

РОССИЙСКАЯ АКАДЕМИЯ НАУК  
ИНСТИТУТ КОСМИЧЕСКИХ ИССЛЕДОВАНИЙ

Пр-1971

М.Н.Боярский, П.В.Житников, М.И.Шевченко

ИНФОРМАЦИОННАЯ СИСТЕМА ДОСТУПА К АРХИВНЫМ  
ДАНЫМ НАУЧНОГО КОСМИЧЕСКОГО ПРОЕКТА АРКАД

Москва  
1997

***М.Н.Боярский, П.В.Житников, М.И.Шевченко Информационная система доступа к архивным данным научного космического проекта АРКАД.***

***M.N.Boyarsky, P.V.Zhitnikov, M.I.Shevchenko. Information System for the Access to the Arcad Space Science Mission Data Archive.***

***В работе описывается информационная система (ИС) научного космического проекта АРКАД. Рассматривается реализация информационной модели системы с использованием СУБД POSTGRES95. Приводится описание программного комплекса, реализующего функциональную модель ИС АРКАД и руководство по использованию WWW-интерфейса пользователя архива данных.***

***The paper describes the Information System of the Arcad Space Science Mission. Realization of the information model of the system using DBMS POSTGRES95 is discussed. The paper also contains the description of the software package created for the Arcad Information System and the data archive WWW-interface user guide.***



## Введение

### Научный космический проект АРКАД

Советско-французский космический проект АРКАД был осуществлен в период с 1981 по 1987 г. и имел целью проведение широких комплексных исследований физических процессов в магнитосфере, в полярной верхней атмосфере и ионосфере и изучение магнитосферно-ионосферных связей [1].

Среди основных направлений научных исследований проекта АРКАД можно назвать:

- исследование дуг и лучей полярных сияний с высоким временным и пространственным разрешением;
- исследование тепловой и горячей плазмы магнитосферы и ионосферы и ее взаимодействия с полярной верхней атмосферой;
- исследования продольных движений частиц в полярной ионосфере и выноса ионов в магнитосферу, то есть изучение явления “полярного ветра” и сверх тепловых пучков ионов;
- исследования взаимодействия магнитосферной плазмы с электромагнитным излучением мощных передатчиков.

Передача информации со спутника осуществлялась по двум телеметрическим системам: по штатной советской телеметрической системе служебного комплекса (ТМС) и дополнительной французской широкополосной телеметрии (ТМФ), входившей в состав научного комплекса [2].

### Необходимость реорганизации информационной системы (ИС) проекта АРКАД

Вся информация, которая передавалась в реальном времени со спутника, была сохранена в виде специальных файлов на 9 - дорожечных 12,7 мм магнитных лентах формата ЕС ЭВМ. Эти магнитные ленты имеют относительно небольшую емкость (до 170 Мбайт, а большинство 20 Мбайт), поэтому количество магнитных лент составило ~1,5 тысячи. Так как с начала проведения проекта прошло достаточно много времени, то возникла необходимость переноса данных проекта со старых 9 - дорожечных магнитных лент на новые современные носители для сохранения информации, уменьшения физического объема, занимаемого архивом, и обеспечения возможности ее будущего использования с применением современных компьютеров и средств распределенного доступа. Было решено в процессе этого переноса создать информационную систему (ИС) архива данных проекта. Рис. 1 наглядно демонстрирует изменения, произошедшие с архивом данных проекта АРКАД в результате реорганизации.

Компонент ИС	Прежнее решение	Новое решение
Архивный носитель	9 - дорожечные ленты	8мм и 4мм ленты
Емкость носителя	20,40,170 Мбайт	5Гбайт и 4Гбайт
Число файлов на носителе	5-15	400-500

<b>Архитектура ИС</b>	<b>Однозвенная (мэйнфрейм)</b>	<b>Трехзвенная клиент-сервер</b>
<b>Рабочее место пользователя</b>	<b>Алфавитно-цифровой терминал</b>	<b>ПК или рабочая станция</b>
<b>Интерфейс пользователя</b>	<b>Отсутствовал</b>	<b>Графический (Web-браузер)</b>
<b>Поиск информации</b>	<b>Ручной по рукописному журналу</b>	<b>Автоматический поиск в базе данных</b>
<b>Способ переноса данных из архива на магнитный диск</b>	<b>Ручной (пользователь просил оператора поставить ленту)</b>	<b>Автоматический (jukebox) или полуавтоматический</b>
<b>Внешний доступ к информации</b>	<b>Отсутствовал</b>	<b>Локальная сеть и Интернет</b>

**Рис.1. Реорганизация информационной системы проекта АРКАД**

На данный момент существуют новые устройства записи данных на магнитные ленты формата 4мм (DAT) и 8мм (Exabyte). Эти ленты, обладая сравнительно невысокой стоимостью, позволяют хранить несколько гигабайт информации (от 2 Гбайт без аппаратного сжатия). Именно они и были выбраны в качестве основных носителей информации, полученной в ходе экспериментов проекта с АРКАД.

При переносе файлов на магнитные ленты нового образца встал вопрос об организации доступа к этим данным. Так как перенос осуществлялся постепенно, то поначалу устраивала следующая схема функционирования. Пользователь, которому в ходе своих исследований были необходимы данные, присылал по электронной почте запрос, содержащий информацию об интересующих его файлах. Этот запрос обрабатывался вручную, после чего стандартными утилитами ОС UNIX (grep, awk, sort и shell-скрипты) осуществлялось сканирование log-файлов, выбор и сортировка номеров лент и порядковых номеров записей на этих лентах, определяющих физическое расположение файлов. После этого нужные ленты устанавливались в считывающее устройство, и осуществлялся перенос выбранных файлов в определенную область жесткого диска.

Но очень скоро встал вопрос об улучшении интерфейса пользователя и оптимизации процесса поиска для ускорения выбора файлов. Спроектированная авторами ИС проекта АРКАД обеспечивает быстрый поиск необходимых файлов и формирование заказа в интерактивном режиме, а также автоматическую обработку заказов.

При разработке ИС проекта АРКАД учитывались следующие общие соображения. Прежде всего, как и в большинстве современных ИС, была принята трехзвенная архитектура клиент-сервер[2]:

- графический интерфейс пользователя
- сервер приложений
- система управления базами данных [3,4] (СУБД), которая обеспечивает оперативный поиск и доступ к информации.

В качестве программы, обеспечивающей графический интерфейс пользователя данной ИС, было решено использовать Web-браузер. Данное решение позволило упростить разработку интерфейса пользователя и дало

возможность отказаться от создания вариантов программы для различных платформ и распространения своих программ среди пользователей. Использование Web-браузера также автоматически обеспечивает возможность работы с ИС с любого удаленного компьютера, подключенного к сети Internet.

Таким образом, основой ИС проекта АРКАД должна была стать СУБД, предоставляющая возможность создания собственного WWW-интерфейса.

## 1. Технические аспекты реализации ИС проекта АРКАД

### 1.1 Обоснование использования трехзвенной архитектуры “клиент-сервер”

Для построения ИС проекта АРКАД была использована трехзвенная архитектура “клиент-сервер” (рис.2). Процесс, выполняемый на компьютере-клиенте (“Компонент представления”), отвечает за интерфейс с пользователем, т.е. это графический интерфейс пользователя. Прикладной компонент реализован как отдельный “сервер приложения”. Доступ к ресурсам осуществляет обычно сервер базы данных. В данной модели [1] прикладной компонент слабо зависит от сервера базы данных. При большом числе пользователей эта модель позволяет добиться существенной экономии ресурсов. Дело в том, что стоимость программ серверов баз данных зависит от числа поддерживаемых пользователей. В данном случае все пользователи подключены к серверу приложения, который является единственным клиентом сервера базы данных.

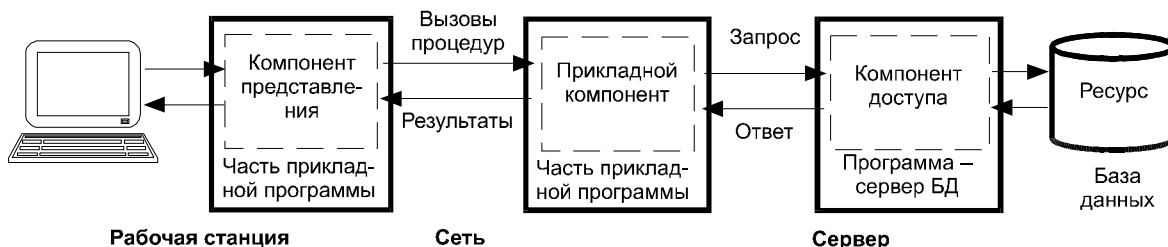


Рис.2. Трехзвенная архитектура “клиент-сервер”.

В ИС проекта АРКАД роль “Компонента представления” играет браузер пользователя, роль “Прикладного компонента” — Web - сервер и CGI - программы, а роль “Компонента доступа” — сервер базы данных.

### 1.2 Выбор Web – сервера

Web - сервер – это программное обеспечение, отвечающее за прием запросов браузеров, поиск указанных файлов и возвращение их содержимого (или выполнение указанных CGI - сценариев и возвращение результатов их выполнения) [6]. Сейчас большинство Web - серверов в Internet работают на UNIX - машинах.

Для использования в ИС проекта АРКАД был выбран Web - сервер Apache в силу того, что Apache считается наиболее эффективным http-сервером из свободно распространяемых.

### 1.3 Выбор СУБД

Как уже было отмечено, для оптимального функционирования ИС АРКАД необходимо использовать систему управления базами данных [4]. Она должна быть размещена на сетевом сервере для того, чтобы пользователи ИС могли работать с общей БД.

Выбор СУБД для ИС проекта АРКАД изначально был ограничен. Желательно было использовать свободно распространяемую в исходных кодах программу, к которой предъявлялись следующие основные требования:

- работа в архитектуре клиент/сервер [1];
- языком запросов должен быть SQL для возможности в будущем перехода на другую СУБД с сохранением разработанного кода.

По сути, выбор делался из двух свободно распространяемых СУБД – miniSQL и POSTGRES95. В ИС проекта АРКАД была использована СУБД POSTGRES95.

Исходный код, руководство по применению и описание POSTGRES95 доступны по следующим Internet адресам:

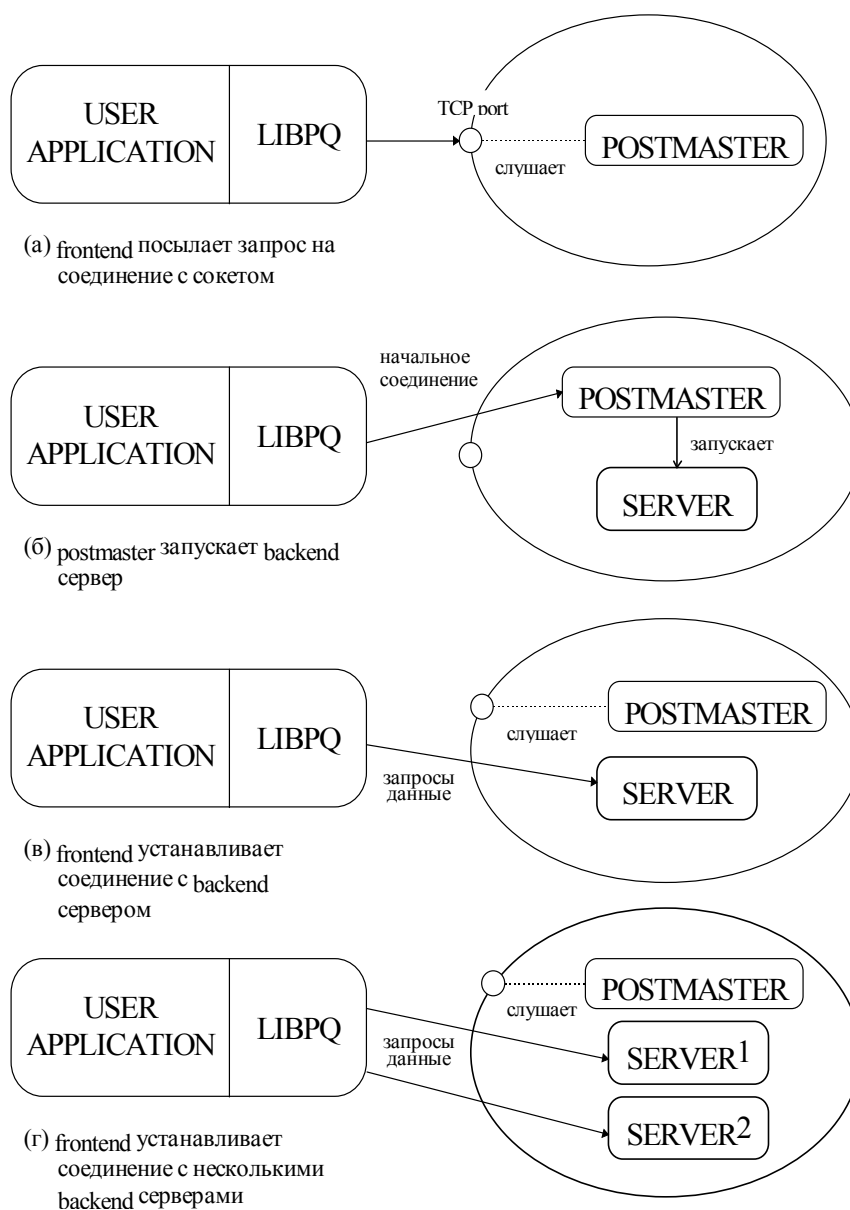
- <ftp://ftp.vnet.net:/pub/postgres95/>
- <http://www.eol.ists.ca:80/~dunlop/postgres95-manual/>
- <http://epoch.CS.Berkeley.EDU:8000/postgres95/index.html>

### 1.4 Основные концепции архитектуры POSTGRES95

POSTGRES95 использует так называемую “process-per-user” модель клиент/сервер. Сеанс работы POSTGRES95 состоит из следующих UNIX процессов:

- Управляющий демон *postmaster*. Он управляет связью между *frontend* и *backend* процессами, а также использованием буфера в разделяемой памяти и выполнением других инициализаций в момент запуска.
- Пользовательское *frontend* приложение (например, *psql*). Запрашивает через *postmaster* соединение с *postgres backend* сервером. Может работать на другом компьютере (клиентской рабочей станции).
- Один или больше *postgres backend* серверов базы данных. Этот процесс и выполняет реальную работу (т.е. обрабатывает запросы). *Postmaster* запускает новый *backend* процесс для каждого *frontend* подсоединения. (*postgres backend* всегда выполняется на сервере).

*Postmaster управляет существующим набором баз данных на одном компьютере. Такой набор баз данных называется “установкой”. Frontend приложение, желающее получить доступ к базе данных, входящей в “установку”, обращается к LIBPQ-библиотеке, LIBPQ посылает по сети запрос на соединение демону postmaster (рис.3а), который в свою очередь запускает новый backend сервер (рис.3б) и устанавливает соединение между вновь созданным backend сервером и frontend процессом (рис.3в). С этого момента frontend процесс и backend сервер взаимодействуют без участия postmaster. Таким образом, postmaster постоянно выполняется, ожидая запросов на соединение, в то время как frontend и backend процессы запускаются и завершаются. LIBPQ-библиотека позволяет одному frontend процессу устанавливать множественные соединения с backend процессами (рис.3г). Однако frontend приложение может быть только однопоточным процессом. Многопоточные frontend/backend соединения не поддерживаются LIBPQ. Одной из особенностей этой архитектуры является то, что postmaster и backend процессы всегда выполняются на одной машине, в то*



*время как frontend приложение может выполняться где угодно.*

**Рис.3. Установление соединения с сервером POSTGRES95.**



## 1.5 Программные интерфейсы POSTGRES95

### 1.5.1 Perl API-интерфейс (pg95perl)

*Pg95perl* является интерфейсом между языком perl версии 5 и СУБД POSTGRES95 (текущая версия *pg95perl5.1.2.0* доступна по <http://www.eol.ists.ca/~dunlop/wdb-p95/>). Все функции *pg95perl5* соответствуют функциям LIBPQ-интерфейса к Postgres95.

### 1.5.2 WWW-интерфейс Postregs95 wdb-p95

Для работы с *wdb-p95* (<http://www.eol.ists.ca/~dunlop/wdb-p95/>) необходимо следующее программное обеспечение:

- Perl (версия не ниже 5.0.0.1);
- *pgperl5 v1.2.0*;
- HTTP сервер.

При установке WDB-P95 нужно учитывать, что Postgres95 сервер, *pgperl* и *httpd* совсем не обязательно должны быть установлены на одной машине. Так как WDB-P95 вызывается *http* демоном, они должны быть на одном хосте. Кроме того, так как WDB-P95 использует в своей работе *Pg.pm – pgperl*, то этот программный пакет должен быть на том же хосте. *Pgperl* в своей работе использует LIBPQ-библиотеку, для того чтобы иметь доступ к любому Postgres95-серверу где угодно в сети, как любой другой Postgres95-клиент. Архитектура иллюстрирована следующим рисунком (рис.4).

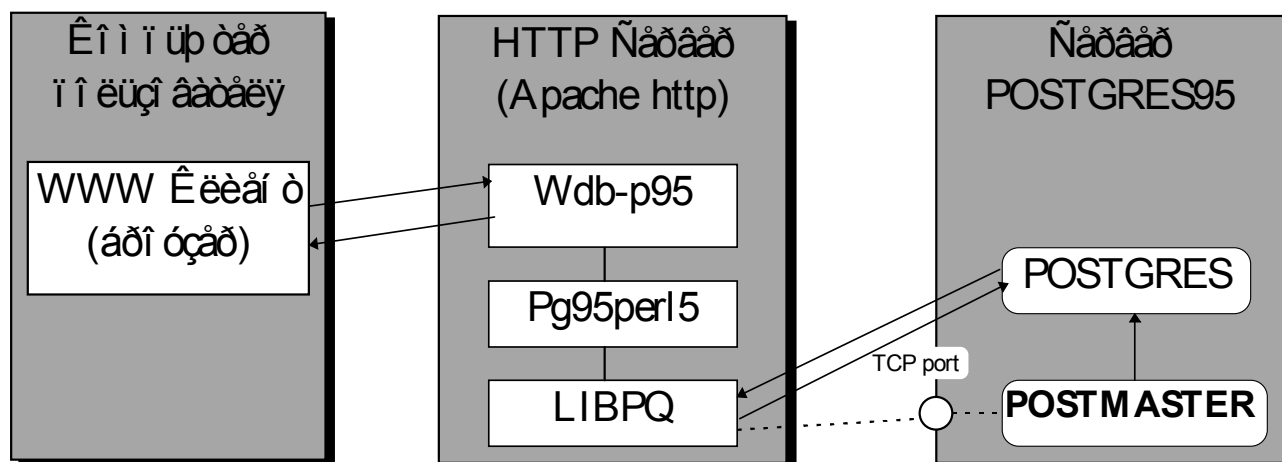


Рис.4. Схема взаимодействия Wdb-p95, браузера и сервера POSTGRES95.

## 2. Реализация информационной модели системы с использованием СУБД POSTGRES95

### 2.1 Структура таблиц, используемых в системе

Для реализации этого проекта, описанного моделью “сущность-связь” [5], в СУБД POSTGRES95 был предложен следующий набор таблиц:

- Таблица “Паспорт файла” (рис.5). По ней пользователь будет производить поиск необходимых файлов.
- Таблица “Архивный файл” (рис.6). Из нее программа управления переносом файлов с магнитных лент в FTP-доступную область берет данные о физическом расположении на магнитной ленте файла, идентификационный номер которого определяется с помощью таблицы “Паспорт файла”.
- Таблица “Доступный файл” (рис.7). В ней фиксируется заказ пользователя, а также содержится информация о файлах, уже находящихся в FTP-доступной области.
- Таблица “Пользователь” (рис.8). Содержит информацию о пользователе: имя пользователя, адрес его электронной почты и время посещения основной WWW - страницы проекта АРКАД.

<i>Атрибуты</i>	<i>Alias</i>	<i>Тип данных</i>	<i>Ключ</i>	<i>Комментарии</i>
<i>ID паспорта</i>	<i>oid</i>	<i>oid</i>	<i>PK</i>	<i>Идентификатор</i>
<i>ID файла</i>	<i>File_ID</i>	<i>oid</i>	<i>FK</i>	<i>Ссылка на oid таблицы “Архивный файл”</i>
<i>№ сеанса</i>	<i>Nsess</i>	<i>int</i>	<i>AK</i>	<i>Номер сеанса</i>
<i>Версия</i>	<i>Vers</i>	<i>char</i>	<i>AK</i>	<i>Версия записи</i>
<i>Формат</i>	<i>Form</i>	<i>char4</i>	<i>AK</i>	<i>Формат данных</i>
<i>Размер</i>	<i>Size</i>	<i>int</i>	—	<i>Размер файла в байтах</i>
<i>Дата начала сеанса</i>	<i>BegDate</i>	<i>date</i>	—	<i>Дата начала сеанса связи</i>
<i>Время начала сеанса</i>	<i>BegTime</i>	<i>time</i>	—	<i>Время начала сеанса связи</i>
<i>Момент начала сеанса</i>	<i>Begmsec</i>	<i>int</i>	—	<i>Момент начала сеанса в микросекундах</i>
<i>Дата окончания сеанса</i>	<i>EndDate</i>	<i>date</i>	—	<i>Дата окончания сеанса связи</i>
<i>Время окончания сеанса</i>	<i>EndTime</i>	<i>time</i>	—	<i>Время окончания сеанса связи</i>
<i>Момент окончания сеанса</i>	<i>Endmsec</i>	<i>int</i>	—	<i>Момент окончания сеанса в микросекундах</i>

**Рис.5. Таблица “Паспорт файла”.**

<i>Атрибуты</i>	<i>Alias</i>	<i>Тип данных</i>	<i>Ключ</i>	<i>Комментарии</i>
<i>ID файла</i>	<i>oid</i>	<i>oid</i>	<i>PK</i>	<i>Идентификатор</i>
<i>№ сеанса</i>	<i>Nsess</i>	<i>int</i>	<i>AK</i>	<i>Номер сеанса</i>
<i>Версия</i>	<i>Vers</i>	<i>char</i>	<i>AK</i>	<i>Версия записи</i>
<i>Формат</i>	<i>Form</i>	<i>char4</i>	<i>AK</i>	<i>Формат данных</i>

<b>Размер</b>	<b>Size</b>	<b>int</b>	—	<b>Размер файла в байтах</b>
<b>Номер ленты в формате DAT</b>	<b>DatLen t</b>	<b>int</b>	—	<b>Номер ленты в DAT формате, на которой записан файл</b>
<b>Номер записи на DAT ленте</b>	<b>DatFile</b>	<b>int</b>	—	<b>Порядковый номер метки на DAT ленте, после которой записан файл</b>
<b>Номер ленты в формате EXABYTE</b>	<b>ExaLen t</b>	<b>int</b>	—	<b>Номер ленты в EXABYTE формате, на которой записан файл</b>
<b>Номер записи на EXABYTE ленте</b>	<b>ExaFile</b>	<b>int</b>	—	<b>Порядковый номер метки на EXABYTE ленте, после которой записан файл</b>
<b>Время создания файла</b>	<b>CreDate</b>	<b>date</b>	—	<b>Время записи файла на ленту</b>

**Рис.6. Таблица “Архивный файл”.**

<b>Атрибуты</b>	<b>Alias</b>	<b>Тип данных</b>	<b>Ключ</b>	<b>Комментарии</b>
<b>ID записи</b>	<b>oid</b>	<b>oid</b>	<b>PK</b>	<b>Идентификатор</b>
<b>ID файла</b>	<b>File_ID</b>	<b>oid</b>	<b>FK</b>	<b>Ссылка на oid таблицы “Архивный файл”</b>
<b>№ сеанса</b>	<b>Nsess</b>	<b>int</b>	<b>AK</b>	<b>Номер сеанса</b>
<b>Версия</b>	<b>Vers</b>	<b>char</b>	<b>AK</b>	<b>Версия записи</b>
<b>Формат</b>	<b>Form</b>	<b>char4</b>	<b>AK</b>	<b>Формат данных</b>
<b>Размер</b>	<b>Size</b>	<b>int</b>	—	<b>Размер файла в байтах</b>
<b>Дата переноса</b>	<b>CreDate</b>	<b>date</b>	—	<b>Дата переноса файла в FTP-доступную область</b>
<b>Флаг выбора</b>	<b>flag_sel</b>	<b>boolean</b>	—	<b>Устанавливается в true, если файл заказывается</b>
<b>Флаг доступности</b>	<b>flag_ftp</b>	<b>boolean</b>	—	<b>Устанавливается в true, если файл доступен</b>
<b>ID пользователя</b>	<b>User_ID</b>	<b>oid</b>	<b>FK</b>	<b>Ссылка на oid таблицы “Пользователь”</b>
<b>Имя файла</b>	<b>FileName</b>	<b>text</b>	—	<b>Имя файла в FTP- доступной области</b>

**Рис.7. Таблица “Доступный файл”.**

<b>Атрибуты</b>	<b>Alias</b>	<b>Тип данных</b>	<b>Ключ</b>	<b>Комментарии</b>
<b>ID пользователя</b>	<b>oid</b>	<b>oid</b>	<b>PK</b>	<b>Идентификатор</b>
<b>Имя пользователя</b>	<b>Name</b>	<b>text</b>	<b>AK</b>	<b>Для составления письма</b>

<i>E-mail</i> пользователя	<i>Emailto</i>	<i>text</i>	—	Для отправки письма
Дата посещения	<i>Date</i>	<i>date</i>	—	Для статистики посещений
Время посещения	<i>Time</i>	<i>time</i>	—	Для статистики посещений

Рис.8. Таблица “Пользователь”.

## 2.2 Заполнение таблиц, используемых в системе

Процессу заполнения базы данных проекта АРКАД предшествовала работа по проверке соответствия между паспортами файлов и их названиями. Для этого были написаны несколько программ на языке Си для считывания паспортов файлов во всех форматах. В одном из форматов данные содержались в форме чисел с плавающей точкой для IBM 360, так что необходимо было написать программу, переводящую числа с плавающей точкой из стандарта IBM 360 в современный стандарт. Кроме того, была проведена работа по переносу файлов с переполненного к тому времени жесткого диска на магнитные ленты.

После всех этих предварительных процедур началась работа по непосредственному заполнению базы данных данными из паспортов файлов, log-файлов, содержащих информацию о номерах магнитных лент и порядковых номерах файлов на этих лентах, и из других источников. Для этого были написаны многочисленные короткие сценарии на C-Shell и Perl.

Порядок заполнения был установлен следующий:

- сначала заполняется таблица “Архивный файл”;
- после этого таблицы “Паспорт файла” и “Доступный файл”, причем значение атрибута “File\_ID” устанавливается в соответствии с таблицей “Архивный файл”, а флаги в таблице “Доступный файл” устанавливаются в FALSE.

Таким образом, была создана связанная система таблиц, две из которых имеют внешний ключ на третью таблицу.

## 3. Программный комплекс, реализующий функциональную модель ИС проекта АРКАД

### 3.1 Состав комплекса

Программный комплекс, реализующий функциональную модель ИС проекта АРКАД, состоит из двух программ, использующих для взаимодействия с пользователем и оператором системы WWW - интерфейс и реализующих, соответственно, “ПРОЦЕСС ЗАКАЗА” и “ПРОЦЕСС ОБРАБОТКИ ЗАКАЗА” функциональной модели ИС проекта АРКАД.

Обе программы используют для взаимодействия с базой данных совокупность процедур *pgperl*, описанную в предыдущей главе. Но при их

разработке были учтены следующие факторы, что привело к различию принципов построения этих программ.

*WWW* - интерфейс пользователя рассчитан на широкий круг пользователей, а, значит, должен быть неприхотлив к программному обеспечению пользователя, поэтому этот программный продукт разрабатывался в расчете на любой браузер, поддерживающий HTML 2.0.

Что касается *WWW* - интерфейса оператора, то он рассчитан на конкретное программное обеспечение, установленное на компьютерах сотрудников Архива данных ИКИ. Для удобства оператора и для обеспечения интерактивного взаимодействия с базой данных помимо обычного программирования с использованием CGI - интерфейса [8] использовался язык программирования JavaScript, что позволило уменьшить сетевой трафик между сервером и клиентом и ускорить работу с интерфейсом.

### **3.2 Программа, реализующая “ПРОЦЕСС ЗАКАЗА” функциональной модели ИС проекта АРКАД и использующая для взаимодействия с пользователем системы WWW - интерфейс**

#### **3.2.1 Общие сведения и функциональное назначение**

Данная программа представляет собой CGI- сценарий, написанный на языке Perl 5.0 с использованием *rpmperl* для взаимодействия с базой данных. За его основу был взят Web - интерфейс к базам данных *POSTGRES95*, *wdb-p95*, который позволяет только осуществлять выбор необходимых элементов базы данных по запросу, составленному пользователем. Созданная программа позволяет после выбора необходимых элементов базы данных формировать в интерактивном режиме и отправлять заказы серверу базы данных, который в свою очередь фиксирует эти заказы в специально созданной для этого таблице. Программа *wdb\_p95* универсальна в том смысле, что она позволяет организовать доступ к данным любой таблицы и фиксировать заказы в другой специализированной таблице. Вся необходимая информация об определенных таблицах содержится в конфигурационном файле *p95-dbi.conf* и в так называемых *Form Definition Files* (файлы с расширением *.fdf*), имена которых передаются программе через URL.

#### **3.2.2 Описание логической структуры**

Программа *wdb\_p95* состоит из следующих основных подпрограмм:

Подпрограмма *ParseFile()*. Считывает содержимое *FDF* файла, интерпретирует данные, в нем содержащиеся. Переводит эти данные в форму, удобную для дальнейшего использования в программе (в ассоциативный массив *%FORM*).

Подпрограмма *PrepareParams()*. Считывает данные из CGI-переменной *QUERY\_STRING* и из стандартного входа *STDIN*. Интерпретирует данные и переводит их в форму, удобную для дальнейшего использования в программе (в ассоциативный массив *%IN*).

Подпрограмма *PrintForm()*. В соответствии с *FDF* файлом генерирует HTML форму для составления запроса к базе данных.

**Подпрограмма DoQuery().** Преобразовывает данные, полученные из формы в SQL-запрос к базе данных. Устанавливает соединение с сервером Postgres95 с помощью вызова процедуры `dbi_connect()`. Посылает SQL-запрос по установленному соединению (`dbi_dosql()`). Возвращает результат запроса построчно посредством процедуры `dbi_nextrow()`. Если `ACTION=select`, вызывает подпрограмму `DoSelection()`. Для вывода результатов запроса в окно браузера вызывает процедуру `PrintTabHead()`, а затем последовательно для каждой строки, возвращенной в виде ассоциативного массива, вызывает процедуру `PrintTabVars()`. Закрывает соединение с сервером Postgres95 вызовом процедуры `dbi_disconnect()`.

**Подпрограмма DoSelection().** По состоянию `CheckButtons`, сообщенному из формы, определяет выбранные файлы, и фиксирует заказ в специальной таблице, имя которой находится в переменной `$FORM{STATUS}`. Фиксация заказа заключается в установлении в `TRUE` значения атрибута “флаг выбора” и в присваивании значению атрибута “`User_ID`” идентификатора пользователя, осуществившего заказ. Причем если заказываемый файл уже доступен, то “флаг выбора” не устанавливается в `TRUE`, а только “`User_ID`” присваивается значению идентификатора пользователя, и дата и время переноса файла устанавливаются в текущие значения. Изменение таблицы производится вызовом процедуры `dbi_update()`.

**Подпрограмма DoSummary().** Устанавливает соединение с сервером Postgres95 с помощью вызова процедуры `dbi_connect()`. Выбирает посредством вызова процедуры `dbi_dosql()` из таблицы, имя которой в переменной `$FORM{STATUS}`, файлы со статусом “доступен” среди файлов, заказанных пользователем в текущем сеансе. Завершает текущий сеанс работы с базой данных. Закрывает соединение с сервером Postgres95 вызовом процедуры `dbi_disconnect()`.

**Подпрограмма PrintTabHead().** Выводит на экран HTML заголовок страницы для результатов запроса. Генерирует в соответствии с FDF файлом заголовок таблицы для вывода данных, являющихся результатами запроса.

**Подпрограмма PrintTabVars().** Выводит на экран строку HTML-таблицы, являющуюся результатом запроса, причем в первой ячейке строки выводится элемент формы `CheckButton`.

### 3.2.3 Структура программы с описанием функций составных частей и связи между ними

Структуру программы наглядно демонстрирует блок-схема на рис.9:

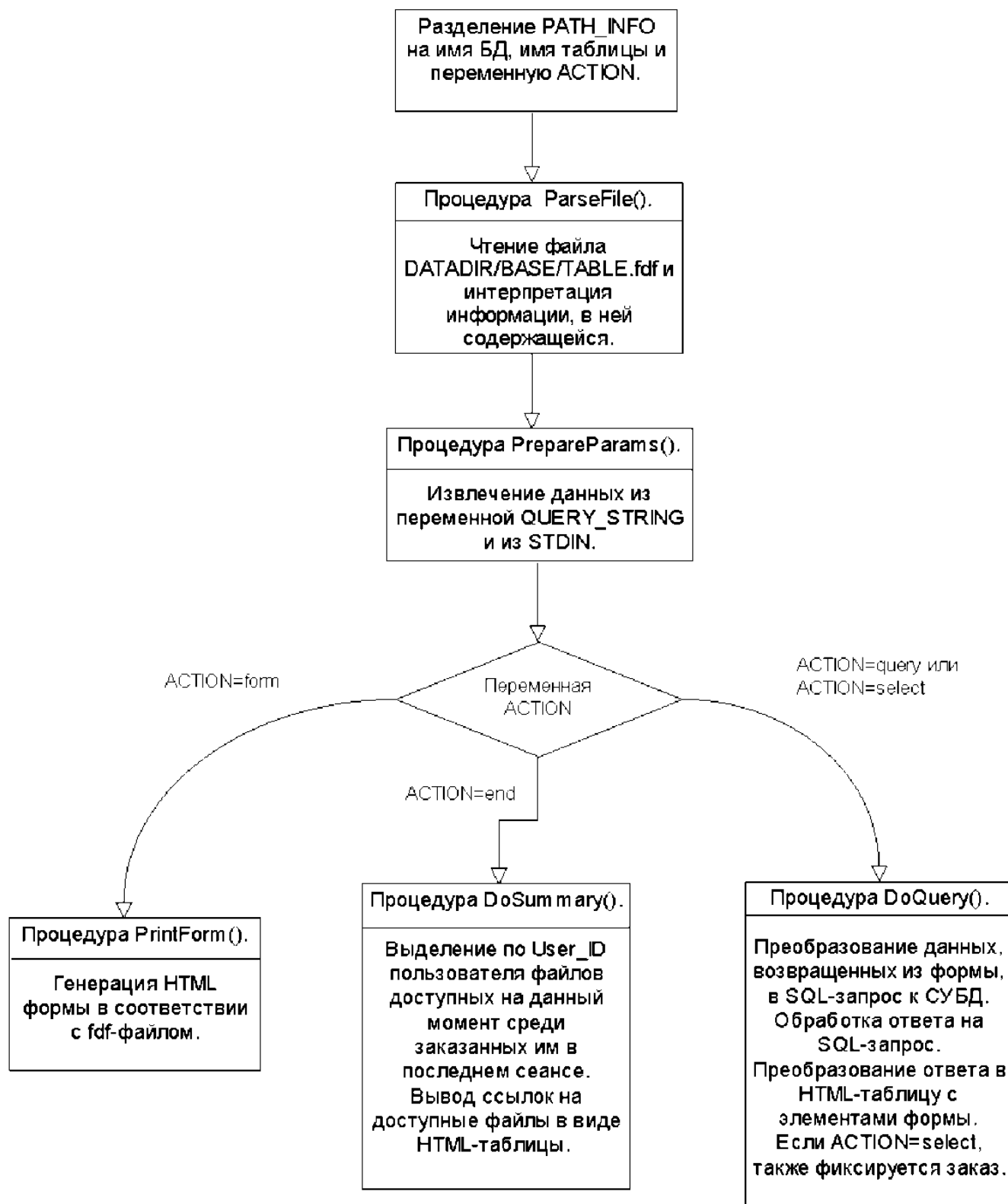


Рис.9. Блок-схема программы Wdb-p95.

### 3.2.4 Связи программы с другими программами

В своей работе программа `wdb-p95` использует Perl интерфейс к `Postgres95`, так называемый `pg95perl5`, для этого в начале программы подгружается `Pg.pm`. Основные процедуры для взаимодействия с `Postgres95` находятся в файле `pg95_dbi.pl`, поэтому он тоже подгружается к основной программе. Кроме того, для упрощения выполнения базовых задач CGI программирования используется модуль для языка `Perl5 CGI.pm` [11].

### 3.2.5 Вызов и загрузка

Для вызова программы необходимо из любого браузера, поддерживающего HTML 2.0 вызвать URL `http://arc.iki.rssi.ru:8001/arcad.html` и выбрать ссылку на интересующий проект.

### 3.2.6 Входные данные

Основная часть информации, необходимой программам CGI, передается им через переменные среды ОС UNIX. Программы могут обрабатывать эту информацию так же, как и значение любой переменной среды (например, посредством ассоциативного массива `%ENV` в Perl).

Так, для данной программы посредством переменной `PATH_TRANSLATED` (соответствует URL) передается имя FDF файла, содержащего всю необходимую информацию о таблице, к которой предоставляется доступ. Формат URL:

`http://arc.iki.rssi.ru:8001/cgi-bin/wdb-p95/DBname/TABLEname/ACTION`

Переменная `QUERY_STRING` служит для передачи данных, внесенных в форму, методом `GET`. Переменная `CONTENT_LENGTH` содержит число байтов, которое необходимо считать из стандартного входа программы `STDIN`, для того чтобы считать данные переданные CGI-сценарию, методом `POST`.

Содержащиеся в FDF файле данные являются входными данными программы, так что опишем формат `Form Definition File`. FDF файл является текстовым файлом со следующими полями:

```
NAME    = tms      # Имя формы для использования в URL
HOST    = stda.iki.rssi.ru # Хост, на котором установлена Postgres95
PORT    = 5432     # Номер порта, на котором выполняется postgres
DATABASE = arcad   # Имя базы данных для запросов
TABLE   = tms     # Имя таблицы для запросов
TITLE   = TMS     # HTML Title для формы и результатов
Q_HEADER = TMS Query Form # Query FORM <H1> tag
R_HEADER = TMS Query Result # Results output <H1> tag
Q_HTML  =         # Дополнительный текст для вывода формы
R_HTML  =         # Дополнительный текст для вывода результатов
SQL     = on     # Включает вывод SQL запроса на экран
STATUS  = tms_ftp # Имя таблицы для фиксации заказа
DOWNLOAD =       # Имя директории, в которую переносятся файлы
#-----
# FIELD (R) Название атрибута
# label  Название для HTML форм и таблиц
# column Колонка базы данных
# type   (R) Тип данных этого поля
```



```

# unitlabel  Пометка для поля для вывода результатов
# help  (E) url для вывода подсказки по этому полю
# length  (R) Длина поля для форматирования
# sortkey  Делает поле сортируемым
# selection  Вводит атрибут в состав альтернативного ключа для
# таблицы, имя которой в переменной STATUS, для режима select
# byorder  Устанавливает ORDER BY для этого атрибута
# url  (E) URL link для использования в выводе результатов
# from_db  (E) Форматирующая функция для перевода данных из
#          формата базы данных
# to_db  (E) Форматирующая функция для перевода данных в
#          формат базы данных
# no_query  Запрещает поле от вывода в форме для запроса
# no_tab  Запрещает поле от вывода в результирующей таблице
# tablen  Поле будет дополнено до этой длины в выходной таблице
# forcetab  Поле будет всегда присутствовать в результирующей
#          таблице
#

```

```

FIELD = oid # Имя атрибута таблицы
label = Select # Название поля для формы
column = oid # Название поля для вывода результатов
type = oid # Тип данных
length = 14
forcetab
no_query

```

```

FIELD = Nsess
label = Session's number
formlabel = Session's number
column = Nsess
type = int
unitlabel = <i>number</i>
length = 10
tablen = 10
#no_query
forcetab
selection
byorder

```

***И так далее для каждого атрибута таблицы.***

***Нужно отметить, что wdb-p95 будет составлять запросы и выводить результаты только для полей, определенных в FDF-файле.***

### **3.2.7 Выходные данные**

***Выходными данными CGI- сценария являются сгенерированные HTML - документы, содержащие элементы формы. Выходные данные wdb-p95 зависят в первую очередь от переменной ACTION. Так, если ACTION=form , то в окно браузера пользователя выводится форма, позволяющая составить запрос к базе данных. Если ACTION=query или ACTION=select , то выводятся результаты запроса в виде таблицы, причем с элементами формы CheckButton напротив каждой строки. Если ACTION=end , то выводятся доступные файлы в виде таблицы, причем напротив каждой строки***

находится ссылка, активизирующая при нажатии получение файла с сервера ИС АРКАД на компьютер пользователя.

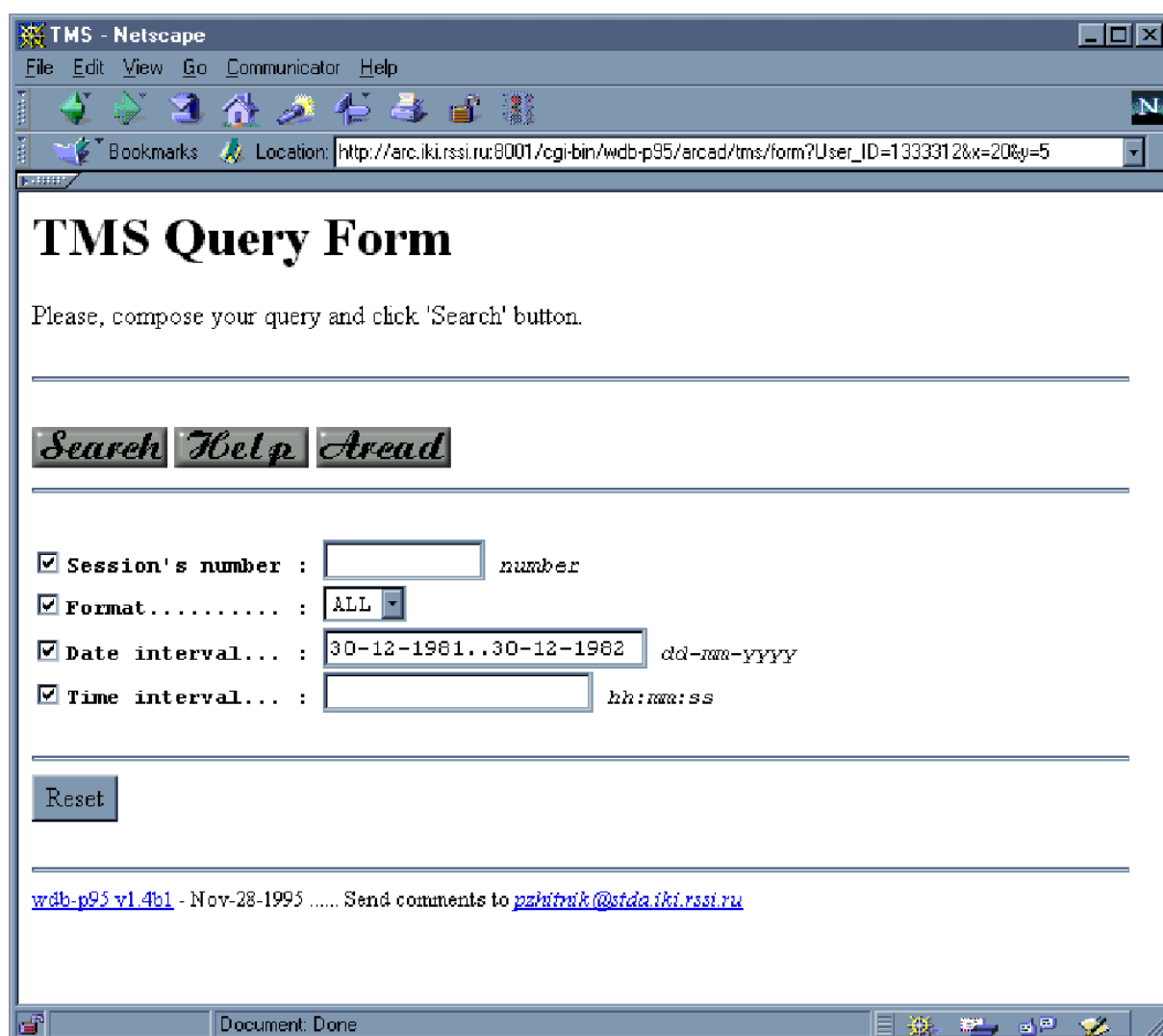
### 3.3 Руководство по использованию WWW – интерфейса пользователя архива данных проекта АРКАД

Для того чтобы воспользоваться этим интерфейсом архива данных проекта АРКАД, пользователь должен прежде всего вызвать URL основной Web - страницы проекта АРКАД (<http://www.iki.rssi.ru/arcad.html>).

На этой странице необходимо перейти по ссылке на интересующую базу данных.

На экране появится форма, содержащая текстовые поля “имя пользователя” и “адрес электронной почты”. В эти текстовые поля необходимо ввести ваше имя и ваш адрес электронной почты, после чего нажать кнопку “Submit” (эта информация нужна для того, чтобы создать уникальный идентификатор пользователя).

На следующей Web - странице следует нажать кнопку “Search” для того, чтобы начать процесс поиска и заказа файлов, или кнопку “Download” для того, чтобы попасть на страницу, с которой можно получить доступные на данный момент файлы.



### **Рис.10. Форма для составления запроса**

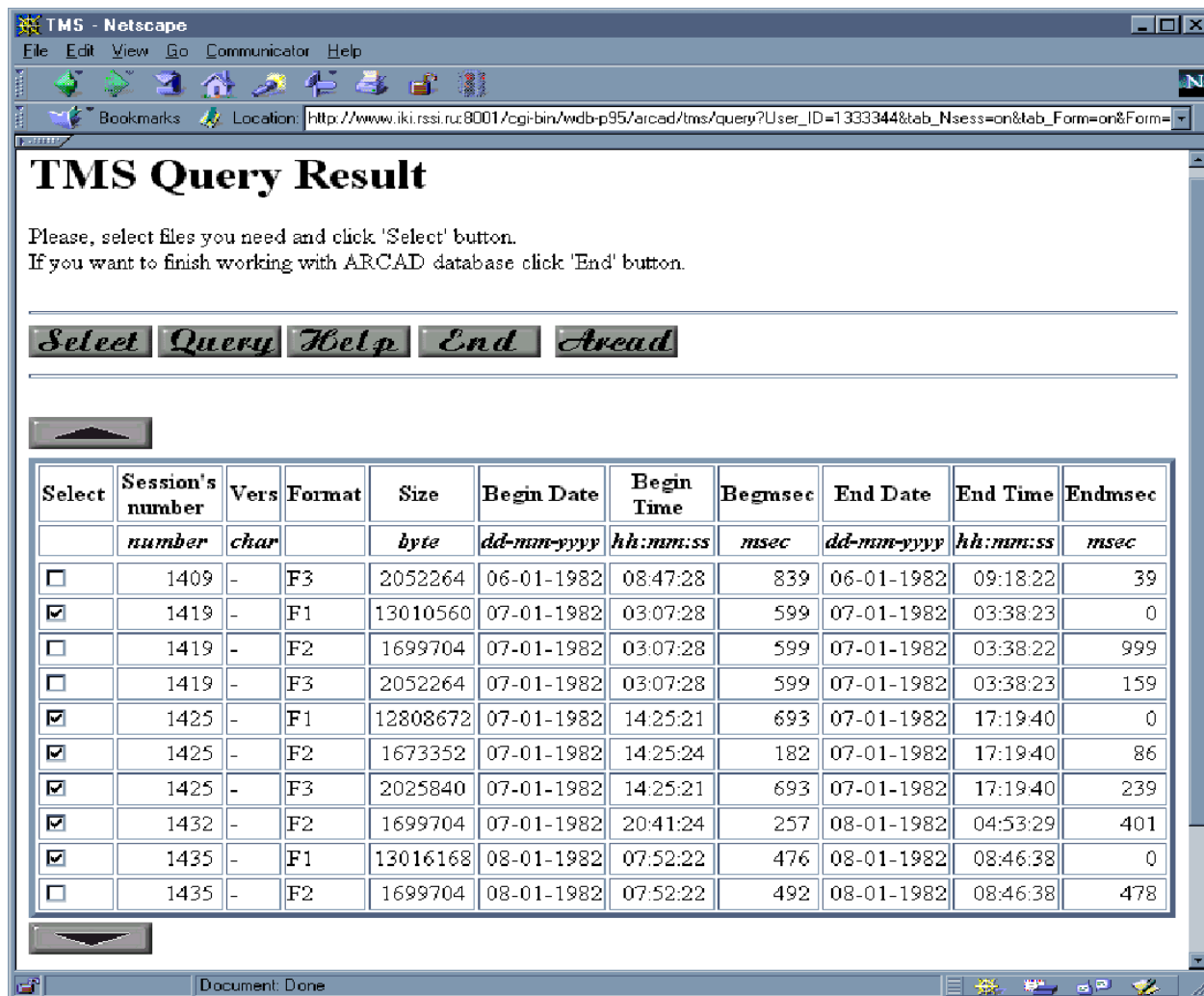
После нажатия кнопки “Search” на экран выведется форма (рис.10), содержащая:

- текстовое поле “номер сеанса”,
- меню выбора формата файлов,
- текстовое поле “дата начала сеанса”
- текстовое поле “время начала сеанса”.

Эту форму необходимо заполнить всеми данными, которые известны пользователю об интересующих его файлах. При этом нужно учитывать, что:

- в поле “номер сеанса” можно вводить как точный номер, так и интервал номеров, отделяя начало интервала от конца двумя точками;
- в поле “дата начала сеанса” нужно вводить дату в формате день/месяц/год, а также интервал дат, отделяя начало интервала от конца двумя точками;
- в поле “время начала сеанса” можно вводить как точное время, так и интервал времени в формате часы:минуты:секунды.

После заполнения формы необходимо нажать на экране кнопку “Search”. Кнопки “ARCAD” и “Help” (рис.10) служат, соответственно, для перехода на домашнюю страницу проекта АРКАД и для получения вспомогательной информации по заполнению формы. Кнопка “Clear” служит для очистки формы.



**Рис.11. Таблица с результатами запроса.**

После нажатия кнопки “Search” на экране появится таблица (рис.11) со строками, удовлетворяющими запросу пользователя. Если число этих строк больше 10, на экран выводятся только первые 10 строк. Чтобы увидеть следующие или предыдущие 10 строк, необходимо нажать кнопку со знаком “Вниз” или “Вверх” соответственно.

Напротив каждой строки находится кнопка. Для того чтобы сформировать заказ на перемещение файла, необходимо нажать кнопку, соответствующую требуемому файлу, после чего нажать кнопку “Select”.

Через некоторое время на экране появится надпись “You have selected files” и такая же таблица. Теперь можно продолжить поиск требуемых файлов.

Для того чтобы изменить запрос к базе данных, нужно нажать кнопку “Query”. Нажатие кнопки “Help” выведет на экран вспомогательную информацию. Кнопка “ARCAD” служит для перехода на основную страницу проекта АРКАД.

Для того чтобы закончить сеанс работы с ИС проекта АРКАД, необходимо нажать кнопку “End”. При этом на экран будут выведены доступные файлы, среди заказанных в данном сеансе. Эти файлы можно получить, нажав “download” напротив нужного файла. Остальные файлы можно будет получить через некоторое время в FTP- доступной области.

## **Список литературы**

- 1. Гальперин Ю.И., Гладышев В.А., Козлов А.И. и др. “Электромагнитная совместимость научного космического комплекса АРКАД-3”. М.: Наука, 1984.**
- 2. Васкевич Д. Стратегии клиент/сервер. Киев: Изд. “Диалектика”. 1996.**
- 3. Кузнецов С.Д.. Введение в СУБД. //Системы управления базами данных. 1995г. №1-4.**
- 4. Ладыженский Г.М.. Система управления базами данных – кратко о главном.//Системы управления базами данных. 1995г. №1-4.**
- 5. Спейнаур С., Куэрсиа В. Справочник Web - мастера. Изд. Группа BHV. 1997.**
- 6. Пин-Шен Чен П. Модель сущность/связь – шаг к единому представлению данных. //Системы управления базами данных. 1995г. №4.**
- 7. Quigley E. PERL by Example. Prentice-Hall, Inc. 1995.**
- 8. Loren Buhle E., Acofа N., Fair Weather R., Ritchey T., Morgan L. Webmaster’s professional reference. New Riders Publishing, Indianapolis, IN.1996.**

## Содержание

<b>Введение</b> .....	<b>2</b>
<b>1. Технические аспекты реализации ИС проекта АРКАД</b> .....	<b>6</b>
1.1 Обоснование использования трехзвенной архитектуры “клиент-сервер” .....	6
1.2 Выбор Web – сервера .....	6
1.3 Выбор СУБД .....	7
1.4 Основные концепции архитектуры POSTGRES95 .....	7
1.5 Программные интерфейсы POSTGRES95 .....	9
1.5.1 Perl API-интерфейс (pg95perl) .....	9
1.5.2 WWW-интерфейс Postregs95 wdb-p95 .....	9
<b>2. Реализация информационной модели системы с использованием СУБД POSTGRES95</b> .....	<b>9</b>
2.1 Структура таблиц, используемых в системе .....	9
2.2 Заполнение таблиц, используемых в системе .....	12
<b>3. Программный комплекс, реализующий функциональную модель ИС проекта АРКАД</b> .....	<b>12</b>
3.1 Состав комплекса .....	12
3.2 Программа, реализующая “ПРОЦЕСС ЗАКАЗА” функциональной модели ИС проекта АРКАД и использующая для взаимодействия с пользователем системы WWW - интерфейс .....	13
3.2.1 Общие сведения и функциональное назначение .....	13
3.2.2 Описание логической структуры .....	13
3.2.3 Структура программы с описанием функций составных частей и связи между ними .....	15
3.2.4 Связи программы с другими программами .....	16
3.2.5 Вызов и загрузка .....	16
3.2.6 Входные данные .....	16
3.2.7 Выходные данные .....	17
3.3 Руководство по использованию WWW – интерфейса пользователя архива данных проекта АРКАД .....	18
<b>Список литературы</b> .....	<b>19</b>

**055(02)2**

**Ротапринт ИКИ РАН  
Москва, 117810, Профсоюзная,  
84/32**

---

**Подписано к печати 24.06.97**

---

**Заказ**

**Формат 70×108/32**

**Тираж 75**

**1 уч.-изд. л.**