ГОСТ Р 34.980.1—92
(ИСО 8571/1—88)

ГОСУДАРСТВЕННЫЙ СТАНДАРТ РОССИЙСКОЙ ФЕДЕРАЦИИ

ИНФОРМАЦИОННАЯ ТЕХНОЛОГИЯ.

ВЗАИМОСВЯЗЬ ОТКРЫТЫХ СИСТЕМ. ПЕРЕДАЧА, ДОСТУП И УПРАВЛЕНИЕ ФАЙЛОМ.

Часть 1. ОБЩЕЕ ОПИСАНИЕ

Издание официальное

69 руб. БЗ 4—92/441

ГОССТАНДАРТ РОССИИ
Москва

УДК 681.224:621.391:006.354 Группа П85

ГОСУДАРСТВЕННЫЙ СТАНДАРТ РОССИЙСКОЙ ФЕДЕРАЦИИ

Информационная технология. ВЗАИМОСВЯЗЬ ОТКРЫТЫХ СИСТЕМ. ПЕРЕДАЧА. ДОСТУП И УПРАВЛЕНИЕ ФАЙЛОМ. Часть 1. ОБЩЕЕ ОПИСАНИЕ

Information technology.
Open systems interconnection
File transfer, access and management
Part 1. General introduction

ГОСТ Р 34.980.1—92 (ИСО 8571/1—88)

ОКСТУ 0034

Дата введения 01.01.94

0. ВВЕДЕНИЕ

Настоящий стандарт входит в комплекс государственных стандартов, обеспечивающих взаимосвязь вычислительных систем. Его отношение к другим стандартам комплекса определено эталонной моделью взаимосвязи открытых систем (ВОС) по ГОСТ 28906. Эталонная модель подразделяет область стандартизации ВОС на ряд уровней спецификаций, каждый из которых предназначен для выполнения определенных функций.

Эталонная модель ВОС обеспечивает совместимость вычислительных систем, выполненных различными изготовителями, имеющих различные системы управления и уровни сложности, созданных в разное время.

Настоящий стандарт определяет файловую услугу и файловый протокол, доступные в рамках прикладного уровня эталонной модели. Данная услуга имеет дело с идентифицируемыми блоками информации, которые могут трактоваться как файлы и могут храниться в открытых системах или передаваться между прикладными процессами.

Настоящий стандарт определяет базисную файловую услугу. Он предоставляет достаточные средства для обеспечения передачи файлов и устанавливает метод доступа к файлу и управление файлами.

Издание официальное

★

Стандарт не определяет интерфейсы к средствам передачи файла и средствам доступа внутри локальной системы.

Примеры использования службы ПДУФ (FTAM) приведены в приложении А, краткое описание объектов информации, определенных в настоящем стандарте, — в приложении Б.

1. НАЗНАЧЕНИЕ И ОБЛАСТЬ ПРИМЕНЕНИЯ

Настоящий стандарт включает общее описание концепций и модулей, представленных в ГОСТ Р 34.980.2 — ГОСТ Р 34.980.4.

2. ССЫЛКИ

ГОСТ 28906 (ИСО 7498) «Системы обработки информации. Взаимосвязь открытых систем. Базовая эталонная модель».

ИСО 8326* «Системы обработки информации. Взаимосвязь открытых систем. Определение базовых услуг сеансового уровня, ориентированного на соединение».

ИСО/ТО 8509* «Системы обработки информации. Взаимосвязь открытых систем. Соглашения по услугам».

ГОСТ 34.980.2 (ИСО 8571/2) «Инфомационная технология. Взаимосвязь открытых систем. Передача, доступ и управление файлом. Часть 2. Определение виртуального файлохранилища».

ГОСТ 34.980.3 (ИСО 8571/3) «Информационная технология. Взаимосвязь открытых систем. Передача, доступ и управление файлом. Часть 3. Определение услуг виртуального файла».

ГОСТ 34.980.4 (ИСО 8571/4) «Информационная технология. Взаимосвязь открытых систем. Передача, доступ и управление файлом. Часть 4. Спецификация файловых протоколов».

ГОСТ 34.981 (ИСО 8649) «Информационная технология. Взаимосвязь открытых систем. Определение услуг сервисного элемента управления ассоциацией».

ГОСТ 34.971 (ИСО 8822) «Информационная технология. Взаимосвязь открытых систем. Определение услуг уровня представления с установлением соединения».

ГОСТ 34.973 (ИСО 8824) «Информационная технология. Взаимосвязь открытых систем. Спецификация абстрактно-синтаксической нотации версии 1 (АСН.1)».

ГОСТ 34.974 (ИСО 8825) «Информационная технология. Взаимосвязь открытых систем. Описание базовых правил кодирования для абстрактно-синтаксической нотации версии 1 (АСН.1)».

---

* До прямого применения данного документа в качестве государственного стандарта распространение его осуществляет секретариат ТК 22 «Информационная технология».

ИСО 9804* «Системы обработки информации. Взаимосвязь открытых систем. Определение элементов услуг прикладного уровня. Совершение, параллельность и восстановление».

ИСО 9834/2* «Системы обработки информации. Процедуры для специальных правил регистрации модели OSI. Часть 2. Регистрация типов документов».

3. ОПРЕДЕЛЕНИЯ ЭТАЛОННОЙ МОДЕЛИ

Настоящий стандарт основывается на концепциях, разработанных в ГОСТ 28906, и использует следующие определенные в нем термины:

прикладной логический объект;
прикладной процесс;
элемент услуг прикладного уровня;
(N) — соединение;
открытая система;
(N) — протокол;
(N) — протокольная управляющая информация;
(N) — протокольный блок данных;
(N) — пункт доступа к услугам;
(N) — адрес пункта доступа к услугам;
(N) — сервисный блок данных;
подуровень;
(N) — данные пользователя;
элемент пользователя.

4. ОПРЕДЕЛЕНИЕ СЕРВИСНЫХ СОГЛАШЕНИЙ

В настоящем стандарте используются следующие термины, определенные в стандарте ИСО/ТО 8509, применительно к файловой услуге:

подтверждение;
индикация;
примитив;
запрос;
ответ;
поставщик услуг;
пользователь услуги.

5. ОПРЕДЕЛЕНИЕ СЛУЖБЫ ПДУФ (FTAM)

Все термины, если не оговорено иначе, относятся к представлению системы, предназначенной для открытой взаимосвязи. Это

* До прямого применения данного документа в качестве государственного стандарта распространение его осуществляет секретариат ТК 22 «Информационная технология»

означает, что термины относятся к виртуальному файлохранилищу, а не к какому-либо реальному файлохранилищу (см. п. 7).

Определения сгруппированы в основные категории и внутри каждой категории упорядочены в алфавитном порядке.

В настоящем стандарте применяются следующие определения:

5.1. Общее описание

5.1.1. Пустой файл (empty file) — файл, содержимое которого состоит только из корневого узла без соответствующего блока данных и без имени узла.

5.1.2. Доступ к файлу (file access) — просмотр, модификация, замена или удаление части содержимого файла.

5.1.3. Содержание сообщения файла (file contents) — блоки данных, имена узлов и информация структурирования, содержащаяся в файле, которые могут быть обработаны во время режима «Открытие файла»; атрибуты файлов не относятся к содержанию сообщения файла.

5.1.4. Административное управление файлами (file management) — создание и уничтожение файлов, просмотр или манипулирование атрибутами файлов.

5.1 5. Передача файла (file transfer) — функция, выполняющая перенос части или всего содержания сообщения файла между открытыми системами.

5.1.6. Иерархическая модель файла (hierarchical file model) — модель внутренней структуры файла, принимающая форму древовидной структуры блоков данных для доступа к поименованному файлу.

5.1.7. Реальный файл (real file) — поименованный набор информации и его атрибуты, которые находятся в реальной системе и для которых отображаются ссылки к виртуальному файлу, осуществляемые в среде взаимосвязи открытых систем.

5.1.8. Реальное файлохранилище (real filestore) — организованный набор файлов, включая их атрибуты и имена, которые находятся в реальной системе и для которых отображаются ссылки к виртуальным файлам в среде открытых систем.

5.1.9. Виртуальный файл, файл (virtual file, file) — явно поименованный набор структурированной информации, имеющий общий набор атрибутов.

5.1.10. Виртуальное файлохранилище (virtual filestore) — абстрактная модель для описания файлов и файлохранилищ и возможных действий над ними. Там, где это не вызывает неоднозначности, этот термин в стандарте ИСО 8571 используется сокращенно как «файлохранилище».

5.2. Архитектура

5.2.1. Режим учета (accounting regime) — период, во время которого используется определенный набор учетной информации.

5.2.2. Блок совершения операций (commitment unit) — набор действий файлохранилища, которые либо завершаются успешно, тогда результат совершения операций будет воздействовать на другие процессы, либо полностью отвергаются, тогда результат не воздействует на другие процессы.

Примечание Отмена действий в блоке совершения операций возможна в любое время до завершения блока совершения операций

5.2.3. Докит (docket) — такой набор информации, который может быть связан с режимом файловой услуги и который должен быть сохранен, если возможно восстановление после ошибки.

5.2.4. Внешняя файловая услуга (external file service) — передача файла, доступ к файлу и управление файлом, как это представляется пользователю файловой услуги.

5.2.5. Пользователь файловой услуги (file service user) — часть логического объекта прикладного уровня, концептуально вызывающая службу ПДУФ (FTAM).

5.2.6. Абонент (initiator) — пользователь файловой услуги, запрашивающий установку режима службы ПДУФ (FTAM).

5.2.7. Внутренняя файловая услуга (internal file service) — услуга, используемая модулем протокола восстановления файла после ошибки для передачи как протокольной управляющей информации восстановления файла после ошибки, так и файловой протокольной управляющей информации без ошибок.

5.2.8. Среда взаимосвязи открытых систем (open systems interconnection environment) — набор определений стандартизированных услуг, протоколов и структур данных, которые применимы для взаимосвязи систем.

5.2.9. Фаза (phase) — период времени, в течение которого обмены протоколами имеют особое назначение, такое как установление или отмена механизма контекста прикладного уровня для каждой фазы множество допустимых сообщений определяется в терминах переходов состояний.

Примечание Логический объект в любой момент времени находится в какой-либо одной фазе.

5.2.10. Значение данных уровня представления (presentation data value) — блок информации, определенной на уровне абстрактного синтаксиса, передаваемый с помощью услуги уровня представления.

5.2.11. Среда реальной системы (real system environment) — аспекты реализации системы, обеспечивающие прикладной процесс внутри реальной системы.

5.2.12. Принимающий логический объект; получатель (receiving entity; receiver) — логический объект, принимающий часть или все содержание сообщения файла во время режима «Передача данных файла».

5.2.13. Режим (regime) — период, в течение которого логический объект находится в одном из возможных своих состояний, при котором допускаются определенные действия.

Примечание Режимы могут быть вложенными.

5.2.14. Ответственный логический объект (responder) — пользователь файловой услуги, допускающий установление режима службы FTAM, запрошенного абонентом.

5.2.15. Возврат в исходное состояние (rollback) — аннулирование несовершенных операций.

5.2.16. Посылающий логический объект, отправитель (sending entity; sender) — логический объект, который посылает часть или все содержание сообщения файла во время режима «Передача данных файла».

5.2.17. Сервисный элемент (service element) — элемент стандартизации, определяющий полную группу функций.

5.2.18. Сервисный примитив (service primitive) — наименьшее определенное взаимодействие между пользователем услуги и поставщиком услуги связи.

5.2.19. Симбиотический сервисный элемент (symbiotic service element) — сервисный элемент, который будет обеспечивать операцию некоторой другой услуги, воспринимая ее семантику и включая части ее абстрактного синтаксиса в определенные места в абстрактном синтаксисе информации, управляющей протоколом первой услуги.

5.3. Схема файлохранилища

5.3.1. Атрибуты взаимодействия (activity attributes) — атрибуты, описывающие условия взаимодействия при использовании файловой услуги. Атрибуты являются локальными по отношению к одному режиму службы ПДУФ (FTAM) (и любому режиму, входящему в него).

5.3.2. Атрибут (attribute) — часть информации, устанавливающая свойство чего-либо и принимающая одно из множеств определенных значений, каждое из которых имеет определенный смысл.

5.3.3. Атрибуты файла (file attributes) — имя или другие идентифицируемые свойства файла.

Примечание. Атрибут файла имеет одно и то же значение в конкретный момент времени для любого пользователя файловой услуги, даже если в этот момент активными являются несколько пользователей.

5.4. Доступ к файлохранилищу

5.4.1. Контекст доступа (access context) — спецификация алгоритма, определяющая подмножество структурной информации и информации пользователя в содержании сообщения файла при чтении файла для передачи или для доступа к файлу.

5.4.2. Элемент данных (data element) — наименьшая часть данных, идентификацию которой необходимо сохранить, когда эта часть данных передана услугой уровня представления. Элемент данных может содержать информацию содержания сообщения файла, структурную информацию файла или протокольную управляющую информацию.

5.4.3. Блок данных (data unit) — наименьший блок содержания сообщения файла, над которым могут выполняться действия файлохранилища. Каждый блок данных связан с узлом структуры доступа к файлу. Блок данных представляет собой последовательность элементов данных.

5.4.4. Блок данных доступа к файлу (file access data unit) — блок структуры доступа к файлу, над которым могут выполняться действия по передаче, удалению, расширению, замене или вставке. Блок данных доступа к файлу может содержать несколько блоков данных или не содержать ни одного блока данных.

5.4.5. Действие файлохранилища (filestore action) — одно из действий, указанных как часть определения виртуального файлохранилища.

5.4.6. Структура доступа к файлу (file access structure) — структура данных файла, которая связывает блоки данных доступа к файлу, допуская идентификацию, описание и обработку блоков данных.

5.5. Структура файла

5.5.1. Дуга (arc) — направленная связь между двумя узлами.

5.5.2. Длина дуги (arc length) — положительное целое число, выражающее разницу в уровнях между порожденным и порождающим узлами.

5.5.3. Порожденный узел (для данного узла) (child of a node) — узел, в котором завершается дуга, выходящая из данного узла.

5.5.4. Лист (leaf) — узел дерева, который не имеет выходящих дуг.

5.5.5. Уровень (узла) (level (of a node) — сумма длин дуг от корня до рассматриваемого узла.

5.5.6. Длинная дуга (long arc) — дуга с длиной, превышающей единичную длину.

5.5.7. Узел (node) — элементарный компонент, из которого строится дерево.

5.5.8. Упорядоченное дерево (ordered tree) — дерево, в котором для каждого узла в дереве упорядочены порожденные узлы.

5.5.9. Порождающий узел (для данного узла) (parent (of a node) — узел, из которого выходит дуга, входящая в данный узел.

5.5.10. Путь (path) — последовательность дуг, которая соединяет один узел с другим таким способом, что каждая дуга проходит в своем определенном направлении.

5.5.11. Корень (root) — единственный узел дерева, который не имеет входящих дуг; он имеет уровень 0.

5.5.12. Родственный узел (для данного узла) (sister (of a node) — узел, который с данным узлом имеет один и тот же порождающий узел.

5.5.13. Поддерево (subtree) — часть дерева, включающая произвольный узел в качестве корня поддерева и все другие узлы, к которым может быть получен доступ с помощью пути из этого корня поддерева.

5.5.14. Последовательность обхода (fraversal sequence) — упорядочение узлов в дереве таким образом, что каждый узел встречается один и только один раз, которое определяется алгоритмом, применяемым ко всем возможным деревьям.

Примечание. В общем случае несколько различных деревьев могут создавать одинаковую последовательность обхода

5.5.15. Дерево (tree) — связанная структура, в которой каждый узел соединяется с другими узлами направленными дугами таким способом, при котором один узел не имеет входящих дуг, а все другие узлы имеют только одну входящую дугу.

5.6. Набор ограничений

5.6.1. Набор ограничений (constraint set) — набор ограничений и уточнений общей модели файла, который определяет более подробную модель, отвечающую требованиям приложений определенного класса.

5.6.2. Модель файла (file model) — модель структуры доступа к содержанию сообщения файла.

5.6.3. Двухмерный (набор ограничений) (flat (constraint set) — набор ограничений, который при применении его к общей иерархической модели файла создает структуру доступа, состоящую из двух уровней (уровни 0 и 1) и имеющую блоки данных только в узлах-листьях и не имеющую блока данных в корневом узле.

5.6.4. Общая иерархическая модель файла (general hierarchical file model) — модель, в которой блоки данных доступа к файлу организованы в иерархическое дерево.

5.6.5. Иерархический набор ограничений (hierarchical constraint set) — набор ограничений, который при применении его к общей иерархической модели файла создает структуру доступа, которая все еще является иерархической, но в которой ограничениями являются формат описаний узла и формат блоков данных.

5.6.6. Неструктурированный набор ограничений (unstructured constraint set) — набор ограничений, который при применении его к общей иерархической модели файла создает структуру доступа, состоящую только из корневого узла с одним блоком данных.

5.7. Типы документов

5.7.1. Сцепление документов (concatenation of documents) — комбинирование двух документов для формирования единого результирующего документа.

5.7.2. Документ (document) — совокупность информации с полностью известным абстрактным синтаксисом, частично известной семантикой и полностью известным набором возможных синтаксисов передачи.

5.7.3. *Тип документа* (*document type*) — спецификация класса документов, которая устанавливает их обязательную семантику, абстрактный синтаксис, синтаксисы передачи и динамику.

5.7.4. Динамика документа (dynamics of document) — свойства сцепления и упрощения документа.

5.7.5. Релаксация документа (relaxation of document) — процесс получения одного документа из другого, используя менее жесткие параметры, описывающие параметры этого документа.

5.7.6. Упрощение документа (simplification of a document) — процесс получения документа из другого, имеющего другой тип, путем исключения информации о структуре документа.

6. СОКРАЩЕНИЯ

В ГОСТ 34.980.2 — ГОСТ 34.980.4 используются следующие сокращения:

БД (DU)	блок данных (data unit)
БДДФ (FADU)	блок данных доступа к файлу (file access data unit)
ВОС (OSI)	взаимосвязь открытых систем (open systems interconnection)
ВФУ (EFS)	внешняя файловая услуга (external file service)
ВнФУ (IFS)	внутренняя файловая услуга (internal file service)
ЗДУП (PDV)	значение данных уровня представления (presentation data value)
ИД (ID)	идентификатор (identifier)
МПВФПО (FERPM)	модуль протокола восстановления файла после ошибок (file error recovery protocol machine)
МФП (FPM)	модуль файлового протокола (file protocol machine)
ПБД (PDU)	протокольный блок данных (protocol data unit)
ПДУФ (FTAM)	передача файла, доступ к файлу и управление файлом (file transfer, access and management)

ПУИ (PCI)	протокольная управляющая информация (protocol control information)
СБДУП (PSDU)	сервисный блок данных уровня представления (presentation service data unit)
СВОС (OSIE)	среда взаимосвязи открытых систем (open systems interconnection environment)
СПиВ (CCR)	совершение, параллельность и восстановление (commitent, concurrency and recovery)
СРС (RSE)	среда реальной системы (real system environment)
СЭПУ (ASE)	сервисный элемент прикладного уровня (application service element)
СЭУА (ACSE)	сервисный элемент управления ассоциацией (association control service element)
ФПБД (FPDU)	файловый протокольный блок данных (file protocol data unit)

РАЗДЕЛ 1. ОБЩИЕ КОНЦЕПЦИИ СЛУЖБЫ ПДУФ.

7. АРХИТЕКТУРНЫЕ ОСНОВЫ МОДЕЛИ ВОС

Целью стандартизации файловой услуги является предоставление возможностей взаимосвязи открытых систем для пользователей файлов, которые хотят передавать информацию, иметь доступ к ней или управлять информацией, которой владеют системы, таким образом, как будто пользователи являются владельцами файлов. Любой, кто выступает в качестве открытой системы и следует указанным файловым протоколам в роли ответственного логического объекта, рассматривается как обеспечивающий виртуальное файлохранилище.

Как и во всех стандартах на модели ВОС используемое здесь архитектурное деление отражает классификацию протоколов. Они не накладывают никаких ограничений на возможные способы структуризации нижележащих реализаций. Единственный пункт, в котором возможно должна быть выполнена проверка на соответствие протокола, — это пункт взаимосвязи.

Настоящий стандарт входит в состав комплекса стандартов, которые взаимосвязаны через базовую модель ВОС (ГОСТ 28906). Аспекты протоколов взаимосвязи, общие для нескольких различных областей применения, определены в отдельных стандартах, а логическая связь различных элементов обеспечивается с помощью определения предоставляемых услуг.

Для понимания предназначения модели ВОС важно ясно различать взаимосвязанные наборы стандартов и их реализацию в аппаратных и программных средствах реальной открытой систе-

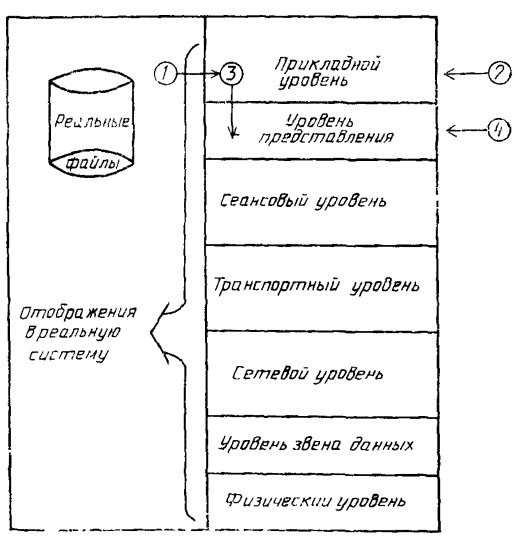
мы, используя протоколы, определенные в стандартах. Это отличие приводит к идентификации двух сред:

а) аспекты реализации в терминах реальных средств и ресурсов, которые обеспечивают прикладной процесс внутри реальной системы, образуют среду реальной системы (СРС);

б) набор определений стандартизованных услуг, протоколов и структур данных, которые обеспечивают взаимосвязь систем, образуют среду взаимосвязи открытых систем (СВОС).

Типичные аспекты поведения прикладного процесса, просматриваемые в среде взаимосвязи открытых систем, образуют логический объект прикладного уровня; функционирование логических объектов прикладного уровня обеспечивается разделенными на уровни функциями связи, определенными в эталонной модели ВОС.

Часть работы, которая должна быть проделана при проектировании реализующей системы, состоит в выборе отображения между элементами набора определений на всех уровнях и в выборе аспектов реализации (черт. 1).



Черт. 1

Примечания:

1. Поток действий между средой реальной системы и средой взаимосвязи открытых систем представляет отображение попыток связаться с элементом

пользователя логического объекта прикладного уровня.

2 Поток действий между логическими объектами прикладного уровня (как протокольная управляющая информация прикладного уровня, так и любые данные пользователя) является логическим потоком информации в согласованном абстрактном синтаксисе, который не зависит от конкретного синтаксиса передачи

3 Поток действий между логическими объектами прикладного уровня и уровня представления является логическим потоком информации, включающим как данные пользователя, так и протокольную управляющую информацию прикладного уровня в одном или нескольких согласованных контекстах уровня представления

4 Поток действий между логическими объектами уровня представления представляет собой данные, закодированные в согласованных синтаксисах передачи и протокольную управляющую информацию уровня представления

8. ОСНОВНЫЕ СВОЙСТВА ФАЙЛОВОЙ УСЛУГИ

8.1. Управление файловой активностью

Для того, чтобы прояснить цели настоящего стандарта, необходимо объяснить способ, с помощью которого управляют файловыми активностями.

Для любой передачи файла или доступа к файлу затрагиваются три логических объекта:

логический объект, который принимает на себя управляющую инициативу;

логический объект, который имеет доступ к виртуальному файлу-источнику;

логический объект, который имеет доступ к виртуальному файлу-адресату.

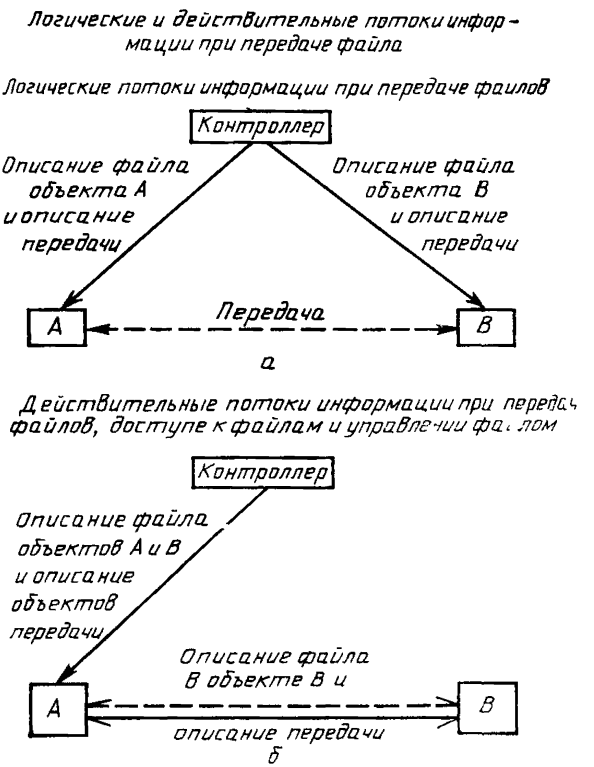
Из контроллера выходят два потока информации:

а) информация, связанная со спецификацией виртуального файла-источника и с ограничениями, накладываемыми на способ, с помощью которого должна быть выполнена передача, посылается в логический объект, имеющий доступ к файлу-источнику;

б) информация, связанная со спецификацией виртуального файла-адресата и с ограничениями, накладываемыми на способ, с помощью которого должна быть выполнена передача, посылается в логический объект, имеющий доступ к файлу-адресату.

Эти потоки и их отношение к передаче представлены на черт 2а

Для упрощения координации и управления передачей файлов предполагается, что контроллер будет проводить эти потоки через один из двух логических объектов файлового протокола, который будет действовать как его исполнитель при выполнении передачи (черт. 2б). Во многих случаях это будет происходить естественно, потому что контроллер и логический объект, инициирующий передачу файла, будут находиться внутри одной и той же системы.



Черт 2

Услуга, определенная здесь, является одной из тех, которые обеспечивают взаимосвязь элементов в объекте *А* и элементов в объекте *В* на черт. 2б, причем объекты *А* и *В* являются пользователями услуги, упомянутыми в определении службы ПДУФ (FTAM).

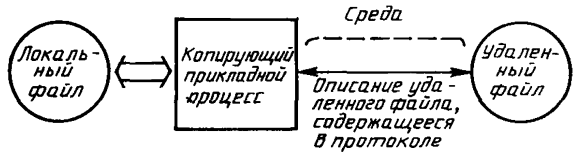
8.2. Асимметрия диалога

Действия, которые должны быть обеспечены файловым протоколом, показывают довольно значительную асимметрию, которая отражается на структуре услуг и протоколов. Асимметрия имеет форму отношений главного управляющего логического объекта и подчиненного.

Во-первых, каждая активность начинается одним пользователем файловой услуги (инициатор *А* на черт. 2б), который должен достигнуть некоторых поставленных целей. Логический объект, связанный с файлохранилищем (ответственный логический объект *В* на черт. 2а), только реагирует на эту инициативу, играя пассивную роль. Это имеет место даже тогда, когда файл передается из одного файлохранилища в другое, потому что про-

токол не обязан содержать информацию о файлохранилище в этом инициирующем логическом объекте. Файловый протокол содержит информацию только о файлохранилище в ответственном логическом объекте. Действие по передаче блоков доступа к файлу в файлохранилище, находящемся в ответственном логическом объекте, может рассматриваться как действие, выполняемое копирующим прикладным процессом, который имеет локальный доступ к одному файлу и удаленный доступ к другому (черт. 3).

Пример диалога между файловыми логическими объектами



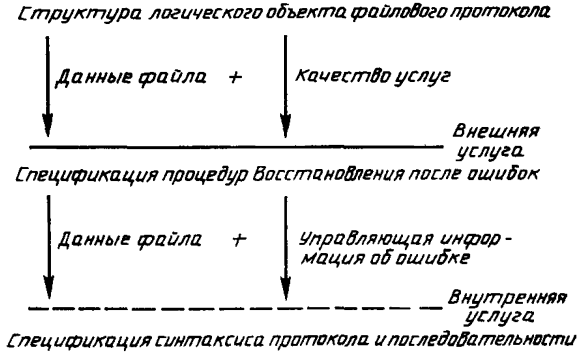
Черт. 3

Вторая асимметрия является более существенной и заключается в том, что при передаче блоков данных доступа к файлу один определенный логический объект является отправителем, а другой — получателем; в любой конкретный момент времени при передаче данных имеется предпочтительное направление потока данных.

8.3. Внешняя и внутренняя файловые услуги

Связь, идентифицированная между абонентом и ответственным логическим объектом в п. 8.2, может быть сама разделена и представлена в модульном виде посредством выделения модулей восстановления после ошибок, выполняя два различных уровня услуг (черт. 4).

Структура логического объекта файлового протокола



Черт. 4

Этими услугами являются:

а) внешняя файловая услуга, при которой пользователь устанавливает свои требования к качеству услуг, но не беспокоясь об обработке ошибок, оставляя эту обязанность поставщику услуг. Передача данных файла моделируется во внешнюю файловую услугу как последовательность безошибочных операций. Таким образом, во внешней файловой услуге не видны корректируемые ошибки или действия по восстановлению после ошибок;

б) внутренняя файловая услуга, используемая протокольным модулем восстановления после ошибок файла. Эта услуга включает примитивы, предоставляющие возможности своим пользователям для обработки ошибок и управления механизмами контрольной точки. Поэтому спецификация протокола, которая устанавливает отношения между внешней и внутренней файловыми услугами, содержит стандартный набор процедур для обработки ошибок, а протокольный модуль, который выполняет эти процедуры, является пользователем внутренней файловой услуги.

Выбор используемых процедур восстановления после ошибок базируется на анализе стоимости службы FTAM и качества услуги связи, запрашиваемой во внешней услуге, и от информации локального административного управления.

8.4. Классы услуг и функциональные блоки

Протокол и услуги службы ПДУФ (FTAM) имеют очень много функций. Они обеспечивают выполнение широкого круга работ, но стоимость реализации всех частей протокола для всех случаев будет соответственно высока. Если бы разным разработчикам был предоставлен выбор функций, которые они могли бы обеспечить, то вероятность того, что этот выбор стал бы одинаковым, будет мала, и соответственно будет мала вероятность взаимосвязи систем.

Чтобы избежать таких проблем, служба ПДУФ (FTAM) определяет два типа функционального выбора. На базовом уровне определяемые функции группируются в функциональные блоки. Реализующая система должна обеспечить функциональный блок либо полностью, либо совсем не обеспечивать. Таким образом ограничивается количество вариантов, которые должны быть выбраны. Определяются модули, которые обеспечивают согласование функциональных блоков при инициализации режима службы ПДУФ (FTAM), это дает возможность двум связанным логическим объектам достигнуть общего представления о доступном наборе функциональных блоков.

Это само собой уменьшает разнообразие, но все еще оставляет значительную свободу. Дальнейшее приближение достигается через определение классов услуг, каждый из которых обеспечивает широкие категории использования. Этими классами являются:

а) класс передачи, который предоставляет возможность перемещения файлов или частей файла между системами, при использовании простых операций с минимумом накладных расходов на протокол до и после передачи данных;

б) класс административного управления, который предоставляет возможность управления виртуальным файлохранилищем посредством последовательности независимых подтверждаемых обменов, но не включая модули передачи файлов;

в) класс передачи и административного управления, который объединяет возможности класса передачи и класса административного управления;

г) класс доступа, который предоставляет возможность инициирующему логическому объекту выполнить последовательность операций с блоками данных доступа к файлу, обеспечивая манипулирование удаленными данными;

д) класс без ограничений, который оставляет выбор функциональных блоков разработчику системы распределенной обработки приложения, предоставляя ему полную гибкость для оптимизации, но не гарантируя наличия общего функционального ядра.

Необходимые классы услуг согласовываются при инициализации режима службы ПДУФ (FTAM).

9. ФУНКЦИИ, СВЯЗАННЫЕ С ФАЙЛОВОЙ УСЛУГОЙ

91. Управление действиями

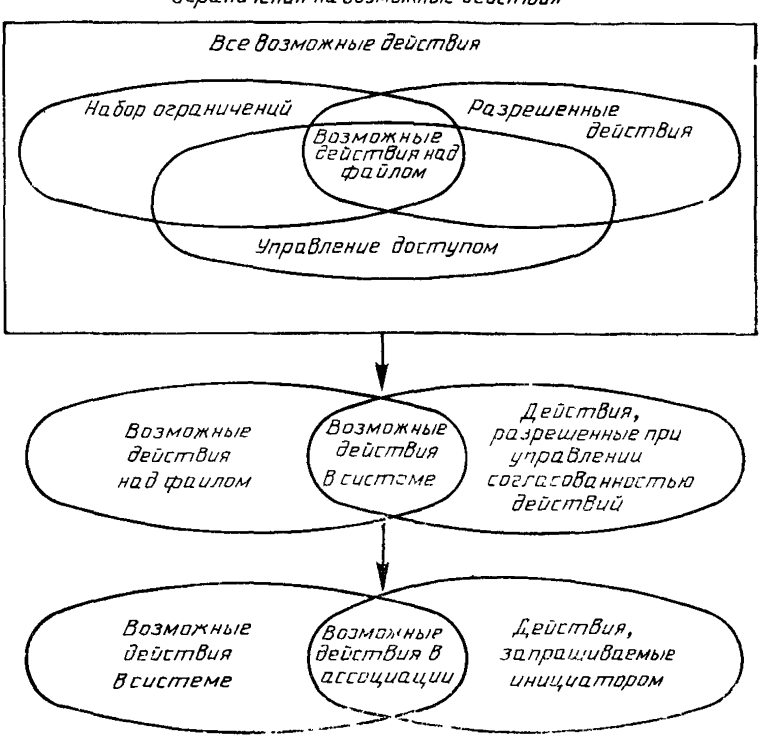
Виртуальное файлохранилище определяет действия, которые могут быть выполнены над файлом. В любой конкретной ситуации доступно только подмножество этих операций. Это подмножество определяется с помощью:

а) атрибутов файла (разд. 2); они указывают действия, соответствующие содержанию сообщения файла (с учетом применяемого набора ограничений), и механизмы локального хранения (посредством атрибута файла «Разрешенные действия») и выражают какие-либо ограничения, накладываемые на управление доступом, влияющие на доступ к файлу;

б) текущего состояния файлохранилища, особенно ограничений, накладываемых совместным доступом при обработке одного и того же самого файла;

в) значений атрибутов взаимодействия, установленных параметрами файловой услуги, если был согласован режим передачи данных.

Каждая из этих причин может ограничить множество возможных действий. Процесс уточнения, приводящий к набору действий, которые фактически могут быть выполнены, представлен на диаграммах Венна на черт. 5.



Черт 5

Первая диаграмма показывает, что действиями, относящимися к файлу, являются только те, которые допускаются структурой файла, как это выражено структурированным набором ограничений, атрибутом «Разрешенные действия» и атрибутом «Управление доступом». Результатом является множество действий, которые в принципе могли бы быть выполнены над файлом.

Полученное в результате множество действий затем еще раз уменьшается, чтобы учесть ограничения, накладываемые какой-либо системой на файлохранилище; наиболее важными из них для доступа к файлу являются ограничения, накладываемые параллельными активностями других пользователей.

И наконец, абонент запрашивает множество действий при установлении режима «Передача данных» и на каждой стадии согласование параметров может приводить к ограничениям либо требуемого множества действий, либо множества действий, которые может допустить система, либо того и другого. Когда ассоциация инициализируется, действия ограничиваются классом услуги и набором согласованных функциональных блоков. Когда

выполняется выбор файла, подмножество этих действий должно стать согласованным результатом из этих разрешенных действий. И, наконец, когда файл открывается, действия, которые должны быть использованы, объявляются находящимися в установившемся режиме обработки.

9.2. Учет

Служба ПДУФ (FTAM) определяет базисный механизм для переноса учетной информации и информации о платежах. С файлом могут быть связаны имена учетных параметров, чтобы отследить ведение расходов, связанных с хранением файла; ведение учета может быть связано с режимами файла для ведения расходов по доступу к информации в файле.

Соответствующие параметры платежей позволяют во время завершения режима обработки файлов выдавать отчеты о понесенных расходах. Учет может быть установлен во время инициализации режима службы ПДУФ (FTAM), но если есть необходимость, он может быть отменен во время выбора определенного файла для того чтобы была возможность оплатить по отдельному счету только за использование этого файла.

Эти модули предоставляют возможность обмена информацией по учету и платежам, но используемая модель ведения учета и модули для управления бюджетом и распределением ресурсов выходят за рамки настоящего стандарта. Рассматриваемые вопросы позволяют охватить многие области деятельности, и служба ПДУФ является только одним из примеров их использования.

9.3. Управление согласованностью действий

Целью введения модулей управления согласованностью действий является обеспечение того, чтобы инициатор имел непротиворечивое представление о файле при наложении ограничений во время совместного доступа к файлу. Эти модули предназначены для того, чтобы предоставить пользователю способ выполнить согласованную последовательность действий без взаимного влияния из-за параллельных доступов.

Индивидуальные действия файлохранилища реализуются таким образом, чтобы любое отдельное действие, например передача данных при чтении или записи, являлось неделимым и последовательным. Это означает, что хотя последовательность действий для множества ассоциаций, в общем случае, не предсказуема, действия в одной ассоциации видны другим ассоциациям только после того, как они закончены.

Управление согласованностью действий обеспечивается исключительно для правильного параллельного выполнения множества задач. Функционирование этого модуля необходимо отличать от функционирования модуля «Управление доступом», который обеспечивает механизмы защиты, и от специфики разрешенных

действий, которые указывают возможности системы, содержащей файлохранилище.

Любое конкретное действие должно быть согласовано с этими тремя модулями управления до того, как оно может быть выполнено.

Обеспечиваются два уровня управления согласованностью действий. Внешний уровень управления доступом ко всему файлу в то время, как внутренний уровень может использоваться для того, чтобы предоставить возможность отдельного управления доступом к отдельным блокам данных доступа к файлу. Внешний уровень или файловый уровень может использоваться во время выбора или создания файла, и управление внешним уровнем отменяется, когда выполняется действие «Отмена выбора файла» или когда файл удаляется. Файл также может модифицироваться во время режима «Открытие файла», если, например, файл сначала открывается для чтения, а затем снова открывается для модификации. Модуль управления согласованностью действий над файлом функционирует в то время, когда файл выбран или открыт, и сохраняется до тех пор, пока не выполнится действие «Отмена выбора файла» или пока файл не будет закрыт.

Управление согласованностью действий с файлом используется отдельно для каждого типа действия, которое может выполняться в виртуальном файлохранилище. Следовательно, имеется независимое управление следующими операциями над файлом: «Чтение», «Вставка», «Замена», «Расширение», «Стирание», «Чтение атрибута», «Изменение атрибута» и «Удаление файла». Это обеспечивает очень мощный набор средств для построения схем согласованности действий, ориентированных на различные применения, включая, в том числе, и простые схемы и схемы, специально предназначенные для специализированных применений. Однако именно из-за их мощности не каждая конфигурация, представленная этими средствами, найдет практическое применение, поэтому при их разработке необходим разумный подход. При каждом режиме «Выбор файла» или «Открытие файла» каждая операция выполняется либо по умолчанию, либо путем указания типа операции одной из следующих категорий:

а) необязательная: я не хочу выполнять эту операцию — другие могут;

б) разделенная: я могу выполнить эту операцию — другие тоже могут;

в) монопольная: я могу выполнить эту операцию — другие не должны;

г) недоступная: никто не может выполнить эту операцию.

Например, абонент в распределенной системе может указать разделенный доступ на чтение и не допускать других операций в

целях защиты общей целостности данных. Любое количество пользователей может затем читать данные во время нормальной работы, но если логический объект административного управления запрашивает файл, требуя монопольный доступ для замены и вставки, чтобы модифицировать его структуру, эта операция не может начаться до тех пор, пока не будут прекращены все другие доступы, и запрещается доступ от других пользователей до тех пор, пока не будет завершена сама эта операция.

Однако, если все пользователи запросили разделенное управление на чтение и монопольное управление на замену для всего файла, только один из пользователей в данный момент времени может иметь доступ к файлу из-за ограничения на операцию замены, и совместная работа не состоится.

Для более четкого разграничения уровней вводится второй тип управления — блокирование блока данных доступа к файлу (БДДФ). Использование этого модуля для блокировки отдельных блоков БДДФ (FADU) согласуется путем выбора соответствующих функциональных блоков во время инициализации режима ПДУФ (FTAM); затем он запрашивается для конкретного файла, когда этот файл открывается.

Если запрашивается блокировка БДДФ (FADU), управление согласованностью действий, указанное во время открытия файла, не применяется сразу (хотя любое запрошенное при выборе файла управление согласованностью действий остается в силе). Вместо этого запрошенное управление применяется на время выполнения каждой запрошенной операции доступа к файлу. Таким образом, запрос на чтение блока БДДФ может в течение выполнения операции «Чтение» ввести в действие разделенное управление согласованностью действий, запрещенное во время открытия файла.

Кроме того, отдельные блоки данных доступа к файлу могут быть заблокированы на некоторый период во время режима открытия файла, ограниченного парой действий доступа к файлу, соответствующим образом отмеченных с помощью некоторого параметра.

Блокировка блока данных доступа к файлу может иметь два состояния — включено или выключено. Если блокировка выключена, действия, указанные как разделенные для целого файла, могут быть выполнены совместно с другими подобными обращениями к файлу. Однако, когда блокировка включена в результате действия пользователя, этот пользователь получает монопольное управление (или никакого доступа в случае, если эта операция не была запрошена предварительно) всем соответствующим блоком данных доступа к файлу. Пользователь не имеет возможности получить управление блоком БДДФ (FADU), если какой-нибудь меньший блок БДДФ (FADU) внутри него забло-

кирован другим пользователем. Пользователь также не может получить управление блоком БДДФ (FADU), если он является частью большего блока FADU, который заблокирован другим пользователем.

Блокировка может быть включена или выключена в соответствии с действиями локализации или действиями по передаче данных; запросы на включение блокировки удовлетворяются до того, как будет выполняться соответствующее действие, в то время как запросы на выключение блокировки удовлетворяются позже. Сами действия, указанные во время выбора файла, как требующие монопольного управления (или не имеющие доступа) для целого файла, не затрагиваются функцией блокировки блоков данных доступа к файлу.

Некоторые системы могут иметь трудности при реализации всех возможностей по управлению согласованности действий. Такие системы могут реализовать более ограниченное управление по сравнению с требуемым, например решение на местном уровне обязать, чтобы все действия имели одинаковый уровень ограничений, равный наиболее ограниченному запросу, но пользователи этих систем отметят соответствующее ухудшение качества услуг.

9.4. Управление доступом

Модули управления доступом, обеспечиваемые настоящим стандартом, основываются на концепции списка управления доступом. Каждый элемент этого списка представляет собой набор операций и ограничений на согласованность действий и набор тестов, которым абонент должен удовлетворять прежде, чем эти операции файлохранилища могут быть выполнены; действия будут разрешены, если условия, заданные каким-либо одним элементом списка, удовлетворяются. Таким образом, список может содержать элементы, каждый из которых разрешает некоторому числу поименованных абонентов читать файл, и элемент, разрешающий тому логическому объекту, который указывает определенный пароль, читать данные из файла и писать данные в файл.

В дополнение к действиям, заданным в элементе, также могут быть включены допустимые комбинации управления согласованностью действий (см. подраздел 9.3). Если они не включены, управление согласованностью действий определяется локально самим файлохранилищем.

Управление согласованностью действий над файлом является более гибким, чем просто ответы «да» или «нет»; это отражается более гибким управлением доступом. Абонент может пожелать выполнить действие, разделяя файл с другими пользователями, имеющими доступ, или абонент может потребовать монопольный доступ. Абонент даже может потребовать, чтобы никакому дру-

гому логическому объекту не разрешалось выполнение действия, которое он сам не уполномочил выполнять. Это обеспечивается с помощью атрибута файла «Управление доступом», который записывает возможность использования каждого варианта управления согласованностью действий отдельно для каждого действия. Управление согласованностью действий допускается только в случае, если оно указано для требуемого определенного типа доступа. Однако абоненту всегда разрешается указать, что именно для него доступ не требуется.

Например, абоненту может быть разрешен запрос на разделение доступа к файлу при чтении и ему может быть предоставлена возможность указать, что ни для кого другого нет доступа для выполнения действия «Удаление файла» на протяжении выполнения операции «Чтение», даже если действие «Удаление файла» само не является разрешенным ни при разделенном, ни при монопольном доступах.

Каждый элемент списка определяет набор разрешенных операций и может указать идентификатор и местоположение абонента (указанием имени логического объекта прикладного уровня), а также любые требуемые пароли для доступа.

При установке режимов службы ПДУФ (FTAM) и режимов «Выбор файла» и «Открытие файла» для различных атрибутов взаимодействия устанавливаются значения, соответствующие возможным элементам списка. В частности, абонент запрашивает выполнение некоторого набора операций, устанавливая атрибут взаимодействия «Текущий запрос доступа» во время установки режима «Выбор файла».

Прежде чем запрошенные действия будут разрешены, соответствующий логический объект просматривает список управления доступом, чтобы определить, совпадает ли значение с каким-либо элементом списка. Если обнаруживается соответствие набору действий и соответствующие проверки удовлетворяются, действия могут быть выполнены; если проверки не удовлетворяются, запрос отклоняется.

Таким образом, в итоге список управления доступом содержит постоянные свойства файла и сохраняется на тот период времени, пока существует файл; проверка доступа выполняется для этого списка всякий раз, когда устанавливается режим, подразумевающий доступ; разрешенный доступ остается действительным на время выполнения этого режима.

9.5. Совершение операций

Настоящий стандарт содержит в протоколе восстановления при ошибках модули для того, чтобы можно было убедиться, что запрошенная передача данных выполнена успешно даже после нескольких сбоев средств связи или систем. Однако имеются требования при выполнении групп или последовательностей дейст-

вий в распределенных прикладных процессах, по которым вся группа действий либо завершается и об этом становится известно, либо происходит возврат в исходное состояние, так что результат выполнения не сохраняется.

Стандарт ИСО 9804, устанавливающий совершение операций, параллельность (согласованность действий) и восстановление ошибок (СПиВ), определяет модули для достижения этих результатов при определенных условиях. Прикладные процессы могут быть составлены комбинированием функций элемента СПиВ (CCR) и службы ПДУФ (FTAM), при этом определяются правила, по которым это делается в случае передачи файла. Это является частным случаем осуществления ассоциации, которую допускает протокол службы ПДУФ (FTAM).

10. ПОСТАВЩИКИ УСЛУГ, ОБЕСПЕЧИВАЮЩИЕ СЛУЖБУ ПДУФ

10.1. Элемент СЭУА — прикладной контекст и среда ПДУФ

Ассоциации прикладного уровня и соответствующие прикладные контексты, необходимые для обеспечения файлового протокола, устанавливаются при помощи сервисного элемента управления ассоциацией, определенного в стандарте ИСО 8649. Необходимые свойства ассоциации и контекста выражаются в терминах сервисных примитивов, с помощью которых инициируются и завершаются отдельные реализации режима службы ПДУФ (FTAM).

Файловый протокол указывает, что эти требования должны удовлетворяться установлением новой ассоциации. В самом деле, первоначально в стандарте по вопросу элемента СЭУА (ACSE) определяется только установление и разъединение ассоциации. Однако, если дополнительные функции элемента СЭУА позволят существующим прикладным ассоциациям возобновляться, чтобы обеспечить среду службы ПДУФ, такие же модули должны быть эквивалентно доступными способами запуска и завершения режима службы ПДУФ.

Свойства прикладной ассоциации составляют прикладной контекст, который имеет имя. Вообще говоря, это имя будет идентифицировать многие различные аспекты прикладного процесса, одним из которых является способность использовать службу ПДУФ. Однако в ГОСТ 34.980.4 определено специальное имя прикладного контекста для случая, когда основная задача — это передача файлов, представляющая активность со своими собственными правами.

В любой момент файловый протокол работает так, что в одной конкретной ассоциации проводится только одна работа с файлом;

если требуется работать с несколькими файлами, устанавливается несколько ассоциаций. По завершении работы с файлом решение начать еще одну работу или разъединить ассоциацию и обеспечивающее ее соединения или модифицировать прикладной контекст, чтобы обеспечить некоторую другую услугу, принимается средствами местного административного управления.

10.2. Услуга уровня представления

Протокольная управляющая информация службы ПДУФ (FTAM) и данные файла объединяются при помощи услуги уровня представления, определенной в ГОСТ 34.971.

Вопросы информации, имеющие смысл для прикладного процесса, не зависят от какого-либо определенного метода кодирования и выражаются с помощью концепции абстрактных синтаксисов. Эти вопросы описываются при помощи определения типов данных как части спецификации абстрактного синтаксиса; они являются единственными аспектами, на которые требуется ссылаться в спецификации прикладных процессов.

Услуга уровня представления управляет представлением информации, имеющей смысл для логических объектов прикладного уровня. Принимая во внимание все элементы, используемые в представлении передаваемой информации, можно назвать три представления. Во-первых, это синтаксис передачи, который является представлением информации, передаваемой между открытыми системами; затем для каждой реальной системы имеется представление информации, используемое внутри этой реальной системы. Все три представления информации представляют единый общий абстрактный синтаксис, соответствующий установленному контексту уровня представления. Однако любой процесс уровня представления должен знать только два из этих представлений — синтаксис передачи и свое собственное локальное представление. Протокол, установленный для взаимодействия между логическими объектами уровня представления, связан только с синтаксисом передачи.

Для любого конкретного прикладного процесса имеют значение некоторые определенные проблемы передаваемой информации; другие проблемы являются только условиями, принимаемыми для того, чтобы предоставить возможность содержать эти указанные аспекты. Различие между главными и второстепенными аспектами является свойством прикладного процесса, и разные прикладные процессы могут отличаться.

Например, прикладной процесс, который передает текстовые сообщения, может заниматься последовательностью слов, формирующих сообщение, но не их точным размещением на распечатываемой странице. С другой стороны, прикладной процесс, который передает текст для наборного устройства, будет заботиться о

каждой детали именно размещения текста. Подобные соображения учитываются и применяются для нетекстовых передач.

Аспекты передачи, которые важны для прикладных процессов, выражаются посредством абстрактного синтаксиса; с точки зрения прикладного процесса, каждый абстрактный синтаксис соответствует единственному контексту уровня представления, и в любой момент времени могут использоваться один или несколько контекстов уровня представления. Когда устанавливается контекст уровня представления между логическими объектами уровня представления, согласовывается подходящий синтаксис передачи между логическими объектами уровня представления. Какие-либо детали уровня представления, не ясные при определении абстрактного синтаксиса, будут согласовываться поставщиком услуги уровня представления. Преобразования, выполняемые логическим объектом уровня представления, ограничены требованием сохранить все аспекты типов данных, определенных в абстрактном синтаксисе.

Набор контекстов уровня представления устанавливается во время инициализации режима службы ПДУФ (FTAM), и это может быть достаточно для предпринимаемых действий. Если должен обрабатываться широкий круг типов файлов, то могут потребоваться средства административного управления контекстами службы уровня представления для изменения набора определенных контекстов, чтобы обеспечить требуемый контекст уровня представления для каждого файла во время его открытия.

Во время передачи отдельных элементов данных прикладной процесс обеспечивает услугу уровня представления значениями данных представления, которые должны быть переданы, типами данных, к которым они относятся, и абстрактным синтаксисом, из которого они взяты; затем для данного контекста уровня представления используется согласованный синтаксис передачи. Выбор протокола прикладного уровня службы ПДУФ (FTAM) подразумевает использование типа абстрактных данных, необходимого для выражения конкретной протокольной управляющей информации (абстрактный синтаксис информации ПУИ службы ПДУФ). ГОСТ 34.974 определяет правила для установления синтаксиса передачи протокольной управляющей информации из абстрактного синтаксиса, которыми должны придерживаться все реализующие системы службы ПДУФ (FTAM). Кроме того, они могут обеспечивать другие стандартные или применяемые на предприятии специфические синтаксисы передачи.

10.3. Услуга сеансового уровня

Логические объекты уровня представления, которые обеспечивают активность службы ПДУФ (FTAM), сами взаимодействуют через услугу сеансового уровня, определенную в стандарте ИСО 8326. Услуга сеансового уровня обеспечивает средства структури-

рования диалога взаимодействия, которые передаются логическим объектам прикладного уровня через логические объекты уровня представления.

Услуга уровня представления может предоставить посредством ниже расположенной услуги сеансового уровня введение точек синхронизации и услуги повторной синхронизации, чтобы обеспечить использование контрольной точки в файле и восстановление при ошибках. Эти услуги дают возможность введения контрольных точек в поток данных пользователя файла, разъединения сеансовых соединений после ошибок и повторной синхронизации механизмов точек синхронизации сеансового уровня до возобновления передачи данных.

Спецификации сеансового уровня также требуют при реализации протокола службы ПДУФ (FTAM) следит за сеансовыми маркерами, управляющими тем, какой логический объект может вводить точки синхронизации.

РАЗДЕЛ 2. ВИРТУАЛЬНОЕ ФАЙЛОХРАНИЛИЩЕ. ОСНОВНЫЕ КОНЦЕПЦИИ

11. ВИРТУАЛЬНОЕ ФАЙЛОХРАНИЛИЩЕ

11.1. Обоснование необходимости использования модели файлохранилища

В существующих реальных системах способы реализации реальных файлохранилищ очень различны. Различные реальные системы имеют широкий круг стилей для описания хранения данных и средств, с помощью которых к данным можно осуществить доступ. Прежде чем использовать в среде ВОС услуги, протоколы и процедуры для передачи файла, доступа к ним и административного управления, необходима общая модель для описания файлов и их атрибутов. Эта модель называется «виртуальное файлохранилище».

Такое определение файлохранилища является очень мощным средством, так как оно дает возможность абсорбировать различия в методах и спецификациях в функцию отображения из открытой системы в реальную открытую систему, в результате чего любая конкретная реальная открытая система может взаимодействовать с другими отличными от нее реальными открытыми системами в терминах, которые могут быть поняты однозначно. Скрытие подробностей работы реальных систем от пользователя внешней связи уменьшает необходимость в модификации существующих реальных систем и, следовательно, уменьшает начальную стоимость разработки взаимодействия открытых систем

Необходимо обойти большие разнообразия в способах обслуживания файла Реализующая система файлового протокола, которая действует в роли абонента, может быть вызвана непосредственно пользователем человеком, подсистемой, обрабатывающей очередь выданных запросов на обработку файлов, или прикладной программой, написанной пользователем Реализующая система файлового протокола, которая действует как ответственный логический объект, может непосредственно иметь доступ к реальному файлохранилищу или взаимодействовать с прикладной программой, написанной пользователем Во всех этих случаях будет использоваться один и тот же файловый протокол

Выражение диалога в терминах виртуального файлохранилища дает возможность взаимосвязи широкого круга реальных систем различной сложности Определение некоторого числа дополнительных функциональных блоков в определениях файловой услуги и нескольких дополнительных групп атрибутов в определениях виртуального файлохранилища делает возможным такой способ работы, при котором простые реальные открытые системы взаимодействуют с более сложными системами, например сложная вычислительная система может связаться с внешней памятью интеллектуального терминала или с устройством ввода вывода Данный метод служит не только для того, чтобы скрыть различия между подобными способами хранения данных, но также и различия в типах или сложности Однако требуются официальные соглашения, которые устанавливают, какие из атрибутов, определенных в настоящем стандарте, действительно обеспечиваются, чтобы потенциальный пользователь мог удостовериться, что любые необходимые свойства, такие как секретность или управление согласованностью действий, обеспечиваются реальным файлохранилищем

11 2 Определение отображения виртуального файлохранилища

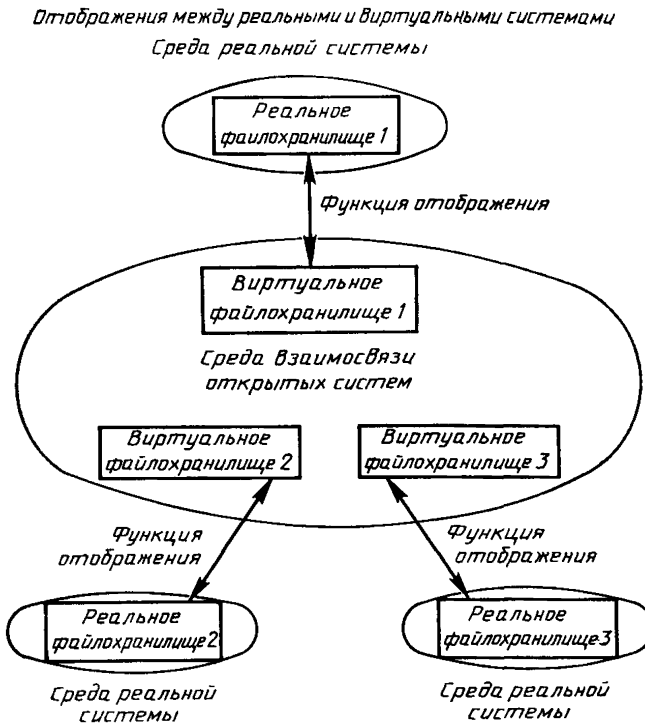
Чтобы использовать файловую услугу и файловый протокол, реализующая система должна связать элементы определения виртуального файлохранилища с имеющейся реальной системой хранения данных

Разработчик, использующий настоящий стандарт, обеспечивает отображение (черт 6) между

а) действиями, блоками данных доступа к файлу, атрибутами файлов и атрибутами взаимодействия, определенными в виртуальном файлохранилище в среде взаимосвязи открытых систем СВОС (OSIE),

б) ресурсами в среде реальных систем СРС (RSE)

Когда протокольные сообщения принимаются логическим объектом прикладного уровня, они интерпретируются в терминах корреспонденций, установленных этим логическим объектом меж-



Черт. 6

ду компонентами виртуального файлохранилища, действиями и теми аспектами среды реальной системы, которые относятся к хранению информации. Отображением поэтому является множество корреспонденций, установленных разработчиком реальной открытой системы.

11.3. Формат виртуального файлохранилища

Определение виртуального файлохранилища формирует схему для описания информации файла. В этом описании файл является логическим объектом, имеющим·

а) единственное имя файла, которое позволяет обращаться к нему без неопределенности;

б) другие описательные атрибуты файла, которые выражают свойства файла, такие как учетная информация, предыстория и так далее;

в) атрибуты файла, выражающие действия, возможные при работе с файлом;

г) атрибуты файла, описывающие логическую структуру и размерности данных, хранящихся в файле;

д) какие-либо блоки данных доступа к файлу, формирующие содержимое этого файла.

Это составляет все аспекты файла, которые могут отслеживаться санкционированным абонентом Если два абонента выполняют запрос на одинаковые аспекты одного и того же файла, они получат одинаковую информацию о свойствах файла при условии, что между этими запросами не была произведена модификация данной информации Эти свойства называются атрибутами файла

Имеются также атрибуты взаимодействия, которые описывают возможность взаимодействия между файлом и соответствующим абонентом, относящиеся к таким свойствам, как аутентификация права пользователя на доступ, варианты передачи данных, суммарная стоимость и тому подобное; имеется независимый набор значений этих атрибутов взаимодействия для каждой действующей активности Это множество атрибутов создается после того, как установится режим службы ПДУФ (FTAM), включая инициализацию файлохранилища, и поддерживается до тех пор, пока этот режим существует, и уничтожается, когда этот режим прекращается.

Некоторые атрибуты файла накладывают ограничения на структуру содержания сообщения файла. Эта структура сохраняется на период существования файла. Однако не все пользователи, имеющие доступ к файлу, могут обращаться к целому файлу. Например, у пользователя может возникнуть необходимость доступа к сложному иерархическому файлу, как будто этот файл является двухмерным, чтобы сформировать итоговые сообщения, или у него нет необходимости получать независимо для всех случаев доступ к наименьшим структурным блокам этого файла Дополнительно, к атрибутам файла, используемым при создании файла и административном управлении файлом, для описания постоянной структуры доступа к файлу имеется специальный контекст доступа, указывающий при передаче данных во время чтения подмножество информации о структуре файла и данных пользователя при передаче блока данных доступа к файлу

11 4 Динамика атрибутов

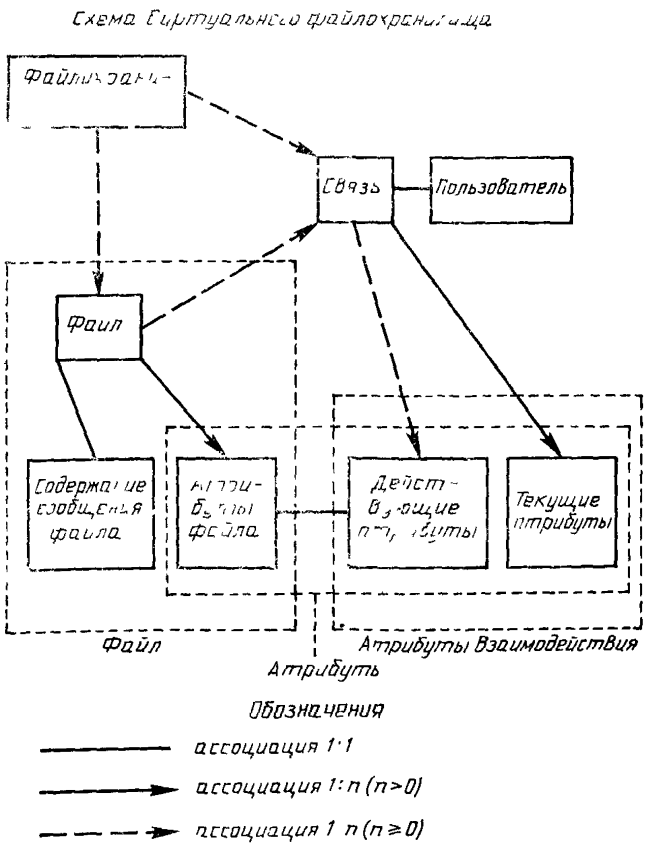
Атрибуты файла отображают текущее состояние файла. Связь между абонентом и ответственным логическим объектом строится по неодинаково разделенной схеме этих атрибутов, отражая знание о них, которое абонент получает от этой конкретной связи Таким образом, в принципе, для каждого атрибута файла существует действующий атрибут, выражающий информацию о нем, которая была передана абоненту через параметры сервисных примитивов службы ПДУФ (FTAM)

Независимо от этого имеются текущие атрибуты, которые описывают данный установленный режим службы ПДУФ; они включают данные об идентификации и местоположении абонента, также результаты согласований в начале различных, ранее установленных режимов.

Действующие и текущие атрибуты вместе называются атрибутами взаимодействия. Это как раз те атрибуты взаимодействия, которые определяют работу модулей файловых протоколов.

11.5. Схема файлохранилища

Предшествующее описание формата виртуального файлохранилища может быть представлено в виде схемы, связывающей различные концепции и принципы их взаимосвязей. Эта схема графически представлена на черт. 7.



Черт. 7

12. СТРУКТУРЫ ФАЙЛОВ

12.1. Категории структуры

Файл может не содержать ни одного или содержать один или несколько идентифицируемых блоков данных. Эти блоки данных связаны логическим образом. Модель файлохранилища, определенная в ГОСТ 34.980.2, обеспечивает древовидную структуру для представления связей между блоками данных, для доступа к этим блокам данных и их идентификации. Эта древовидная структура называется структурой доступа к файлу. Каждый узел структуры доступа к файлу не имеет или имеет один связанный с ним блок данных. Структура информации с точки зрения пользователя файловой услуги может отличаться от структуры доступа к файлу, и пользователь файловой услуги должен отображать свою семантику в структуру доступа к файлу.

Примечание. В общем случае эти отношения могут быть либо последовательными, либо иерархическими, сетевыми или реляционными Осознана необходимость в других моделях, помимо иерархических Такие модели не включены в настоящий стандарт, но будущие дополнения или переработки возможно включат дополнительные модели (например для сетевых или реляционных баз данных) или будут указаны ссылки на модели, определенные в других стандартах

Единицей данных, в терминах которой выполняются операции над содержанием сообщения файла, является «Блок данных доступа к файлу (БДДФ)», который сам может быть структурирован и содержит блоки данных с определенным абстрактным синтаксисом. Подмножество информации структуры доступа к файлу и данные пользователя, передаваемые в конкретном действии «Чтение», определяются контекстом доступа, указанным для этого действия.

Имеются четыре аспекта структуры файла, каждый из которых несет различную информацию об этом файле:

а) структура доступа к файлу — описывает состав файла в виде блоков данных доступа к файлу;

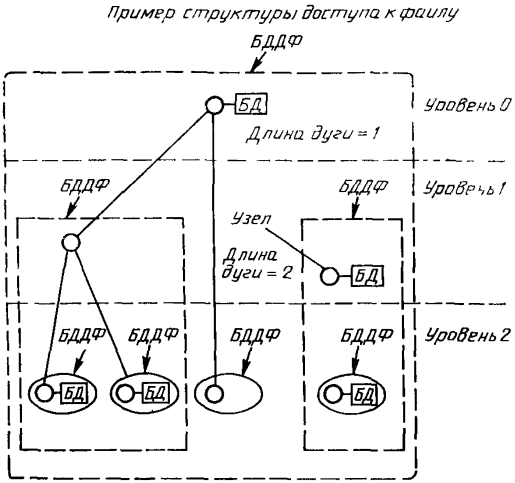
б) структура уровня представления — описывает абстрактную структуру блоков данных, которая определяется внутри структуры доступа к файлу;

в) структура передачи — описывает упорядочение блоков данных доступа к файлу при передаче;

г) структура идентификации — описывает назначение имен узлам в структуре доступа к файлу и идентификацию передаваемых блоков данных доступа к файлу.

12.2. Структура доступа к файлу

Структура доступа к файлу является в первую очередь статическим видом файла. На черт. 8 показан пример структуры доступа к файлу в виде дерева. Структура доступа к файлу опре-



Черт. 8

деляет, к каким частям файла имеется независимый доступ при использовании службы ПДУФ (FTAM).

Использование иерархической структуры для границ доступа к файлу не препятствует прикладным процессам представлять неиерархические логические структуры, используя ссылки в содержании сообщения блоков данных доступа к файлу.

12.3. Структура уровня представления

Структура уровня представления описывает отношения между элементами данных, блоками данных и содержанием сообщения файла. Из нее выводятся правила для передачи информации содержания сообщения файла с помощью услуги уровня представления. Следующее описание указывает, как выполняются правила при разделении информационного содержания сообщения файла на блоки данных и элементы данных. Подробная спецификация приведена в ГОСТ 34.980.2 и в любом определении типа документа, относящегося к файлу. Другими словами, описание указывает, как структура записи реальных файлов может быть отображена в структуру файла службы ПДУФ (FTAM) и как значения данных уровня представления отображаются в примитивы P-DATA (блоки данных услуги уровня представления СБДУП. Предполагается, что файл определен с помощью нотации абстрактного синтаксиса АСН.1:

а) содержание сообщения файла и информация о структуре состоят из последовательностей элементов «Элемент данных»;

б) имеется однозначное соответствие между элементами «Элемент данных», определенными в модуле БДДФ (FADU) нотации АСН.1 (определенном в ГОСТ 34.980.2), и значениями данных, передаваемых к поставщику и от поставщика услуг службы ПДУФ (FTAM) через сервисный примитив F-DATA;

в) в этих элементах «Элемент данных» элементы «Элемент данных содержания сообщения файла» (данные пользователя) обычно определяются так же, как один из типов АСН.1;

г) если реальный файл со структурой записи должен быть передан как неструктурированный файл, основным типом будет тип CHOICE (ВЫБОРОЧНЫЙ ТИП) нотации АСН.1 для различных типов записи в файле (если файл содержит только один тип записи, то тип CHOICE (ВЫБОРОЧНЫЙ ТИП), конечно, будет отсутствовать). Если службой FTAM должен быть обеспечен доступ к отдельным записям, то будет указано, что в каждом элементе «Блок данных» должно быть только одно значение базисного типа.

Примечание. Например, для типа документа FTAM 1 (неструктурированный текстовый файл), типом элемента «Элемент данных содержания сообщения файла» является «Строка графических символов» и предполагается, что каждый элемент типа «Графическая строка» соответствует одной строке текста;

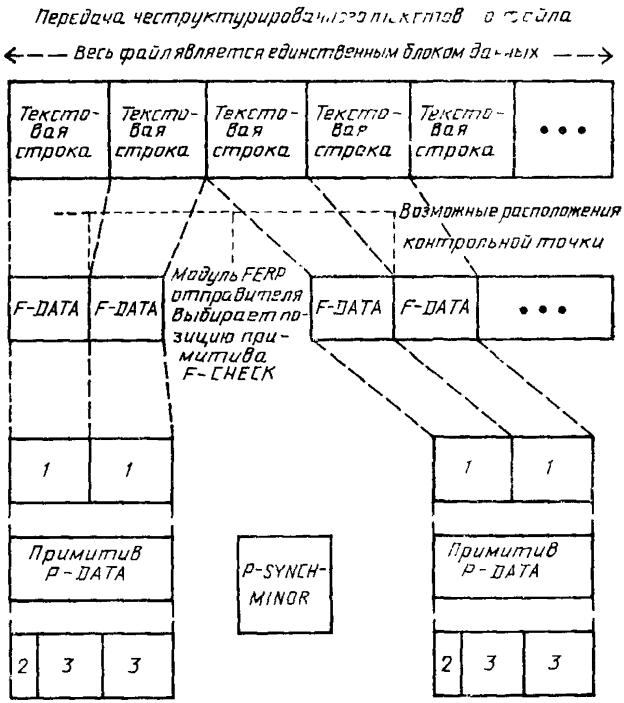
д) протокол уровня представления позволяет собирать значения данных уровня представления в соответствующие группы для передачи. Степень такого группирования определяется выбором варианта отправителя; это дело данной системы, минимизирующей накладные расходы на передачу. Получатель не знает заранее, сколько значений базисного типа будет введено в каждый примитив P-DATA;

е) контрольные точки могут вставляться только между примитивами P-DATA, таким образом размер группы, упомянутой выше в пункте д), определяется положением контрольных точек, вставляемых во время передачи файла;

ж) отметим, что более подробно структурированный файл передается как «группа значений элементов типа «Элемент данных» (ГОСТ 34.980.2). Каждый элемент типа «Элемент данных» соответствует одному значению данных примитива F-DATA, и некоторые из них могут быть отображены в один примитив P-DATA (ГОСТ 34.980.4). Последовательность элементов типа «Элемент данных» описывается с использованием нотации АСН.1, но для типов SEQUENCE (ПОСЛЕДОВАТЕЛЬНОСТЬ) и SEQUENCE OF (ПОСЛЕДОВАТЕЛЬНОСТЬ ИЗ), имеющих место в расширенном наборе правил для элементов типа «Поддерево», «Порождение» и «Блок данных» не выполняется кодирование в передаваемом потоке, так как эти структурированные определения

формируют вторичную точку входа, и на них не ссылаются файловые протоколы.

Отношения между элементами данных, файловыми услугами, значениями данных уровня представления и сервисными примитивами уровня представления и их кодированием указаны на черт. 9, на котором представлена передача неструктурированного текстового файла.



FERPM—модуль протокола восстановления файла после ошибок
1 — значение данных уровня представления; *2* — идентификатор контекста уровня представления; *3* — кодирование значения данных уровня представления

Черт 9

13. НАБОРЫ ОГРАНИЧЕНИЙ

Общая иерархическая структура, описанная в разд. 12, может представлять очень широкий ряд различных фактических структур файла. Однако любому отдельному прикладному процессу будет необходимо конкретное множество структур, а любая реальная система, вероятно, обеспечивает ограниченный набор типов файлов с ограничениями, накладываемыми на способ мо-

дификации файлов. Чтобы выразить эти ограничения, вводится понятие «Набор ограничений». Набор ограничений устанавливает ограничения, накладываемые на область допустимых структур, и определяет, как основные действия по доступу к файлу могут модифицировать структуру без изменения ее основной сущности. Наборы ограничений, отражающие некоторые общие типы файлов, включены в настоящий стандарт, но впоследствии могут быть определены и зарегистрированы другие наборы ограничений.

Каждый набор ограничений идентифицируется значением идентификатора объекта, установленным настоящим стандартом или какими-либо другими механизмами, определенными в ГОСТ 34.973.

Настоящий стандарт определяет наборы ограничений для моделирования ряда широко используемых структур, которые являются подмножествами общих иерархических структур. Например, различные ограничения, накладываемые на имя узла, отражают различные классы индексной структуры. Определены следующие наборы ограничений, накладываемые на:

а) неструктурированный файл, в котором существует единственный блок данных без имени;

б) последовательный двухмерный файл, в котором имеется последовательность непоименованных блоков данных; это ограничение может быть, например, использовано для моделирования последовательных входных и выходных файлов программ на языке Фартран;

в) упорядоченный двухмерный файл, в котором имеется последовательность поименованных блоков данных; определяется обработка блоков данных с одинаковыми именами;

г) упорядоченный двухмерный файл с уникальными именами, в котором имеется последовательность уникально поименованных блоков данных;

д) упорядоченный иерархический файл, в котором допускается общая иерархия для того, чтобы моделировать использование многоиндексных систем; вставка основана на позиции в этих индексах;

е) общий иерархический файл, в котором допускается общая иерархия с полным управлением по размещению новых узлов при модификации структуры;

ж) общий иерархический файл с уникальными именами, при котором к набору ограничений на общий иерархический файл добавляется гребование об использовании уникальных имен.

14. ТИПЫ ДОКУМЕНТОВ

Содержание сообщения файлов может быть нескольких типов. Можно отдельно указывать модель файла, набор ограничений и

абстрактный синтаксис блоков данных. Однако это сложно, и предпочтительнее указывать семантику, абстрактный синтаксис, синтаксис передачи и динамику структуры единым образом. Это выполняется с помощью определения «Тип документа».

Определения типов документов могут квалифицироваться группой из одного или нескольких параметров. Эти параметры позволяют эффективно сразу определить семейство тесно связанных типов документов. Например, единственное определение может охватывать некоторое число типов текстовых документов с определенным набором или с максимальной длиной строки, задаваемых параметрами.

Для некоторых прикладных процессов может быть достаточным ограниченный вид структурированного файла во время чтения данных от этого процесса. Определение типа документа может включать объявление других типов документа, которые выводятся из данного с некоторой потерей информации; этот процесс известен как «Упрощение исходного типа документа». Подобно этому определение может допускать изменение некоторых параметров, чтобы уменьшить ограничения; этот процесс известен как «Релаксация исходного типа документа».

Тип документа регистрируется международной или национальной официальной регистратурой (см. стандарт ИСО 9834/2 для определения процедур регистрации) в связи с общим интересом или из-за его важности по отношению к указанной группе прикладных процессов. Объект регистрации получает имя, являющееся значением идентификатора объекта.

Тип документа состоит, в основном, из набора связанных операторов, некоторые из которых указывают на их целевое использование. Этот набор включает:

а) идентификацию типа документа и всех объектов информации, на которые он ссылается;

б) его целевую область приложения; типы документов, указанные в определении виртуального файлохранилища и используемые для взаимосвязи файлов, являются объектами настоящего стандарта;

в) те параметры, которые относятся к типу документа;

г) ограничения на семантику, которые должны применяться при интерпретации документа;

д) структуру доступа к файлу документа в терминах модели файла и набора ограничений, которые должны применяться;

е) абстрактный синтаксис или синтаксисы информации о структуре и блоках данных внутри структуры;

ж) способ, которым информация формируется в последовательность значений данных уровня представления и процедуры для их передачи;

з) синтаксис или синтаксисы передачи, при которых определенные абстрактные синтаксисы могут быть объединены;

и) детали операций, обрабатывающих тип документов, определенных в соответствующих сервисных элементах прикладного уровня СЭПУ (ASE). Например, они могут включать результат сцепления двух экземпляров типа документа или способы, при которых структура доступа к файлу может быть упрощена путем рассмотрения его в качестве экземпляра документа более простого типа.

Концепция «Тип документа» может применяться рекурсивно, позволяя выполнить пошаговое уточнение для последующих более незначительных классов возможных документов. Таким образом, например, может быть определен тип документа, который предоставляет свободу соглашений по множеству абстрактных синтаксисов, допускаемых в блоках данных. Затем на этот тип документа могут быть сделаны ссылки в ряде типов документов, которые разделяют одинаковые общие свойства, но каждый из которых добавляет различные ограничения на используемые абстрактные синтаксисы.

РАЗДЕЛ 3. ОБЗОР ФАЙЛОВОЙ УСЛУГИ И ФАЙЛОВОГО ПРОТОКОЛА

15. ФАЙЛОВАЯ УСЛУГА

Файловая услуга и обеспечивающий ее протокол относятся к поэтапному созданию рабочей среды, в которой могут иметь место активности, запрашиваемые абонентом. При диалоге в порядке очередности файловая услуга:

а) позволяет абоненту и ответственному логическому объекту (файлохранилищу) получать информацию друг о друге, включая идентификацию, исходя из названий логических объектов прикладного уровня;

б) идентифицирует необходимый файл;

в) устанавливает атрибуты, описывающие файл и передачу данных большого объема, которые должны иметь место в этой активности;

г) включает административное управление файлом;

д) определяет позицию блока данных FADU в структуре доступа к файлу, к которому должно быть обращение;

е) вставляет, замещает, расширяет или уничтожает один или несколько полных блоков данных FADU.

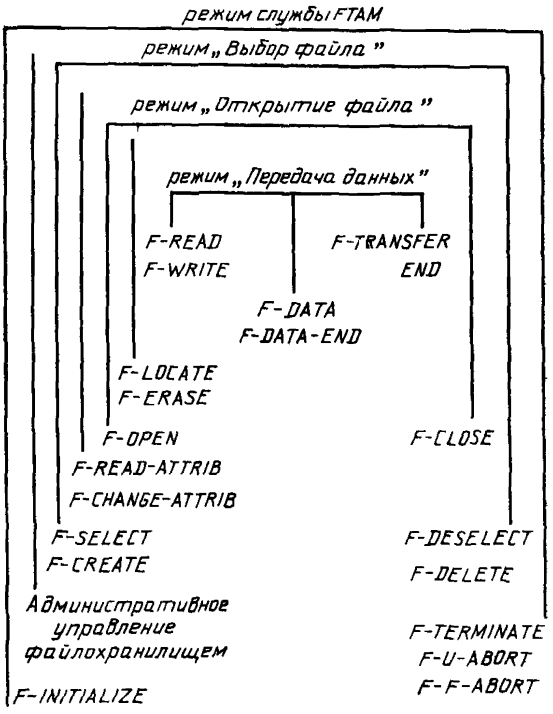
Эти шаги создают различные части набора контекстов файлов. Период времени, в течение которого некоторая часть общего состояния, поддерживаемого пользователями услуги, является допус-

тимой, называется режимом. Когда в процессе развития разделенное состояние расширяется, создается вложение соответствующих режимов. Однако обычно между установлением режима на двух концах соединения должна быть временная задержка.

Период времени, в течение которого протокольные обмены имеют определенное назначение, такое как установка или освобождение прикладного контекста, называется фазой. Для каждой фазы определяется набор допустимых сообщений, исходя из переходов состояний. В любое время каждый логический объект, участвующий в режиме службы ПДУФ (FTAM), находится в одной определенной фазе; фазы не могут быть вложенными одна в другую; активность выполняется через последовательность фаз.

Фазы вводятся в последующих подразделах в таком порядке, в котором они будут иметь место во время типового использования файловой услуги.

Вложение режимов показано на черт. 10.



Черт. 10

15.1. Фаза «Инициализация режима службы ПДУФ»

Фаза «Инициализация режима службы ПДУФ (FTAM)» является первой в любом примере файловой услуги. Во время этой фазы устанавливается прикладной контекст и начальное состояние режима службы ПДУФ, гарантирующие, что ассоциация нижележащего прикладного уровня связывает правильные адреса, согласуемые доступные классы услуг и функциональные блоки. Также устанавливается специфическая информация службы ПДУФ, такая как информация санкционирования и учетная информация, необходимые для успешного выполнения последующих операций файлохранилища.

15.2 Фаза «Административное управление файлохранилищем»

Операции в фазе «Административное управление файлохранилищем» находятся на стадии изучения.

15.3. Фаза «Выбор файла»

Во время выполнения фазы «Выбор файла» устанавливается режим «Выбор файла», идентифицируется или создается уникальный файл, для которого будут выполняться операции в последующих фазах. Фаза «Выбор файла» продолжается до тех пор, пока она явно не будет заменена фазой «Отмена выбора файла» или фазой «Завершение режима службы ПДУФ». Операции, выполняемые в последующих фазах, следовательно, относятся к выбранному файлу и не содержат явной идентификации файла. Фаза «Выбор файла» может включать создание нового файла с указанными свойствами.

Выбор файла по настоящему стандарту осуществляется по имени файла. Запрос на выбор файла может включать информацию управления доступом к указанному файлу.

Примечание. В некоторых последующих дополнениях к настоящему стандарту выбор файла может быть выражен в виде ряда ограничений, накладываемых на атрибуты файлов, а не на имя файла (ГОСТ 34.980.2), значения которых служат для идентификации требуемых файлов

15.4. Фаза «Административное управление файлом»

Операции, выполняемые в фазе «Административное управление файлом», дают возможность пользователю файловой услуги выполнять действия административного управления над выбранным файлом. Например, могут быть прочитаны или отредактированы некоторые атрибуты файла.

15.5. Фаза «Открытие файла»

Во время выполнения фазы «Открытие файла» устанавливается режим «Открытие файла», при котором имеет место передача блоков данных доступа к файлу. Во время этой фазы устанавливаются средства передачи данных, включая некоторые спе-

циальные контексты уровня представления, необходимые для передачи.

15.6. Фаза «Доступ к данным»

Целый файл представляет собой единственный блок данных доступа к файлу, но в нем могут существовать более малые блоки данных БДДФ (FADU). Действия над содержанием сообщения файла переносятся и на блоки данных БДДФ (FADU). Блоки данных содержат, в основном, набор элементов данных, но эти элементы данных не могут быть затребованы или отредактированы независимо.

Фаза «Доступ к данным» представляет период времени, в течение которого выполняется некоторое количество операций (для класса передачи только одна). Этими операциями являются либо операции по передаче данных большого объема, либо операции управления. Каждая операция по передаче данных большого объема содержит:

а) оператор, указывающий операцию, которая должна выполняться, задавая тип операции и идентифицируя блок данных доступа к файлу, для которого эта операция должна применяться:

б) любую необходимую передачу данных;

в) завершение обмена.

Операции управления являются простыми командами и ответами. Определяемыми операциями являются:

г) операция «Определение местоположения», которая указывает узел внутри структуры файла; этот узел является корнем некоторого блока данных доступа к файлу;

д) операция «Стирание», которая удаляет блок данных FADU из структуры файла.

15.7. Фаза «Закрытие файла»

Фаза «Закрытие файла» завершает режим «Открытие файла», установленный во время фазы «Открытие файла»; она заканчивает период, в котором возможна передача данных. Фаза «Закрытие файла» дает возможность обмена информацией о состоянии в конце режима «Открытие файла». Она обеспечивает подтверждение того, что все требуемые передачи данных были завершены либо успешно, либо со сбоем; если завершение успешное, предшествующие действия не будут заново вызываться. При успешном закрытии файл содержит результат всех действий, выполненных во время фазы «Передача данных», несмотря на сбои связи или прикладного уровня, которые могли иметь место впоследствии. Сбой после открытия и перед успешным закрытием оставляет файл в неопределенном состоянии; состояние файла может быть установлено заново при помощи процедур восстановления при ошибках.

15.8 Фаза «Отмена выбора файла»

Фаза «Отмена выбора файла» отменяет режим «Выбор файла», установленный во время фазы «Выбор файла» Фаза «Отмена выбора файла» может включать стандартные последовательности учета. Она сохраняет, в сущности, любую идентификацию аутентификации или идентификацию учета, установленные до фазы «Выбор файла». Фаза «Отмена выбора файла» может вызвать уничтожение выбранного файла.

15.9 Фаза «Завершение режима службы ПДУФ (FTAM)»

Эта фаза аннулирует режим, совместно используемый абонентом и ответственным логическим объектом, отменяя любой режим аутентификации и режим учета Кроме того, в дальнейшем не остается никакого режима, связанного с файлом; активность файловой услуги завершается.

16. МОДУЛИ, ИСПОЛЬЗУЕМЫЕ В ФАЙЛОВОМ ПРОТОКОЛЕ

16 1 Модуль состояний протокола

Основные действия базисного протокола заключаются в установлении и завершении ряда вложенных режимов В самом внутреннем вложенном режиме передача данных является простым однонаправленным потоком с возможной вставкой контрольных точек Следовательно, реализация протокола может быть точно описана с помощью модуля состояний, непосредственно смоделированного по подобию кратких описаний и состояний, приведенных в ГОСТ 34 980 4 (ИСО 8571/4).

Стоимость и сложность реализующей системы точно соответствуют сложности модуля состояний, на котором она базируется. Классы услуг и функциональные блоки должны быть, следовательно, выбраны таким образом, чтобы дать простой модуль состояний с наименее возможной взаимозависимостью между функциональными блоками.

16 2 Группирование протокольных блоков данных

Один из способов, при котором механизм состояния упрощается, является введение обязательных групп в класс услуг передачи Для того, чтобы взаимосвязать протокольные блоки данных, которые устанавливают режимы «Выбор файла» и «Открытие файла», требуются соответствующие реализации; подобное ограничение применяется отдельно к тем блокам, которые завершают взаимосвязь Пороговый параметр в начале взаимосвязи устанавливает количество блоков, которое должно быть успешно реализовано до того, как будут выполнены запрашиваемые действия Реализации класса услуг передачи требуются для установле-

ния порогового параметра на группирование таким образом, чтобы либо все требуемые режимы устанавливались бы, либо все запросы на установление не достигали бы успеха (т. е. действия должны быть приведены в исходное состояние).

Результатом такого группирования является то, что протокольные блоки данных, связанные с выбором, управлением атрибутами и открытием файла, могут быть рассмотрены, с точки зрения разработчиков, в качестве подполей в одной передаче данных. Механизм состояния для инициализации некоторого режима файла имеет только два стабильных состояния (инициализированное и открытое) и два состояния ожидания.

Однако с помощью выполнения группирования, необязательного в доступе, и классами услуг без ограничений появляется большая гибкость для обеспечения широкого круга приложений в вышепредставленном совместном способе.

16.3. Прозрачность

Для поставщика услуг уровня представления и данные пользователя файла и протокольная управляющая информация службы ПДУФ (FTAM) являются последовательностью значений данных уровня представления. В общем случае, требуется наличие возможности передавать произвольные файлы без неопределенности между данными пользователя и протокольной управляющей информацией.

Это означает, что должны быть предприняты шаги для предохранения произвольного ряда значений данных уровня представления в данных пользователя файла, которые могут быть синтаксически взяты в качестве протокольной управляющей информации, которая так будет интерпретироваться.

Решение, принятое в протоколе службы ПДУФ (FTAM), состоит в том, чтобы выделить использование двух контекстов уровня представления. Первым контекстом уровня представления, используемым в режиме службы ПДУФ (FTAM), считается такой, который должен быть контекстом уровня представления протокольной управляющей информации ПУИ (PCI) службы ПДУФ (FTAM), а в другом контексте уровня представления значения данных уровня представления могут быть явно идентифицированы как отличающиеся от протокольной управляющей информации ПУИ (PCI) режима службы ПДУФ (FTAM) через модули протокола уровня представления, даже если они взяты из того же абстрактного синтаксиса и являются синтаксически хорошо сформированными блоками данных протокола.

Так как модуль прозрачности базируется на контексте уровня представления, который является примером использования абстрактного синтаксиса, прозрачность обеспечивается даже при условии, если данные пользователя файла представлены в абстрактном синтаксисе протокольной управляющей информации

ПУИ (PCI) режима службы ПДУФ (FTAM) Следовательно, должны были бы существовать два особых контекста уровня представления, соответствующих этому абстрактному синтаксису

16 4 Ввод контрольной точки

Контрольные точки вводятся в поток данных пользователя файла в точках, выбираемых отправителем, чтобы дать возможность выполнить операции по рестарту и функции восстановления при ошибках И отправитель и получатель должны хранить информацию о местонахождении в файле активных контрольных точек (т е контрольные точки, введенные или полученные, но еще не замеченные из за подтверждения более поздней контрольной точки)

В общем случае также необходимо сохранить информацию, связывающую точку в структуре файла, в которую вставляется контрольная точка Эта информация разделяется на две категории

а) семантическая информация, не выраженная явно в содержании сообщения файла, такая как позиция в структуре доступа к файлу или иные связи между блоками данных, такие как ссылки к указателям и идентификаторам, она не требует дополнительной информации, которая не сохраняется с файлом,

б) информация о состоянии интерпретации абстрактного синтаксиса или синтаксиса передачи Если где-либо была допущена возможность введения контрольной точки, то состояние интерпретации соответствующего синтаксиса передачи также необходимо сохранить Эта информация потенциально является дополнительной информацией, выраженной явно в хранимом файле, потому что трансформация местного синтаксиса, управляемая услугой уровня представления (такая как переупорядочивание полей в прикладных процессах, ориентированных на запись), может сделать невозможным вывод принятого состояния из хранимого в памяти состояния, или действительно запомнить всю информацию, прежде чем будет заполнен блок семантики

Чтобы избежать необходимости при реализации обеспечивать этот специальный второй тип информации, позиционирование контрольных точек ограничивается таким образом, чтобы они находились только в границах семантики заполненных блоков, определенных в абстрактных синтаксисах

При наличии информации о структуре файла службы ПДУФ (FTAM) эти ограничения выражаются в нотации АСН 1 с помощью выражения структуры файла и содержания сообщения файла как ряд связанных элементов данных, каждый из которых кодируется независимо Абстрактный синтаксис протокола ссылается на эти элементы с помощью одного символа, расширением которого является тип CHOICE (ВЫБОРОЧНЫЙ ТИП) из доступных элементов данных Взаимосвязь семантик со струк-

турной точки зрения определяется использованием нотации АСН.1, исходя из типа данных, представляющих полную иерархическую модель файла, но этот тип данных никогда не упоминается в абстрактном синтаксисе протокола и, таким образом, не кодируется.

Подчиняясь этим ограничениям, контрольные точки размещаются в позициях, определяемых отправителем, который может быть или не может быть связан с границами памяти в той или другой системе. Получатель должен быть готов принять контрольные точки в любой допустимой позиции в файле.

16.5. Диагностика и результаты

Протокол обеспечивает два уровня информации, зависящие от успешного завершения или завершения со сбоем запрошенных операций.

Первый уровень представляет собой индикацию на самом общем уровне, в зависимости от завершения режима самих модулей протокола; это сигнализируется с помощью двух параметров результата, указывающих:

а) успешное выполнение каких-либо запрошенных передач модуля протокола;

б) успешное выполнение каких-либо запрошенных действий файлохранилища.

Например, при удалении файла, передача протокола отмены выбора файла завершается всегда успешно, но действие файлохранилища по удалению файла может закончиться со сбоем.

Значение параметра «Результат действия» может указывать либо успешное завершение, либо сбой, после которого возможно восстановление (которое может инициировать протокол восстановления при ошибках), либо постоянную ошибку.

Вторым источником информации является параметр «Диагностическое сообщение», который содержит информацию, более ориентированную на пользователя, с подробными объяснениями и более сложной структурой. Анализ этого параметра не является обязательным для продолжения выполнения протокола.

16.6. Обработка докита и энергонезависимая память

Настоящий стандарт определяет процедуры восстановления при ошибках, которые допускают, чтобы действия по передаче данных были продолжены после возникновения ошибки связи или ошибки системы, которые вызывают разрыв соединения уровня представления и ассоциации прикладного уровня. Однако для того, чтобы восстановление при ошибках имело место, совокупность информации, которая описывает действие и стадию, достигнутую при передаче данных, должна быть сохранена неповрежденной. При ошибках, которые разрушают информацию в системе,

выдается сообщение, кто с кем связан и почему после них не может быть восстановлена работа.

Совокупность информации, которая должна быть сохранена, называется докитом. Определение такого термина не означает, что при реализации необходимо концентрирование информации в одном месте, а просто указывает, что весь набор обладает определенной степенью постоянства.

Тип памяти, используемый для хранения докита, будет зависеть от требований к надежности, установленных пользователем файловой услуги. Для любого типа памяти имеется некоторая небольшая вероятность ошибки, даже если она связана только с физическим повреждением в обеспечиваемом оборудовании. Разработчик распределенного прикладного процесса может выбрать любую технологию энергонезависимой памяти, которая обеспечивает необходимый уровень надежности эксплуатируемых систем.

16.7. Модули восстановления при ошибках

Целью этих процедур, определенных в настоящем стандарте, является обеспечение непрерываемой файловой услуги для пользователей внешней файловой услуги, с исправлением ошибки с помощью обмена между модулями протокола таким образом, чтобы пользователь не осознавал, что имело место что-либо из неординарного.

Ошибки могут разрушить обеспечивающую взаимосвязь или в одной из конечных систем может возникнуть сбой и проведен рестарт; в другом случае сохраняется в исправности ассоциация высокого уровня между взаимодействующими прикладными процессами, потому что оба хранят соответствующие докиты. Протокол обработки ошибок восстанавливает обеспечиваемые соединения и состояние передачи данных, имевшееся до сбоя, базируясь на информации, сохраненной в доките, и затем продолжает взаимосвязь.

Пользователь файловой услуги предоставляет модуль протокола обработки ошибок для файла, информацию, необходимую для выбора модулей, точно описанных посредством параметра «Качество услуги» и с помощью местных способов. Модуль восстановления файла при ошибках, действующий в качестве пользователя внутренней файловой услуги, таким образом может передать пользователю функциональные блоки восстановления или рестарта (или никакие из них) для использования в указанной связи.

ПРИЛОЖЕНИЕ А
Справочное

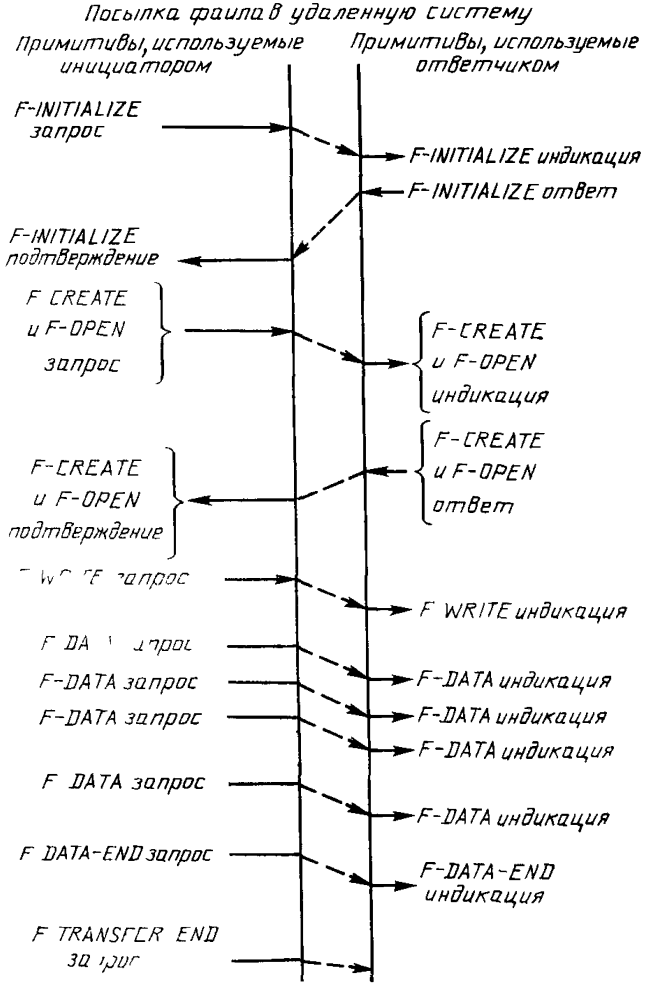
ПРИМЕРЫ ИСПОЛЬЗОВАНИЯ СЛУЖБЫ ПДУФ

А1. Введение

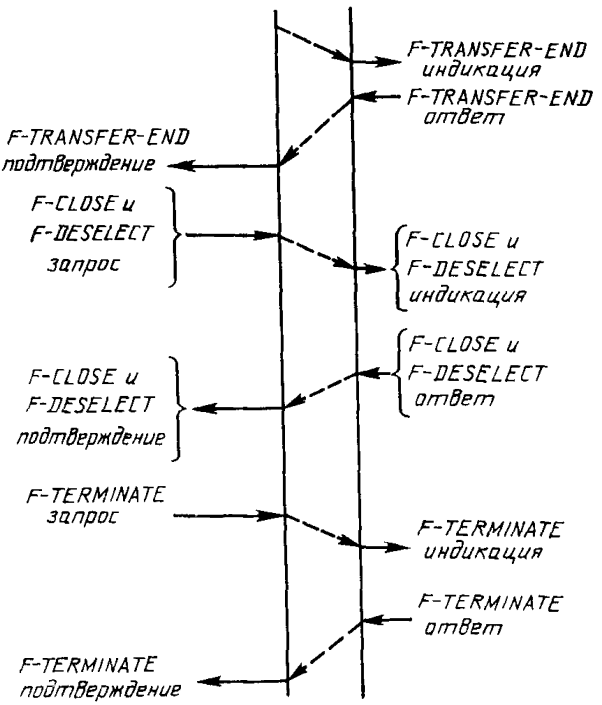
Это приложение предоставляет набор ситуации в которых могла бы использоваться служба ПДУФ (FTAM), и указывает протокольные обмены которые могут иметь место (предполагая, что обмен проходит без ошибок) Область применения службы ПДУФ (FTAM) чрезвычайно широкая и существует значительно большее число вариантов использования чем представлено в приведенном здесь примере

А.2. Передача целого файла в удаленную систему

Предположим, что пользователь в одной системе хочет передать файл коллеге в другую систему через общую сеть передачи данных При этом выполняются следующие шаги (см черт 11)



Черт 11, лист 1



Черт. 11, лист 2

а) пользователь-инициатор запрашивает ассоциацию прикладного уровня. Он предоставляет информацию, необходимую для адресации удаленной системы и для самоидентификации. Реализующая служба выбирает необходимое качество услуг и контексты уровня представления на основе своей системной управляющей информации и выбирает класс передачи функциональными блоками записи и ограниченного управления. Выбор функциональных блоков восстановления при ошибках и рестарта будет зависеть от соотношения стоимость/производительность;

б) система-ответчик выполняет всю необходимую процедуру аутентификации местоположения вызывающей системы, идентификации пользователя-инициатора, предполагаемый учет и разрешает соединение;

в) система-инициатор посылает спецификации файла, который должен быть создан в удаленной системе, и спецификации среды, в которой должна выполняться пересылка данных. Примитивы создания и открытия файла указываются отдельно в настоящем стандарте, чтобы было возможно более сложное использование, но в этом простом примере эти примитивы всегда сцеплены и посылаются как единая операция;

г) система-приемник принимает эту спецификацию;

д) система-инициатор посылает как непрерывный поток пересылок спецификацию операции «запись», которая должна быть выполнена, а также содержание сообщения файла, маркер конца данных и запрос на подтверждение конца передачи;

е) система-ответчик подтверждает получение содержания сообщения файла;

ж) система-инициатор разъединяет соединение с вновь созданным файлом; аналогично здесь указано сцепление отдельно указанных компонентов;

з) система-ответчик подтверждает, что файл был освобожден;

и) система-инициатор запрашивает разъединение ассоциации с удаленной системой;

к) система-ответчик подтверждает, что режим службы ПДУФ (FTAM) освобожден;

л) система-инициатор разрывает обеспечивающие связи и записывает всю необходимую учетную информацию и информацию журнала регистрации.

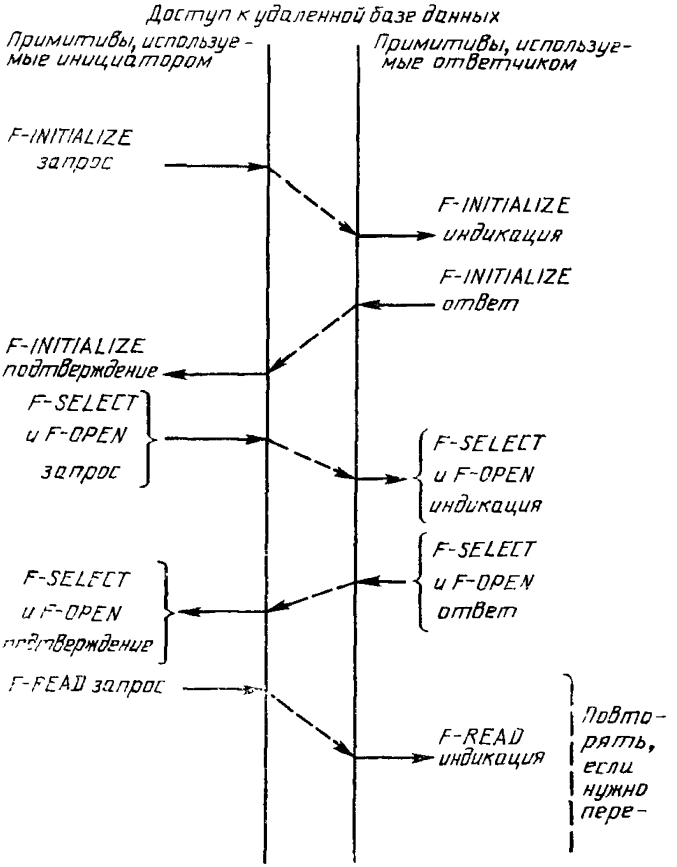
А.3. Доступ к удаленной базе данных

Предположим, что пользователь хочет получить ряд справок из удаленной базы данных. Необходимо выполнить следующие шаги (см. черт. 12):

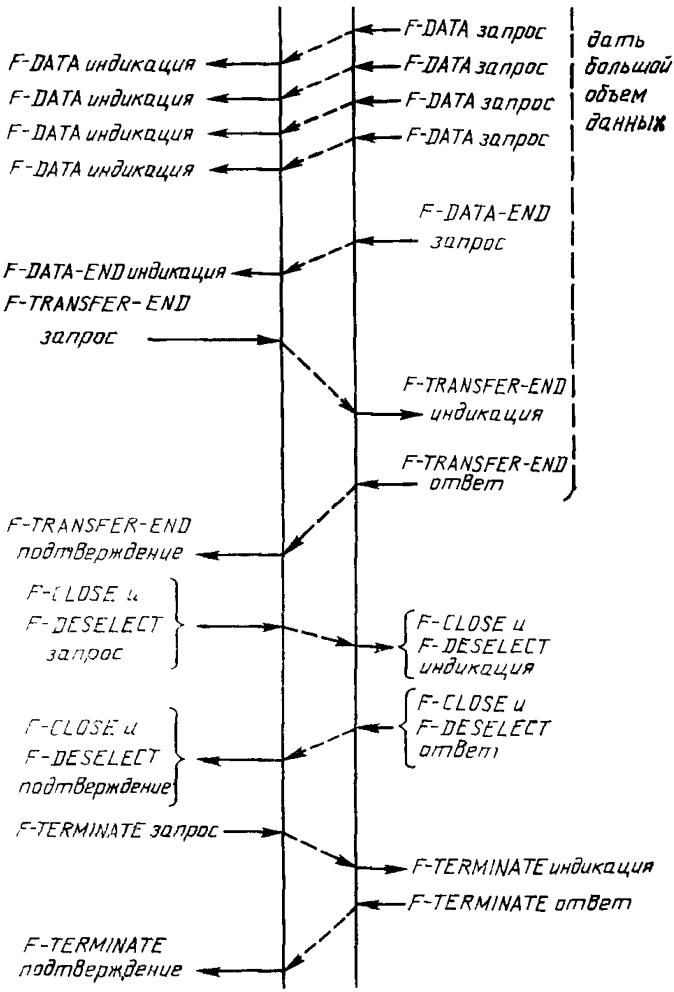
а) инициировать ассоциацию, как в шагах а) и б) разд. А.2, но выбирая только функциональный элемент для чтения;

б) выполнить шаги по инициализации файлов, как в шагах в) и г) разд. А.2. Вместо предоставления спецификации файла пользователя инициирующая система называет поле информации внутри адресуемой базы данных;

в) система-инициатор посылает запрос на чтение данных, используя запрос к базе данных в качестве идентификатора блока данных доступа к файлу в двухмерном файле нотации возможных ответов;



Черт. 12 лист 1



Черт. 12, лист 2

г) система-ответчик возвращает результат запроса в качестве содержания сообщения запрошенного блока данных доступа к файлу;

д) система-инициатор указывает конец транзакции, а система-ответчик подтверждает это;

е) дальнейшие запросы обрабатываются подобным образом в порядке поступления по мере необходимости;

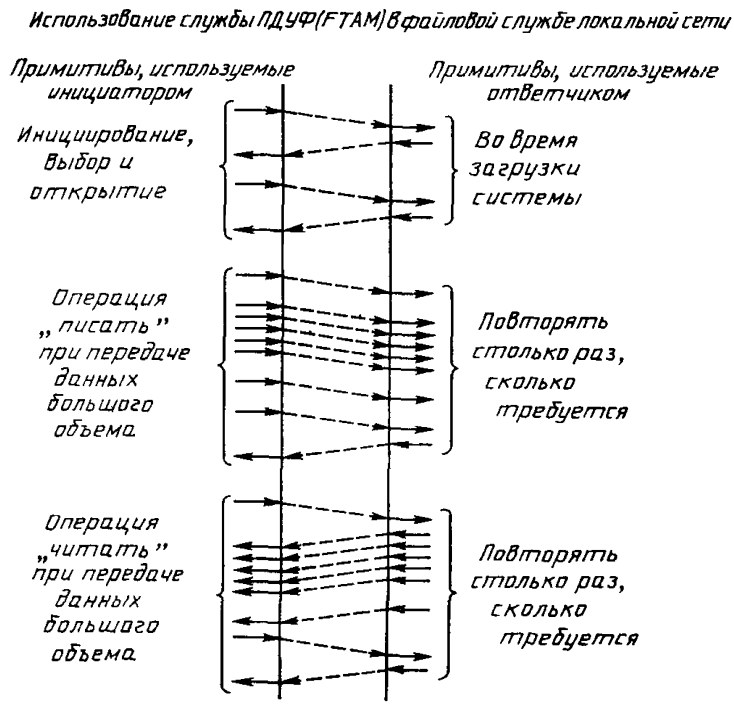
ж) система-инициатор разрывает связь с информационной базой, как указано в шагах ж) и з) разд А.2;

з) ассоциация и обеспечивающие ее ресурсы освобождаются, как описано в шагах от и) до к) разд А.2

А.4. Файловая служба локальной вычислительной сети

Предположим, что малая система без диска, подсоединенная к локальной вычислительной сети в качестве рабочей станции, использует центральную файловую службу для обработки ее рабочих файлов Эти файлы организуются

операционной системой рабочей станции внутри области файлового пространства, рассматриваемой файловой службой как единый неупорядоченный файл прямого доступа Предпринимаются следующие шаги (см. черт. 13):



Черт. 13

а) в процессе инициализации системы, когда загружается ее операционная система, рабочая станция устанавливает соединение с файловой службой и открывает свой рабочий файл, как описано в шагах а) — г) разд. А.2, и в это время выбирая класс доступа файла и функциональный элемент как на чтение, так и на запись;

б) если операционной системе нужно прочитать данные из своего файлового пространства или записать данные в свое файловое пространство, она использует процедуры передачи на чтение или запись данных большого объема, уже описанные в разд. А 3 и А 2. Поскольку средства связи локальной вычислительной сети не подвержены большому числу ошибок, отпадает необходимость в процедуре восстановления при ошибках;

в) последовательность операций чтения и записи продолжается до тех пор, пока работает операционная система Если происходит сбой системы, файловая служба откажется от соединения. Соединение будет устанавливаться заново во время восстановления работоспособности операционной системы рабочей станции Затем во время инициализации система будет заново устанавливать целостность образа диска.

А.5. Активность административного управления файлами

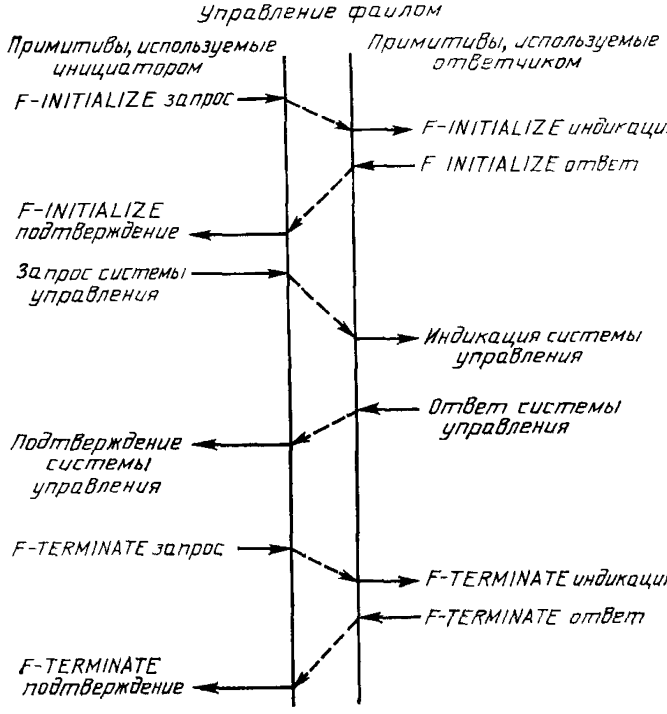
Предположим, системная управляющая программа файловой службы, упоминавшаяся в разд А 4, хочет определить размер рабочего файла для одной из нескольких рабочих станций и изменить счетчик, в котором подсчитываются затраты памяти Выполняются следующие шаги (см черт 14):

а) выполняется инициализация соединения и выбирается класс управления файлом,

б) система управления посылает непрерывным потоком команды для идентификации и выбора файла, запроса значения атрибута файла «Размер файла», установления значения атрибута файла «Учет» и затем отмены режима выбора файла;

в) файловая служба отвечает на каждую из этих команд одним ответом;

г) система управления завершает соединение и выдает результаты в системную управляющую программу



Черт. 14

Примечание На указанном выше чертеже термины «запрос системы управления», «индикация», «ответ» и «подтверждение» используются для представления группы соответствующей последовательности примитивов F-SELECT, F READ-ATTRIBUTE, F-CHANGE-ATTRIBUTE и F-DESELECT

ПРИЛОЖЕНИЕ Б
Справочное

КРАТКОЕ ОПИСАНИЕ ОБЪЕКТОВ ИНФОРМАЦИИ, ОПРЕДЕЛЕННЫХ В СТАНДАРТЕ

Б 1 Введение

Настоящий стандарт устанавливает несколько значений для идентификатора объекта типа АСН 1 Различные определения приведены в настоящем стандарте вместе для упрощения работы с ними Однако в других частях ГОСТ 34 980 могут появиться определения не вошедшие в это приложение

Размещение значений идентификатора обьекта в настоящем стандарте организуется в два этапа То есть сначала компонент идентификатора вводится для некоторого множества объектов а затем вводится дополнительный компонент идентификатора для отдельных объектов внутри этого множества Этот процесс выполняется просто через использование структуры компонентов определенной в ГОСТ 34 973

Б 2 Группы объектов

Группы обьектов определены следующим образом

а) прикладные контексты,
б) абстрактные синтаксисы
в) модели файлов
г) наборы ограничений
д) типы документов

Примечание Правила для создания синтаксиса передачи из определенных абстрактных синтаксисов определяются при помощи идентификатора «Объединение — Международная организация по стандартизации ISO — Международный консультативный комитет по телеграфии и телефонии CCITT», определенного в ГОСТ 34 974

Б 3 Идентификаторы объектов

Специально определенные идентификаторы обьектов даны в следующих подразделах

Б 3 1 Прикладные контексты

ГОСТ 34 980 (ИСО 8571)	прикладной контекст	(1)	iso ftam	(1)

Б 3 2 *Абстрактные синтаксисы*

ГОСТ 34 980 (ИСО 8571)	абстрактный синтаксис	(2)	ftam pci	(1)
ГОСТ 34 980 (ИСО 8571)	абстрактный синтаксис	(2)	ft m-fadu	(2)
ГОСТ 34 980 (ИСО 8571)	абстрактный синтаксис	(2)	неструктурированный текст	(3)
ГОСТ 34 980 (ИСО 8571)	абстрактный синтаксис	(2)	неструктурированный двоичный	(4)

Б 3 3 *Модели файлов*

ГОСТ 34 980 (ИСО 8571)	модель файла	(3)	иерархическая	(1)

Б 3 4 *Наборы ограничений*

ГОСТ 34 980 (ИСО 8571)	набор ограничений	(4)	неструктурированный	(1)
ГОСТ 34 980 (ИСО 8571)	набор ограничений	(4)	последовательный двухмерный	(2)
ГОСТ 34 980 (ИСО 8571)	набор ограничений	(4)	упорядоченный двухмерный	(3)

ГОСТ 34.980 (ИСО 8571)	набор ограничений	(4)	упорядоченный двухмерный с уникальными именами (4)
ГОСТ 34.980 (ИСО 8571)	набор ограничений	(4)	упорядоченный иерархический (5)
ГОСТ 34.980 (ИСО 8571)	набор ограничений	(4)	общий иерархический (6)
ГОСТ 34.980 (ИСО 8571)	набор ограничений	(4)	общий иерархический с уникальними именами (7)

Б.3.5. *Типы документов*

ГОСТ 34.980 (ИСО 8571)	тип документа	(5)	неструктурированный текстовый (1)
ГОСТ 34 980 (ИСО 8571)	тип документа	(5)	последовательный текстовый (2)
ГОСТ 34.980 (ИСО 8571)	тип документа	(5)	неструктурированный двоичный (3)
ГОСТ 34.980 (ИСО 8571)	тип документа	(5)	последовательный двоичный (4)
ГОСТ 34.980 (ИСО 8571)	тип документа	(5)	простой иерархический (5)

ИНФОРМАЦИОННЫЕ ДАННЫЕ

1. ПОДГОТОВЛЕН И ВНЕСЕН Техническим Комитетом ТК 22 «Информационная технология»

2. УТВЕРЖДЕН И ВВЕДЕН В ДЕЙСТВИЕ Постановлением Госстандарта России от 19.08.92 № 990
Настоящий стандарт подготовлен методом прямого применения международного стандарта ИСО 8571/1—88 «Системы обработки информации. Взаимосвязь открытых систем. Передача, доступ и управление файлом. Часть 1. Общее описание» и полностью ему соответствует

3. Срок проверки — 1998 г., периодичность проверок — 5 лет

4. ВВЕДЕН ВПЕРВЫЕ

5. ССЫЛОЧНЫЕ НОРМАТИВНО-ТЕХНИЧЕСКИЕ ДОКУМЕНТЫ

Обозначение отечественного НТД, на который дана ссылка	Обозначение соответствующего международного стандарта	Номер раздела пункта приложения, в котором дана ссылка
ГОСТ 28906—91	ИСО 7498—84	Введение, 3, 7
—	ИСО 8326—87*	10 3
—	ИСО 8509—87*	4
ГОСТ 34 980 2—92	ИСО 8571/2—88	1, 5 12 3, 15 3
ГОСТ 34 980 3—92	ИСО 8571/3—88	1, 5
ГОСТ 34.980 4—92	ИСО 8571/4—88	1, 5, 10 1, 12 3, 16.1
ГОСТ 34 981—91	ИСО 8649—88	10 1
ГОСТ 34 971—91	ИСО 8822—88	10 2
ГОСТ 34 973—91	ИСО 8824—87	13, приложение Б
ГОСТ 34 974—91	ИСО 8825—87	10 2, приложение Б
—	ИСО 9804—90*	9 5
—	ИСО 9834/2—90*	14

* До прямого применения данного документа в качестве государственного стандарта распространение его осуществляет секретариат ТК 22 «Информационная технология».

СОДЕРЖАНИЕ

Редактор *В. М Лысенкина*
Технический редактор *В. Н. Малькова*
Корректор *А. И. Зюбан*

Сдано в наб 23 09 92 Подп. к печ. 28.12 92 Усл п л 3,25 Усл. кр -отт. 3,36. Уч.-изд. л. 3,60. Тираж 423 экз

Ордена «Знак Почета» Издательство стандартов, 123557, Москва, ГСП, Новопресненский пер., 3.
Калужская типография стандартов, ул. Московская, 256. Зак 2103