

ОРДЕНА ЛЕНИНА
ИНСТИТУТ ПРИКЛАДНОЙ МАТЕМАТИКИ
им. М.В.Келдыша РАН

А.В. Воронков, С.Л. Головков

Архив пакета прикладных программ «РЕАКТОР»

Москва
2005

А.В. Воронков, С.Л. Головков
Архив пакета прикладных программ «РЕАКТОР»

Аннотация

Описывается подсистема хранения и анализа данных пакета прикладных программ «РЕАКТОР». Эта подсистема (Архив) предназначена для организации счета с продолжением, обмена информацией между различными функциональными частями пакета, обеспечения пользователя удобными и мощными средствами анализа результатов счета.

Описывается архитектура Архива, библиотека подпрограмм для работы с Архивом и Редактор Архива, представляющий пользователю средства для анализа данных, хранящихся в Архиве.

A.V. Voronkov, S.L. Golovkov
Archive of the application package «REACTOR»

Abstract

The subsystem of data storage and analysis of the application package «REACTOR» is described. This subsystem (called «Archive») is intended for organizing the calculation with the continuation, information transfer between different functional parts of package, providing users with the convenient and powerful facilities for analysis of the results of calculation. Are described the architecture of Archive, subroutine library for the work with the Archive and the Archive Editor, presenting to user tools for the data analysis.

Работа выполнена при финансовой поддержке Российского Фонда Фундаментальных Исследований (проект № 05-01-08004-офи_э «Разработка и внедрение программного пакета полномасштабного моделирования основных нейтронно-физических и тепло-гидравлических процессов в ядерных реакторах различного типа и назначения»)

Содержание

Введение	4
1. Архив пакета «РЕАКТОР»	5
2. Архитектура Архива	6
2.1. Дисковый архив	7
2.2. Оперативный архив	8
3. Интерфейс функциональных модулей	9
3.1. Внешняя среда	10
3.2. Создание Архива	10
3.3. Открытие Архива	10
3.4. Запись в Архив	11
3.5. Чтение из Архива	12
3.6. Удаление архивного массива	13
3.7. Заккрытие Архива	13
3.8. Печать оглавления Архива	14
3.9. Файл конфигурации Архива	15
4. Редактор Архива	16
4.1. Основная панель Редактор Архива	17
4.2. Просмотр содержимого массива	18
4.3. Визуализация	20
4.3.1. Использование системы Gnuplot	21
4.3.1.1. Стили визуализации	21
4.3.1.2. Нормировка	22
4.3.1.3. Выбор координаты и плоскости	22
4.3.1.4. Режимы демонстрации	23
4.3.2. Использование системы Tecplot	24
4.3.2.1. Режим IJ-Ordered Data	24
4.3.2.2. Режим Finite-Element Data	25
4.4. Удаление массивов и сжатие Архива	28
4.5. Создание нового Архива	30
Заключение	31
Литература	32

Введение

Пакет прикладных программ «РЕАКТОР» [1] разработан в Институте прикладной математики им. М.В.Келдыша РАН и в течение многих лет эксплуатируется в ИПМ РАН и ряде других организаций. Пакет «РЕАКТОР» позволяет проводить численный расчет как стационарного состояния, так и расчет кампании ядерного реактора. Расчет может проводиться в одно-, двух- и трехмерных геометриях, в различных приближениях по энергии (одно-, мало- и многогрупповое), с использованием различных приближений уравнения переноса (диффузионное, P1, кинетическое).

Пакет представляет собой совокупность большого числа независимых программ – **функциональных модулей**. Каждый такой модуль решает логически завершенный фрагмент общей задачи, например, ввод геометрической информации, подготовка групповых констант, расчет нейтронных полей, выгорание и т.д. Прикладной специалист для решения своей задачи должен сформировать и выполнить цепочку модулей, каждый из которых в качестве входных данных использует выходные данные предшествующих модулей.

Пакет «РЕАКТОР» включает в свой состав не только модули, выполняющие собственно расчеты, но и большой набор компонент, обеспечивающих эффективную эксплуатацию пакета. Фактически счетные модули пакета «погружены» в «управляющую среду», предоставляющую пользователю (прикладному специалисту) удобные сервисы для управления пакетом на всех этапах работы с задачей.

В настоящей статье описывается один из компонентов «управляющей среды» пакета прикладных программ «РЕАКТОР», а именно **подсистема хранения и анализа данных (Архив)**.

Архив является интегрирующим элементом пакета «РЕАКТОР». Он представляет собой специализированную базу данных, содержащую числовые массивы. Каждый модуль считывает из Архива необходимые ему данные и записывает в него результаты своей работы.

Один раз Архив уже «пережил» коренную модернизацию, связанную с переходом от ЭВМ БЭСМ-6 к персональным компьютерам и рабочим станциям, что повлекло за собой отказ от сложной иерархии запоминающих устройств (магнитные ленты, диски, барабаны). Цели, которые были поставлены и достигнуты при той модернизации, состояли в достижении машинной независимости, надежности и простоте сопровождения пакета [2, 3].

Выросшие функциональные возможности пакета (полномасштабное моделирование нейтронно-физических и тепло-гидравлических процессов, потвальный расчет и т.п.), новые вычислительные возможности (распределенная вычислительная среда, суперкомпьютеры), возросшие требования пользователей (в частности, необходимость сложного анализа результатов) вызвали новые требования к системе хранения числовых данных и,

следовательно, обусловили необходимость разработки новой версии Архива пакета «РЕАКТОР».

В настоящей статье описывается новая версия Архива. Описана архитектура Архива, библиотека подпрограмм для работы с Архивом и Редактор Архива, представляющий собой графический интерфейс, обеспечивающий доступ прикладного специалиста к содержимому Архива.

1. Архив пакета «РЕАКТОР»

Основное назначение Архива пакета «РЕАКТОР» состоит в обеспечении:

- организации счета с продолжением;
- обмена информацией между различными функциональными частями пакета;
- возможности анализа результатов счета.

С точки зрения пользователя Архив представляет собой *базу данных*, предназначенную для хранения *одномерных массивов* (целых или действительных чисел). Запись в Архив и чтение из Архива возможно как целиком массива, так и его фрагментов.

С архитектурной точки зрения Архив представляет собой совокупность *архивных объектов*, каждый из которых предназначен для хранения одного массива. Архивный объект имеет *уникальный идентификатор (ID)*, представляющий собой пятерку значений {"имя", "номер варианта", "индекс 1", "индекс 2", "индекс 3"}. "Имя" представляет собой строку длиной не более шести символов, а "номер варианта", "индекс 1", "индекс 2" и "индекс 3" – целые числа.

Смысл целочисленных компонент ID архивного объекта состоит в том, что они позволяют в пределах одного Архива разбивать всю информацию на отдельные непересекающиеся порции. Одну такую порцию составляет информация, относящаяся к определенному счетному варианту, указываемому компонентой "номер варианта". "Явные" индексы, значения которых определяет пользователь по своему усмотрению, чаще всего используется для указания номера плоскости, номера группы, номера итерации, номера шага по времени и т.д.

Каждый архивный объект представляет собой набор элементов оглавления и связанную с этими элементами совокупность файловых записей, не обязательно смежных.

С точки зрения файловой системы Архив представляет собой один или несколько *бинарных файлов прямого доступа*. В момент создания Архива определяется – будет ли он одно- или много-файловым.

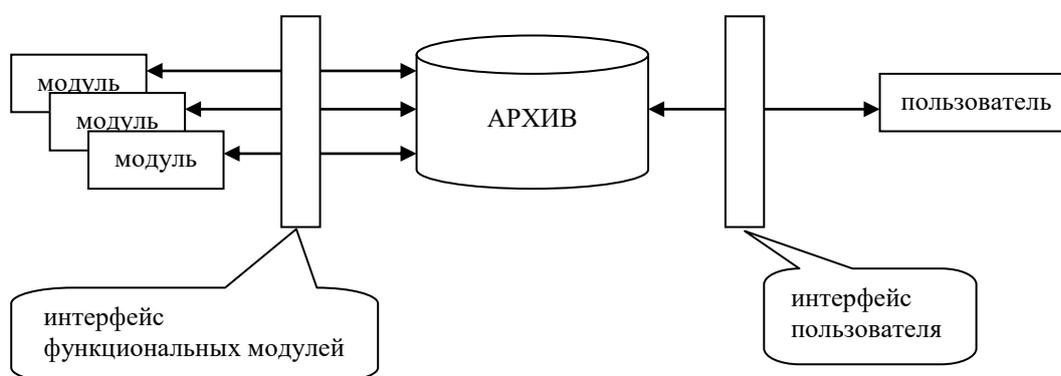
Если создается одно-файловый Архив, то его размер определяется допустимым с точки зрения операционной системы размером файла. Если создается много-файловый Архив, то при его создании указывается максимальный размер отдельного файла. В этом случае при достижении

текущим архивным файлом максимально допустимого размера создается новый архивный файл.

При создании Архива его имя совпадает с именем архивного файла. Например, для Архива “Archiv” создается файл Archiv. Если это много-файловый Архив, то по мере записи в него данных появляются новые архивные файлы: Archiv_001, Archiv_002, Archiv_003 и т.д.

С Архивом «работают» как модули расчетного пакета, так и пользователи. Поэтому были разработаны и реализованы два интерфейса для доступа к содержимому Архива:

- интерфейс функциональных модулей,
- интерфейс пользователя.



Интерфейс функциональных модулей реализован в виде библиотеки фортран-подпрограмм, включающей, например, такие подпрограммы: createArchive (создание нового Архива), openArchive (открытие Архива), hWriteI (запись целочисленного массива), closeArchive (закрытие Архива) и т.п.

Интерфейс пользователя («Редактор Архива»), представляет собой графический интерфейс, функционирующий в среде Windows и обеспечивающий широкий спектр возможностей по доступу к содержимому Архива и манипуляций с этим содержимым. В частности, Интерфейс позволяет просматривать оглавление Архива и содержимое архивных массивов, визуализировать архивные массивы, удалять массивы из Архива и сжимать Архивы, копировать массивы из одного Архива в другой и т.п.

2. Архитектура Архива

При открытии Архива данные, хранящиеся в нем (все или часть), перемещаются в оперативную память. Таким образом, во время работы счетного модуля Архив представляет собой совокупность структур данных, расположенных как на диске, так и в оперативной памяти компьютера. При завершении работы (при закрытии Архива) данные, находящиеся в оперативной памяти, вновь перемещаются на диск.

Часть Архива, которая размещается в оперативной памяти, называется **оперативным архивом**, а та часть, которая размещается на диске - **дисквым архивом**.

Распределение данных между оперативной и дисковой памятью, а также логика работы с этими запоминающими устройствами зависит от «стратегии» функционирования Архива. Эта стратегия (называемая **архивной стратегией**) задается с помощью специальной директивы файла конфигурации ARCHIV.CFG (описание структуры конфигурационного файла и содержащихся в нем директив приведено в разделе 3.9).

В настоящее время реализованы две архивные стратегии:

- стратегия «DISK», суть которой состоит в том, что при открытии Архива в оперативную память перемещается только оглавление Архива, а все операции чтения/записи выполняются с дисквым архивом;
- стратегия «RAM», суть которой состоит в том, что при открытии Архива в оперативную память (в оперативный архив) перемещается оглавление и содержимое Архива; все операции доступа к Архиву (чтение, запись, создание и т.п.) выполняются над оперативным архивом, а при закрытии Архива оглавление и содержимое оперативного архива перемещаются в дисквый архив.

2.1. Дисквый архив

Дисквый архив представляет собой файл (файлы) прямого доступа, в котором выделены три части: **заголовок**, **оглавление** и **область хранения** (рис. 2.1).



Рис.2.1. Структура дисквого архива

Начало первого архивного файла занято заголовком. Следом за заголовком располагается последовательность чередующихся **блоков оглавления** и **областей хранения**. Размер каждого блока оглавления

фиксирован. Размер областей хранения произволен (с точностью до записи) и зависит от размеров записываемых в Архив массивов.

Первый блок оглавления начинается сразу за заголовком Архива. Если в некоторый момент произойдет исчерпание памяти, выделенной для оглавления, то создается следующий блок оглавления. Все блоки оглавления связаны в однонаправленный список.

Заголовок содержит следующую информацию: имя версии архивных программ, имя архива, размер заголовка, размер блока оглавления, указатель на последнюю запись текущего блока оглавления, указатель на текущую запись текущего блока оглавления, указатель на текущий элемент оглавления текущего блока оглавления, указатель на первую свободную запись Архива и т.п.

Блок оглавления представляет собой последовательность **элементов оглавления** фиксированного размера. Элементы оглавления имеют тип, определяющий ту роль, которую они играют в организации оглавления. Элементы оглавления могут быть **служебными** и **рабочими**. Служебные типы элементов оглавления используются для организации структуры оглавления.

Рабочие типы элементов оглавления описывают архивные объекты. Такой элемент содержит **идентификатор объекта**, **служебную информацию** (тип хранимого массива, дата и время создания или последнего обновления архивного объекта, длина зоны хранения), **описание зоны хранения** (имя архивного файла, адрес зоны хранения, длина зоны хранения).

Область хранения дискового архива представляет собой последовательность зон хранения архивных объектов. Каждая зона хранения состоит из одной или нескольких смежных записей архивного файла. Адресом зоны является номер первой записи.

2.2. Оперативный Архив

Архив, расположенный в оперативной памяти (оперативный архив), представляет собой набор динамически выделяемых областей оперативной памяти, предназначенных для хранения архивных объектов и описателей этих объектов.

С архитектурной точки зрения оперативный архив представляет собой односвязный список описателей архивных объектов, с каждым из которых связан список описателей зон хранения на диске и может быть связана зона хранения в оперативной памяти (оперативная зона хранения). На рис.2.2 представлена структура оперативного Архива.

На этом рисунке квадраты с надписью “оз” представляют описатели зон хранения на диске. Так, на рисунке первый описатель архивного объекта имеет один описатель зоны хранения, а второй описатель архивного объекта имеет два описателя зон хранения на диске (т.е. второй массив хранится в двух несмежных областях Архива).

Оперативный архив

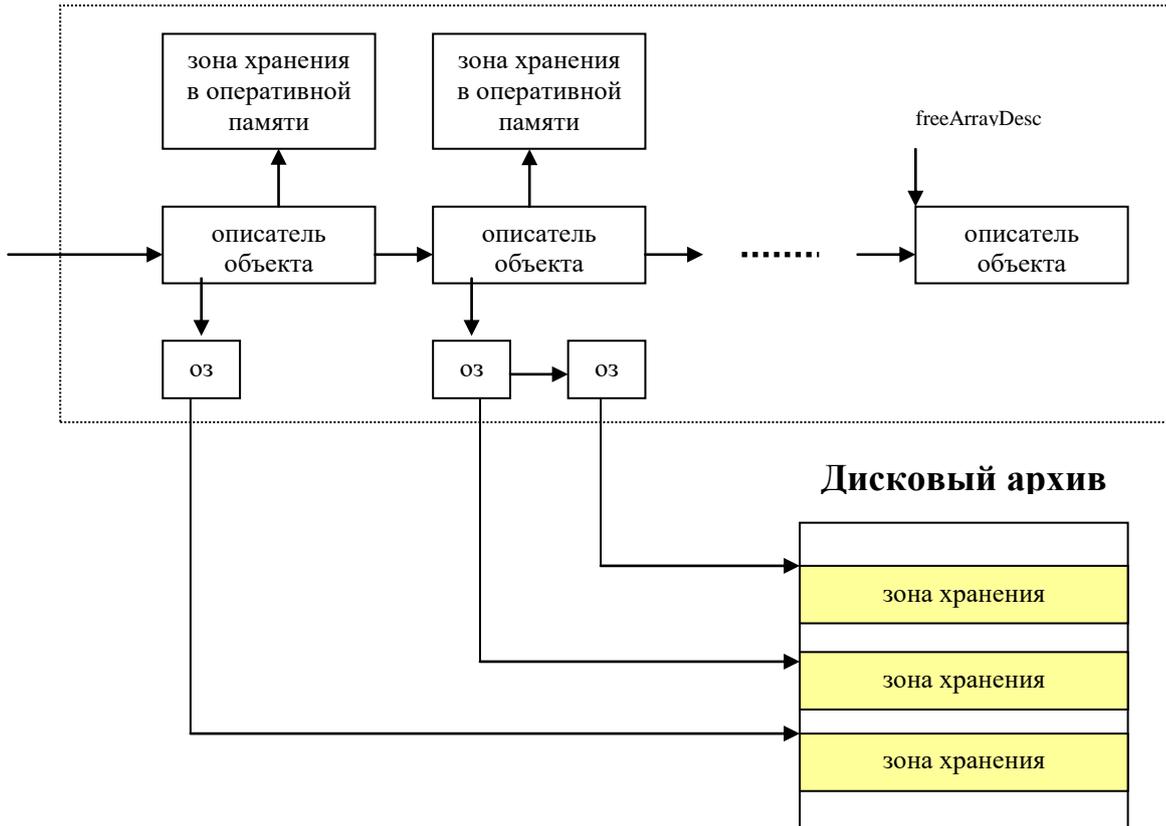


Рис.2.2. Структура оперативного архива

Если используется архивная стратегия «DISK», то зоны хранения в оперативной памяти не создаются; вся работа осуществляется с дисковым архивом. Если используется архивная стратегия «RAM», то в оперативной памяти создаются зоны хранения и вся работа осуществляется только с ними (т.е. только с оперативным архивом); при необходимости создаются описатели зон хранения дискового архива, но сами зоны хранения на диске не создаются (они будут созданы только при закрытии Архива).

3. Интерфейс функциональных модулей

Как уже говорилось, Интерфейс функциональных модулей представляет собой набор подпрограмм, написанных на языке Фортран 90. Библиотека этих подпрограмм содержится в следующих файлах:

confArchive.f90	конфигурационный модуль, описывающий структуру архивного файла
createArchive.f90	подпрограммы создания, открытия и закрытия Архива
ioArchive.f90	подпрограммы чтения/записи архивных массивов
serveArchive.f90	подпрограмма печати оглавления Архива
auxArchive.f90	вспомогательные подпрограммы

3.1. Внешняя среда

Архивные подпрограммы используют в процессе своей работы значения двух common-переменных, определенных в счетном модуле, вызывающем эти подпрограммы:

```
COMMON /ntask/ntask  
COMMON /namsec/namsec
```

Переменная **ntask** определяет номер варианта. Архивные подпрограммы имеют доступ только к тем архивным массивам, в идентификаторе которых значение компоненты «номер варианта» совпадает со значением переменной **ntask**.

Переменная **namsec** содержит имя выполняемого модуля и используется для определения имени диагностического файла (out-файла) при возникновении необходимости вывода какого-либо сообщения. Если архивная подпрограмма обнаружит, что значение переменной **namsec** не определено, то вывод сообщений будет выполняться в файл с именем **ioSystem.out**.

3.2. Создание Архива

Для создания Архива предназначена подпрограмма **CreateArchive**. Ее вызов имеет следующий формат:

```
call CreateArchive
```

Имя создаваемого Архива подпрограмма берет из конфигурационного файла **ARCHIV.CFG**, который содержит параметры создаваемого Архива. Конфигурационный файл может содержать лишь имя создаваемого Архива, оставляя остальные параметры на усмотрение системы. Если конфигурационный файл отсутствует, то подпрограмма создает Архив с именем **ARCHIV**.

3.3. Открытие Архива

Для открытия Архива предназначена подпрограмма **OpenArchive**. Ее вызов имеет следующий формат:

```
call OpenArchive
```

Имя открываемого Архива подпрограмма определяет также как и подпрограмма **CreateArchive**: если существует конфигурационный файл **ARCHIV.CFG**, то имя Архива берется из него, если же конфигурационного файла нет, то делается попытка открыть Архив с именем **ARCHIV**.

Явный вызов подпрограммы **OpenArchive** не является обязательным, поскольку все подпрограммы доступа к Архиву проверяют «открытость» Архива и в случае необходимости сами вызывают подпрограмму **OpenArchive**.

3.4. Запись в Архив

Для записи массивов в Архив предназначены следующие подпрограммы:

hWriteI - запись целочисленного массива

hWriteIP - запись вырезки целочисленного массива

hWriteR - запись массива действительных чисел

hWriteRP - запись вырезки массива действительных чисел

Формат обращения к этим подпрограммам имеет следующий вид:

hWriteI(Name, Array, Ind1, Ind2, Ind3, Length)

hWriteIP(Name, Array, Ind1, Ind2, Ind3, Length, k1, k2)

hWriteR(Name, Array, Ind1, Ind2, Ind3, Length)

hWriteRP(Name, Array, Ind1, Ind2, Ind3, Length, k1, k2)

где	character(len=6)	:: Name	имя архивного объекта
	dimension(:)	:: Array	имя записываемого массива
	integer	:: Ind1, Ind2, Ind3	индексы
	integer	:: Length	длина записываемого массива
	integer	:: k1, k2	границы вырезки массива

Если указанный архивный объект не существует, то он создается. Если указанный архивный объект существует, но его размера недостаточно для размещения записываемого массива (вырезки массива), то размер архивного объекта увеличивается на необходимую величину.

Элементы создаваемого (увеличиваемого) архивного объекта инициализируются нулевыми значениями.

Запись вырезки массива выполняется следующим образом (рис.3.1):

- вырезка $[1, k2-k1+1]$ из массива с именем Array и длиной Length записывается в архивный объект {Name, ntask, Ind1, Ind2, Ind3} в позиции $[k1, k2]$

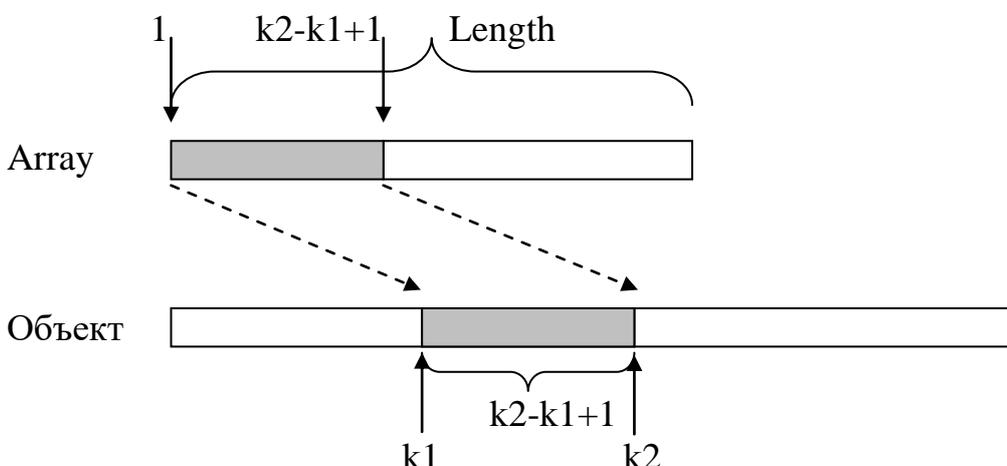


Рис.3.1. Запись вырезки массива в архивный объект

Пример: `call hWriteI('TGROUP', TGROUP, 0, 0, 0, len(TGROUP))`
 Подпрограмма запишет в Архив под именем 'TGROUP' (более точно – в архивный объект с уникальным идентификатором { 'TGROUP', ntask, 0, 0, 0 }) массив целых чисел TGROU**P**.

3.5. Чтение из Архива

Для чтения массивов из Архива предназначены следующие подпрограммы:

hReadI – чтение целочисленного массива
 hReadIP – чтение вырезки целочисленного массива
 hReadR – чтение массива действительных чисел
 hReadRP – чтение вырезки массива действительных чисел

Формат обращения к этим подпрограммам имеет следующий вид:

hReadI (Name, Ind1, Ind2, Ind3, Array, Length)
 hReadIP (Name, Ind1, Ind2, Ind3, Array, Length, k1, k2)
 hReadR (Name, Ind1, Ind2, Ind3, Array, Length)
 hReadRP (Name, Ind1, Ind2, Ind3, Array, Length, k1, k2)

где character(len=6) :: Name имя архивного объекта
 dimension(:) :: Array имя массива-назначения
 integer :: Ind1, Ind2, Ind3 индексы
 integer :: Length длина массива-назначения
 integer :: k1, k2 границы вырезки

Чтение вырезки массива осуществляется следующим образом (рис. 3.2):

- значения, находящиеся в позициях [k1, k2] архивного объекта {Name, ntask, Ind1, Ind2, Ind3}, помещаются в позиции [1, k2-k1+1] массива с именем Array и длиной Length

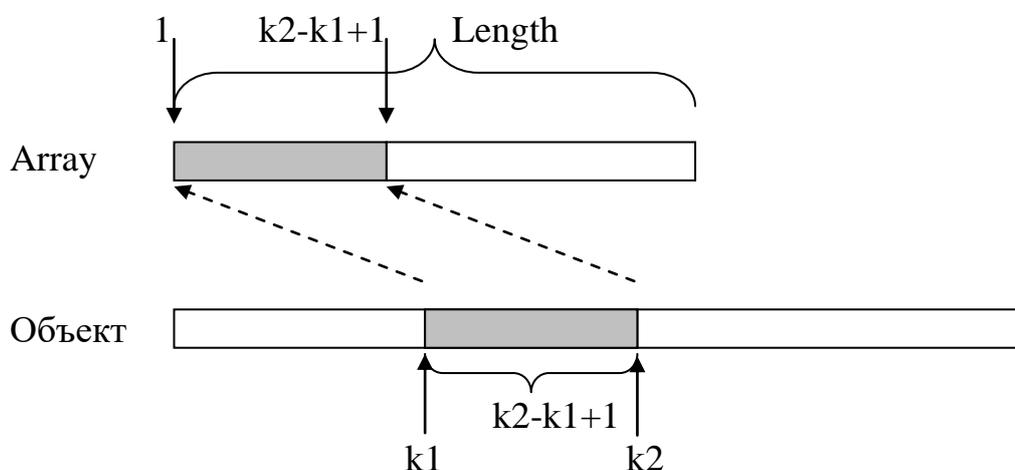


Рис.3.2. Чтение вырезки массива из архивного объекта

Пример: `call hReadIP('CONSTS', 1,0,0, NRO, len(NRO), 21,27)`
 Подпрограмма прочитает данные, хранящиеся в позициях [21,27] архивного объекта с уникальным идентификатором {'CONSTS', ntask,1,0,0}, и поместит их в позиции [1,7] массива целых чисел NRO.

Для совместимости со старыми версиями модулей пакета РЕАКТОР в библиотеку Архива включены дополнительные подпрограммы чтения массивов из Архива:

ghReadI - чтение целочисленного массива
 ghReadIP - чтение вырезки целочисленного массива
 ghReadR - чтение массива действительных чисел
 ghReadRP - чтение вырезки массива действительных чисел

Эти подпрограммы отличаются от приведенных выше тем, что среди их параметров отсутствует параметр Length (длина массива-назначения). Эти подпрограммы являются «опасными», поскольку при записи в массив они не проверяют выход за границы массива-назначения. Поэтому нужно стараться избегать их применения.

3.6. Удаление архивного массива

Удаление архивного массива (архивного объекта) осуществляется с помощью вызова подпрограммы **deleteArray**, вызов которой имеет следующий формат:

```
call deleteArray( Name, arInd1, arInd2, arInd3 )
```

где character(len=6) :: Name имя архивного объекта
 integer :: Ind1,Ind2,Ind3 индексы

Отметим, что реального уменьшения объема Архива не происходит; указанный архивный массив просто «вычеркивается» из оглавления. Для того, чтобы «сжать» Архив, необходимо использовать Редактор Архива, описанный в разделе 4.

3.7. Закрытие Архива

Закрытие Архива осуществляется с помощью вызова подпрограммы **CloseArchive**:

```
call CloseArchive
```

Выполнение подпрограммы состоит в перемещении всех данных из оперативного архива в дисковый архив, освобождении динамической памяти, занимаемой оперативным архивом и закрытии архивных файлов.

3.8. Печать оглавления Архива

Для «печати» оглавления Архива предназначена подпрограмма **PrintArchive**. Ее вызов имеет следующий формат:

```
call PrintArchive
```

Подпрограмма выводит в текущий out-файл содержимое оглавления Архива в следующем виде:

```
ARCHIVE (DISK): ARCHIV
version: h4.01
```

```
#####
# N : N : Name : I1 : I2 : I3 : Length : Date : Time : Type #
# : Var: : : : : : elem : rec : of rec : of rec : #
#-----#
# 1 : 0 : KAN : 0 : 0 : 0 : 12 : 1 : 26.03.04 : 22.25.05 : int #
# 2 : 0 : TASK : 0 : 0 : 0 : 38 : 1 : 26.03.04 : 22.25.05 : int #
# 3 : 1 : CHART : 0 : 0 : 0 : 600 : 3 : 26.03.04 : 22.25.05 : int #
# 4 : 1 : KARTAP : 1 : 0 : 0 : 1519 : 6 : 26.03.04 : 22.25.05 : int #
# 5 : 1 : KRTBLK : 1 : 0 : 0 : 1519 : 6 : 26.03.04 : 22.25.05 : int #
# . . . . .
# 168 : 1 : FLUX3S : 26 : 0 : 0 : 24304 : 95 : 09.04.04 : 21.30.51 : real #
# 169 : 1 : FLUX3R : 26 : 0 : 0 : 24304 : 95 : 09.04.04 : 21.30.51 : real #
# 170 : 1 : SFTWRK : 0 : 0 : 0 : 24304 : 95 : 09.04.04 : 21.30.51 : real #
# 171 : 1 : SFTN3D : 0 : 0 : 0 : 24304 : 95 : 09.04.04 : 21.30.51 : real #
#####
```

```
Total Number of Elements = 2694602
Total Number of Records = 10566
```

Колонки таблицы имеют следующее значение:

N	порядковый номер архивного объекта
N Var	номер варианта
Name	имя архивного объекта
I1,I2,I3	индексы 1, 2 и 3
Length	длина архивного объекта
elem	- в числе элементов массива
rec	- в числе записей
Date of rec	дата создания или последней модификации
Time of rec	время создания или последней модификации
Type	тип хранимых данных

Если был задан режим сбора статистики, то вслед за выдачей оглавления Архива будут выведены статистические данные об использовании Архива. Эти данные включают число операций записи целочисленных массивов, общее время записи целочисленных массивов, число операций записи действительных массивов и т.п.

Печать оглавления осуществляется вне зависимости от того, открыт Архив или нет. В первом случае (Архив открыт, оглавление берется из оперативной памяти) «шапка выдачи» имеет вид:

ARCHIVE (MEMORY): <имя Архива>

Во втором случае (Архив закрыт, оглавление берется с диска) «шапка выдачи» имеет вид:

ARCHIVE (DISK): <имя Архива>

3.9. Файл конфигурации Архива

Некоторые характеристики Архива и режимы его функционирования определяются значениями **конфигурационных параметров**, указанными с помощью **файла конфигурации ARCHIV.CFG**. Этот файл используется программой создания Архива (CreateArchive) и программой открытия Архива (OpenArchive). Остальные программы получают значения конфигурационных параметров из глобальных переменных, значения которых устанавливаются при создании или открытии Архива.

Правила использования конфигурационного файла ARCHIV.CFG таковы:

- если файл отсутствует, то значения конфигурационных параметров (включая имя Архива) устанавливаются по умолчанию;
- то же самое происходит, если конфигурационный файл есть, но в нем отсутствует описание параметров Архива;
- если описание Архива есть, то с его помощью переопределяются только те конфигурационные параметры, для которых в описании Архива имеются соответствующие директивы.

Не вдаваясь в подробности, приведем пример конфигурационного файла, комментируя его строки:

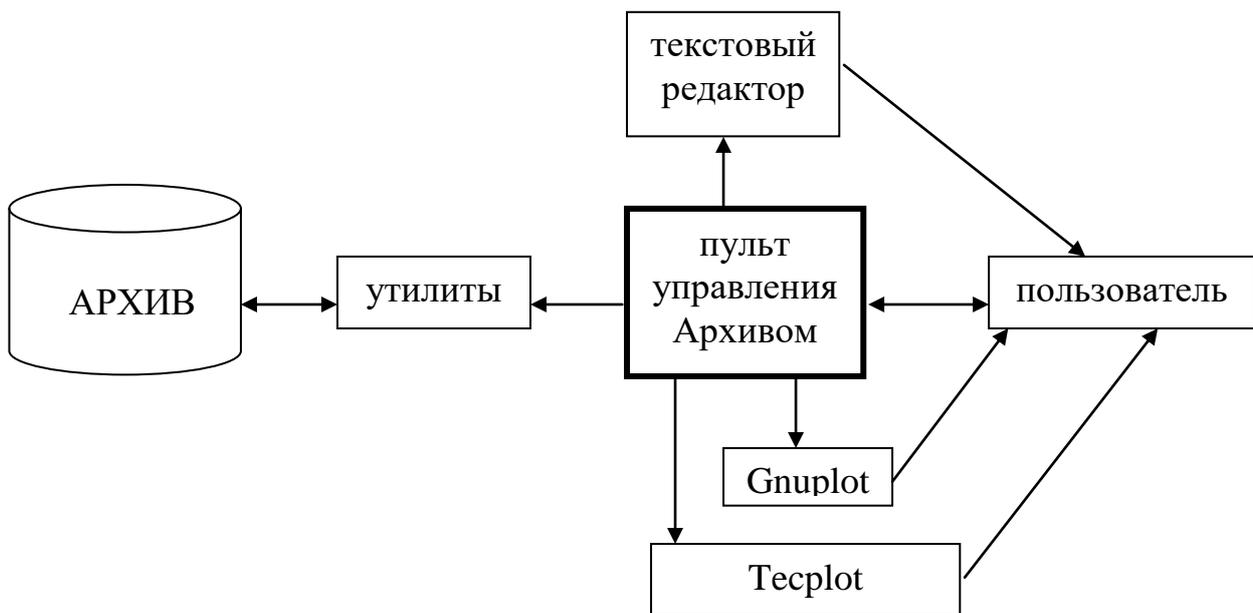
===ARCHIVE NEWARC	определяет имя Архива - NEWARC
SIZE_HEAD= 1	определяет размер заголовка (число записей)
SIZE_BLOCK_CONTENT= 10	определяет размер блока оглавления
N_UNIT=900	определяет диапазон устройств ввода-вывода
STRATEGY=DISK	определяет архивную стратегию
MAXFILESIZE=300	определяет предельный размер архивного файла
STATISTIC=NO	определяет режим сбора статистики
===END	

4. Редактор Архива

Подсистема «Редактора Архива» представляет собой графический интерфейс пользователя для работы с Архивами пакета прикладных программ «РЕАКТОР». Редактор Архива функционирует в среде Windows98/XP и обеспечивает широкий спектр возможностей по доступу к содержимому Архива и манипуляций с этим содержимым. В частности, Редактор Архива позволяет

- просматривать оглавление Архива;
- просматривать в различных режимах содержимое архивных массивов, в том числе и с предобработкой (выделение плоскостей и др.);
- визуализировать некоторые типы архивных массивов с помощью двух графических систем (Gnuplot и Tecplot);
- удалять массивы из Архива и сжимать Архивы;
- создавать новые Архивы и копировать массивы из одного Архива в другой;
- менять представление Архива.

Архитектура Редактора Архива имеет следующий вид:



Ядром Редактора Архива является компонент «Пульт управления Архивом». Этот компонент написан на языке С и предоставляет пользователю требуемую функциональность.

При необходимости Ядро вызывает «внешние» инструменты, например текстовый редактор для просмотра содержимого архивного массива или графическую систему для построения графика, интересующего пользователя.

Для выполнения каких-либо манипуляций с Архивом (например, чтения оглавления или фрагмента массива, удаления или вставки массива и т.п.), «Пульт управления Архивом» запускает **утилиты** - автономные программы, написанные на языке Фортран. В свою очередь, утилиты для работы с Архивом пользуются стандартными подпрограммами Интерфейса функциональных модулей.

Общение между Ядром и утилитами организовано с помощью текстовых файлов: Ядро записывает в файл «здание» утилите, затем запускает утилиту, а после завершения ее работы считывает из файла результаты. Решение о такой организации доступа к содержимому Архива принято по следующим соображениям:

- обеспечение высокой степени независимости Редактора Архива от реализации Архива. Изменение структуры или иных характеристик Архива, как правило, требует только перекомпиляции программ-утилит;
- обеспечение высокой степени мобильности Редактора Архива. Поскольку утилиты написаны на том же языке, что и функциональные модули пакета, то там, где работает расчетный пакет, будет работать и Редактор Архива;
- обеспечения возможности работы Редактора Архива в распределенной вычислительной среде, поскольку Ядро Редактора Архива («Пульт управления Архивом») и утилиты могут работать на разных компьютерах (например, Ядро – на персональном компьютере, утилиты – на удаленном суперкомпьютере).

4.1. Основная панель Редактора Архива

После запуска Редактора Архива на экране появляется основная панель - «Пульт управления Архивом». Допустим, мы нажали на этой панели кнопку «Открыть...», и на появившейся панели выбора файла выбрали некий Архив.

Если выбранный файл является Архивом, то в большом окне «Пульта управления Архивом» появится оглавление этого Архива, в верхних полях появится информация об Архиве, и активизируются некоторые кнопки (рис. 4.1).

Два больших окна этой панели имеют следующее назначение:

- верхнее окно предназначено для работы с существующим Архивом,
- нижнее окно предназначено для создания нового Архива.

Все манипуляции с Архивом осуществляются с помощью средств, находящихся в верхней части панели (верхнее окно, поля над этим окном и кнопки, расположенные слева от окна).

Нижняя часть панели (нижнее окно, кнопка «Создать» и поле ввода имени) предназначены для создания новых Архивов и контроля копирования массивов в новый Архив.

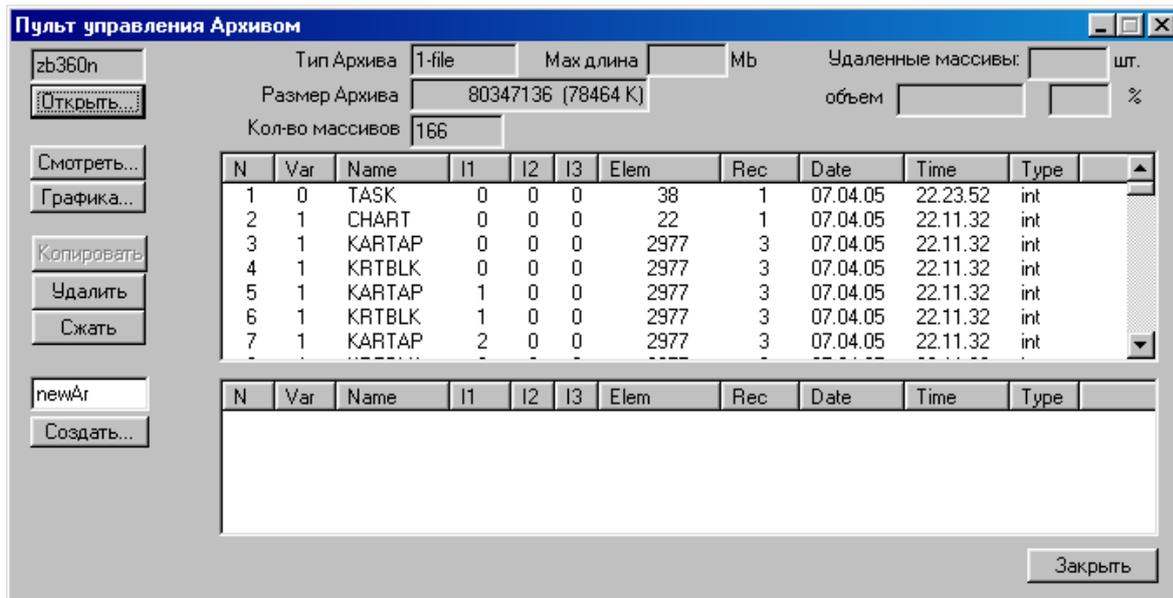
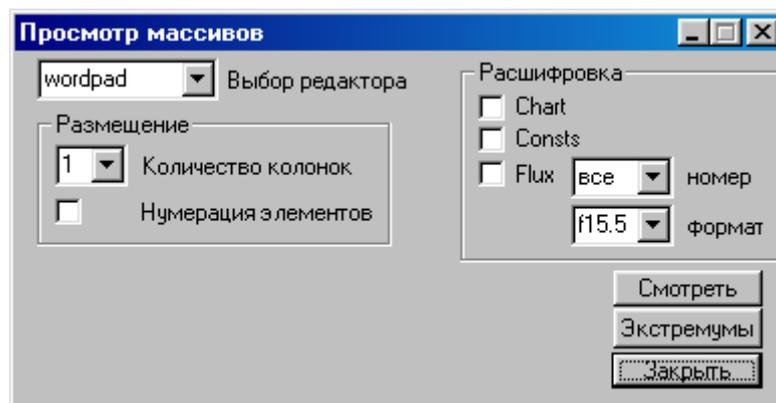


Рис. 4.1. Основная панель Редактора Архива

Некоторые управляющие элементы основной панели вызывают вспомогательные инструменты.

4.2. Просмотр содержимого массива

Для просмотра содержимого массивов необходимо вызвать панель «Просмотр массивов». Нажмем для этого кнопку «Смотреть...». На экране появится требуемая панель:



Для просмотра массива необходимо пометить интересующий массив (помечаем мышкой номер массива, находящийся в самой левой колонке с именем N) и нажать кнопку «Смотреть».

Управляющие элементы панели «Просмотр массивов» позволяют выбрать требуемое представление просматриваемого массива. Например, можно выбрать редактор, количество колонок, признак нумерации элементов, формат представления чисел и режим.

Возможны три режима просмотра содержимого массива: *простой просмотр*, *просмотр с дешифрацией* и *просмотр экстремумов*.

В режиме простого просмотра вызывается текстовый редактор, и в окне редактора появляются числовые значения выбранного массива. Можно указать количество колонок вывода и/или необходимость нумерации элементов массива.

В режиме просмотра экстремумов в окне редактора появляются только экстремальные значения массива. Если мы ищем экстремумы в массиве типа FLUX, то экстремумы определяются для каждой плоскости по отдельности, и найденные экстремумы сопровождаются координатами элемента сетки, в которой экстремум был найден:

Режим просмотра с дешифрацией может потребовать довольно сложной обработки архивных массивов. Например, для массива типа FLUX производится разбивка совокупности элементов массива на плоскости, пометка элементов каждой плоскости координатами и, возможно, вычленение отдельных плоскостей. На следующем скриншоте приведен пример окна редактора, содержащего отдельную плоскость массива:

```
# Archive: C:\VORON2005\CONVERTER\TEST-NEW2x2\zb360n
# Array: FLUX3R Type=rea Var= 1 Ind1= 7 Ind2= 0 Ind3=
# Array Length: 190528 elements

# Plane Length: 2977

# Plane: 13 -----
      1      1      0.52075E+00
      2      1      0.68414E+00
      3      1      0.70208E+00
      4      1      0.72955E+00
      5      1      0.72297E+00
      6      1      0.73380E+00
      7      1      0.72611E+00
```

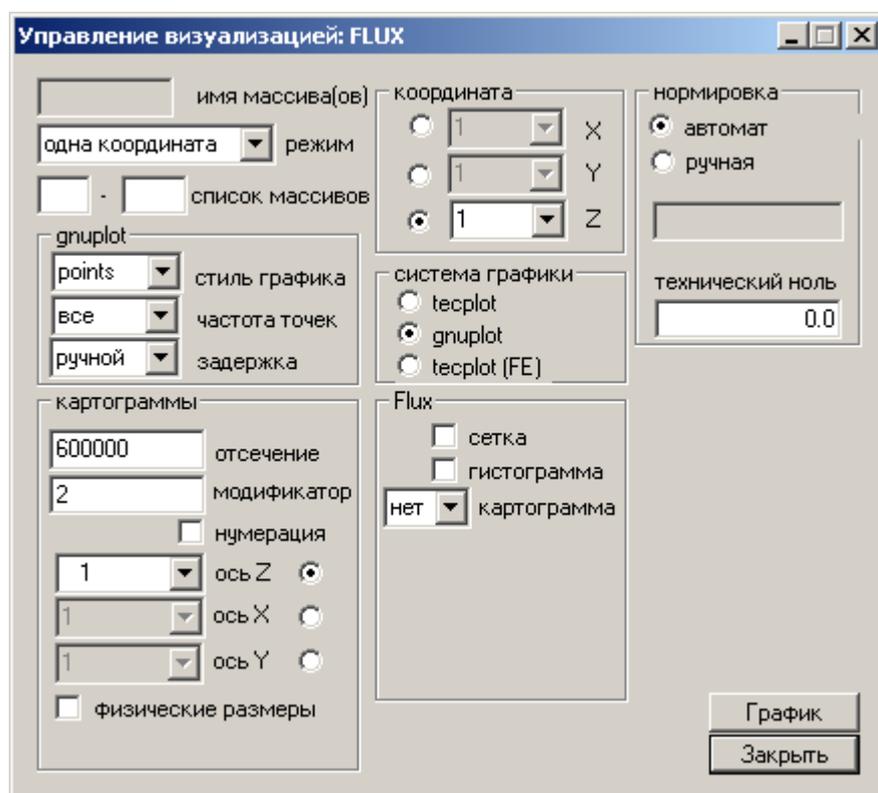
Первые три строчки выдачи описывают общие характеристики массива (эти строки выдаются во всех режимах просмотра):

- полное имя архивного файла;
- имя массива и его идентификационные параметры (тип, номер варианта, индексы);
- длина массива (количество элементов массива).

Далее следуют две строки, специфичные для режима дешифрации массива типа FLUX: длина плоскости и номер просматриваемой плоскости.

4.3. Визуализация

В настоящей версии Редактора Архива реализована визуализация массивов типа FLUX (поля) и KARTAP (картограммы). Для того чтобы визуализировать массивы, необходимо вызвать панель «Управление визуализацией». Нажмем кнопку «Графика...» и на экране появится искомая панель:



Управляющие элементы этой панели позволяют управлять выдачей графики на экран или в файл. В частности, позволяют выбрать графическую систему: Gnuplot или Tecplot.

Наличие двух графических систем обуславливается тем обстоятельством, что Gnuplot является простой и «легкой» системой, а Tecplot – мощной и «тяжелой». Поэтому на стадии предварительного анализа результатов счета разумно пользоваться первой системой, быстро просматривая серии графиков; если же какой-то график вызвал интерес, то его можно анализировать, используя всю «мощь» системы Tecplot.

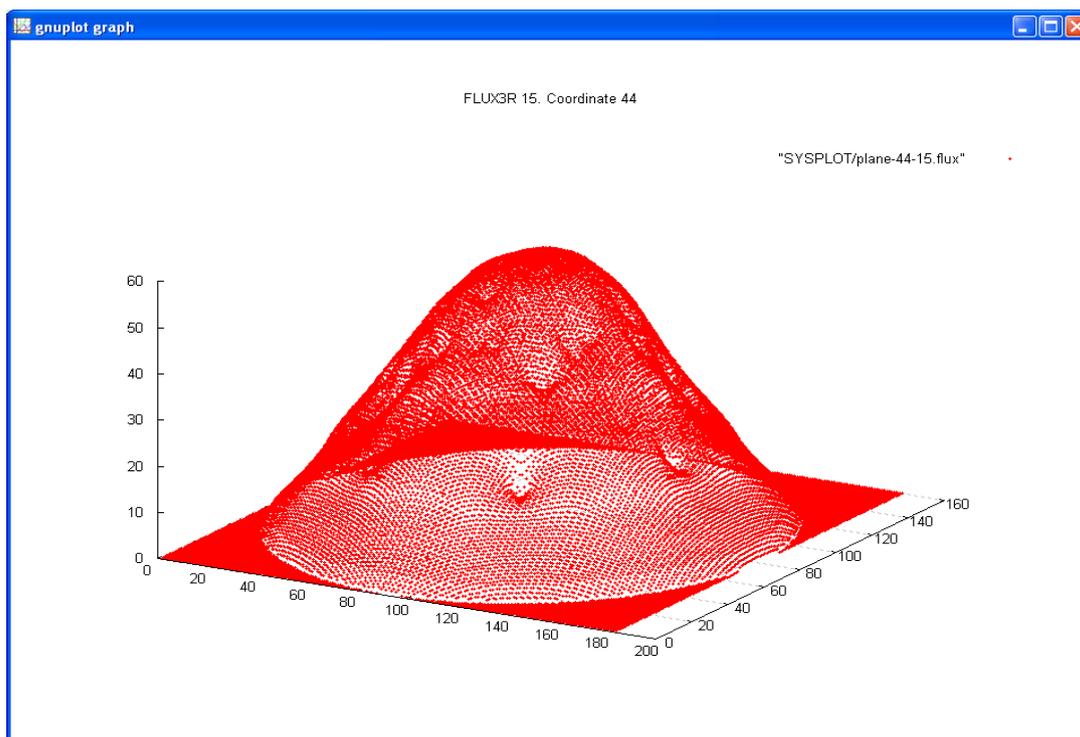
Поскольку система Gnuplot является свободно распространяемым продуктом, а Tecplot – коммерческий продукт, то для сохранения «лицензионной чистоты» реализована следующая логика использования этих инструментов Редактор Архива.

Если выбрана система Gnuplot (пометкой соответствующего флажка), то Редактор Архива автоматически вызывает Gnuplot для визуализации требуемого массива. Если же выбрана система Tecplot, то Редактор Архива ограничивается формированием файла в формате Tecplot, который пользователь должен сам загружать с помощью системы Tecplot.

4.3.1. Использование системы Gnuplot

Начнем рассмотрение возможностей управления визуализацией, используя более простую систему Gnuplot.

Пометим некоторый массив типа FLUX и выберем в поле координаты «Z» некоторое значение (например, 44) и нажмем кнопку «График». На экране появится окно системы Gnuplot, содержащее графическое представление выбранной плоскости выбранного массива:



В этом окне кроме самого графика индицируется

- имя архивного массива (FLUX3R с индексом 15) и номер визуализируемой плоскости (Coordinate 44);
- имя файла, содержащего визуализируемую информацию: “SYSLOT/plane-44-15.flux”.

Заметим, что имя файла однозначно идентифицирует визуализируемый объект.

Изображение можно «развернуть» нужным ракурсом, можно скопировать в буфер (клавиши Alt+PrintScrn) для последующей вставки в отчет или выполнить какие-либо манипуляции с помощью меню Gnuplot, расположенного в левом верхнем углу окна.

4.3.1.1. Стили визуализации

Для управления видом графиков на панели «Управление визуализацией: FLUX» предусмотрены поля «стиль графика» и «частота точек».

Поле «частота точек». По умолчанию значение этого поля «все», что означает – на график выводятся все точки, соответствующие элементам массива. Если же график получился слишком «густой», то можно сделать его более «редким». Для этого установим в поле «частота точек» иное значение. Например, если значение этого поля будет равно «2», то на график будут выдаваться каждый второй элемент массива.

Поле «стиль графика». По умолчанию значение этого поля «points», однако его можно сменить на «lines» и тогда график будет отрисовываться линиями.

4.3.1.2. Нормировка

Для управления нормировкой графиков на панели «Управление визуализацией: FLUX» предусмотрены переключатель «автомат - ручная» и поле «технический ноль».

Переключатель «автомат - ручная». По умолчанию значение этого переключателя - «автомат». В этом случае график строится по точным значениям элементов массива. Если же переключиться в положение «ручная» и ввести в активизированное поле некоторое значение (значение вводится в виде числа с фиксированной точкой), то при построении графика все значения элементов массива будут умножаться на введенное число.

Поле «технический ноль». Используемые графические системы (Gnuplot и Tecplot) требуют (во всяком случае, в некоторых важных режимах), чтобы значения графика были определены в прямоугольной области. Однако область определения значений графика в общем случае не является прямоугольной (она определена массивом LSTR). Поэтому Редактор Архива вынужден «доопределять» строящуюся функцию. В тех точках прямоугольной области, где функция не определена, Редактор Архива использует значение «технический ноль».

По умолчанию это значение равно нулю (0.0), однако в ряде случаев пользователю требуется отличать «настоящий» ноль от технического. Для этого и предназначено обсуждаемое поле. Если в него ввести некоторое значение (значение вводится в виде числа с фиксированной точкой), то при построении графика в тех точках прямоугольной области, где функция не определена, Редактор Архива будет использовать именно это значение.

4.3.1.3. Выбор координаты и плоскости

Выше мы построили плоскость с номером 44 по координате Z. Такой способ построения определяется полем «режим», в котором стоит значение «одна координата» и выбором значений в группе «координата», где переключателем отмечено поле Z, в котором указан номер плоскости.

С помощью полей группы «координата» мы можем выбрать другую требуемую плоскость. Заметим, что Редактор Архива автоматически

определяет число плоскостей по каждой координате и подставляет их списки в соответствующие поля.

4.3.1.4. Режимы демонстрации

Выше мы познакомились с режимом «ручной» (он устанавливается по умолчанию). В этом режиме необходимо выбрать требуемый массив (поставив мышкой), координату и номер плоскости (с помощью соответствующих переключателя и поля). Выбрав нужный массив и нужную плоскость, нажимаем кнопку «График» и получаем на экране требуемую картинку.

Поле «режим» позволяет просматривать серии графиков. Для этого необходимо выбрать один из следующих режимов

- режим «**все координаты**» позволяет просмотреть графики всех плоскостей выбранного массива;
- режим «**все массивы**» позволяет просмотреть графики выбранной плоскости всех массивов.

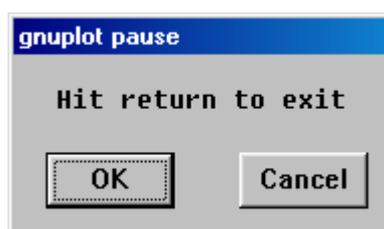
Установив режим «все координаты», поставим мышкой нужный массив и нажмем кнопку «График». На экране появится окно с графиком первой плоскости и панелька «gnuplot pause». Нажав кнопку «ОК», получим окно с графиком второй плоскости и т.д. Процесс прервется либо при исчерпании списка плоскостей, либо при нажатии кнопки «Cancel». Номер просматриваемой плоскости индицируется на графике.

Установив режим «все массивы», необходимо указать номер плоскости (в соответствующем поле группы «координата») и диапазон значения первого индекса массивов (в полях «список массивов»).

Введем, например, в поля «список массивов» значения «20» и «26», а в поле координаты «Z» значение «51». Тогда, нажав кнопку «График» мы получим на экране график 51-ой плоскости 20-ого массива FLUX3R, затем график 51-ой плоскости 21-ого массива и т.д. до 26-ого массива включительно.

Если список массивов в Архиве не «сплошной» (например, отсутствует 23-ий массив FLUX3R), то ничего страшного не произойдет. Редактор Архива после показа 22-ого массива перейдет к показу 24-ого массива.

Поле «задержка». По умолчанию значение этого поля «ручной». Это означает, что в режимах «все координаты» и «все массивы» после предъявления очередного графика появляется панелька «gnuplot pause»:



Нажатием кнопки «ОК» пользователь инициирует предъявление очередного графика; нажатием кнопки «Cancel» прерывает процесс.

Если же в поле «задержка» установить некоторое число, например «1», то панелька «gnuplot pause» появится только после первого предъявления графика. Если пользователь, убедившись, что предъявляется то, что нужно, и настроив ракурс показа, нажмет кнопку «ОК», то дальнейший показ будет автоматическим. При выборе задержки «1» очередной график будет появляться через одну секунду.

4.3.2. Использование системы Tecplot

Как уже упоминалось выше, если с помощью переключателя «система графики» мы переключимся на систему Tecplot, то Редактор Архива ограничится только подготовкой файлов, содержащих данные для построения графиков в формате этой системы.

Tecplot позволяет строить графики двумя способами: «IJ-Ordered Data» и «Finite-Element Data». В соответствии с этим переключатель «система графики» имеет два положения:

- tecplot для режима «IJ-Ordered Data»
- tecplot(FE) для режима «Finite-Element Data»

4.3.2.1. Режим IJ-Ordered Data

В этом режиме данные для Tecplot готовятся в виде двумерного массива, где индексы I и J являются целочисленными значениями. Трехмерное представление графика Tecplot строит самостоятельно.

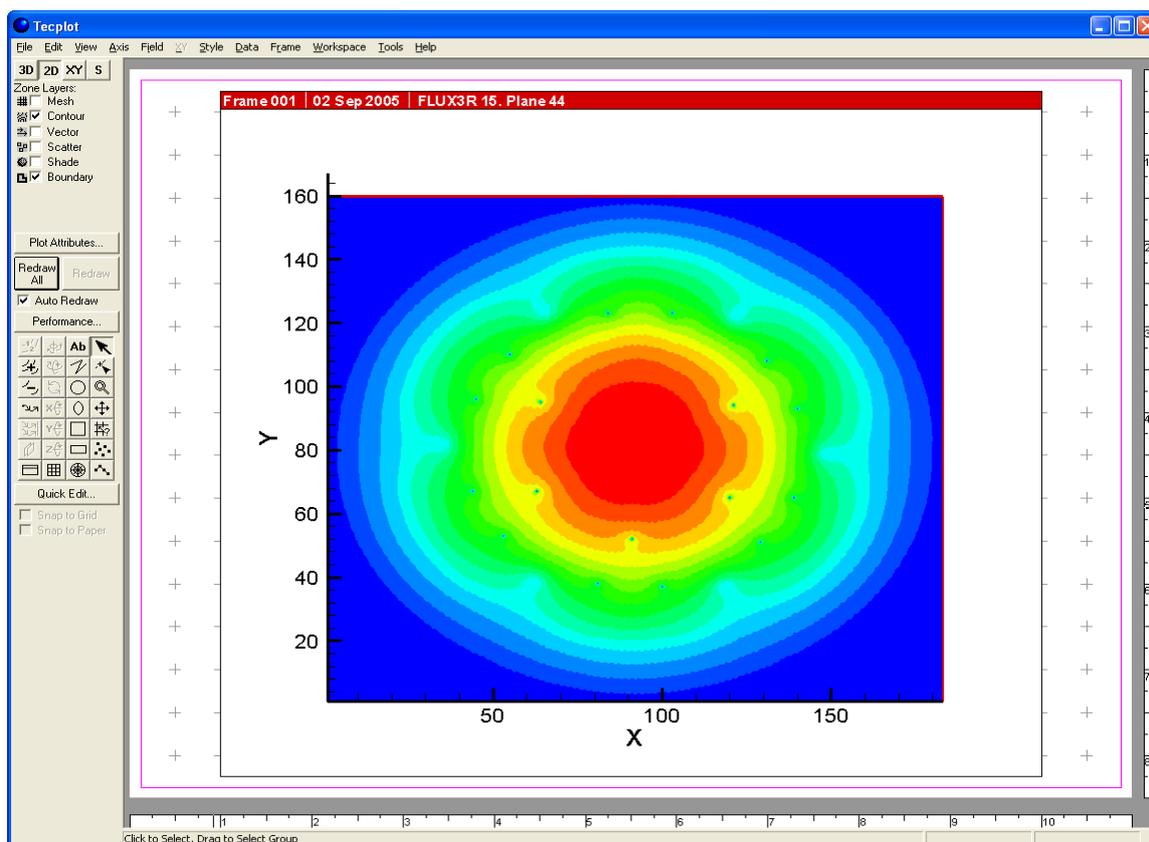
Переключимся в положение «tecplot» и нажмем кнопку «График». Созданный в результате работы Редактора Архива файл находится в каталоге SYSLOT (как, впрочем, и все файлы, предназначенные для визуализации). Имя этого файла указывает, какая именно плоскость подготовлена для визуализации. Пусть, например, имя файла таково:

plane-44-15.dat

Расшифровка имени: plane - используется координата Z
 44 - номер плоскости (координаты)
 15 - первый индекс архивного файла
 dat - файл имеет формат Tecplot

Запустим систему Tecplot и загрузим указанный файл. На экране появится окно, содержащее двумерное представление искомого графика.

Заметим, что имя визуализируемого файла (вместе со значением первого индекса) и номер плоскости индицируются на титульной строке фрейма:



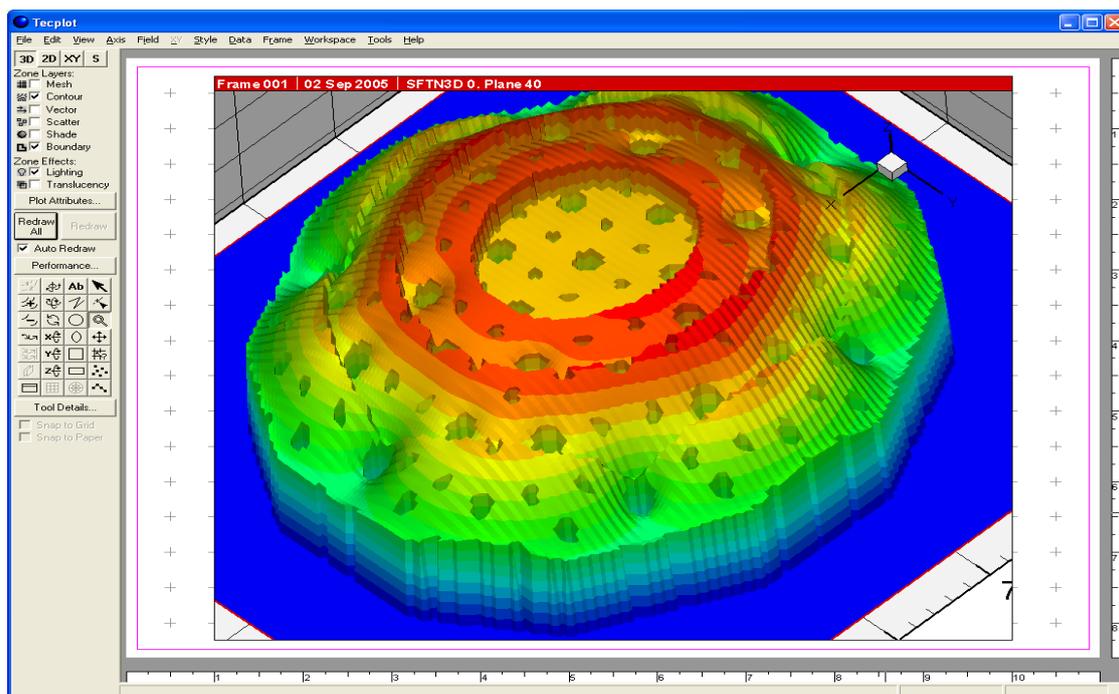
Над построенным графиком можно выполнять все манипуляции, допустимые системой Tecplot – строить линии уровня, менять представление и т.п.

4.3.2.2. Режим Finite-Element Data

В этом режиме данные для Tecplot готовятся в виде коллекции трехмерных точек, вслед за которыми идут инструкции о соединении этих точек для построения поверхности. Для построения поверхности Редактор Архива использует треугольники.

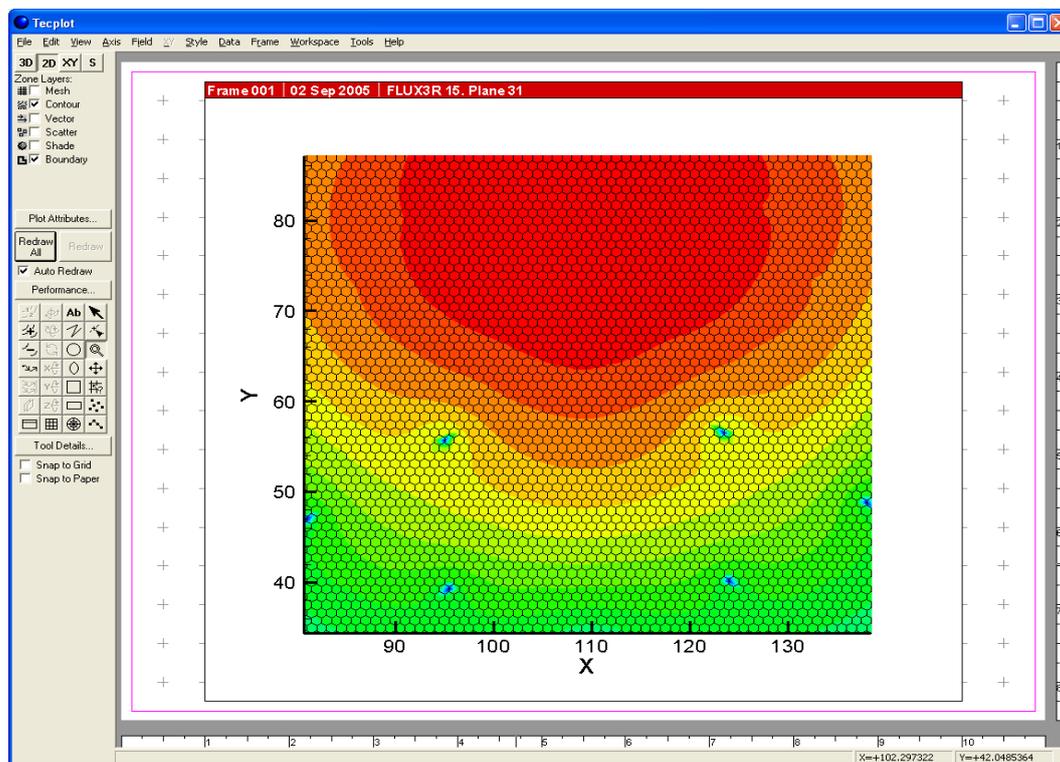
Установим переключатель системы графики на панели «Управление визуализацией: FLUX» в положение «tecplot(FE)». Выберем архивный массив (например, SFTN3D с индексом 0), выберем координату 40 и нажмем кнопку «График».

В каталоге SYSPLOT появится файл planeFE-0-40.dat, содержащий подготовленные данные для визуализации (имя файла – planeFE – указывает на использование режима Finite-Element). Загрузим этот файл в Tecplot и рассмотрим его в представлении «3D»:



На панели управления визуализацией находится группа «Flux», содержащая флажки «сетка», «гистограмма» и поле «картограмма». Эта группа позволяет управлять совмещением графика поля с картограммами.

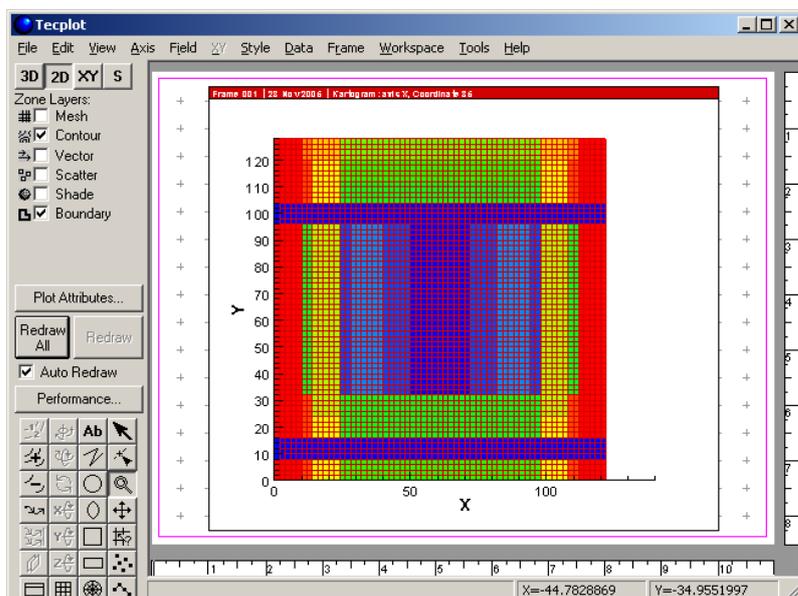
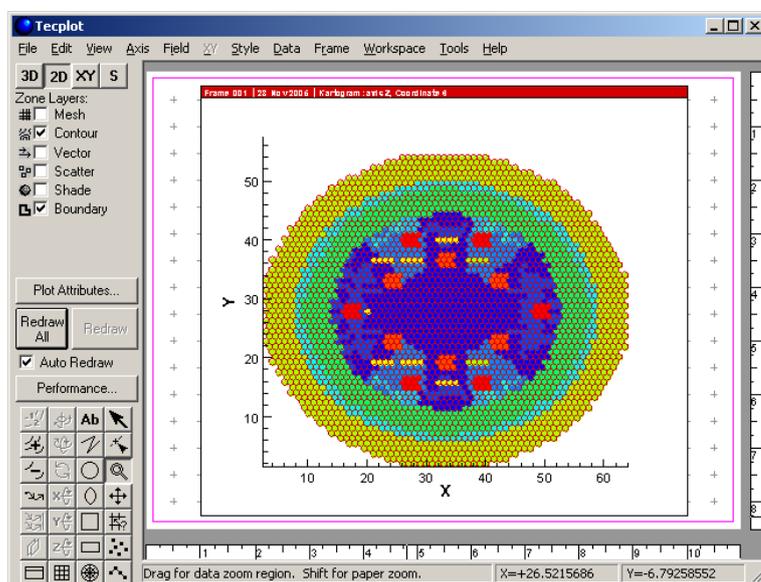
Если, например, установить флажок «сетка», то искомый график будет совмещен с гексагональной сеткой. Например, окно с графиком может выглядеть следующим образом:



Установка флажка «гистограмма» приведет к тому, что графики будут строиться без «сглаживания». А выбор в окне «Картограмма» некоторого значения (в это окно система помещает список всех картограмм, находящихся в Архиве) приведет к тому, что ячейки сетки будут помечены номерами вещества, взятыми из выбранной картограммы.

Для просмотра картограмм необходимо в окне «Пульт управления» пометить соответствующий массив картограммы, выбрать в группе «картограммы» панели «Управление визуализацией» ось просмотра (X, Y или Z) и установить при необходимости флажки «нумерация» (вывод в ячейки картограммы номеров вещества) и «физические размеры» (при построении сетки учитываются реальные размеры ячеек).

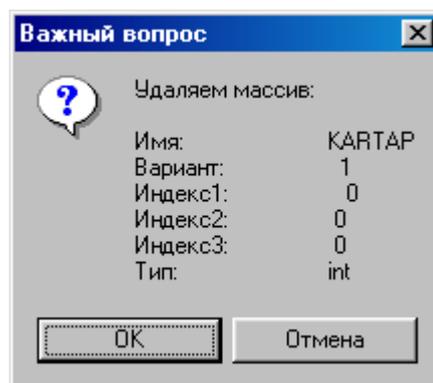
На следующих двух скриншотах приведены примеры картограмм, построенной по осям Z и X (без нумерации и учета физических размеров):



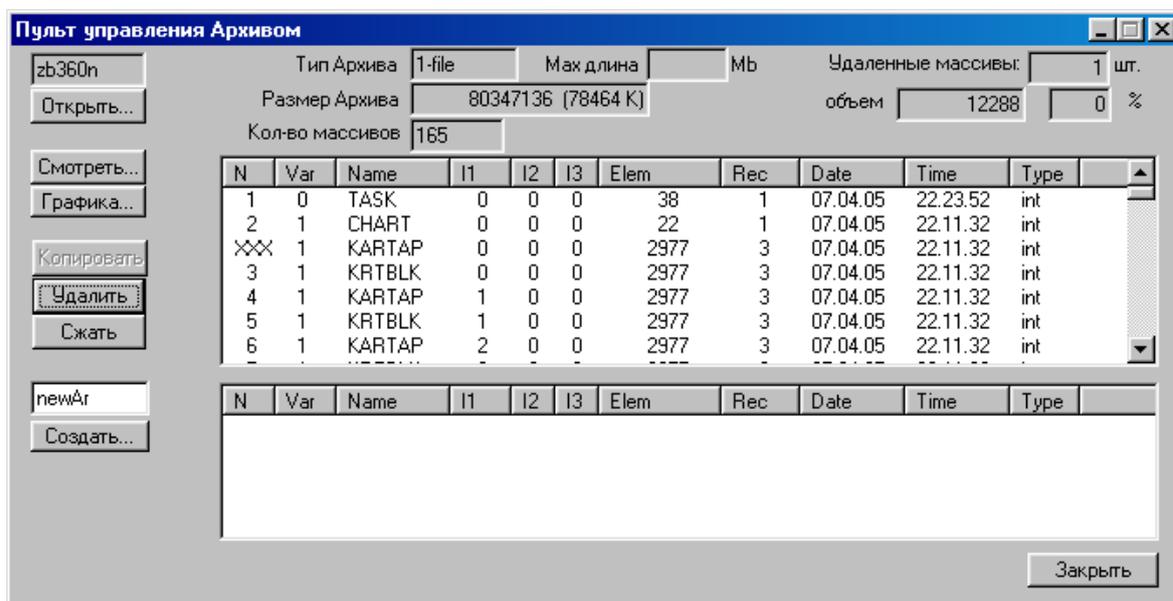
4.4. Удаление массивов и сжатие Архива

Из Архива можно удалить ненужные массивы и тем сократить размер Архива.

Для удаления массива необходимо пометить номер массива (помечаем мышкой самую левую колонку с именем N; пометим, например, массив с номером 3) и нажать кнопку «Удалить». На экране появится сообщение



Если мы нажмем кнопку «OK», то содержимое большого окна главной панели приобретет следующий вид:



Вместо номера выбранного массива появятся литеры «XXX», означающие, что этот массив «вычеркнут» из оглавления Архива.

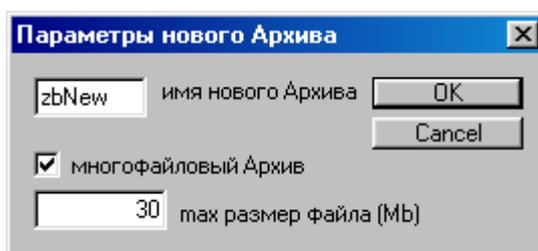
Заметим, что значение поля «Кол-во массивов» сократилось на единицу, а в полях «Удаленные массивы», «Объем» и «%» появились значения, индицирующие соответственно количество удаленных массивов, их суммарный размер (в байтах) и долю от общего размера Архива (в процентах).

Заметим также, что при удалении массивов реального уменьшения размера Архива не происходит – происходит только коррекция оглавления Архива.

Допустим, что мы «вычеркнули» еще несколько ненужных массивов. Теперь мы можем выполнить процедуру сжатия Архива. Для этого нажмем кнопку «Сжать». На экране появится панель параметров нового Архива: в поле «имя нового Архива» мы вводим имя (по умолчанию оно установлено в «XXXXXX»). Имя Архива не должно превышать шести символов.

Флажок «многофайловый Архив» определяет – будет ли новый Архив одно- или много-файловым. Если флажок установлен, то в поле «max размер файла (Mb)» необходимо ввести предельную величину отдельного файла Архива (в мегабайтах).

Допустим, мы хотим, чтобы новый сжатый Архив имел имя «zbNew» и был бы много-файловым с предельным размером одного файла в 30 мегабайт. Введем необходимые параметры в поля панели:



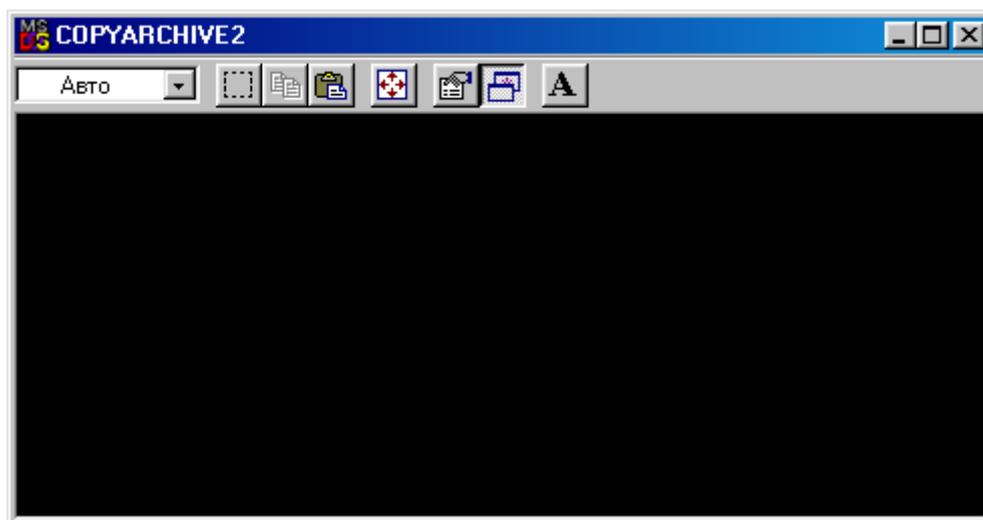
и нажмем кнопку «ОК».

Если Архив с указанным именем уже существует, то на экране появится сообщение об этом и никаких действий произведено не будет.

Если же указанное имя «свободно», то будет создан новый Архив, и теперь окно главной панели будет индексировать оглавление нового сжатого Архива.

Замечание

Отметим следующее обстоятельство. Сжатие Архива (а равно все остальные манипуляции с Архивом) осуществляется с помощью «рабочих утилит» – программ, написанных на Фортране и вызываемых Редактором Архива в виде независимых процессов. Такие процессы порождают свои собственные окна. Обычно появление и исчезновение этих окон «ненаблюдаемо», поскольку время выполнения утилит мало. Однако процесс сжатия большого Архива может занять какое-то время. В течение этого времени на экране «висит» окно рабочей утилиты:



Не следует обращать на него внимания, поскольку оно исчезнет при завершении работы утилиты.

Аналогичные окна появляются и висят на экране при выполнении и других «тяжелых» операциях, например, при визуализации больших массивов.

Кнопку «Сжать» можно использовать для изменения представления Архива. Например, если мы хотим много-файловый Архив преобразовать в одно-файловый, то для этого нужно открыть много-файловый Архив, нажать кнопку «Сжать», на появившейся панели «Параметры нового Архива» сбросить флажок «многофайловый Архив» и нажать кнопку «ОК».

Аналогично можно превратить одно-файловый Архив в много-файловый или изменить размеры файлов много-файлового Архива.

4.5. Создание нового Архива

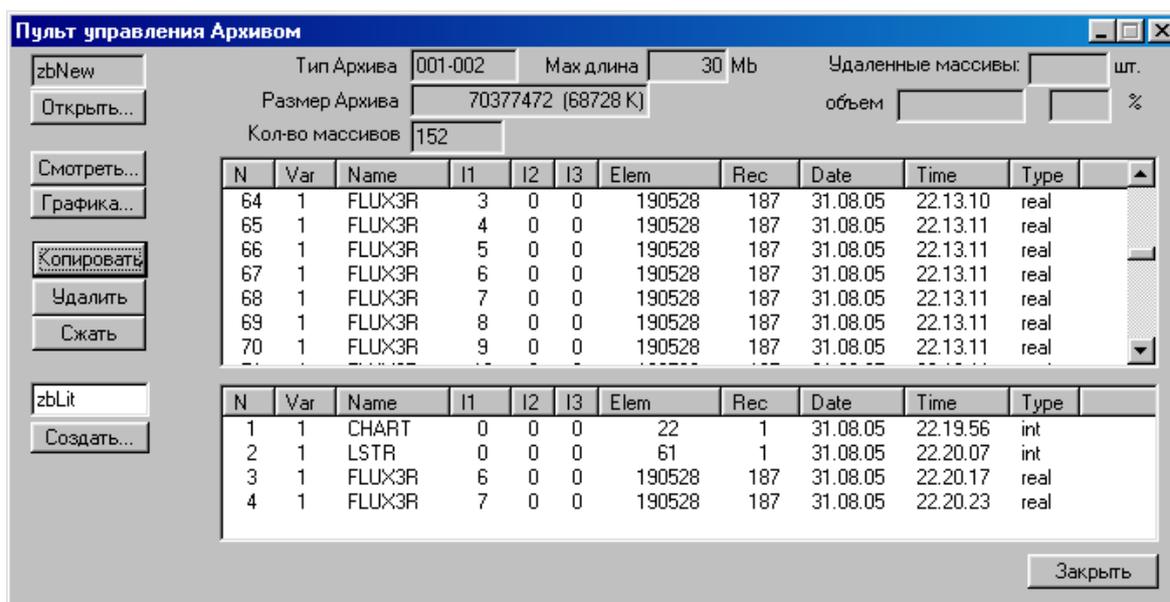
Редактор Архива позволяет создавать новые Архивы и копировать в них массивы из уже существующего Архива.

Введем в поле, расположенное над кнопкой «Создать...», имя нового Архива (например, zbLit) и нажмем эту кнопку. На экране появится уже знакомая панель «Параметры нового Архива». Введем необходимые параметры (например, сбросим флажок «многофайловый Архив» – мы хотим создать одно-файловый Архив) и нажмем кнопку «ОК».

В том же каталоге, в котором находится просматриваемый Архив (zbNew), будет создан новый Архив. На панели пульта управления это индицируется тем, что теперь активизировалась кнопка «Копировать».

Пометим какой-либо массив существующего Архива (в верхнем окне) и нажмем кнопку «Копировать». Видим, что имя этого массива появилось в поле

оглавления созданного Архива zbLit. Продолжая этот процесс, мы отбираем в новый Архив нужные нам массивы:



Можно открыть в верхнем окне другой Архив и скопировать какие-то массивы из этого другого Архива а Архив zbLit и т.д. То есть в созданном Архиве можно «собрать» данные из нескольких различных Архивов.

Заключение

В настоящем документе описана новая версия Архива пакета прикладных программ «РЕАКТОР».

Необходимость разработки новой версии Архива была обусловлена существенным ростом «алгоритмической мощности» пакета «РЕАКТОР», появлением новых вычислительные возможности (распределенная вычислительная среда, суперкомпьютеры), а также возросшими требованиями пользователей (в частности, в отношении необходимости сложного визуального анализа результатов).

Новая версия сохраняет функциональную совместимость со старой версией Архива:

- сохраняются архитектурные принципы построения Архива и логика его функционирования;
- сохраняется функциональность архивной библиотеки;
- «почти» сохраняется синтаксис вызовов архивных подпрограмм.

Новая версия архивных подпрограмм имеет следующие основные отличия от старой версии:

- позволяет использовать в качестве среды хранения архивных данных не только дисковое пространство, но и оперативную память компьютера;

- обеспечивает большую гибкость в настройке структуры Архива на требования счетных модулей;
- обладает меньшей зависимостью от структур данных счетных модулей.

Кроме того, для новой версии Архива разработан и реализован инструмент непосредственного доступа пользователя к содержимому Архива – Редактор Архива.

Создан также инструмент для конвертации «старых» Архивов в «новые» Архивы (автономная утилита «Конвертер»). К сожалению, в данном документе для описания этого инструмента не хватило места.

Несмотря на то что Архив исторически развивался как часть пакета «РЕАКТОР», программы и инструменты работы с Архивом представляют и самостоятельный интерес, так как могут быть использованы совершенно независимо от пакета «РЕАКТОР» в научных проектах, реализуемых на языке Фортран, для создания наглядного и простого способа обмена данными между программно-независимыми модулями, организации счета с продолжением, хранения больших объемов информации во внешней памяти и т. п.

Содержащиеся в данном документе описание библиотеки подпрограмм и Редактора Архива могут служить Инструкцией пользователя для работы с новой версией Архива, в частности, при разработке новых модулей пакета.

Литература

1. A.Voronkov, V.Arzhanov. REACTOR – Program System for Neutron-Physical Calculations. Proc. International Topical Meeting: Advances in Mathematics, Computations, and Reactor Physics, USA, Vol5, April 28 – May 2, 1991.
2. А. В. Воронков, В. И. Аржанов. Принципы построения пакета РЕАКТОР. Препринт ИПМ им. М.В.Келдыша АН СССР, 1995.
3. А. В. Воронков, В. И. Аржанов. Машинно-независимый архив для решения прикладных задач математической физики. Препринт ИПМ им. М.В.Келдыша АН СССР, 1996, № 7, 32 стр.