



Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет
имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ Информатика и системы управления

КАФЕДРА Программное обеспечение ЭВМ и информационные технологии

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №2 **По дисциплине «Типы и структуре данных»**

Название: Длинная арифметика

Студент Дремин Кирилл

Группа ИУ7 – 36Б

Тип лабораторной работы: Учебная

Вариант № 1

Преподаватель Барышникова Марина Юрьевна

Описание условия задачи

Необходимо создать таблицу, заполненную не менее чем 40 записями с вариантной частью. Требуется произвести поиск информации по заданному вариантному полю, упорядочивание таблицы и оценку эффективности упорядочивания различными методами.

Описание ТЗ

Описание исходных данных и результатов

Программа оперирует записями о книгах, содержащими автора, издательство, название книги, количество страниц, далее, в зависимости от жанра литературы:

1. Техническая - отрасль хозяйства, вид тех. литературы (переведенная или отечественная), год издания
2. Художественная литература (пьеса, рассказ или стихи)
3. Детская литература (сказка или детские стихи)

Ключевым значением записи выбрано количество страниц.

Также программа оперирует таким понятием, как таблица ключей, она представляет собой таблицу, соответствующую таблице записей, однако хранящей только индекс элемента в основной таблице и его ключевое значение.

Описание задачи, реализуемой программой

Программа предоставляет доступ к различным операциям над таблицей: загрузка/выгрузка таблицы в/из файла, генерация случайных данных для тестирования, вставка и удаление записей, сортировка таблицы/таблицы ключей, вывод таблицы, вывод таблицы ключей, вывод таблицы по ключам, замер эффективности работы алгоритмов сортировки.

Также программа может осуществлять поиск по запросу - поиск отечественной технической литературы, по заданной отрасли хозяйства и году издания.

Способ обращения к программе

Обращение к программе происходит путём работы с текстовым меню через консоль, доступные пункты меню:

1. Выход из программы
2. Создать файл, случайно заполненный данными
3. Считать данные из файла в память
4. Записать текущие данные в файл
5. Вывести данные из памяти
6. Добавить запись
7. Удалить запись
8. Провести замеры времени сортировки двумя способами
9. Отсортировать таблицу
10. Отсортировать таблицу ключей
11. Вывести таблицу
12. Вывести таблицу ключей

13. Вывести таблицу по ключам

14. Вывести отечественную тех. литературу по указанной отрасли указанного года

Описание возможных аварийных ситуаций и ошибок пользователя

1. В случае превышения допустимого числа символов для строковых полей (либо ввода пустой строки), произойдет потеря информации. Для фамилии автора и издательства: 15 символов, для названия - 31 символ.
2. Ввод недопустимых данных (отрицательные числа там, где требуются положительные, например), произойдет ошибка ввода и пользователю будет необходимо заново повторить попытку ввода
3. Ошибки открытия файла на чтение/запись
4. Ошибка выделения памяти

Описание внутренних СД

Программа оперирует таблицей записей следующего вида:

```
typedef struct
{
    /// Фамилия автора
    char author[AUTHOR_NAME_LEN];
    /// Издательство
    char publisher[PUBLISHER_NAME_LEN];
    /// Название книги
    char title[TITLE_NAME_LEN];
    /// Количество страниц
    unsigned int page_count;
    /// Жанр книги
    genre_t genre;
    /// Специфичные для жанра данные
    union {
        /// Техническая литература
        struct {
            /// Отрасль хозяйства
            industry_type industry;
            /// Вид технической литературы
            tech_type_t type;
            /// Год издания
            unsigned short int year;
        } tech;
        /// Художественная литература
        struct
        {
            /// Вид художественной литературы
            fiction_type_t type;
        } fiction;
        /// Детская литература
        struct
        {
            /// Вид детской литературы
            child_type_t type;
        } child;
    } literature_kind;
} book_t;
```

Для реализации таблицы ключей используется следующая структура данных:

```
/// Структура для хранения ключей таблицы
typedef struct
{
    /// Индекс в базовой таблице
    unsigned int base_table_index;
    /// Число страниц
    unsigned int page_count;
} table_key_t;
```

Сравнение алгоритмов сортировки

При проведении замеров эффективности использовались 3 варианта сортировки таблицы, для замеров сортировка повторялась 100.000 раз: а) Быстрая сортировка всей таблицы б) Быстрая сортировка с использованием таблицы ключей в) Сортировка вставками с использованием таблицы ключей

Эффективность сортировки а) взята за 100% Тогда результаты сравнения эффективности примут следующий вид:

2 записи в таблице

Сортировка	Время	Память
а	100.00%	100.00%
б	100.00%	91.30%
в	140.00%	91.30%

10 записей в таблице

Сортировка	Время	Память
а	100.00%	100.00%
б	185.29%	91.30%
в	75.90%	91.30%

20 записей в таблице

Сортировка	Время	Память
а	100.00%	100.00%
б	159.42%	91.30%
в	43.14%	91.30%

40 записей в таблице

Сортировка	Время	Память
а	100.00%	100.00%
б	168.00%	91.30%
в	28.23%	91.30%

Выводы

Использование типа с вариантной частью позволяет более эффективно компоновать различные схожие друг с другом структуры данных, взамен на некоторое усложнение работы с ними и увеличение среднего размера. Исходя из полученных в ходе тестирования эффективности сортировки таблицы записей результатов, можно сделать вывод, что использование дополнительной таблицы ключей для сортировки позволяет гораздо эффективнее сортировать таблицу записей, за счёт уменьшения размера перемещаемых объёмов памяти. Потеря эффективности по памяти при этом незначительна и стремится к нулю при увеличении размера одной записи.

Ответы на вопросы

1. Как выделяется память под вариантную часть записи?

Объём записи под вариантную часть записи высчитывается, как максимум размера среди возможных вариантов.

2. Что будет, если в вариантную часть ввести данные, не соответствующие описанным

Так как вариантная часть по сути предоставляет лишь разный интерфейс доступа к одному и тому же участку памяти, внесение некорректных данных приведёт к неверной их интерпретации и потере смысла значения.

3. Кто должен следить за правильностью выполнения операций с вариантной частью записи?

Контроль за правильностью выполнения операций с вариантной частью записи осуществляется программистом.

4. Что представляет собой таблица ключей, зачем она нужна?

Таблица ключей представляет собой вспомогательную таблицу записей, хранящих в себе индексы соответствующих записей в исходной таблице, а также значение ключевого для них поля. Таблица ключей может быть тривиально построена на основе исходной таблицы, а в дальнейшем - может быть отсортирована и использована для вывода отсортированной исходной таблицы, путём использования индексов записей исходной таблицы.

5. В каких случаях эффективнее обрабатывать данные в самой таблице, а когда - использовать таблицу ключей?

Обработка с использованием таблицы ключей эффективнее на любом размере таблицы при условии, что размер записи в таблице ключей сильно меньше размера записи в таблице, так как затраты времени на перемещение данных будут сильно снижены при небольших дополнительных затратах памяти

6. Какие способы сортировки предпочтительнее для обработки таблиц и почему?

Для обработки таблиц предпочтительно использовать сортировку с минимальной возможной алгоритмической сложностью, а также не сортировать таблицу целиком, а создавать небольшие вспомогательные таблицы, которые смогут быть отсортированы гораздо быстрее, и по которым можно будет получить отсортированную версию исходной таблицы.