Восьмая Международная Конференция «Крым 2001»

Конференция проводится в рамках мероприятий ИФЛА 2001 г.

Библиотеки и ассоциации в меняющемся мире: новые технологии и новые формы сотрудничества

Тема 2001 года:

Производители и пользователи печатной и электронной информации на пути к информационному обществу

Труды конференции

Том 1

СУДАК

(основная программа)

Ялта, Алушта, Феодосия, Керчь, Старый Крым

(выездные заседания)

Автономная Республика Крым, Украина

9-17 июня 2001

Издательство ГПНТБ России
Москва, 2001

О новой файловой структуре данных CDS/ISIS

On New File Structure of Data in CDS/ISIS

Про нову файлову структуру даних CDS/ISIS

Сбойчаков К.О., Бродовский А.И.
Государственная публичная научно-техническая библиотека России, Москва, Россия
Мазов Н.А., Жижимов О.Л.
Объединенный институт геологии, геофизики и минералогии СО РАН, Новосибирск, Россия

Konstantin O. Sboychakov and Alexander I. Brodovsky
Russian National Public Library for Science and Technology, Moscow, Russia
Nikolai A. Mazov and Oleg L. Zhizhimov
Russian National Public Library for Science and Technology,
Joint Institute of Geology, Geophysics and Mineralogy. Siberian Division of the Russian Academy of Sciences, Novosibirsk, Russia

К.О. Сбойчаков, А.І. Бродовський
Державна публічна науково-технічна бібліотека Росії, Москва, Росія
М.А. Мазов, О.Л. Жижимов
Об'єднаний інститут геології, геофізики і мінералогії СВ РАН, Новосибірськ Росія

Рассматриваются вопросы построения новой файловой структуры для известной системы CDS/ISIS. Обсуждаются недостатки «старой» структуры, достоинства «новой» при ведении баз данных под управлением CDS/ISIS. Приводится структура основных файлов системы, описывается динамическая библиотека процедур и функций для работы с новой файловой структурой системы.

The problems of new file structure design in the known system CDS/ISIS are considered. The disadvantages of «old» structure and advantages of the «new» one are considered within the CDS/ISIS databases support. The structure of system main files is introduced and the dynamic procedures and operational functions library intended to work with new system file structure are proposed.

Розглядаються питання побудови нової файлової структури для відомої системи CDS/ISIS. Обговорюються недоліки «старої» структури, переваги «нової» при введенні баз даних під управлінням CDS/ISIS. Наводиться структура основних файлів системи, описується динамічна бібліотека процедур і функцій для роботи з новою файловою структурою системи.

В качестве системы управления базами данных (СУБД) для обработки информационно-библиотечных массивов во многих организациях как в России, так и за рубежом используется некоммерческая система

CDS/ISIS [1]. На основе этой СУБД разработаны различные автоматизированные информационные и библиотечные системы, существенно расширяющих возможности базовой системы CDS/ISIS [2 — 4], одной из известных и наиболее распространенных в настоящее время является система ИРБИС. Для многих организаций эта система стала основным программным средством ведения баз данных научно-технической информации, электронных каталогов их библиотек, а также для решения других технологических задач. Однако, существующая структура и организация файловой системы CDS/ISIS не вполне позволяет решать те задачи, которые стоят сегодня перед библиотеками и информационными органами, ввиду различных ограничений, накладываемых системой. Тем не менее, до настоящего времени практически не существует систем, способных конкурировать с CDS/ISIS в плане функциональности. В силу ряда обстоятельств, версия системы для DOS и Windows в ЮНЕСКО с конца 1998 года не развивается. Хотя эта ниша и заполнена разработанной в BIREME (Бразилия) [5, 6] полнофункциональной динамически загружаемой библиотеки ISIS.DLL для создания Windows приложений и CGI-приложение WWWISIS.EXE для работы с базами данных CDS/ISIS, однако эти продукты также ориентированы на работу со старой версией файловой системы и следовательно они не смогут снять всех тех ограничений, с которыми в настоящее время сталкиваются специалисты, эксплуатирующие базы данных системы CDS/ISIS. В связи с этим возникает практическая потребность пересмотреть существующую файловую структуру на предмет удовлетворения требованиям сегодняшнего дня и снятия практически всех ограничений, присущих версии системы для DOS, с учетом накопленного многолетнего опыта эксплуатации системы.

Ниже приведены новые структуры файлов системы.

Запись данных основного файла (.MST).

Запись данных основного файла является записью переменной длины (по структуре подобна записи ISO-2709), состоящей из трех частей: лидера записи, справочника и собственно полей данных переменной длины.

В таблице 1. показана структура лидера записи, который имеет фиксированную длину и состоит из следующих целых чисел:

Таблица 1

Содержание	Число бит	Пояснение
MFN	32	номер записи файла документов
MFRL	32	длина записи (всегда четное число)
MFB_LOW	32	ссылка назад (младшее слово)
MFB_HIGH	32	ссылка назад (старшее слово)
BASE	32	смещение (базовый адрес)
NVF	32	Число полей в записи
STATUS	32	статус записи

Как видно из таблицы, смещение на предыдущий вариант записи имеет длину 6 байт, **MFB_LOW** — младшее слово и **MFB_HIGH** — старшее полуслово в полном смещении к записи.

В поле статуса записи **STASTUS** в настоящее время задействованы следующие значения битов:

BIT_ALL_ZERO (0) — предыдущий вариант записи;
BIT_LOG_DEL (1) — логически удаленная запись;
BIT_NOTACT_REC (8) — неактуализированная запись;
BIT_LAST_REC (32) — последний экземпляр записи.

Общая длина лидера записи содержит 28 байт.

В таблице 2. показана структура элемента справочника записи для конкретного появления поля. Количество элементов справочника определяется полем **NVF**, определенного в лидере записи. Элемент справочника имеет фиксированную длину и состоит из трех целых чисел.

Таблица 2

Содержание	Число бит	Пояснение
TAG	32	метка поля записи
POS	32	смещение на позицию первого символа данного поля в разделе полей переменной длины (в физической записи смещения упорядочены по возрастанию, у первого поля значение POS = 0)
LEN	32	длина поля в байтах

Таким образом, общая длина справочника записи в байтах будет 12 * **NVF** байт, а поле **BASE** в лидере записи всегда будет равно 26 + 12 * **NVF**.

Поля данных переменной длины помещаются сразу за справочником в порядке, указанном в нем, одно за другим без разделителей.

Управляющая запись основного файла (.MST).

Первая запись в основном файле записей — управляющая запись, которая формируется в момент определения базы данных или при ее инициализации. Содержимое этой записи поддерживается автоматически. В таблице 3. представлено ее содержание.

Таблица 3

Содержание	Число бит	Пояснение
CTLMFN	32	Резерв
NXTMFN	32	номер записи файла документов, назначаемый для следующей записи, создаваемой в базе данных
NXT_LOW	32	младшее слово смещения на конец файла MST
NXT_HIGH	32	старшее слово смещения на конец файла MST
MFTYPE	32	смещение (базовый адрес)
RECCNT	32	Число полей в записи
MFCXX1	32	Резерв
MFCXX2	32	Резерв
MFCXX3	32	индикатор блокирования базы данных (> 0 — база данных блокирована, =0 — нет)

Таким образом управляющая запись содержит 36 байт.

Запись данных основного файла (.XRF).

Во время создания и/или модификации записей основного файла документов вычисляется индекс, который определяет позицию каждой записи. Индекс хранится в файле перекрестных ссылок (с расширением **XRF**). Таким образом, самым файл перекрестных ссылок представляет собой таблицу индексов на записи основного файла документов. Первая ссылка в файле соответствует записи файла документов с относительным номером 1, вторая — 2 и т. д. Каждая ссылка состоит из трех полей. В таблице 4. показана структура индекса в файле перекрестных ссылок.

Таблица 4

Содержание	Число бит	Пояснение
XRF_LOW	32	смещение на запись (младшее слово)
XRF_HIGH	32	смещение на запись (старшее слово)
XRF_FLAGS	32	индикатор записи (флаги)

В настоящей версии задействованы следующие биты в индикаторе записи **XRF_FLAGS**:

BIT_LOG_DEL (1) — логически удаленная запись;

BIT_PHYS_DEL (2) — физически удаленная запись;

BIT_ABSENT (4) — несуществующая запись;

BIT_NOTACT_REC (8)- неактуализированная запись;

BIT_LOCK_REC (64)- заблокированная запись.

Далее рассмотрим технику обновления базы данных.

Новые записи всегда добавляются в конец основного файла документов с позиции, которая определяется размером файла документов. Присваиваемый номер записи файла документов выбирается из поля **NXTMFN** управляющей записи. При добавлении записи **NXTMFN** возрастает на 1. Кроме этого, создается новая ссылка на эту новую запись в файле перекрестных ссылок с флагами — **BIT_NEW_REC** *or* **BIT_NOTACT_REC**. Поле **STATUS** новой записи в файле документов имеет значение **BIT_LAST_REC**. Флаг **BIT_NOTACT_REC** указывает на то, что новая запись должна быть в дальнейшем затем проинвертирована.

При модификации запись всегда записывается в конец файла документов с позиции, которая определяется размером файла документов. Поле **STATUS** последней версии записи в файле документов имеет значение **BIT_LAST_REC** *or* **BIT_NOTACT_REC**. Поле **STATUS** старой версии записи в файле документов обновляется и становится равным **BIT_ALL_ZERO** *or* **BIT_NOTACT_REC**. Кроме этого, создается новая ссылка на эту новую версию записи в файле перекрестных ссылок с флагом — **BIT_NOTACT_REC**. Ссылка назад в новой версии записи — поля **MFB_LOW, MFB_HIGH** — указывает на предыдущую версию записи, не зависимо от того, была ли старая версия записи проинвертирована.

Для того чтобы иметь возможность отката после проведения инвертирования записи, в отличие от стандартной системы CDS/ISIS ссылка назад **MFB_LOW**, **MFB_HIGH** не обнуляется. После инвертирования флаг **BIT_NOTACT_REC** обнуляется.

Удаление записи рассматривается как модификация со следующими дополнительными установками для этой записи:

В основном файле документов в поле **STATUS** записи взводится флаг **BIT_LOG_DEL**;

В файле перекрестных ссылок в поле **XRF_FLAGS** записи взводятся флаги **BIT_LOG_DEL** и **BIT_NOTACT_REC**.

Как можно было увидеть выше, при модификации записей файла документов его объем возрастает и возможны потери дисковой памяти, которую нельзя использовать в дальнейшем. Средство реорганизации позволяет сжать файл документов.

Во время фазы копирования файла документов создается файл копии *(.BKP)*, который в дальнейшем может быть использован при восстановлении основного файла документов. Структура и формат файла копии те же, что и для основного файла документов *(.MST)*, за исключением того, файл перекрестных ссылок не требуется, так как все записи размещаются последовательно без пропусков. Записи, помеченные как удаленные, в файл копии не записываются. Так как только последняя версия записи записывается в файл копии, выполнить операцию копирования невозможно, если в базе есть хотя бы одна неинвертированная запись.

Во время фазы восстановления основного файла документов последовательно читается файл копии и создается основной файл документов *(.MST)* и файл перекрестных ссылок *(.XRF)*.

Далее будет рассмотрена структура инверсного файла. В новой структуре инверсный файл состоит из трех физических файлов, два из которых содержат словарь поисковых терминов (в структуре бинарного дерева) и третий содержит список ссылок, соответствующих каждому термину.

В бинарном дереве файл с расширением **N01** содержит узлы дерева и файл с расширением **L01** — листья. Записи с листьями указывают на файл ссылок **IFP**.

Физически взаимосвязи между файлами — **N01** и **L01**- обеспечиваются ссылками, которые представляют собой относительные адреса соответствующих записей. Относительный адрес это порядковый номер записи в данном файле. Структура записи одинакова для **N01** и **L01** файлов. Размер (длина) записи зависит от реализации (512; 1024; 2048; 4096). Адрес корневой записи файла **N01** сохраняется как номер первой записи. Смещение на запись в файле **IFP** сохраняется в файле **L01** и имеет длину 8 байт.

Эти файлы содержат в себе индексы словаря поисковых терминов и состоят из записей (блоков) постоянной длины. Записи состоят из трех частей: лидера, справочника и ключей переменной длины. В таблице 5. показана структура лидера в файле N01, L01.

Таблица 5

Содержание	Число бит	Пояснение
NUMBER	32	номер записи (начиная с 1) в N01 номер первой записи равен номеру корневой записи дерева
PREV	32	номер предыдущей записи (если нет = -1)
NEXT	32	номер следующей записи(если нет = -1)
TERMS	16	число ключей в записи
OFFSET_FREE	16	смещение на свободную позицию в записи (от начала записи)

Справочник это таблица, определяющая поисковый термин. Каждый ключ переменной длины, который есть в записи, представлен в справочнике одним входом

В таблице 6. показана структура справочника.

Таблица 6

Содержание	Число бит	Пояснение
LEN	16	длина ключа
OFFSET_KEY	16	смещение на ключ (от начала записи)
BLOCK	32	ссылка на запись файла N01 или файла L01
FLAG	32	смещения на ссылочную запись в IFP

Следует заметить, что поле **BLOCK** — в **N01** файле это ссылка на запись файла **N01** (если **BLOCK** > 0) или файла **L01** (если **BLOCK** < 0), у которой 1-й ключ равен данному. Положительное значение **BLOCK** определяет ветку индекса иерархически более низкого уровня. Самый низкий уровень индекса (**BLOCK** < 0) соответствует ссылкам на записи (листья) файла **L01**. Поле **BLOCK** — в **L01** файле это младшее слово 8 байтового смещения на ссылочную запись в **IFP**.

Поле **FLAG** — в **N01** файле всегда 0, а в **L01** файле это старшее слово 8 байтового смещения на ссылочную запись в **IFP**.

Ключи переменной длины записываются, начиная с конца записи, так что порядок входов, соответствующих им, определяется алфавитным порядком ключей. Сами ключи располагаются вплотную друг к другу без разделителей в порядке поступления на запись.

Файл ссылок на термины словаря (.IFP).

Файл содержит список ссылок для каждого термина словаря. Список ссылок может быть представлен в двух различных форматах. Выбор формата размещения ссылок осуществляется при загрузке словаря из файла **LK1** (этот файл формируется после отбора и сортировки терминов) в зависимости от общего числа ссылок для данного термина. Обычный формат — это заголовок блока и набор упорядоченных ссылок. При превышении определенного числа ссылок (MIN_POSTINGS_IN_BLOCK — в данной реализации 255) формат включает специальный блок и набор блоков обычного формата размер которых определяется по следующей схеме: блоки 4, 8, 16, 32 Кб для общего числа ссылок соответственно 256-32000; 32000-64000; 64000-128000; 128000 и более. Такая схема оптимизирует работу с диском в процессе инвертирования записи в базах данных, характеризующихся большим количеством ссылок на термин.

Запись состоит из заголовка и упорядоченного набора ссылок.

В таблице 7. рассмотрен формат заголовка инверсного файла в обычном формате.

Таблица 7

Содержание	Число бит	Пояснение
NXT_LOW	32	Младшее слово смещения на следующую запись
NXT_HIGH	32	Старшее слово смещения на следующую запись
TOTP	32	Общее число ссылок для данного термина (в первой записи) или число ссылок в данном блоке
SEGP	32	Число ссылок в данном блоке
SEGC	32	Вместимость записи в ссылках

В таблице 8. рассмотрен формат ссылки инверсного файла в обычном формате.

Таблица 8

Содержание	Число бит	Пояснение
PMFN	32	Номер записи
PTAG	32	Идентификатор поля
POCC	32	Номер повторения
PCNT	32	Номер термина в поле

В этом случае специального формата первой записью является специальный блок, который представляет собой заголовок (обыкновенного формата) и набор входов следующего формата, как показано в таблице 9.

Таблица 9

Содержание	Число бит	Пояснение
POSTING	32	1-я ссылка из записи обыкновенного формата
NXT_LOW	32	Младшее слово смещения на следующую запись (если нет = 0)
NXT_HIGH	32	Старшее слово смещения на следующую запись (если нет = 0)

Записи, на которые ссылается специальный блок связаны между собой как описано выше. При чем общее количество ссылок для данного термина сохраняется только в специальном блоке.

При выполнении актуализации инверсного файла могут создаваться новые дополнительные записи при добавлении новых ссылок. В этом случае создается новая запись размером равным общему количеству ссылок, если нет специального блока, и размером, равным количеству ссылок в данной записи, если есть. Новая запись создается таким образом, чтобы не нарушалась возрастающая последовательность следования ссылок. Новая запись связывается с уже существующими записями посредством поля **NXT_LOW**, ссылки распределяются равномерно между старой и новой записью.

Для реализации описанной структуры авторами разработана специальная динамическая библиотека, реализующая в полном объеме функции нижнего уровня по доступу к базам данных. Отличительной чертой указанной функций этой библиотеки при работе с базами данных, является платформенная независимость на уровне данных.

Библиотека включает все стандартные функции СУБД ISIS кроме функций поиска (SEARCH) и сортировки (SORT), реализованных отдельно. В том числе все специальные функции, необходимые для администрирования баз данных — копирование/восстановление основного файла документов, разгрузка/загрузка словаря и сортировка терминов словаря. Библиотека не использует каких-либо дополнительных классов и рассчитана на использование в многопотоковом режиме в среде Windows. Размер блока файла словаря 2048 байт, что дает среднее значение числа терминов в блоке < 100 и среднее число итераций в алгоритме бинарного поиска $\mathrm{Log}_2(100) = 7$.

Литература

1. Пакет прикладных программ CDS/ISIS/M версия 2.3: Методические материалы и документация по пакетам прикладных программ. Вып. 70. — М.: МЦНТИ, 1991. — 257 с.
2. Бродовский А.И. Программные средства, расширяющие возможности ППП CDS/ISIS и их применение для автоматизации библиотечно-информационных процессов в ГПНТБ России // Науч. и техн. б-ки. — 1995. — № 2. — С. 24 — 34.
3. Web-ориентированная информационно-поисковая система для доступа к базам данных CDS/ISIS / Мазов Н.А., Малицкий Н.А., Баженов С.Р., Жижимов О.Л. // Науч. и техн. б-ки. — 2000. — № 2. — С. 52 — 57.
4. Мазов Н.А., Жижимов О.Л. Интеграция Z39.50 и CDS/ISIS: состояние и перспективы развития // Науч. и техн. б-ки. — 2000. — № 5. — С. 76 — 79.
5. BIREME/PAHO/WHO. ISIS Application Program Interface (ISIS_DLL): V er. 5.0. Sao Paulo, Brasilian, Aug. 1997. [http://www.bireme.br/isis/I/isisdll.htm].
6. BIREME/PAHO/WHO. WWWISIS: Ver. 3.0. Sao Paulo, Brasilian, Oct. 1997. [http://www.bireme.br/isis/I/wwwi.htm].