

КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ  
ІМЕНІ ТАРАСА ШЕВЧЕНКА

# Звіт

з лабораторної роботи №1

на тему:

“Робота зі структурованими файлами з індексно-  
довільним доступом”

Виконав студент 2 курсу

Групи К-24

Факультету комп’ютерних наук

та кібернетики

Минько Вадим Леонідович

Київ, 2024

## Мета

Мета роботи – навчитися працювати зі структурованими файлами без використання СУБД на основі лише мови C++ і розробити для цього програму.

## Основні принципи виконання роботи

### Структура записів

В даній лабораторній роботі я працював з двома таблицями предметної області “телебачення”: телепередачі, телеведучі. Зв’язок між таблицями є “один-до-багатьох”, в якому master - це телепередача, slave - телеведучий.

Структура master-запису TVshow:

- uint32\_t id - первинний ключ, визначається автоматично
- char[50] name
- char[50] category
- int64\_t first\_host - адреса першого телеведучого певної програми у slave-файлі.
- bool exists - визначає чи видалений логічно запис чи ні

Структура slave-запису TVhost:

- uint32\_t ssn - первинний ключ, вводиться вручну
- char[100] email
- char[25] phone\_number
- char[50] first\_name
- char[50] second\_name
- char[25] birth\_date
- int64\_t next - адреса наступного ведучого телепрограми у зв’язному списку
- bool exists - визначає чи видалений логічно запис чи ні

master-записи мають індексний доступ: при знаходженні майстера за індексом йде ітерація по індексному списку і при знаходженні пари з цим індексом береться значення адреси.

slave-записи мають рекурсивний доступ, адже вони зберігаються у зв’язному списку, початок якого прив’язаний до їхнього майстера.

Також програма містить масиви сміття для видалених master- та slave-записів.

### **Робота з файлами**

У файлах master.bin та slave.bin містяться master та slave записи відповідно. master-записи мають індексний доступ, тому для зберігання пар індексу та адреси відповідного запису маємо ще один файл masterind.bin. При запуску програми ці файли відкриваються без змін. Винятком є masterind.bin, в якому одразу зчитуємо індексний список і очищуємо. Заповнюємо masterind.bin новим списком вже при завершенні виконання програми. Решту файлів заповнюємо під час реорганізації при досягненні певної кількості сміття і також при закритті програми.

### **Команди виконання**

- insert-m - користувач вводить значення атрибутів master-запису, id визначається як останній id запису+1, дані записуються у master-файл на місце кінця файлу або адреси останнього логічно видаленого запису у масиві сміття, пара індексу та адреси запису у файлі записуються в індексний список.
- insert-s - користувач вводить значення атрибутів slave-запису, далі вводить id майстера до якого буде належати, дані записуються у slave-файл аналогічним чином, як у майстера, поле first\_host майстера присвоює значення адреси нового запису, а попереднє значення first\_host присвоюється адресі наступного slave-запису в нового запису.
- get-m - читає за індексом та виводить всі атрибути певного master запису. Інші варіанти: get-m ind attribute - виводить тільки певний атрибут певного master-запису, get-m all - виводить всі атрибути всіх записів.
- get-s - читає за ssn та виводить всі атрибути певного slave-запису. Інші варіанти: get-s ind attribute - виводить тільки певний атрибут певного slave-запису, get-s all - виводить всі атрибути всіх записів, get-s m ind - виводить всіх атрибути всіх записів, які належать певному майстеру.
- del-s, del-m - читає і виконує логічне видалення, тобто присвоює полю exists значення false та записує адресу в масив сміття. В del-m також виконується логічне видалення всіх slave-записів, які належать master-запису.

- `update-m`, `update-s` - користувач вводить первинний ключ, назва атрибуту та значення. Команда читає і оновлює значення певного атрибуту.
- `calc-m`, `calc-s` - обчислює кількість існуючих записів та виводить його.
- `ut-m`, `ut-s` - виводить всі атрибути, включаючи службові, всіх записів, включаючи логічно видалені.
- `help` - виводить список команд і що вони виконують.
- `exit` - вихід з програми.

## **Реорганізація**

Реорганізація виконується, якщо кількість елементів масиву сміття у `master` або `slave` не менше 5, а також при завершенні виконання програми. Реорганізація проводиться таким способом: для `master` та `slave` створюється новий тимчасовий файл, у цей тимчасовий файл записуються тільки не видалені логічно записи. Основний файл очищується і з тимчасового файлу записується весь його вміст в очищений основний файл. Тимчасовий файл далі видаляється. При реорганізації також оновлюються адреси в індексному масиві майстера та зв'язних списках `slave`-записів. Таким чином, логічно видалені записи фізично видалилися та основний файл тепер містить тільки не видалені логічно записи.