

20. Алексеев А. В., Беляков И. М., Бочков А. И., Евдокимов В. В., Ириничев Е. П., Морозов В. Ю., Москвин А. Н., Нуждин А. А., Пепеляев М. П., Резчиков В. Ю., Сучкова В. В., Шарифуллин Э. Ш., Шемякина Т. В., Шумилин В. А. Методика САТУРН-2005. Математические модели, алгоритмы и программы решения многомерных задач переноса частиц и энергии // Вопросы атомной науки и техники. Сер. Математическое моделирование физических процессов. 2013. Вып. 4. С. 17–30.

21. Бабанов А. В., Бельков С. А., Бондаренко С. В., Ватулин В. В., Винокуров О. А., Гречишкина И. Н., Змушко В. В., Измайлова Т. Б., Митрофанов Е. И., Рябикина Н. А., Шамраев Б. Н. Методика МИМОЗА-НДЗД. Расчет трехмерных задач спектрального переноса излучения // Вопросы атомной науки и техники. Сер. Математическое моделирование физических процессов. 2012. Вып. 2. С. 64–72.

ПОДСИСТЕМА ДЛЯ ОБРАБОТКИ НЕСЕТОЧНЫХ ДАННЫХ ПО ЗАДАЧЕ И ЕЕ ИНТЕГРАЦИЯ В КОМПЛЕКС ЛОГОС

*М. В. Артамонов, С. В. Величко, В. И. Дерюгин, Н. А. Захарова,
Н. А. Мустаева, А. А. Резяпов*

Российский Федеральный Ядерный Центр –
Всероссийский НИИ экспериментальной физики, г. Саров

В рамках работ по созданию и внедрению отечественного базового программного обеспечения для комплексного имитационного моделирования на супер-ЭВМ во ВНИИЭФ ведется разработка программного комплекса ЛОГОС. Комплекс состоит из различных прикладных программных пакетов для моделирования физических процессов и препостпроцессора «ЛОГОС.ПреПост», который является интерактивным интерфейсом для подготовки расчетной модели, управления счетом и анализа результатов.

Функционально работу пользователя в комплексе (процесс имитационного моделирования) можно разделить на три этапа: подготовка расчетной модели, проведение расчета, обработка результатов расчетов.

Для задания и сопровождения расчета математической задачи необходимы не только данные распределенные на расчетной сетке, но и данные, не привязанные к расчетной сетке, необходимые для задания параметров математической модели и тактики проведения расчета. Например, блок управления счетом, наборы констант (уравнения состояния веществ, константы кинетики), список счетных процессов и их параметров и т. д. Такие компоненты (или **объекты**) представляют собой целостные, логически связанные данные и так же, как и сеточные данные, **нуждаются в сохранении и обработке**. Данный доклад посвящен решению этой задачи.

Так как комплекс ЛОГОС объединяет различные прикладные пакеты и общий препостпроцессор, то необходимо обеспечить интерфейс для обмена данными между ними. В настоящее время есть два основных подхода к созданию файлового интерфейса:

- использование известных форматов (k, yaml, xml, и т. д.);
- разработка своего формата хранения.

Первый подход уже используется, однако, с ростом требований к математическим моделям, известные форматы перестали удовлетворять ряд методик.

1. Общая концепция подсистемы

Для решения обозначенной проблемы были выработаны требования к подсистеме хранения и обозначены ее ключевые свойства:

Ключевой доступ. Каждому объекту при создании сопоставляется уникальный текстовый ключ, используемый для его идентификации. При работе с параметрами объекта данные могут запрашиваться по ключам, предоставляя удобный доступ к ним без необходимости знать внутреннюю организацию и порядок их хранения.

Самодокументированность объекта. Объект хранит в себе блок служебной информации (*дескриптор*), содержащий информацию обо всех элементах данных и минимальную информацию о свойствах отображения параметра в сервисных средствах.

Динамичность. Состав данных (структура, типы и значения элементов данных) объекта определяется и может изменяться пользователем произвольным образом (как программно, так и интерактивно). При этом сервисные средства автоматически обеспечивают операции чтения/записи и визуализации/редактирования объекта.

Поддержка иерархии. Объект может иметь дочерние объекты. Это позволяет организовать описание логически связанных данных практически любой сложности.

Поддержка шаблонов. Подсистема предоставляет возможность хранить базу данных описаний различных объектов для использования в качестве шаблонов.

Концепция рассматриваемой подсистемы основана на использовании для хранения компонентов задачи универсальной сущности, содержащей описание формата хранящихся данных и самих данных. Такая сущность предоставляет возможность объединить логически связанные данные и использовать единые инструменты для обработки, в том числе импорта/экспорта в бинарный файл.

Этой сущностью является «*Структура*». Она содержит два основных блока: атрибутивная запись и таблица основных записей. Атрибутивная запись в каждом экземпляре «структуры» может быть только одна, она хранит общую информацию всего экземпляра объекта. Основных записей, как правило, более одной и их количество может изменяться.

Состав объекта «*структура*» представлен на рис. 1.

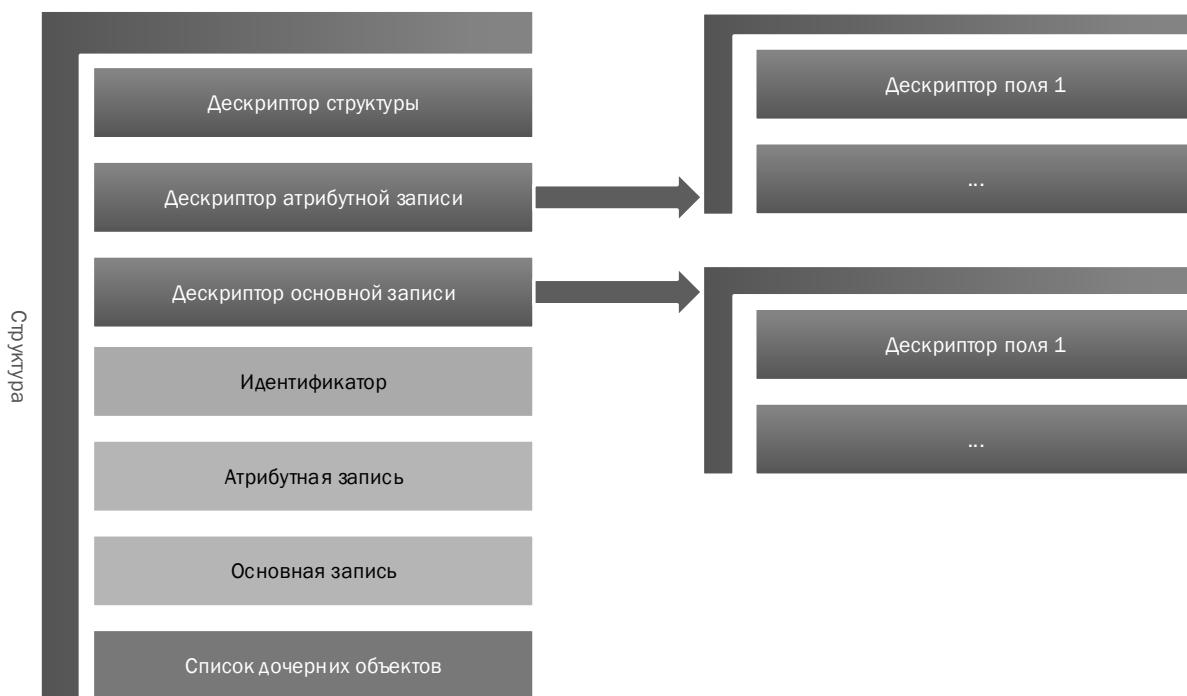


Рис. 1. Состав объекта «структура»

Экземпляры объектов «структура» группируются по ключевому имени типа (например, счетные программы или наборы констант), а совокупность этих групп формируют сущность «задача».

В качестве контейнера для внешнего хранения используется формат EFR [1], который создан во ВНИИЭФ и применяется в ядерных центрах России.

Разработанную подсистему хранения мы назвали библиотекой DBS.

2. Инфраструктура проекта библиотеки DBS

Библиотека рассчитана на разные группы разработчиков, поэтому для ее использования была организована необходимая инфраструктура, а именно: подготовлен проектный файл библиотеки для сборки как для ОС Windows, так и для ОС Linux, настроена система непрерывной интеграции, реализовано два программных интерфейса библиотеки: на языках C++ и Фортран.

Одна из основных задач библиотеки — это импорт и экспорт данных по задаче в файл. Для поддержки совместимости со старыми версиями файлов было введено три уровня версии (три числа). Первый уровень представляет собой версию архитектуры, второй обеспечивает версию интерфейсов библиотеки, а третий обеспечивает версию исправлений внутренних ошибок. Таким образом, файлы, записанные старой версией по второму уровню, должны читаться в новой версии, а записанные старой версией только по третьему уровню, должны читаться и новой версией и более старой.

Организовано модульное тестирование библиотеки. А также организовано автоматическое формирование программных пакетов для передачи разработчикам ЛОГОС.ПреПост и расчетных модулей.

Таким образом, получаем состав проекта библиотеки, представленный на рис. 2.

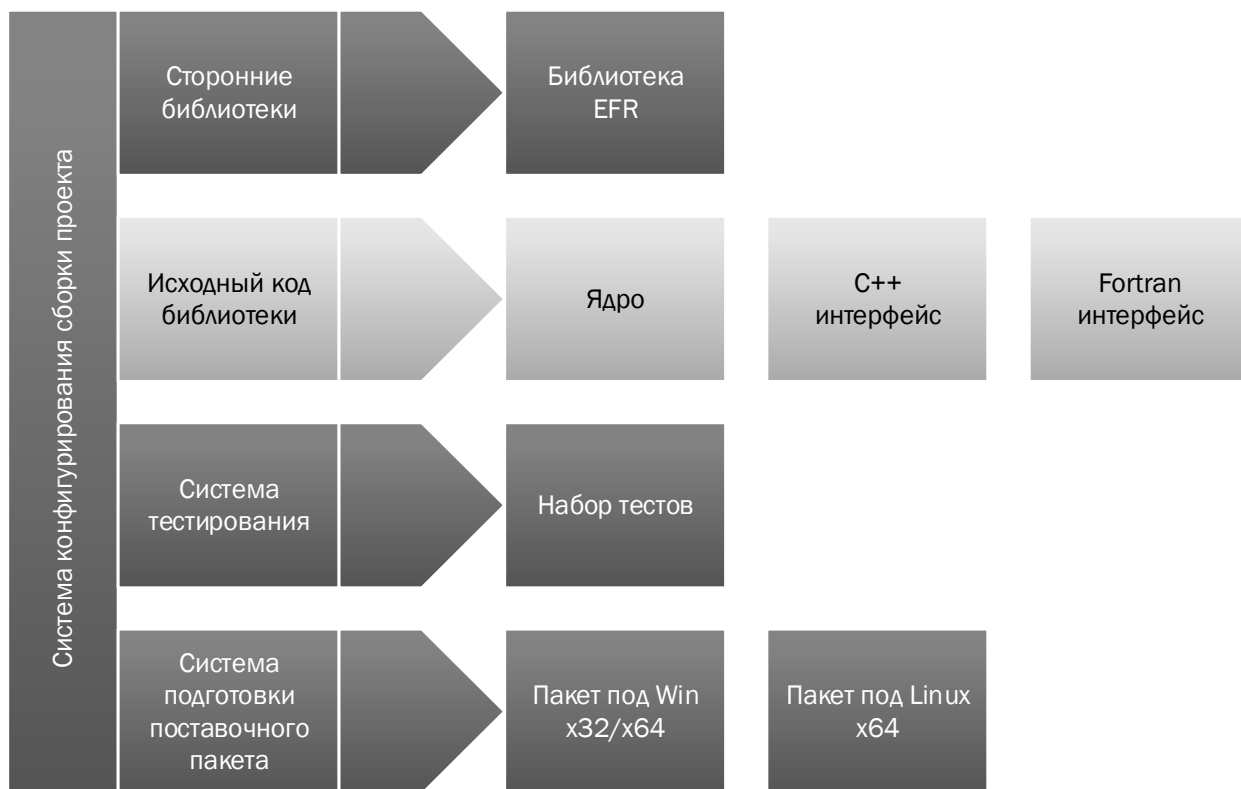


Рис. 2. Состав проекта библиотеки

Библиотека включает в себя ядро и две реализации программных интерфейсов. Два интерфейса потребовались, так как ЛОГОС.ПреПост реализован преимущественно на C++, а расчетные модули – чаще всего на Fortran. На рис. 3 изображена схема обмена данными по задачи между ЛОГОС.ПреПост и расчетными модулями.

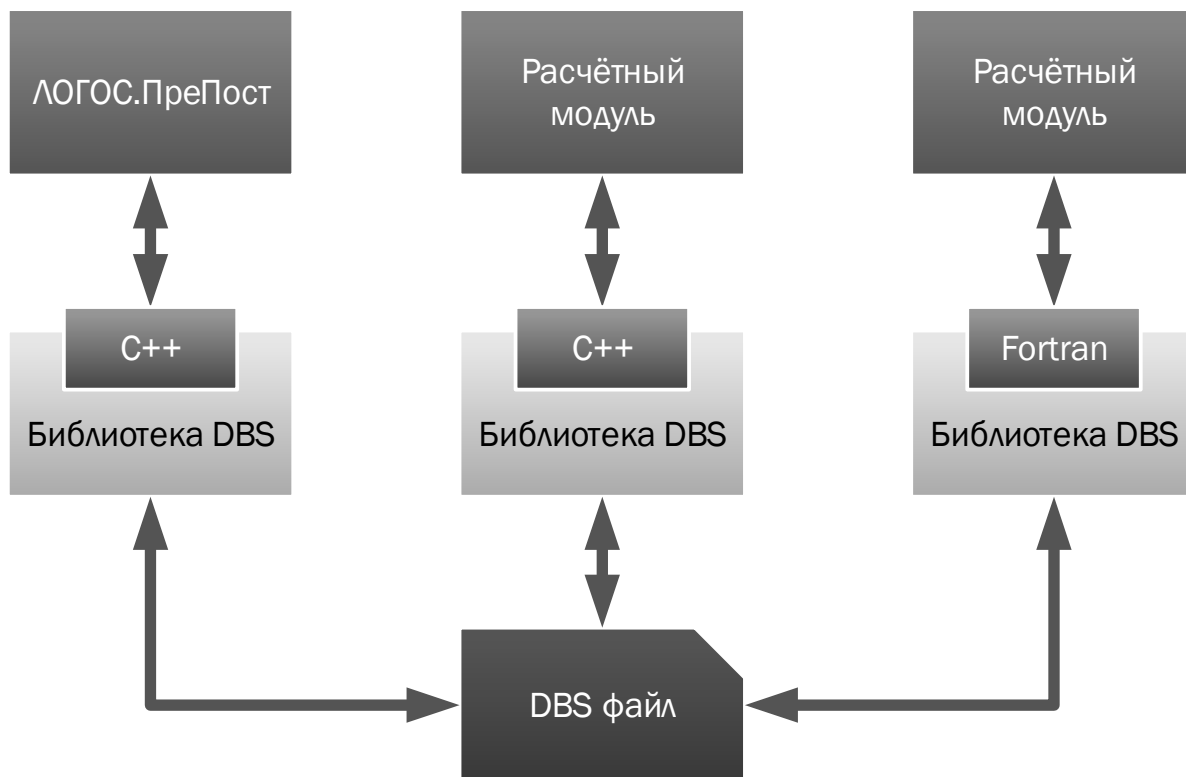


Рис. 3. Схема взаимодействия модулей с библиотекой

3. Интеграция в ЛОГОС.ПреПост

Следующим шагом после создания проекта стала интеграция библиотеки DBS в ЛОГОС.ПреПост. В соответствии с общими правилами сторонних для ЛОГОС.ПреПост библиотек, были подготовлены скрипты автоматического подключения и обновления до актуальной версии библиотеки.

В ЛОГОС.ПреПост имеется своя модель представления несеточных данных, которая была представлена на конференции в 2012 году [2]. В конфигурационном файле задается описание компонент (объектов ЛОГОС.ПреПост), по нему в препостпроцессоре отображается графический интерфейс для работы с данными и осуществляется импорт/экспорт данных.

С учетом многолетнего опыта работы со средствами импорта/экспорта данных [3] была поставлена задача реализовать общий, **универсальный алгоритм** чтения/записи несеточных данных формата DBS для различных объектов (см п. 1), принадлежащих различным моделям. Для этого использовано проверенное решение [3] с использованием конфигурационного файла, в котором описаны правила хранения объекта в файле DBS. Эти правила включают в себя необходимость наличия объекта в файле, а также необходимость конвертирования данных объекта, так как в общем случае объекты, описанные в модели данных ЛОГОС.ПреПост [2], могут отличаться от объектов, необходимых в модели расчетного модуля.

Задачу обмена данными формата DBS решают соответствующие модули импорта и экспорта данных в ЛОГОС.ПреПост. На рис. 4 схематично изображено преобразование данных из объектов модели данных ЛОГОС.ПреПост в модель данных расчетного модуля и обратно.

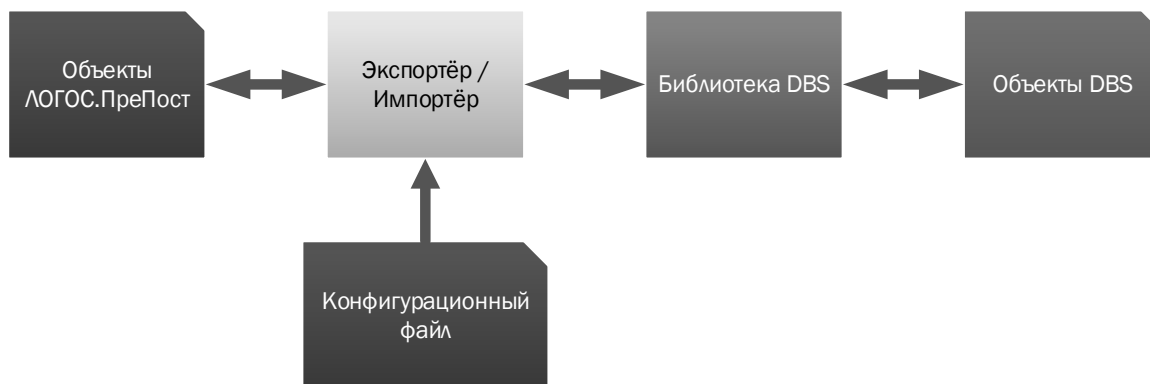


Рис. 4. Схема преобразования объектов моделей данных

Конфигурационный файл содержит информацию о преобразовании ключевых типов структур, условия записи параметров, способ преобразования объектов и другие.

Алгоритм записи объектов ЛОГОС.ПреПост для расчетного модуля, с использованием библиотеки DBS, выглядит следующим образом. Создаем «задачу», в которую будут помещены объекты. Затем создаем пустую «структуру» и по описанию объекта ЛОГОС.ПреПост [2] создаем описание (*дескрипторы*) атрибутной или основной записи. При описании поля указываются имя поля, его уникальный ключ-идентификатор, тип значения, параметры отображения. Далее структура помещается в задачу как шаблон. Теперь по этому шаблону можно создавать структуры с данными и добавлять их в задачу.

После добавления в задачу всех структур вызывается сохранение, при этом библиотека произведет упаковку данных и запишет их в файл.

4. Интеграция в расчетный модуль

Интегрирование библиотеки DBS в Логос-Прочность [4] осуществляется стандартными средствами Фортрана и на текущий момент возможны варианты использования динамических и статических библиотек.

Часть расчетного модуля для работы с объектами DBS-файлов, основные блоки которого схематично представлены на рис. 5, можно описать следующим образом. На основе процедур библиотеки DBS разработан базовый пакет подпрограмм для работы с данными стандартных и произвольных типов данных. Они, в свою очередь, являются основой для реализации подпрограмм инициализации внутренних данных расчетного модуля.



Рис. 5. Иерархия подпрограмм для работы с объектами DBS в Логос-Прочность

На рис. 6 изображена последовательность чтения и инициализации данных для счета. Чтению объектов DBS предшествует чтение из EFR-файла [1] счетных данных (Сетка, узловые и ячеечные величины), а также другая необходимая информация (декомпозиция, версия и т. д.). На этапе чтения объектов DBS-файла производится инициализация следующих данных: параметры счетных программ, тактика счета, наборы констант веществ, интегральные характеристики и т. д.

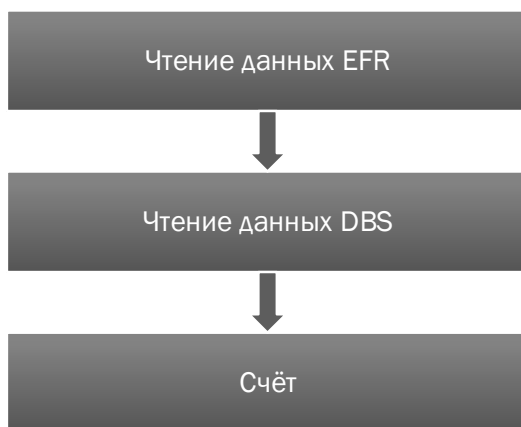


Рис. 6. Последовательность чтения данных для счета

В настоящий момент перечень объектов DBS-файла для нужд Логос-Прочность не является исчерпывающим и будет расширяться, также как и инструментарий для их чтения/записи и инициализации.

5. Заключение

Реализованная подсистема обеспечивает хранение логически связанных данных, предоставляет программный интерфейс для обработки этих данных, в том числе импорта/экспорта в бинарный файл. Подсистема поставляется в виде пакета, готового к подключению в сторонние проекты. В настоящее время текущая версия библиотеки DBS внедрена в ЛОГОС.ПреПост, что позволило подготовить тестовые математические модели для методики Логос-Прочность и передать их расчетный модуль через файл формата DBS, а также загрузить данные в ЛОГОС.ПреПост после работы расчетного модуля.

Литература

1. Олесницкая К. К., Антипин И. А., Петрова М. А. Библиотека ЕФР как средство эффективного доступа к файловым данным на гибридных вычислительных системах и суперкомпьютерах // XV Международная конференция «Супервычисления и математическое моделирование»: сб. трудов. Саров, 13–17 октября, 2015. С. 346–354.
2. Иванов К. В., Анищенко А. А. ЛОГОС.ПреПост. Архитектурные решения на уровне бизнес-логики // XIV Международная конференция «Супервычисления и математическое моделирование»: сб. трудов. Саров, 1–5 октября, 2012. С. 329–333.
3. Дерюгин В. И., Дюпин В. Н., Иванов К. В., Санталов С. А. ЛОГОС.ПреПост. Форматы и структуры данных // XIV Международная конференция «Супервычисления и математическое моделирование»: сб. трудов. Саров, 1–5 октября, 2012. С. 222–225.
4. Авдеев П. А., Александрова О. А., Артемова Е. О. и др. Обзор возможностей моделирования задач прочности с использованием пакета программ ЛОГОС // XV Международная конференция «Супервычисления и математическое моделирование»: сб. трудов. Саров, 13–17 октября, 2015. С. 9–17.