|  |  |
| --- | --- |
| **Gerb-BMSTU_01** | **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**

**«ОБРАБОТКА ТАБЛИЦ»**

Студент Светличная Алина Алексеевна

Группа ИУ7 – 33Б

Проверил Барышникова Марина Юрьевна

*2021 г.*

# **Описание технического задания**

Условие задачи:

* возможность добавления записей в конец таблицы и удаления записи по значению указанного поля
* просмотр отсортированной таблицы ключей при несортированной исходной таблице
* вывод упорядоченной исходной таблицы;
* вывод исходной таблицы в упорядоченном виде, используя упорядоченную таблицу ключей
* вывод результатов сравнения эффективности работы программы при обработке данных в исходной таблице и в таблице ключей;
* вывод результатов использования различных алгоритмов сортировок.

**Вариант №1:** Ввести список литературы, содержащий фамилию автора, название книги, издательство, количество страниц, вид литературы (1: техническая – отрасль, отечественная, переводная, год издания; 2: художественная – роман, пьеса, стихи; 3: детская – сказки, стихи). Вывести список отечественной технической литературы по указанной отрасли

Входные данные:

Пункт меню, введенный по следующим правилам:

0. Вывод правил хранения и записи структур.

1. Загрузить список книг из файла.

2. Добавить книгу в конец таблицы.

3. Удалить книгу из таблицы по количеству страниц.

4. Посмотреть отсортированный (по количеству страниц) массив ключей.

5. Вывести упорядоченную (по количеству страниц) таблицу.

6. Вывести упорядоченную таблицу (по количеству страниц), используя упорядоченный массив ключей.");

7. Вывести результаты сравнения эффективности программы при обработке таблицы и массив ключей.

8. Вывести список отечественной технической литературы по указанной отрасли указанного года.

9. Вывести таблицу.

10. Выйти из программы.

Таблица записей, хранящаяся в следующем виде:

1) На первой строке файла записывается количество записей.

2) Далее с каждой новой строки записывается одна запись.

3) Все записи имеют одинаковую первичную сигнатуру.

3.1) Фамилия автора.

3.1.1) Программой непредусмотрено никаких символов кроме букв в данном поле.

3.1.2) Регистр имеет значение.

3.2) Наименование книги.

3.2.1) Программой непредусмотрено никаких символов кроме букв и подчеркиваний в данном поле.

3.2.2) Регистр имеет значение.

3.3) Издательство.

3.3.1) Программой непредусмотрено никаких символов кроме букв и подчеркиваний в данном поле.

3.3.2) Регистр имеет значение.

3.4) Количество страниц.

3.4.1) Целое число.

3.5) Тип литературы.

3.5.1) Один из трех (technical, fiction, children).

3.5.2) Регистр имеет значение.

4) В зависимости от типа литературы содержится дополнительная информация.

4.1) Technical");

4.1.1) Наименование отсрасли (аналогично фамилии автора)");

4.1.2) Отечественность книги (если книга отчественная в данной позиции стоит 1, если нет - 0)");

4.1.3) Зарубежность книги (если книга зарубежная в данной позиции стоит 1, если нет - 0)");

4.1.3.1) Программа не обрабатывает случаи нескольких нулей в выборных полях");

4.1.4) Год издания книги (аналогично количеству страниц)");

4.2) Fiction");

4.2.1) Роман (выборное поле)");

4.2.2) Пьеса (выборное поле)");

4.2.3) Поэзия (выборное поле)");

4.3) Children");

4.3.1) Сказка (выборное поле)");

4.3.2) Стих (выборное поле)");

5. Правила для записи полей не полностью распространяются на пункт меню 2 (правила исправлены подсказками на экране)

Выходные данные:

Соответствуют пункту меню

Действие программы (основное):

Исследование различных сортировок (bubble sort, Qsort) таблиц различными путями (работа напрямую с таблицей, с массивом ключей).

Обращение к программе:

Запускается через терминал командой ./main.exe в директории с программой.

Возможные аварийные ситуации:

1. Некорректный номер меню (возможен только от 0 до 10)
2. Ошибочные входные файлы (пустой файл, несуществующий файл)
3. Некорректный ввод данных

* Некорректные символы в любом поле в файле (меню 1)
* Некорректные символы при добавлении новой книги (меню 2)
* Некорректные символы при удалении книги (меню 3)

1. Переполнение таблицы (максимальное количество записей - 100)

# **Описание структур данных**

**action\_type**  – переменная для записи выбранного пункта меню

\* MAX\_LEN = 101

Поля структуры, описанной ниже, для хранения информации о технической книге:

branch – техническая отрасль

domestic – признак отечественной книги

translate – признак зарубежной книги

year – год издания книги

**typedef struct**

**{**

**char branch[MAX\_LEN];**

**short int domestic;**

**short int translate;**

**short year;**

**} technical\_t;**

Поля структуры, описанной ниже, для хранения информации о художественной книге:

novel – признак романа

play – признак пьесы

poetry – признак поэзии

**typedef struct**

**{**

**short int novel;**

**short int play;**

**short int poetry;**

**} fiction\_t;**

Поля структуры, описанной ниже, для хранения информации о детской книге:

story – признак сказки

poetry – признак стиха

**typedef struct**

**{**

**short int story;**

**short int poetry;**

**} children\_t;**

Поля объединения, описанного ниже, для хранения дополнительной информации то типу книги:

technical – техническая книга

fiction – художественная книга

children – детская книга

**typedef union**

**{**

**technical\_t technical;**

**fiction\_t fiction;**

**children\_t children;**

**} type\_t;**

Поля структуры, описанной ниже, для хранения всей информации о книге:

author – фамилия автора

name – наименование

publish\_house – издательство

size\_book – количество страниц

is\_type – тип книги

type – объединение дополнительной информации по типу

**typedef struct**

**{**

**char author[MAX\_LEN];**

**char name[MAX\_LEN];**

**char publish\_house[MAX\_LEN];**

**short size\_book;**

**char is\_type[MAX\_LEN];**

**type\_t type;**

**} books\_t;**

Поля структуры, описанной ниже, для хранения массива ключей:

size\_book – количество страниц

id – индекс в основной таблице

**typedef struct**

**{**

**short size\_book;**

**short id;**

**} keys\_t;**

Поля структуры, описанной ниже, для таблицы, объединяющей исходную таблицу и таблицу ключей:

books – структура информации о книге

keys – структура для хранения ключей

size\_book – количество записей в таблице

**typedef struct**

**{**

**books\_t \*books;**

**keys\_t \*keys;**

**short size;**

**} table\_t;**

# **Описание алгоритма**

# **Функции ввода**

1. short int read\_action(short \*action\_type) – читает номер меню
   * \*action\_type – указатель на переменную для записи номера меню
2. short read\_file\_size(table\_t \*table, FILE \*f) – читает количество записей в файле
   * \*table – указатель на таблицу, объединяющую исходную таблицу и таблицу ключей
   * \*f – файловая переменная
3. short read\_books\_author(table\_t \*table, FILE \*f, short int i) – читает фамилию автора
   * \*table – указатель на таблицу, объединяющую исходную таблицу и таблицу ключей
   * \*f – файловая переменная
   * i – индекс записи
4. short int read\_books\_name(table\_t \*table, FILE \*f, short int i) – читает наименование книги
5. short int read\_books\_publish\_house(table\_t \*table, FILE \*f, short int i) – читает издательство
6. short int read\_books\_is\_type(table\_t \*table, FILE \*f, short int i)– читает тип литературы
7. short int read\_technical(table\_t \*table, FILE \*f, short int i) – читает всю дополнительную информацию о технических книгах
8. short int read\_fiction(table\_t \*table, FILE \*f, short int i) – читает всю дополнительную информацию о художественных книгах
9. short int read\_children(table\_t \*table, FILE \*f, short int i) – читает всю дополнительную информацию о детских книгах
10. short read\_size\_book\_to\_del(table\_t \*const table, int \*size\_book) – читает количество страниц для удаления
    * \*table – указатель на таблицу, объединяющую исходную таблицу и таблицу ключей
    * \*size\_book – указатель на переменную для записи количества страниц
11. short read\_branch(char \*brach\_str) – читает отрасль для вывода списка пункта меню 8
    * \*brach\_str i – указатель на переменную для записи прочитанной отрасли
12. short reade\_year(int \*year\_str)– читает год для вывода списка пункта меню 8
    * \*year\_str– указатель на переменную для записи прочитанного года

# **Функции вывода**

1. void print\_rules() – выводит правила ввода и хранения записей
2. void print\_menu() – выводит правила пункты меню
3. void print\_table\_keys(const table\_t table) – выводит таблицу ключей
   * table – таблица, объединяющая исходную таблицу и таблицу ключей
4. void print\_table(const table\_t table, short int keys) – выводит на экран исходную таблицу
   * table – таблица, объединяющая исходную таблицу и таблицу ключей
   * keys – ключ, определяющий выводится таблица подряд или отсортированная
5. void print\_sorts\_vs\_results(int64\_t total\_ticks, short sort\_type, short table\_type) – выводит информацию о сортировке по нескольким ключам
   * total\_ticks – время выполнения сортировки
   * sort\_type – ключ, определяющий метод сортировки
   * table\_type – ключ, определяющий тип сортировки среди таблицы
6. void print\_by\_condition(const table\_t table, int i) – выводит информацию только о технических книгах
   * table – таблица, объединяющая исходную таблицу и таблицу ключей
   * i – индекс элемента в таблице

# **Функции сортировки**

1. void swap(short int table, void \*val1, void \*val2) – меняет местами два элемента в таблице
   * table – ключ, определяющий тип таблицы (основная таблица, таблица ключей)
   * \*val1 – указатель на первый элемент
   * \*val2 – указатель на второй элемент
2. int comparator\_keys(const void\* val1, const void\* val2) – сравнивает количество станиц двух элементов в таблице ключей
   * \*val1 – указатель на первый элемент
   * \*val2 – указатель на второй элемент
3. int comparator\_table(const void \*val1, const void \*val2)– сравнивает количество станиц двух элементов в основной таблице
   * \*val1 – указатель на первый элемент
   * \*val2 – указатель на второй элемент
4. void bubble\_sort(const int size, void \*arr, int (\*comparator)(const void \*val1, const void \*val2), short int table, size\_t size\_bytes)– сравнивает количество станиц двух элементов в основной таблице
   * size – количество записей
   * \*arr – массив для сортировки
   * table – ключ, определяющий тип таблицы (основная таблица, таблица ключей)
   * size\_bytes – количество байтов под одну запись
   * сompartor – указатель на функцию сравнения элементов

# **Функции времени**

1. uint64\_t tick(void) – засекает время выполнения сортировок

# **Функции для работы с таблицами**

1. void clear\_table(table\_t \*const table) – очищает таблицу при перезаписи
   * \*table – указатель на таблицу, объединяющую исходную таблицу и таблицу ключей
2. short load\_data(table\_t \*const table, FILE \*f, short start, short end) – заполняет таблицу данными из файла
   * \*table – указатель на таблицу, объединяющую исходную таблицу и таблицу ключей
   * \*f – файловая переменная
   * start – индекс начала таблицы
   * end – индекс конца таблицы
3. void update\_keys(table\_t \*const table, const short i, const short id, const short size\_book) – обновляет таблицу ключей после чтения таблицы
   * \*table – указатель на таблицу, объединяющую исходную таблицу и таблицу ключей
   * i – индекс записи в таблице ключей
   * id – индекс записи в основной таблице
   * size\_book – количество страниц в записи
4. short upload\_from\_file(table\_t \*const table) – вызывает все функции чтения файла
   * \*table – указатель на таблицу, объединяющую исходную таблицу и таблицу ключей
5. short load\_data\_one(table\_t \*const table, FILE \*f, short start, short end) – читает одну запись для добавления в таблицу
   * \*table – указатель на таблицу, объединяющую исходную таблицу и таблицу ключей
   * \*f – файловая переменная
   * start – индекс начала таблицы
   * end – индекс конца таблицы
6. short add\_to\_end(table\_t \*const table) – добавляет запись в конец таблицы
   * \*table – указатель на таблицу, объединяющую исходную таблицу и таблицу ключей
7. short check\_size(short size) – проверяет, не пуста ли таблица
   * size – количество записей в таблице
8. void del\_by\_pos(table\_t \*const table, int del\_ind\_keys, int del\_ind\_table) – удаляет книгу по количеству страниц
   * \*table – указатель на таблицу, объединяющую исходную таблицу и таблицу ключей
   * del\_ind\_keys – индекс записи в таблице ключей
   * del\_ind\_table – индекс записи в основной таблице
9. short find\_by\_size\_book(table\_t \*const table, int size\_book) – находит все книги с определенным количеством страниц
   * \*table – указатель на таблицу, объединяющую исходную таблицу и таблицу ключей
   * size\_book – количество страниц для поиска
10. short del\_by\_size\_book(table\_t \*const table) – вызывает все функции для удаления книги по количеству страниц
    * \*table – указатель на таблицу, объединяющую исходную таблицу и таблицу ключей
11. short qs\_vs\_bubble(table\_t \*const table) – сравнивает сортировки по методу и типу таблицы
    * \*table – указатель на таблицу, объединяющую исходную таблицу и таблицу ключей
12. short find\_books(table\_t \*const table, char \*const branch\_str, const int year\_str) – ищет технические книги по заданной отрасли по заданному году
    * \*table – указатель на таблицу, объединяющую исходную таблицу и таблицу ключей
    * \*branch\_str – указатель на строку с заданной отраслью
    * \*year\_str – указатель на переменную с заданным годом
13. short find\_book\_by\_condition(table\_t \*const table) – вызывает все функции для пункта меню 8
    * \*table – указатель на таблицу, объединяющую исходную таблицу и таблицу ключей
14. short int do\_action(short int action\_type, table\_t \*table) – вызывает функции согласно пункту меню
    * \*table – указатель на таблицу, объединяющую исходную таблицу и таблицу ключей
    * action\_type – номер заданного пункта меню

**Тесты**

|  |  |  |
| --- | --- | --- |
| Аварийная ситуация | Код ошибки | Уведомление |
| Пункт меню - w | ERROR\_SYMBOLS | В записи пункта меню встречены некорректные символы |
| Пункт меню - 12 | ERROR\_MENU\_NUMBER | Некорректный пункт меню |
| Пустой файл | ERROR\_OPEN\_FILE | Невозможно открыть файл |
| Количество записей - r | ERROR\_FILE\_SIZE | В записи количества записей встречены некорректные символы |
| Количество записей - 0 | ERROR\_FILE\_SIZE | Некорректное количество записей |
| Фамилия автора – Esenin& | ERROR\_SYMBOLS | В фамилии автора встречены некорректные символы |
| Название книги – Consuel\*\*o | ERROR\_SYMBOLS | В названии книги встречены некорректные символы |
| Издательство – AST# | ERROR\_SYMBOLS | В названии издательства встречены некорректные символы |
| Тип книги – technical8 | ERROR\_SYMBOLS | В названии типа книги встречены некорректные символы |
| Название отрасли – math% | ERROR\_SYMBOLS | В названии отрасли технических книг встречены некорректные символы |
| Отечественная ли книга - w | ERROR\_SYMBOLS | Обозначение отечественности книги некорректно |
| Переведенная ли книга - w | ERROR\_SYMBOLS | Обозначение перевода книги некорректно |
| Год написания книги – 196r | ERROR\_SYMBOLS | Год написания книги некорректен |
| Обозначение жанра 'роман' – er | ERROR\_SYMBOLS | Обозначение жанра 'роман' некорректно |
| Обозначение жанра 'пьеса' – 2w | ERROR\_SYMBOLS | Обозначение жанра 'пьеса' некорректно |
| Обозначение жанра 'поэзия' – 1\_ | ERROR\_SYMBOLS | Обозначение жанра 'поэзия' некорректно |
| Обозначение жанра 'поэзия' – 0% | ERROR\_SYMBOLS | Обозначение жанра 'поэзия' некорректно |
| Обозначение жанра 'стихи' – t | ERROR\_SYMBOLS | Обозначение жанра 'стихи' некорректно |
| В пункте удаления книги по количеству страниц введено – 567\_ | ERROR\_SYMBOLS | Введены некорректные символы |
| Количество страниц - 0 | ERROR\_SIZE\_BOOK | Некорректное количество страниц |
| В пункте поиска книги по заданной отрасли (и заданному году) введено – “\n” | ERROR\_INPUT | Ошибка чтения |
| Заданная отрасль – math() | ERROR\_SYMBOLS | В названии введенной отрасли встречены некорректные символы |
| В пункте поиска книги по заданному году (и заданной отрасли) введено – qwerty | ERROR\_INPUT | Введенный год некорректен |
| Вызывается пункт меню о добавлении книги при полной таблице (установленное значение - 1000) | TABLE\_IS\_FULL | Невозможно добавить книгу, таблица переполнена |
| Любой пункт меню вызван до заполнения таблицы | TABLE\_IS\_EMPTY | Таблица пуста |
| При поиске книг по заданной отрасли заданного года введены параметры книг, которых нет в таблице | NO\_MATCHES | Книг с такими параметрами не найдено |
| При удалении книги по количеству страниц введено такое количество страниц, которого нет в таблице | NO\_BOOK | Книг с таким количеством страниц нет в таблице |

**Обработка результатов сортировок**

**Время сортировки (в секундах \* ):**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Количество записей | Сортировка пузырьком | | QuickSort | |
| Таблица | Массив ключей | Таблица | Массив ключей |
| 10 | 42533 | 15267 | 151147 | 14867 |
| 50 | 913787 | 394907 | 781320 | 76720 |
| 150 | 8129453 | 2847333 | 3702987 | 362120 |
| 500 | 108551327 | 35716507 | 6547600 | 569520 |

**Объем занимаемой памяти (в байтах):**

|  |  |  |
| --- | --- | --- |
| Количество записей | Таблица | Массив ключей |
| 10 | 5160 | 40 |
| 50 | 25800 | 200 |
| 150 | 77400 | 600 |
| 500 | 258000 | 2000 |

|  |  |  |  |
| --- | --- | --- | --- |
| Количество записей | Занимаемая память в % массива ключей от всей таблицы | % роста быстроты сортировки массива ключей по сравнению с таблицей (сортировка пузырьком) | % роста скорости сортировки массива ключей по сравнению с таблицей (QuickSort) |
| 10 | ~0,7% | ~278% | ~1016% |
| 50 | ~0,7% | ~231% | ~1018% |
| 150 | ~0,7% | ~285% | ~1025% |
| 500 | ~0,7% | ~303% | ~1149% |

# **Контрольные вопросы**

1. Как выделяется память под вариантную часть записи?

В языке си вариативная часть структуры реализована с помощью объединений (union). Память выделяется под наибольшее по памяти из полей.

2. Что будет, если в вариантную часть ввести данные, несоответствующие описанным?

Результат будет системно-зависимым и трудно предсказуемым. Возможно, произойдет приведение типов.

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

Ответственность за правильность проведения операций целиком и полностью лежит на программисте.

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

Таблица ключей представляет собой таблицу, в которой находится два столбца: номер ячейки в исходной таблице и значение выбранного поля исходной таблицы. Ее необходимость заключается в оптимизации программы, потому что такая сортировка быстрее.

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

Если необходима быстро действенность, а память относительно неважный ресурс, то используется таблица ключей. Если же наоборот необходимо наименьшее количество памяти, то используется сама таблица, которую также можно использовать, когда в ней содержится малое количество полей.

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

Для таблиц с большим количеством записей предпочтительно использовать стандартные и устойчивые способы сортировки, со средним временем обработки O(n\*log n), такие как QuickSort. Если же в таблице не так много записей, то предпочтительнее использовать простые алгоритмы сортировки, например известную всем сортировку пузырьком.

# **Вывод**

В ходе лабораторной работы я познакомилась с таблицами и способами их обработки. Полученные результаты ярко показывают, что при затратах менее, чем в 1% (в общем случае цифра будет в разы больше, около 20 -30%, однако в моей лабораторной работе выбрана структура данных, хранящая несколько строк, против таблицы ключей, хранящей всего лишь два целых коротких числа), мы получаем увеличение скорости как минимум в 2 раза (при быстрой сортировке удалось добиться и вовсе 10-кратного увеличения). Отсюда можно сделать вывод, что обработка таблицы, кажущейся неочевидной обработкой таблицей ключей, является оптимальной.