|  |  |
| --- | --- |
| 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 **ПРОГРАММНАЯ ИНЖЕНЕРИЯ (ИУ7)**

НАПРАВЛЕНИЕ ПОДГОТОВКИ **09.03.04** Программная инженерия

**Отчет**

|  |  |
| --- | --- |
| **По лабораторной работе №** | 2 |

**Название:**

Записи с вариантами, обработка таблиц

**Дисциплина:** Типы и структуры данных

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Студент | ИУ7-36Б |  | | А.А. Жаворонкова |
|  | (Группа) | |  | (И.О. Фамилия) |
|  |  | |  |  |
| Преподаватель: | Никульшина Т. А. | |  |  |

Москва, 2022

Описание условия задачи

Ввести список абонентов, содержащий фамилию, имя, телефон,

адрес, статус (личный – дата рождения: день, месяц, год;

служебный – должность, организация). Найти всех друзей, которых

необходимо поздравить с днем рождения в ближайшую неделю.

Описание ТЗ

*Описание исходных данных и результатов*

*Исходные данные:* файл с информацией об абоненте, пункт меню (см. описание задачи). Каждая запись содержит: фамилию, имя, номер телефона, адрес, для личной записи: дата рождения (день, месяц, год); для служебной записи: должность, организация.  
*Результаты:* таблица абонентов, таблица ключей, таблица эффективности, выборка, абоненты по таблице ключей.

*Описание задачи, реализуемой программой*

Меню:

0) Выйти из программы

1) Считать таблицу из файла

2) Вывести данные

3) Вывести таблицу ключей

4) Добавить запись

5) Удалить запись

6) Отсортировать записи по номеру телефона (быстрая/медленная сортировки)

7) Отсортировать таблицу ключей (быстрая/медленная сортировки)

8) Вывести данные по таблице ключей

9) Вывести таблицу эффективности

10) Вывести список людей, у которых день рождения в ближайшую неделю

11) Вывести меню

*Способ обращения к программе*

Вызов программы происходит через терминал (main.exe). Дальнейшая работа с программой выполняется при помощи меню, выводимого на экран.

*Описание возможных ошибок пользователя*

* Ввод некорректного пункта меню
* Ввод имени несуществующего файла
* Ввод имени файла, содержащего данные в некорректном формате
* Ввод записи в некорректном формате
* При удалении записи – ввод несуществующего номера
* При выборе сортировки – ввод неверного значения

Описание внутренних структур данных

Для хранения данных об абоненте используется массив структур abonent\_t. Структура abonent\_t имеет следующее содержание:

char surname[SURN\_MAX + 1]; // фамилия SURN\_MAX = 50

char name[N\_MAX + 1]; // имя N\_MAX = 50

int number[NUMB\_LEN]; // номер телефона NUMB\_LEN = 11

char adress[ADR\_MAX + 1]; // адрес ADR\_MAX = 100

type\_t type; // тип записи: личная, служебная

status\_t status; // вариантная часть

Для хранения ключей таблицы используется массив структур key\_t. Структура key\_t имеет следующее содержание:

unsigned int position; // позиция элемента в исходной таблице

int number[NUMB\_LEN]; // номер телефона NUMB\_LEN = 11

Временная эффективность и затраты памяти

*Сортировка таблицы с 40 полями*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Сортировка | Быстрая всей таблицы | Обычная всей таблицы | Быстрая по ключам | Обычная по ключам |
| Время, нс | 37 670 | 460 764 | 8 064 | 72 687 |
| Память, б | 13 320 | 13 320 | 1 776 | 1 776 |

*Сортировка таблицы с 400 полями*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Сортировка | Быстрая всей таблицы | Обычная всей таблицы | Быстрая по ключам | Обычная по ключам |
| Время, нс | 452 448 | 42 897 688 | 155 816 | 5 996 157 |
| Память, б | 133 200 | 133 200 | 17 760 | 17 760 |

Данные таблицы были получены путем измерения среднего времени работы алгоритма. Из таблицы видно, что сортировка таблицы по ключам происходит быстрее, однако затраты памяти для сортировки по ключам выше, так как необходимо хранить массив ключей, помимо основного массива.

При сортировке таблицы с 40 полями обычная сортировка всей таблицы выполняется дольше быстрой на 92%. При сортировке по ключам обычная сортировка выполняется дольше быстрой на 89%. При этом сортировка по ключам на 66% выполняется быстрее сортировки всей таблицы, но требует на 13% больше памяти.

При сортировке таблицы с 400 полями обычная сортировка всей таблицы выполняется дольше быстрой на 99%. При сортировке по ключам обычная сортировка выполняется дольше быстрой на 97%. При этом сортировка по ключам на 79% выполняется быстрее сортировки всей таблицы, но требует на 13% больше памяти.

Вывод

Несмотря на затраты по памяти, сортировка таблицы по ключам получается выгоднее сортировки всей таблицы. Дополнительная таблица – таблица ключей требует 13% от памяти занимаемой исходной таблицей, однако выигрыш во времени получается в 3-5 раз, чем при сортировке без таблицы ключей. При увеличении размера таблицы растет эффективность (92 – 99; 89 – 87) использования сортировки через таблицу ключей. Таким образом, при больших размерах таблицы целесообразнее использовать сортировку по ключам.

Ответы на вопросы

1. Как выделяется память под вариативную часть записи?  
Объем памяти, необходимый для записи с вариантами складывается из длин полей фиксированной части и максимального по длине поля вариантной части.

2. Что будет, если в вариативную часть ввести данные, несоответствующие с описанными?  
В таком случае нельзя точно сказать, что произойдет.

3. Кто должен следить за правильностью выполнения операций, с вариативной частью записи?  
За правильностью выполнения операций над вариативной частью должен следить программист.

4. Что представляет из себя таблица ключей и зачем она нужна?  
Таблица ключей представляет из себя массив структур, каждый элемент которого содержит значение какого-либо ключа и его позицию в исходной таблице.

5. В каких случаях эффективно обрабатывать данные в самой таблице, а когда – использовать таблицу ключей?   
В случае, если нам часто приходится сортировать таблицу с большим количеством полей, следует использовать таблицу ключей. Если же полей немного, то можно обойтись и без выделения дополнительной памяти для таблицы ключей.

6. Какие способы сортировки предпочтительнее для обработки таблиц и почему?  
Предпочтительнее те способы сортировки, которые не переставляют уже отсортированные элементы, так как перестановка элементов таблицы занимает много времени. Например, метод сортировки вставками.