровня,

Записи файла ПТН также содержат два адреса. Один является адресом записи в файле РМ, другой - адресом записи файла ПП, соответствующей изделию, к которому относится технологическая операция. Кроме того, всякая запись из файла СИ и ПТН содержит адреса двух записей этого же файла, а именно: адрес следующей записи в первичной цепочке и адрес следующей записи в цепочке использования.

Конфигурация технических средств. Для работы системы интеграции и обработки данных необходим полный комплект технических средств соответствующей модели ЕС ЭВМ (начиная с ЕС-1020 и выше).

Следует отметить, что накопители на магнитных дисках необходимы для размещения компонентов операционной системы и файлов данных пользователя. Минимальное требование - два накопителя для ДЭС, три накопителя для ОС. Реорганизация связующего файла требует наличия двойного объема самого связующего файла плюс объем связанного с ним файла.

## 1.6. Банк данных универсальной структуры (БАНК)

Система "БАНК" принадлежит к информационным системам с базовым языком и представляет собой комплекс средств, предназначенных для управления базой данных, в которой состав данных и их взаимосвязи определяются пользователем.

Конфигурация системы. К средствам системы "БАНК" относятся: метод организации хранимых данных (набор правил), позволяющий пользователю разработать структуру файлов базы данных в соответствии с установленными требованиями;

пакет прикладных программ, реализующих универсальные процедуры доступа к файлам базы данных, а также некоторые служебные функции;

язык описания структуры файлов и обращение к универсальным процедурам.

База данных. База данных может состоять из одного или нескольких (до пяти) файлов и занимает одно или несколько запоминающих устройств на магнитных дисках. Каждый файл является самостоятельной частью базы данных, поэтому пользователь имеет возможность работать с ним так как со всей базой.

Организация файлов — произвольная. Физической единицей данных является страница; ее длина определяется разработчиком с учетом структуры файла и физических характеристик диска. Общее количество страниц во всех файлах базы не должно превышать 65535.

Способ организации данных позволяет отобразить иерархические связи, структуру типа графа, перекрестные связи между последовательными массивами и т.д.

Основной единицей, хранимой в системе информации, является запись. Запись состоит из одного или нескольких полей данных, относящихся к одному или нескольким ключевым полям (ключам). Кроме полей данных, состав и характеристики которых задает пользователь, записи базы данных содержат служебные поля (поля идентификации и поля цепей), которые создаются и обрабатываются только системой "БАНК".

База данных может включать в себя разнородные данные, которые объединяются между собой в группы при помощи цепных связей.

Логической единицей данных является запись. База данных может включать в себя записи различных типов, т.е. различного состава и длины. Может быть использовано до 255 типов записей, длина которых ограничена только размером дорожки.

Связь между записями реализуется с помощью цепей. Цепь - это логическая последовательность записей в виде замкнутой петли, в которой каждая запись содержит адрес следующей записи. Запись может входить в любое количество цепей.

Каждой записи пользователь должен присвоить уникальное имя и тип. Имя может содержать от одного до восьми символов и обязательно начинаться с латинской буквы. Присвоенный записи тип хранится в поле идентификации записи. Тип используется системой для связи с ее описанием. Количество записей одного типа не ограничено. Физически записи объединяются в блоки фиксированной длины-страницы. Каждой записи при помещении в файл присваивается логический адрес, который состоит из номера страницы и относительного адреса страницы.

Далее в запись включаются, по крайней мере, одно поле цепи на каждую цепь, в которую входит запись. Значением поля цепи является код ссылки другой записи, представляющей собой относительный логический адрес записи данных. Остальную часть записи составляют поля данных, определенные пользователем. Их количество ограничено лишь максимальным размером записи (длиной дорожки).

Программное обеспечение. Программные средства системы "БАНК" включают в себя блок универсальных процедурных модулей, которые реализуют доступ к данным. Универсальность процедур состоит в том, что они могут настраиваться на конкретную структуру базы данных. Система обеспечивает язык описания структуры файлов и язык обращения к процедурам доступа. Кроме того, программные средства включают набор макроопределений. Язык макрокоманд соответствует

требованиям входного языка макрогенератора ДЭС/ЕС. Параметры всех макрокоманд являются ключевыми. Система "БАНК" включает также программу формирования таблиц, резидентную программу и обслуживающие программы.

Для создания базы данных пользователь должен:

— разработать структуру базы данных, исходя из требований своих задач и возможностей банка данных;

— оценить требуемый объем памяти на МД, выбрать длину страницы и определить количество страниц в базе данных;

— подготовить МД при помощи обслуживающей программы;

— подготовить исходные данные для создания базы данных;

— разработать программу создания файла или воспользоваться обслуживающей программой реорганизации.

Система "БАНК" разработана Научно-исследовательским институтом управляющих машин и систем (г.Пермь). Нормальное функционирование системы "БАНК" обеспечивается стандартным набором устройств модели ЕС 1020. Необходимость подключения дополнительных МД определяется объемом информации, включенной в базу данных.

## 1.7. Математическое обеспечение банка данных (ОКА)

Система управления информацией ОКА – это обрабатывающая система программ, работающая под управлением ОС ЕС, которая обеспечивает реализацию как средних, так и больших по объему баз данных самых различных применений. Эта система может работать как в режиме оперативной обработки, так и в режиме пакетной обработки, причем обработка сообщений может вестись одновременно в двух режимах. Система ОКА является мощным средством для разработки АСУ крупных предприятий, научно-производственных объединений, отраслей. Система ОКА может использоваться совместно с другими системами информации.

Назначение системы. Основными назначениями системы ОКА являются:

обеспечение методами организации данных для создания, взаимосвязи и ведения больших баз данных и для использования этих баз в различных областях применения;

обеспечение средствами обработки и обслуживания баз данных в режиме пакетной обработки;

обеспечение возможности расширения баз данных и их приспособляемости к телеобработке и, режиму передачи данных;

обеспечение эффективной подсистемы телесвязи для организации и создания системы с большим объемом обрабатываемой информации и малым временем ожидания.

База данных. База данных системы ОКА состоит из записей данных. Запись базы данных представляет собой совокупность элементов фиксированной длины, называемых сегментами.

Сегмент представляет собой часть базы данных, которая содержит одно или более логически связанных полей данных. Существует два основных вида сегмента: корневой и зависимый.

Корневой сегмент - это сегмент, имеющий наивысшую иерархию в записи данных. Каждая запись базы данных должна иметь только один корневой сегмент.

Корневой сегмент содержит данные, которые доступны всем пользователям, обрабатывающим запись данных.

Зависимый сегмент - это сегмент, полное иерархическое значение которого зависит от корневого сегмента, т.е. зависимый сегмент всегда находится на более низком иерархическом уровне, чем корневой сегмент.

Система ОКА имеет две конфигурации: систему баз данных /ДВ/ и систему баз данных со средствами передачи данных /ДВ/ДС/.

Система ДВ обрабатывает входные рабочие потоки в пакетном режиме; система ДВ-ДС является проблемно-ориентированной и планирует работу на основе входных сообщений. Эта система создана путем объединения модулей системы ДВ и модулей средств обеспечения передачи данных /ДС/.

Система ОКА состоит из двух основных частей:

средств обработки данных;

средств передачи данных.

Средства обработки данных. Возможность обработки данных обеспечивается в системе ОКА с помощью языка ВЕТА. Обработка базы данных заключается в ее определении, создании, доступа к ней и обслуживании.

Все возможности обработки данных, которыми располагает язык ВЕТА, могут использоваться в режиме пакетной обработки и телеобработки.

Язык ВЕТА обеспечивает независимость прикладных программ от методов доступа, от физической организации памяти и от характеристик тех устройств, в которых хранятся нужные данные. Эта независимость обеспечивается с помощью общей символической,

программной связи, с помощью описаний баз данных, внешних по отношению к прикладной программе.

Язык ВЕТА обеспечивает исключение избыточных данных и облегчает совместное использование общих данных, т.е. способствует интеграции данных с минимальной избыточностью.

В случае, если анализ данных пользователя показывает, что все данные невозможно разместить в одной общей базе данных, то благодаря языку ВЕТА можно организовать информацию в виде нескольких баз данных.

Общая символическая программа связи и описание баз данных позволяет прикладной программе запрашивать язык ВЕТА для следующих целей:

- отыскания конкретного сегмента /GET UNIQLLE /;
- отыскания следующего сегмента последовательности /GETNEXT /;
- исключения данных из имеющегося сегмента /REPLACE /;
- замены данных в имеющемся сегменте /DELETE /;
- включения нового сегмента /INSEPT /,

Общая символическая программа связи оперирует со следующими языками: КОБОЛ, ПЛ/I и АССЕМБЛЕР.

Описание базы данных создается на основе выдаваемых пользователем операторов, которые определяют логическую структуру данных и физическую организацию каждой базы данных. Эти операторы являются входными данными для вспомогательной программы системы ОКА, выполняемой в автономном режиме. Результатом выполнения этой программы будет создание и заполнение описания базы данных в определяемой пользователем библиотеке описания баз данных. Это описание баз данных позволяет языку ВЕТА "отобразить" используемую в прикладной программе логическую структуру базы данных в физической организации данных, используемой программой управления данными операционной системы ОС ЕС. Логическую структуру данных

можно "переотобразить" в другую физическую организацию, не изменив при этом программу. К этой базе данных могут быть добавлены данные другой области применения, но это не повлечет за собой изменения первоначальных прикладных программ.

Язык BEGA позволяет использовать две основные физические организации памяти: иерархическую последовательную, иерархическую прямую.

Иерархическая последовательная организация использует два метода доступа: иерархический последовательный /HSAM/; иерархический индексно-последовательный /HISAM/.

Иерархическая прямая организация использует тоже два метода доступа: иерархической прямой /HDAM/, иерархической индексно-прямой /HIDAM/.

Основное различие между иерархической прямой и иерархической последовательной организацией заключается в характере связей между сегментами и в методах доступа к данным.

В качестве основы для /HSAM/ используется последовательный метод доступа ОС ЕС /BSAM/, физические блоки /переменное число/, занимаемые определенной физической записью, базы данных, связаны между собой путем смежного расположения в памяти.

BSAM обеспечивает прямое обращение только к первому сегменту первой физической записи базы данных.

Основу HISAM составляет индексно-последовательный метод доступа ОС ЕС /ISAM/.

Каждая физическая запись базы данных берет начало в логической записи ISAM, в которой помещается столько сегментов физической записи базы данных, сколько она может вместить. Если листа для всех сегментов не хватает, то используется один или несколько блоков последовательного метода доступа для переполнения /OSAM/. Логическую запись ISAM и все физи-

ческие блоки *OSAM* базы данных связывают прямые адреса. Наборы данных, с которыми оперируют *OSAM*, определяются как стандартные физические последовательные наборы данных ОС ЕС.

Сегменты иерархической прямой организации, представляющие одну физическую запись базы данных хранятся в одном или нескольких физических блоках. Однако все сегменты этой физической записи, а не физические блоки, содержащие эту запись, связаны прямыми адресами. Все сегменты физической записи связаны как с сегментами того же типа так и с типами смежных сегментов путем прямой адресации. Физические блоки не связаны прямой адресацией.

Метод доступа *HIDAM* в качестве основы использует *OSAM*.

Метод доступа *HIDAM* используется для индексного доступа к иерархической прямой организации. Индекс *HIDAM* образуется с помощью *ISAM*. в каждой логической записи *OSAM* содержится ключ и прямой адрес корневого сегмента. Запоминание всех сегментов физически осуществляется в блоках *OSAM*. Запоминание индексов осуществляется в блоках *ISAM*. Поскольку набор данных для индексов запоминается отдельно от набора данных сегментов, можно производить неоднократное индексирование или реорганизацию индексов отдельно от данных.

Средства передачи данных. При использовании системы ОКА в режиме телеобработки язык ВЕГА обеспечивает интерфейс сообщений, поступающих с конечных устройств ввода-вывода. Этот интерфейс используется при всех обращениях к базе данных.

Исходя из требований пользователя, определенных типом решаемых задач, входные сообщения могут быть обработаны немедленно или же с некоторым интервалом. Один из терминалов /ЕС-8502/ или /ЕС-857Д/ действует в качестве основного терминала системы, с которого пользователь осуществляет управление иницированием повторного запуска по контрольным точкам, работой конечных устройств

и обработкой входных сообщений. Связь между отдельными терминалами или одной или несколькими программами может осуществляться в диалоговом режиме. Поэтому в системе ОКА предусматривается использование специального участка памяти для сохранения информации при диалоге. Этот участок может выделяться в оперативной памяти или же в памяти с прямым доступом.

Система ОКА инициирует <sup>И</sup> выполнение программ обработки сообщений в соответствии с типами полученных сообщений. Все приемлемые для системы типы сообщений заранее определяются и проверяются с помощью 8-элементного кода в первом сегменте сообщений. Когда разрешенное сообщение полностью получено и поставлено в очередь, об этом сообщается программе планирования. Выбор сообщений из очереди делается на основе приоритетов. Обработка сообщений начинается после того, как программа планирования выделит необходимые ресурсы. Допускается планирование двух или нескольких программ обработки сообщений, которые одновременно могут обновлять общую базу данных. Но две или более программы, которые предназначены для обновления одного типа сегмента: не могут планироваться одновременно.

Вспомогательные программы. В системе ОКА разработаны следующие вспомогательные программы:

программа определения систем, образующая блоки управления, используемые системой ОКА для определения конкретной среды, в которой будет работать программа пользователя;

программа обеспечения защиты данных и программа, создающая блоки управления, которые описывают требования пользователя к защите информации в процессе обработки данных;

программа анализа системного журнала, выдающая записи, касающиеся типа сообщений и работы терминала;

программа образования блока программной спецификации образующая управляющие блоки для определения характеристик

терминала и баз данных, предназначенных для использования конкретной, прикладной программы;

программа генерации описания базы данных, образующая управляющие блоки, предназначенные для описания каждой базы данных;

программа загрузки и реорганизации базы данных, представляющая собой обобщенную программу, способствующую образованию и реорганизации базы данных;

программа вывода на печать и записи в память базы данных, используемая при реконструкции базы данных;

программа восстановления базы данных, используемая для сбора и применения различных модификаций базы данных, которые ранее регистрировались в системном журнале и которые должны использоваться при реконструкции базы данных.

Конфигурация технических средств. Минимальный набор средств для работы системы ОКА помимо средств ввода-вывода, которые требуются для ОС ЕС, прикладных программ, не входящих в систему ОКА, и самой базы данных включают в себя:

один накопитель на магнитной ленте: ЕС-5010, необходимый для распределения и обслуживания ресурсов системы;

500 цилиндров ЗУ на дисках ЕС-5050 для хранения программ и рабочие области или эквивалентную память на дисках ЕС-5061.

В качестве дополнительных устройств ввода-вывода, используемых системой в режиме телеобработки, необходим, по крайней мере, один комплекс ЕС-8502, один комплекс ЕС-8501 и два накопителя на магнитной ленте ЕС-5010. Дополнительно могут быть использованы ЕС-8563, ЕС-8570, ЕС-8511.

Система ОКА в минимальной конфигурации требует оперативная память емкостью 128К для пакетного режима и емкостью 256К для режима телеобработки.

## I.8. Анализ и сравнительные характеристики рассмотренных систем

Рассмотрев существующие системы по формированию и управлению базой данных следует отметить наглядные тенденции в их развитии:

управление базой данных в первую очередь заключается в организации интегрированного накопления, хранения и поиска информации;

система должна обеспечивать возможность корректировки базы данных, т.е. включение новой информации или изменение устаревших данных;

система должна иметь возможность реорганизации базы данных через определенные интервалы времени;

система должна обладать аппаратом контроля правильности ввода исходной информации, так как она работает с большим объемом исходных данных. Модули контроля данных, должны не только выявлять имеющиеся при вводе данных ошибки, но и диагностировать их таким образом, чтобы пользователь мог легко внести исправления;

система управления базой данных должна быть независима от модели данных.

Эти общие тенденции, которые были замечены при рассмотрении конкретных систем.

Анализируя и сравнивая данные системы можно отметить их достоинства и специфические характеристики:

в ГВВ (НСИ-Г-ДОС)

входной файл может содержать информацию о документах разной структуры, в то время как архивный файл в ОСМО содержит информацию о документах однородной структуры;

в ГВВ требуется генерация рабочего модуля на каждый новый файл (или изменение старого), а в системах ОСМО, НСИ-I-ДОС достаточно заменить описание файла;

система НСИ-I-ДОС не располагает системными модулями контроля входной информации, а возлагает это на блоки пользователя;

система НСИ-I-ДОС может одновременно обслуживать только один файл (или выполнять одну свою функцию), ОСМО позволяет обслуживать одновременно до семи файлов;

в ГВВ отсутствует возможность печати основных файлов, такие возможности имеются в ОСМО и НСИ-I-ДОС;

система СИОД нацелена на конкретное применение (задана структура базы данных), НСИ-I-ДОС, БАНК, ОСМО, ОКА-универсальные системы;

система НСИ-I-ДОС (ОСМО) обладают свойствами независимости прикладных программ от выбранной структуры базы данных;

все рассмотренные системы (за исключением ОКА, СИОД) поддерживают иерархическую модель базы данных;

системы БАНК, ОКА, являются системами с базовым языком;

все системы имеют возможность работы с языком пользователя - Ассемблер, система ОСМО также и с языком ПЛ/I, а система СИОД, ОКА также еще и с языком КОБОЛ.

## 2. ОСОБЕННОСТИ ПРОЕКТИРОВАНИЯ ВНУТРИМАШИННОГО ИНФОРМАЦИОННОГО ОБЕСПЕЧЕНИЯ

### 2.1. Характерные особенности ИО в традиционных АСОД

В РММ по теме 6/4.10 "Разработать руководство по проектированию машинной реализации информационной базы АСУС" определены входы в процессы проектирования, сами процессы и результаты проектирования внутримашинного ИО для традиционных систем обработки данных. Характерные особенности таких систем:

информационная база на машинных носителях организуется для каждой отдельной задачи или для комплекса задач, а не для системы в целом;

база представляет собой набор отдельных файлов данных, имеющих большой процент избыточной информации;

каждый файл определяется как со стороны пользователя конкретной задачи, так и со стороны физического хранения данных как набор однородных записей фиксированных структур;

процедуры обработки файлов: ввод, контроль, переконфигурация, сортировка, распечатка и т.д. реализуются стандартными (или специальными) программами жестко связанными с программами пользователя;

изменение структур записей, появление новых приложений требуют создания новых файлов или корректировки программ;

Такие особенности систем организации и обработки данных определяют специфические особенности по их проектированию. Проектирование внутримашинного информационного обеспечения ведется в основном автономно для каждой конкретной задачи или комплекса задач. Разработчик (информационщик) самостоятельно разрабатывает все информационные процедуры внутримашинного ИО, начиная от проектирования информационной базы: модели внутримашинного информа-

ционного языка, состава, содержания и структур файлов данных, служебных выходных документов и моделей языка запросов и оканчивая проектированием обработки данных в ЭВМ: блок-схемы информационной увязки задач, проектные решения по внутримашинным процедурам ведения и обработки данных. В качестве критерия эффективности ИО принимались минимум затрат на разработку программного обеспечения соответствующей задачи (комплекса задач) и минимум эксплуатационных затрат. Ограничениями служат затраты на разработку самого ИО, а также требования, заданные на входе: необходимость использования существующих в системе массивов или обеспечение возможности перехода к ним в дальнейшем, заданные структуры массивов на машинных носителях.

## 2.2. Общие характеристики баз данных, требования по их организации и обработки в условиях БД

Одной из основных частей любого автоматизированного банка данных является база данных. Ее назначение в системе - быть центром накопления общих данных сложной структуры, используемых разными пользователями в различных задачах.

База данных составляется из структур различного типа, причем структуры одного типа могут состоять или конструироваться из структур других типов. Наиболее распространенными являются структуры: элемент данных (поле), сегмент (группа элементов), запись (блок, ), файл, база данных.

Элемент - это элементарная логически неделимая единица информации. Для обращения к элементу используется имя.

Сегмент - это совокупность элементов. Сегмент имеет имя, которое служит для обращения к нему и для задания отношений между сегментами при описании базы.

*Они  
применяются  
для  
структур!*

**Запись** - совокупность сегментов, в которой один и только один сегмент, определяющий запись не подчиняется в иерархическом отношении другому сегменту. Каждому типу записи присваивается уникальное имя, которое используется в программе пользователя.

**Файл** - совокупность записей. В файл могут быть включены записи различных типов. Для записей файла возможны ссылки друг на друга, а также ссылки на записи других файлов.

**База данных** - совокупность файлов. Доступ к файлам осуществляется посредством имен. Файлы могут быть связаны между собой некоторыми отношениями.

Между элементами данных (в основном сегментами) могут устанавливаться различного рода отношения, которые определяют структуру баз данных: сетевую, иерархическую, инвертированную.

К элементам данных (в основном к записям) могут применяться различные методы доступа: последовательный, индексно-последовательный, прямой и т.д.

Организация набора данных должна удовлетворять следующим требованиям:

централизованное хранение данных на носителях;

минимум избыточности элементов информации в системе;

возможность поэтажного создания и расширения базы;

целостность данных;

минимальные затраты на обработку данных (затраты на создание самих программ по обработке, а также эксплуатационные затраты);

безопасность и секретность данных и др.

Для обеспечения гибкости использования данных важным является также два аспекта в организации баз данных. Во-первых данные должны быть независимыми от программ пользователя, использующих их таким, образом, чтобы могла наращиваться или перестраиваться

без изменения программ. Во-вторых, процедуры запроса и поиска требуемой информации в базе данных должны точно реализовываться не трудоемкими для написания программами на обычном языке программирования.

Для реализации указанных требований применяются различные типы организации данных, средства описания данных (ЯОД) и средства обработки данных (ЯМОД).

Существует в основном три типа организации данных:

файловая, связанная с такими представлениями о данных, какими их понимают прикладные программы (логическая подмодель базы данных для конкретного приложения);

глобальная логическая организация данных, связанная со структурой всей базы данных, на основании которой могут быть получены различные файловые организации (логическая модель базы данных);

физическая организация — физическое представление данных и расположение их на запоминающих устройствах (физическая модель базы данных).

Определение логических моделей и подмоделей баз данных выполняется с помощью языковых и программных средств представления БД пользователю. Обычно каждый банк имеет свой ЯОД. Оттранслированные с ЯОД описания файлов (схемы) хранятся в базе данных. Наличие таких схем позволяет реализовать принцип независимости данных.

Организация физической базы данных зависит от используемых технических средств, используемых средств физического поиска записей — индексов, указателей, цепочек и т.д.

Централизация хранения информации, перераспределение акцента из обработки данных на их хранение привело к необходи-

месте появления лица (группы лиц), ответственного за сохранность данных, осуществляющего контроль за ними — администратора банка данных.

Администратор выполняет функции по созданию баз данных, управлению ими, модификации структур данных в соответствии с требованиями новых приложений, таким образом, какой он считает лучшим для всей системы.

### 2.3. Особенности проектирования внутримашинного ИО в условиях наличия БД

При проектировании внутримашинного ИО в условиях наличия БД возникают в основном три группы вопросов; методические, организационные, технологические. Эти группы вопросов обусловлены наличием особенностей самих БД: баз данных, СУБД и других элементов банков данных.

Особенности вопросов методического характера в том, что в настоящее время недостаточно в литературе материалов, связанных с БД и именно в части их проектирования и использования. Наличие обширных публикаций по информационному обеспечению по традиционным системам обработки данных, рассмотрение многоплановых вопросов повышения его эффективности, наличие методик по проектированию — это в основном результат накопленного опыта по работе с этими системами. Совсем по-другому обстоит дело в условиях наличия БД. Центральной проблемой создания любого автоматизированного банка данных является конструирование конкретной базы данных. Вместе с тем в советской печати эти вопросы освещены недостаточно. Первые опыты построения БД показали, что создание эффективных баз данных представляет собой длительный и трудоемкий процесс. Сегодня разработка и внедрение базы данных

в АСУ с использованием уже готовых СУБД занимает несколько лет. В результате - у разработчиков нет достаточного опыта в решении вопросов проектирования внутримашинного информационного обеспечения в новых условиях.

В настоящей работе проведен анализ нескольких банков данных. Они отличаются в основном возможностями СУБД, возможностями организации типов баз данных, и др. Повторению некоторых процедурных функций или структур хранения данных банки дополняют друг друга. В то же время это не их достоинство, а недостаток. В дальнейшем различие между банками данных разных типов будет определяться в основном их мощностью (или производительностью, набором дополнительных сервисных процедур и т.д.), а не структурами данных или методами доступа, которыми они располагают.

Особенностью существующих БД является и то, что все они не являются идеальными банками. То-есть полностью независимые (от прикладных программ) базы данных бывают также редки как и полностью избыточные. Наиболее эффективны в настоящее время такие базы данных, в которых структура выбрана с целью выполнения определенной последовательности операций с данными, а не для произвольного, не связанного с функциями обработки информации. Вопросы зависимости информационной базы от прикладных программ связаны с возможностями программного обеспечения. В одном случае пользователю (прикладному программисту) достаточно сообщить имя элемента данных (записи, которую он хочет получить). При наличии другого программного обеспечения он должен сообщать идентификаторы элементов данных и имя набора, в которых элементы данных содержатся. В худшем случае программист должен знать физическую организацию данных, и задавать машинный адрес данных. Если отсутствует независимость данных ему необходимо

знать также точный физический формат записи.

Важным элементом проектирования внутримашинного ИО является также установление четкого взаимодействия между разработчиком (информационщиком) и администратором данных, так как и тот и другой отвечает за эффективную организацию базы данных, но у каждого разная ответственность. У первого — за эффективность базы для конкретного приложения. У второго — за эффективность базы во всей системе. В этом плане возникают вопросы как организационного, так и технологического характера.

Перечисленные особенности информационного обеспечения определяют особенности его проектирования.

#### 2.4. Уточнение входов в процессы проектирования

Проектирование внутримашинного информационного обеспечения в условиях наличия СУБД осуществляется двумя разработчиками: администратором банка данных и информационщиком, занимающимся вопросами ИО конкретного приложения. При этом администратор банка решает только вопросы проектирования информационной базы, а информационщик — проектирования базы и обработки данных в ЭВМ.

В технологии баз данных работа по их организации производится в несколько этапов:

предварительное описание базы данных и ее формирование на носителях информации;

описание частей баз данных, необходимых для конкретных приложений;

описание процедур обработки данных с использованием языковых и программных конструкций.

При проектировании информационной базы администратор использует справочники данных, создаваемые в системе со всеми

необходимыми характеристиками, а также инструкции по созданию баз данных, поставляемых каждым БД.

Разработчик информационного обеспечения по конкретному приложению решает группы вопросов, предусмотренных в РИМ. При этом учитываются особенности использования конкретных СУБД, генераторов отчетов, модульных программ следующим образом:

Входы в процессы проектирования. Задаются требования по использованию на входе элементов данных, имеющихся в базе в форме, которую допускает заданная СУБД (например, задание только списка ~~звездных~~ элементов данных, списка элементов и имен файлов, где содержатся данные или же списка элементов имен файлов и форматы записей). Аналогично требования по результирующим (в данной задаче) элементам данных. Перечень основных возможных процедур обработки данных, реализуемых данным СУБД требования применения генераторов отчетов или модульных систем программирования для формирования выходных документов.

Процесс проектирования ивнутримашинного ИО. При определении состава, содержания и структуры баз данных учитывать возможности СУБД и требования на входе по представлению информационных элементов. В зависимости от указанных факторов производится выбор носителей и определения размещения на них информации.

Уточняются процедуры по обработке информации, СУБД, генераторами отчетов или модульными программами.

Результаты процессов проектирования. Уточненные составы и характеристики элементов данных в соответствии с требованиями, принятыми в данной системе. Указываются процедуры по обработке информации, где применяются средства СУБД, генераторов отчетов и модульных систем программирования.

### 3. МЕТОДЫ ВЫБОРА СМО ДЛЯ СОЗДАНИЯ, ВЕДЕНИЯ И ИСПОЛЬЗОВАНИЯ ИНФОРМАЦИОННОЙ БАЗЫ АСУ

В первой главе данного отчета были рассмотрены общие характеристики систем по формированию и управлению базой данных, дан краткий сравнительный анализ рассмотренных систем, но не были приведены критерии, которыми следует руководствоваться при выборе той или иной системы. В этой главе приведены и охарактеризованы те критерии, которыми следует руководствоваться при выборе систем по формированию и управлению базой данных.

#### 3.1. Критерии выбора систем.

При решении проблемы выбора систем по формированию и управлению базой данных стоят следующие задачи: характеристика информационной базы предприятия; характеристика возможностей и ограничений СУБД.

Характеристика информационной базы предприятия. Освещая этот вопрос следует указать объем информации для обработки, структуру предприятия или движения информации, структуру данных и идентификаторов (если на предприятии уже функционирует АСУ), набор технических средств, которыми располагает предприятие.

#### Характеристика возможностей и ограничений СУБД.

Класс поддерживаемых структур. Характеризуется какую модель базы данных поддерживает СУБД – иерархическую, сетевую, реляционную и т.д. Указываются ограничительные размеры базы данных.

Специализация. В этом пункте следует указать на какой род деятельности нацелена СУБД, т.е. проблемно ориентирована она или нет.

Функциональная изобретательность. Следует указать обладает ли система свойствами включать в работу заданный набор присущих ей функций.

Независимость прикладных программ от структуры данных. Описывается как влияет изменение структуры данных на программы пользователя.

Средства обеспечения конфиденциальности данных. Раскрывается вопрос возможен ли доступ к определенной части базы данных или ко всей базе.

Наличие средств контроля. В этом пункте следует указать имеет ли система средства контроля при формировании информационной базы данных, указать те методы которыми располагает система.

Конфигурация технических средств. Указывается какие технические средства минимально необходимы для работы СУБД и какая конфигурация наиболее эффективна.

Тип операционной системы. Указывается под управлением какой операционной системы работает СУБД.

Средства запроса для конечного пользователя. Указывается имеет ли система фазы диалога с базой данных, возможность формирования справок, запросов. Указывается на каком языке происходит диагностика работы СУБД.

Параллельность работы. Характеризуется возможность работы СУБД параллельно с несколькими базами данных.

Дополнительный сервис пользователям. Указывается какими дополнительными возможностями обладает система по ведению базы данных.

Физическая организация данных. Следует указать методы доступа принятые в СУБД, средства ускоряющие доступ (т.е. те возможности, которые дополняют возможности операционной системы).

Режим доступа. Указывается в каком режиме может работать система - в пакетном, мультидоступ.

Класс системы. В этом пункте следует указать является ли данная система замкнутой, системой с базовым языком или гибридной. Если система с базовым языком, то необходимо дать качественную оценку языку манипулирования данными и языку описания данных, то есть указать прост или сложен язык в описании и манипулировании данными. Если система замкнутая, то необходимо указать на сложность или простоту операторов по генерации системы.

Средства обеспечения целостности данных. В этом вопросе следует указать наличие в системе возможностей по сохранению предыдущего состояния базы, наличие контрольных точек и т.д.

Дополнительные факторы. Здесь следует остановиться на перспективах развития системы, т.е. предполагается ли ее совершенствование, устранение замеченных недостатков, ошибок. Указать на наличие интерфейса с другими БД. Указать производительность системы, т.е. время на получение данных при заданной конфигурации технических средств. Указать сроки освоения системы, принимая во внимание наличие технической документации, возможность сопровождения организации-разработчика. Указывается с каким из языков пользователя может работать система.

### 3.2. Характеристика возможностей и ограничений СУБД

Рассмотрев в предыдущем пункте вопросы, характеризующие информационную базу данных АСУ и критерии по выбору систем формирования и ведения базы данных, построим таблицу, в которой отразим характеристики рассмотренных выше систем (таблица I).

Таблица I

## Характеристика систем

№ п/п	Характеристика системы	Обозначение системы					
		Г В В	НСИ-1-ДОС	О С М О	С И О Д	Б А Н К	О К А
1	2	3	4	5	6	7	8
1.	Класс поддерживаемых структур	иерархические	иерархические	иерархические	сетевые	иерархические, сетевые	иерархические, сетевые
2.	Специализация	универсальная	универсальная	универсальная	проблемно ориентирована	универсальная	универсальная
3.	Функциональная избирательность	функционально избирательная	функционально избирательная	функционально избирательная	нет	нет	функционально избирательная
4.	Независимость прикладных программ от структуры данных	зависимы от структуры данных	не зависят от структуры данных	независимы от структуры данных	зависимы от структуры данных	зависима от структуры данных	зависима от структуры данных, независима от памяти устройств
5.	Средства обеспечения конфиденциальности данных	возможен доступ к части базы	доступ ко всей базе	доступ ко всей базе	доступ ко всей базе	возможен доступ к части базы данных	возможен доступ к части базы данных

Продолжение таблицы I

1	2	3	4	5	6	7	8
6. Наличие средств контроля	есть средства контроля	проверка на формальную правильность (контроль с подключением блоков пользователя)	есть средства контроля	средств контроля не имеет	минимальный набор средств контроля	минимальный набор средств контроля	
7. Конфигурация технических средств	для ДОС: 2 ММД, 64К для ОС: 3 ММД, 128К	3 ММД, 64К	полный комплект тех. средств ЕС-1020 и выше	полный комплект тех. средств ЕС-1020 и выше Для ОС: 3 ММД	полный комплект тех. средств ЕС-1020 и выше	4 ММД, 128К-256К	
8. Тип операционной системы	ДОС, ОС	ДОС	ДОС (см.п.16)	ДОС, ОС	ДОС	ОС	
9. Средства запроса для конечного пользователя	печать файлов (архивных)	выдача справок (печать основных файлов)	выдача справок, запросов на уровне файлов	выдача справок	выдача справок	выдача справок, запросов	
- диагностика	русский	русский	русский	русский	английский	английский	
10. Параллельность работы	невозможна	невозможна	невозможна	невозможна	невозможна	работа возможна	

Продолжение таблицы I

I	2	3	4	5	6	7	8
II. Дополнительный сервис пользователям	не имеет	не имеет	имеет: программы поиска, сортировки и т.д.	не имеет	имеет	имеет	
12. Физическая организация данных	файлы последовательной и прямой организации	файлы последовательной и индексно-последовательной организации	файлы последовательной и прямой организации	файлы последовательной и индексно-последовательной организации	файлы прямой организации	файлы последовательной и прямой организации	
13. Режим доступа	пакетный	пакетный	пакетный	пакетный	пакетный	пакетный	пакетный мультидоступ
14. Класс системы	замкнутая	замкнутая, не требует генерации	замкнутая	замкнутая	с базовым языком	с базовым языком	
15. Средства обеспечения целостности данных	не имеет	не имеет	хранение предыдущего поколения	хранение предыдущего поколения	хранение предыдущего поколения	наличие контрольных точек (наличие программы восстановления базы данных)	
16. Дополнительные факторы: - развитие системы	нет	нет	развивается и дорабатывается для ОС	развивается разработка III	нет	нет	

Продолжение таблицы I

I	2	3	4	5	6	7	8
- наличие интерфейса	нет	нет	нет	нет	нет	нет	есть
- осваиваемость системы	3 - 5 месяцев	4 - 5 месяцев	5 - 8 месяцев	6 - 8 месяцев	8 - 10 месяцев	10 - 14 месяцев	
- возможность работы с языком пользователя	Ассемблер	Ассемблер	Ассемблер ПЛ/Г	Ассемблер ПЛ/Г, КОБОЛ	Ассемблер	Ассемблер ПЛ/Г, КОБОЛ	

#### 4. ВЫВОДЫ И ПРЕДЛОЖЕНИЯ ПО ВЫБОРУ СМО ДЛЯ ФОРМИРОВАНИЯ, ВЕДЕНИЯ И УПРАВЛЕНИЯ ИНФОРМАЦИОННОЙ БАЗОЙ АСУ ДСК

Анализ разработанных систем по формированию, ведению и управлению базой данных позволяет сделать следующие выводы:

4.1. В настоящее время существуют разработанные системы, позволяющие эффективно решать вопросы организации, хранения, ведения и управления информационной базой АСУ. К таким системам относятся ОСМО, ОКА. Эти системы различны по своему классу, возможностям, срокам освоения, но в то же время эти системы, которые могут работать совместно.

Объемы выпускаемой продукции на домостроительных комбинатах страны различны, отсюда и объем информации, циркулирующей на предприятии, различен. Различна также и техническая оснащенность вычислительной техникой этих предприятий.

4.2. Поэтому, можно отметить следующие пути организации информационной базы при помощи рассмотренных систем.

4.2.1. При объеме выпускаемой продукции на ДСК не более 400 т. м куб. сборного железобетона (объем нормативно-справочной информации примерно равен 10-25 Мбайт) можно рекомендовать для применения систему ОСМО. Система не требует большой технической оснащенности предприятия, большого срока на ее освоение.

4.2.2. Для предприятий, где объем продукции более 400 т. м куб. сборного железобетона (объем нормативно-справочной информации более 25 Мбайт) ОСМО можно рекомендовать для формирования и ведения нормативно-справочной информации. Для управ-

ления базой данных, централизованного хранения и ведения плановой и учетной информации рекомендуется использовать систему ОКА. ОКА также может быть использована для передачи данных между объектами пользователя. Внедрять системы рекомендуется поэтапно: на первом этапе систему ОСМО (которой свойственны легкость освоения, простота внедрения, эксплуатации), на втором этапе систему ОКА.

4.3. Важное значение для практического создания на конкретном предприятии базы данных имеет определение структуры и состава базы данных. При этом необходимо решить следующие проблемы: определение фонда данных, необходимого для решения всех функциональных задач АСУ, определение перечня и структуры файлов, определение связей между файлами. Решение указанных проблем нельзя оставлять за пользователями различных предприятий. Для АСУ строительства необходимо разработать типовую базу данных, процессы обработки информации, т.е. необходимо типизировать программное, информационное обеспечение. Это значительно ускорит практическую реализацию в АСУ принципа БД, ускорит внедряемость систем АСУ на предприятиях.

## Л И Т Е Р А Т У Р А

1. Тимофеев Б.Б., Литвинов В.А. Структура и организация банка данных в АСУ промышленным предприятием. УСИМ, 1974, № 6
2. Богданов А.Ф. и др. Отечественный и зарубежный опыт проектирования банков данных (обзор). Латв. МНТИ, Рига, 1976
3. Арткевич А.А., Баточко В.В. Создание и использование банков данных в АСУП.
4. Информационные системы общего назначения (Аналитический обзор систем управления базами данных). Пер. с англ., М., "Статистика", 1975
5. Гольштейн Б.Д. Об основных понятиях системы управления базами данных (тематический обзор). В сб. Алгоритмы и организация решения экономических задач. Вып. 2, М., "Статистика", 1973
6. Сергиенко И.В., Тарасюк И.Н., Тукалевская Н.И. Автоматизированные системы обработки данных. Киев, "Наукова думка", 1976
7. Келехсаев А.А., Беляев А.П. Системы интеграции и обработки данных СИОД1, СИОД2. "Статистика", 1977
8. Агафонов В.Н. и др. Генератор программы ввода данных для ЕС ЭВМ. М., "Статистика", 1976
9. Зайцев Н.Г., Возняк В.В. и др. Общесистемное математическое обеспечение типовой АСУ на базе ЕС ЭВМ. Редакция 2.0 Ротопринт НИИАСС, 1978

10. Материалы семинара "Вопросы разработки и эксплуатации систем обработки данных на базе ЕС ЭВМ". Москва, МДНП имени Ф.Э.Дзержинского, 1978
11. Рекомендации по применению. Научно-производственное объединение "Центрпрограммистем", Калинин, 1978
12. Цалепко М.Ш. Реляционные модели баз данных. В сб. "Алгоритмы и организация решения экономических задач", М., "Статистика", 1977, № 9, 10.
13. Королев М.А. Информационные системы и структуры данных. М., "Статистика", 1977
14. Дж.Мартин Организация баз данных в вычислительных системах. М., "Мир", 1978