

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РФ
ГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ
«ДОНСКОЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ
УНИВЕРСИТЕТ»

Кафедра "Программное обеспечение вычислительной
техники и автоматизированных систем"

ОРГАНИЗАЦИЯ АБСТРАКТНОГО БЛОЧНОГО ДИСКОВОГО
ПРОСТРАНСТВА ВНУТРИ ФАЙЛА ДАННЫХ

Методические указания к выполнению лабораторной работы
№5 по дисциплине «Операционные системы и оболочки»

Ростов-на-Дону, 2011 г.

Составитель: к.т.н., доц. Долгов В.В.

Организация абстрактного блочного дискового пространства внутри файла данных: методические указания к выполнению лабораторной работы №5 – Ростов н/Д: Издательский центр ДГТУ, 2011. – 7 с.

В методической разработке рассматриваются принципы организации, учета и управления блочным пространством в файловых системах. Даны задания к лабораторной работе помогающие закрепить на практике полученные знания. Методические указания предназначены для студентов специальностей 010503 "Математическое обеспечение и администрирование информационных систем".

Печатается по решению методической комиссии факультета «Информатика и вычислительная техника».

Рецензент: к.т.н., доц. Гранков М.В.

Научный редактор: д.т.н., проф. Нейдорф Р.А.

© Издательский центр ДГТУ, 2011

Все современные файловых системы (ФС) хранят файлы в виде последовательности блоков фиксированного размера, расположенных в различных частях носителя информации (чаще всего внешнего диска). При такой организации операционная система (ОС) должна знать, какие блоки данных на носителе являются занятыми, а какие – свободными. Классическими способами учета занятости любого пространства, разбитого на блоки фиксированного размера, являются связный список блоков, список цепочек блоков и битовая карта. Инициализация служебных областей, создание битовых карт или списков пустых блоков совместно с начальной структурой всей ФС является одной из причин необходимости т.н. форматирования носителя.

Связный список пустых блоков

При учете пустого пространства с помощью связного списка номеров свободных блоков. Такой список располагается в отдельных, специально предназначенных для этого блоках ФС, в каждом из которых все пространство отведено под номера пустых блоков, а один из номеров (как правило, первый или последний) содержит номер блока, содержащий продолжение списка. Блоки, содержащие такой список, могут располагаться либо в начале ФС, что может потребовать резервирования достаточно большого объема пространства и не является оптимальным, либо располагаться в произвольных блоках ФС (рис. 1). Достоинство произвольного расположения заключается в том, что размер списка (т.е. количество блоков занятых под него) напрямую зависит от количества свободных блоков. Таким образом, общий объем служебной информации в таких ФС подстраивается под их текущее состояние.

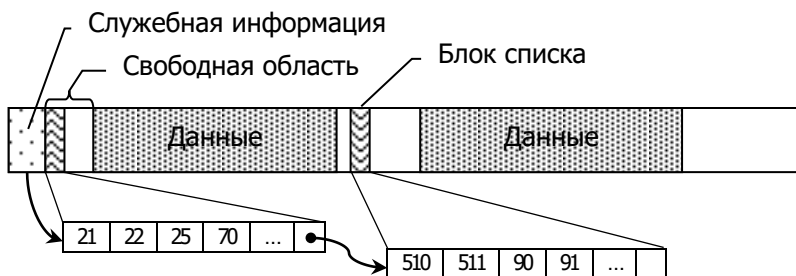


Рисунок 1. Произвольное расположение блоков списка

При таком способе хранения количество ссылок на пустые блоки зависит от размера блока и от возможного их максимального количества. Так, для ФС с количеством блоков не превышающих 2^{24} и размером блока в 4Кб в одном блоке может храниться 1364 ссылки (плюс одна, указывающая на продолжение списка).

Для работы с такой структурой, ФС достаточно (минимально достаточно) держать в оперативной памяти один блок из списка. Когда ФС необходимо выделить пустые блоки под данные файла, их номера выбираются из списка, содержащегося в памяти. При удалении файла его блоки освобождаются и добавляются в список. Когда блок с указателями заполняется полностью, он записывается на диск. При этом перед записью он разбивается на два блока, каждый из которых содержит только половину списка исходного блока. Такая операция позволяет существенно сократить количество обращений к носителю, т.к. остающийся в памяти блок готов как к добавлению в него новых указателей, так и к изъятию из него блоков под новые данные [1].

Связный список цепочек блоков

При использовании связного списка для хранения перечня свободных блоков ФС очень часто возникает ситуация, когда свободными оказываются несколько (иногда довольно много) подряд расположенных блоков. В этом случае целесообразно хранить информацию о них не в виде простого списка, а в виде списка цепочек блоков, описывая каждую из них парой вида «начальный блок»–«количество блоков в цепочке». Т.е., например, вместо последовательности списка «34,35,36,37,38,39,682,683,684,560, ...» записывать пары «(34, 6); (682, 3); ...».

Такой подход может существенно сократить длину списка и уменьшить частоту обращений ФС к носителю для записи/чтения служебной информации.

Битовая карта блоков

Наиболее компактной структурой для хранения информации о занятых/пустых блоках на носителе, несомненно, является битовая карта. Например, в случае, когда количество блоков файловой системы не превышает числа 2^{24} (что для блоков

размером в 4Кб обеспечивает размер ФС в 64Гб), необходимо всего 512 блоков или 2Мб информации, что для современных ФС не является проблемой.

Битовая карта является структурой фиксированного размера, где для каждого блока на носителе отводится ровно один бит. Из-за постоянства размера такая карта, как правило, храниться в специальной служебной области файловой системы (рис. 2) и иногда в целях надежности дублируется.

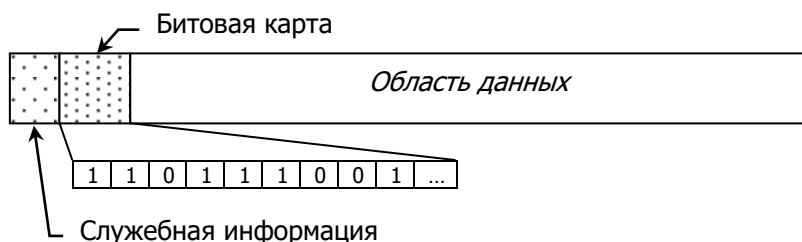


Рисунок 2. Пример расположения битовой карты внутри ФС

ЗАДАНИЕ К ЛАБОРАТОРНОЙ РАБОТЕ

Создать набор библиотечных функций, позволяющих организовать внутри указанного файла блочное пространство с блоками заданного размера. Считать, что размер блока может быть равен степени числа 2 (два), но не менее чем 2^{10} (1 Кб) и не более чем 2^{16} (64 Кб), а количество блоков не будет превышать $2^{32}-1$. Созданная библиотека должна содержать функции для выполнения следующих задач:

- инициализация блочного пространства в указанном файле (входными данными должен служить дескриптор доступа к файлу и размер блока в создаваемом пространстве);
- выделение заданного количества пустых блоков (входом функции служит количество запрашиваемых блоков и/или перечень номеров блоков «желательных» для выделения; выходом – перечень номеров выделенных блоков);
- освобождение ранее выделенных блоков (вход – перечень блоков подлежащих освобождению);

- запись данных в указанные блоки (входом данной функции служит массив данных и список блоков, в которые он должен быть записан, если список блоков пуст, то библиотека должна самостоятельно выделить блоки под данные; результатом работы функции должен быть список блоков, в которые были помещены данные);
- чтение данных из заданных блоков (на вход подается список блоков и область памяти, куда необходимо прочитать данные);
- получение информации о состоянии блочного пространства: размер блока данных; общее количество блоков; количество пустых блоков; размер памяти, занятой под служебную информацию; размер транзакционного кэша;
- начало транзакционной сессии (в процессе работы сессии все изменения, производимые в блочном пространстве должны кэшироваться в оперативной памяти);
- завершение (запись, фиксация) транзакции (при записи транзакции все внесенные изменения должны быть записаны на физический носитель);
- откат транзакции (все внесенные изменения отбрасываются).

Способ хранения информации о свободных блоках формируемого блочного пространства выбрать на основании варианта из таблицы 1.

Таблица 1. Варианты заданий к лабораторной работе №5

№ варианта	Способ хранения информации о свободных блоках
1	Связный список пустых блоков
2	Связный список цепочек блоков
3	Битовая карта блоков

На основании данной библиотеки написать программу, демонстрирующую ее возможности на практике.

КОНТРОЛЬНЫЕ ВОПРОСЫ К ЛАБОРАТОРНОЙ РАБОТЕ

1. Какие способы учета пустых/занятых блоков вы знаете?
2. Почему важно, чтобы изменения, вносимые в служебные области дискового пространства, были выполнены как единое целое?
3. Что дает ограничение количества блоков до $2^{32}-1$?
4. Почему необходима инициализация блочного пространства (форматирование)?
5. В чем плюсы и минусы кэширования всех изменений, производимых в процессе транзакции?
6. Какие способы расположения списка пустых блоков вы знаете?

ЛИТЕРАТУРА

1. Таненбаум Э. Современные операционные системы. 2-е изд. – СПб.: Питер, 2002. – 1040 с.: ил.
2. Дейтел Г. Введение в операционные системы: В 2-х томах. Пер. с англ. – М: Мир, 1987. – 359с.

Редактор А.А. Литвинова

ЛР № 04779 от 18.05.01.	В набор	В печать
Объем 0,5 усл.п.л., уч.-изд.л.	Офсет.	Формат 60х84/16.
Бумага тип №3.	Заказ №	Тираж 140. Цена

Издательский центр ДГТУ
Адрес университета и полиграфического предприятия:
344010, г. Ростов-на-Дону, пл. Гагарина, 1.