



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 **«ЗАПИСИ С ВАРИАНТАМИ. ОБРАБОТКА ТАБЛИЦ»**

Студент Кладницкий А. Б.

Преподаватель

Группа ИУ7 – 32Б

2022 г.

1. Описание условия задачи

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).

Создать таблицу, содержащую не менее 40-ка записей (тип – запись с вариантами (объединениями)). Упорядочить данные в ней по возрастанию ключей, двумя алгоритмами сортировки, где ключ – поле pages (количество страниц), используя: а) саму таблицу, б) массив ключей. Возможность добавления и удаления записей в ручном режиме обязательна. Осуществить поиск информации по варианту: вывести список отечественной технической литературы по указанной отрасли указанного года.

2. ТЗ

1. Исходные данные:

- Файл, содержащий в себе текстовую БД. Разделитель по умолчанию – символ “|”. Одна запись занимает одну строку. В файле записи хранятся в не отформатированном виде.
- Запись состоит из следующих полей:
 - Имя автора книги (до 30 символов)
 - Название книги (до 30 символов)
 - Издательство (до 30 символов)
 - Кол-во страниц
 - Тип
 - tech (техническая)
 - Отрасль (до 20 символов)
 - Тип
 - domestic (отечественная)
 - transfered (переводная) Тип
 - Год выпуска
 - art (художественная)
 - Тип
 - novel (роман)
 - play (пьеса)
 - poetry (стихи)
 - child (детская)
 - Тип
 - fairy (сказки)
 - poetry (стихи)
- Коды для управления меню
- Данные, для работы с меню

2. Результирующие данные:

- В зависимости от выполняемой задачи, отсортированная таблица, выведенная в консоль или файл; отфильтрованные элементы таблицы.

3. Задача программы:

- Загрузка БД из файла
- Добавление в БД новой записи (вставка по индексу)
- Удаление записи по значению
- Запись БД в файл
- Сортировка (2 штуки)
- Фильтрация по условию
- Вывод таблицы
- Вывод таблицы ключей

4. Способ обращения к программе:

- Запуск через терминал (./app.exe), работа через меню.

5. Возможные ошибки:

- Ошибка работы с меню: неверный код
- Ошибка работы с файлом: не существует или другая проблема
- Ошибка ввода
- Ошибка выделения динамической памяти
- Неверный формат строки
- Неверный формат ввода целого числа
- Неверный тип записи
- Неверный индекс (для удаления и вставки)
- Пустой результат фильтрации
- Обращение к незагруженной БД

3. Описание внутренних структур данных

Тип данных db_t:

```
typedef struct db
{
    keys_t *keys;
    record_t *recs;
    size_t len;
} db_t;
```

Тип содержит в себе динамические массивы типа record_t (запись) и keys_t (доп. ключ), а также их длину.

```
typedef struct keys
{
    int id;
    int pages;
} keys_t;
```

Тип keys_t содержит в себе значение id (индекс) и значение поля, по которому происходит сортировка (pages)

```
typedef struct record
{
    char author[MAX_AUTHOR + 1];
    char title[MAX_TITLE + 1];
    char publish[MAX_PUBLISH + 1];
    int pages;
    enum book_kind kind;
    union kind
    {
        book_tech_t tech;
        book_art_t art;
        book_child_t child;
    } book;
} record_t;
```

Тип `record_t` хранит в себе значения текстовых полей и объединение трех типов (виды книг по условию)

```
MAX_AUTHOR = 30
```

```
MAX_TITLE = 30
```

```
MAX_PUBLISH = 30
```

Типы для книг представлены следующим образом:

```
enum tech
```

```
{
```

```
    DOMESTIC,  // Отечественная
```

```
    TRANSFERED  // Переводная
```

```
};
```

```
typedef struct book_tech
```

```
{
```

```
    char industry[MAX_INDUSTRY + 1]; // Отрасль
```

```
    enum tech kind;  // Тип
```

```
    int year;  // Год издания
```

```
} book_tech_t;
```

```
enum art
```

```
{
```

```
    NOVEL,  // Роман
```

```
    PLAY,  // Пьеса
```

```
    POETRY  // Стихи
```

```
};
```

```
typedef struct book_art
```

```
{
```

```
    enum art kind;  // Тип
```

```
} book_art_t;
```

```
enum child
```

```
{
```

```
    FAIRY,  // Сказка
```

```
    CPOETRY  // Стихи
```

```
};
```

```
typedef struct book_child
{
    enum child kind; // Тип
} book_child_t;
```

```
MAX_INDUSTRY = 20
```

4. Алгоритм

1. Узнать, нужно ли использовать доп. ключи при работе с таблицей.
2. Вывести меню
3. Получить ответ от пользователя: какой пункт меню выполнять?
4. Выполнять команды пользователя, пока не был введен 0 или достигнуто другое условие выхода из главного цикла.

5. Набор тестов

Негативные тесты:

	Описание	Входные данные	Результат
1.	Некорректный пункт меню	-1, uhuv и т.д.	ERR_MENU
2.	Несуществующий файл	non.txt	ERR_FILE
3.	Файл без доступа	не хватает прав на взаимодействие	ERR_FILE
4.	Неверная длина строки в файле	Фамилия автора больше 30	ERR_BAD_STR
5.	Неверный формат целого числа в файле	er34	ERR_BAD_INT
6.	Отсутствие разделителей	Файл без “ ”	ERR_BAD_STR
7.	Пустая строка в файле	Пустая строка	ERR_BAD_STR
8.	Неверный тип книги	TecH	ERR_BAD_ENUM
9.	Неверный тип tech	testing	ERR_BAD_ENUM
10.	Неверный тип art	game	ERR_BAD_ENUM
11.	Неверный тип child	сpoetry	ERR_BAD_ENUM
12.	Неверный индекс для удаления	<0 или больше длины	ERR_BAD_INDEX
13.	Неверный индекс для вставки	<0 или больше длины	ERR_BAD_INDEX
14.	Нет записей по итогам фильтрации	значение отрасли и года	ERR_EMPTY_FILTER
15.	Ошибка записи в выходной файл	файл не создался	ERR_FILE
16.	Ошибка выделения динамической памяти	???	ERR_ALLOC
17.	Неоткрытая БД	вызов пункта меню до открытия файла (1)	ERR_UNINIT_DB

Позитивные тесты:

	Описание	Входные данные	Результат
1.	Открытие файла	название файла	Файл открыт, данные прочитаны в БД
2.	Вставка записи	строка-запись, индекс	В БД вставлена строка по заданному индексу
3.	Удаление записи	индекс	Из БД удалена строка

			по заданному индексу
4.	Запись БД в новый файл	название файла	БД сохранена в файл
5.	Сортировка	-	БД отсортирована по условию
6.	Быстрая сортировка	-	БД отсортирована по условию
7.	Фильтрация	запрос фильтрации (год и отрасль)	Поля БД, удовлетворяющие фильтру, выведены
8.	Вывод БД	-	Печать БД в терминале
9.	Вывод меню	-	Повторный вывод меню в терминал

6. Оценка эффективности

Время:

Всего записей	Пузырек			Qsort		
	Таблица	Ключи	Относит. эффективность	Таблица	Ключи	Относит. эффективность
40	58	33	175%	14	10	140%
80	248	118	210%	47	19	247%
120	493	253	194%	41	31	132%
160	867	427	203%	56	39	143%
480	7941	3476	228%	165	120	138%

Память:

Всего записей	Таблица	Ключи	%
40	5440	320	5,9%
80	10880	640	5,9%
120	16320	960	5,9%
160	21760	1280	5,9%
480	65280	3840	5,9%

7. Вывод

Приобрел навыки работы с типом данных «запись» («структура») содержащим вариантную часть, и с данными, хранящимися в таблицах. Оценил относительную эффективность программы (в процентах) по времени и по используемому объему памяти в зависимости от используемого алгоритма и от объема сортируемой информации.

Вариантные записи удобно использовать, когда одновременно должны быть использованы данные только одного из полей. Это позволяет экономить место.

Использование доп. таблицы ключей ускоряет сортировку при больших объемах данных. При этом, для нее необходимо затрачивать дополнительную память.

8. Контрольные вопросы

1. Как выделяется память под вариантную часть записи?
Размер памяти определяется размером максимального поля. При этом память общая, ее делят все поля.
2. Что будет, если в вариантную часть ввести данные, несоответствующие описанным?
При компиляции тип данных не проверяется, ответственность возлагается на программиста.
3. Кто должен следить за правильностью выполнения операций с вариантной частью записи?
Программист (см. вопрос 2)
4. Что представляет собой таблица ключей, зачем она нужна?
Дополнительная структура из двух массивов. В одном – идентификаторы (индексы), в другом сортируемые ключевые значения.

Ее использование ускоряет работу сортировки, но более затратное по памяти.

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

При маленьком размере таблицы, выгоднее обрабатывать ее саму. Если в структуре содержится много записей и полей, то выгоднее пользоваться таблицей ключей.

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

Для сортировки таблицы ключей эффективнее сортировки, работающие с наименьшей сложностью работы.

Для сортировки непосредственно записей эффективнее сортировки с наименьшим числом перестановок.