

Лабораторна робота №3. Робота з файлами

Мета і задачі:

Навчитися створювати та відлагоджувати програми, у яких здійснюється робота з файлами (зчитування/запис) на мові програмування C.

Теоретичні відомості і методичні вказівки

Файл – це впорядкована послідовність однотипних компонент, розташованих на зовнішньому носії. Файли призначені для зберігання інформації, а обробка цієї інформації здійснюється програмами.

Використання файлів доцільно у випадку:

- довготривалого зберігання даних
- доступу різних програм до одних і тих же даних
- обробки великих масивів даних, які неможливо повністю розмістити в оперативній пам'яті комп'ютера

У мові C використовуються такі функції бібліотеки `stdio.h` для роботи з текстовими файлами: **`fopen`**, **`fscanf`**, **`fprintf`**, **`fclose`**.

Для роботи з текстовими файлами спочатку потрібно оголосити файлову змінну, яка є покажчиком на файл :

```
FILE * <файлова змінна>;
```

Наприклад, оголошення файлової змінної `f` :

```
FILE *f;
```

Потім потрібно відкрити файл і зв'язати його з файловою змінною використавши функцію **`fopen`**:

```
файлова змінна = fopen ("шлях до файлу", «режим  
доступу»);
```

Наприклад, відкриваємо файл `in.txt` для зчитування:

```
f = fopen("D:\\in.txt", "r");
```

Функція **`fopen`** повертає покажчик на структуру типу `FILE` при успішному відкритті файлу, і `NULL` в протилежному випадку

В таблиці 1 вказані режими доступу, які застосовуються у функції `fopen`.

Таблиця 1

Режим доступу	Опис
---------------	------

r	Файл відкривається для читання
r+	Файл відкривається для читання і запису
w	Відкривається пустий файл для запису
w+	Відкривається пустий файл для читання та запису
a	Файл відкривається для дозапису в кінець файлу
a+	Файл відкривається для читання та дозапису в кінець файлу

Режим відкриття може також містити символи t (текстовий файл) и b (двійковий файл), вказуючи на тип файлу, який відкривається: rb, wb, ab, rt, at, rb+, wb+, ab+ тощо.

Для перевірки існування чи правильності відкриття файлу можна використати такий фрагмент коду:

```
if ((fopen("D:\\in.txt", "r"))==NULL)
{printf("Помилка при відкритті файлу");
return 0;
}
```

Приклад відкриття файлу out.txt для запису:

```
f2 = fopen("D:\\out.txt", "w");
```

Загальний прототип функції зчитування з файлу **fscanf**:

```
fscanf(файлова змінна, стрічка форматування, список адрес змінних);
```

Приклад, зчитування з файлу даних у змінні *a* і *b* типу int:

```
fscanf(f,"%i %i",&a, &b);
```

Загальний прототип функції запису у файл **fprintf**:

```
fprintf(файлова змінна, стрічка форматування, список змінних);
```

Приклад, запису у файл значень змінних *a* і *b* типу int:

```
fscanf(f2,"%i %i",a, b);
```

Загальний прототип функції закриття файлу **fclose**:

```
fclose(файлова змінна);
```

Приклад, закриття файлу:

```
fclose();
```

Порядок виконання і звітування

1. Створити програму на мові C згідно варіанту використавши середовище програмування Dev-C++ 4.0:
 - для збереження даних про планшетні сканери описати структуру наступного вигляду:

```
struct scan_info{  
    char model [25];      //найменування моделі  
    int price;            // ціна  
    double x_size;        // горизонтальний розмір  
    області сканування  
    double y_size;        // вертикальний розмір  
    області сканування  
    int optr;             // оптичний дозвіл  
    int grey;             // число градацій сірого  
};
```
 - структура файлу: спочатку у файлі розміщається значення типу int, що визначає кількість зроблених у файлі записів; далі в кожній новій стрічці розміщуються записи про сканери.
2. Відкомпілювати та відлагодити програму.
3. Розробити набір тестів і перевірити роботу програми на них.
4. Відповісти на контрольні запитання.
5. Зробити висновки.
6. Звіт по лабораторній роботі має складатися з титульної сторінки, лістингів програм, висновків по роботі.

Варіанти завдань

Варіант 1.

1. Написати функцію, що записує у текстовий файл дані про сканер із приведеної структури.
2. Написати функцію, що витягає з цього файлу дані про сканер у структуру типу `scan_info`. Обов'язковий параметр - номер необхідного запису. Функція повинна повертати нульове значення, якщо читання пройшло успішно, і -1 у протилежному випадку.
3. Привести приклад програми, що створює файл із даними про сканери, (дані вводяться з клавіатури) - 6-8 записів, і виводить на дисплей дані про запитуваний запис.
4. Усі необхідні дані для функцій повинні передаватися їм як параметри. Використання глобальних змінних у функціях не допускається.

Варіант 2.

1. Написати функцію, що записує в текстовий файл дані про сканер з приведеної структури.
2. Написати функцію, що сортує записи в описаному вище текстовому файлі по одній з наступних характеристик: ціна або число градацій сірого. Обов'язковий параметр - ознака, що задає критерій сортування.
3. Привести приклад програми, що створює файл із даними про сканери (дані вводяться з клавіатури) не менш восьми записів і здійснює його сортування.
4. Усі необхідні дані для функцій повинні передаватися їм як параметри. Використання глобальних змінних у функціях не допускається.

Варіант 3.

1. Написати функцію, що сортує записи в описаному вище текстовому файлі по найменуванню моделі сканера.
2. Привести приклад програми, що створює файл із даними про сканери, (дані вводяться з клавіатури) не менш восьми записів, і здійснює його сортування.
3. Усі необхідні дані для функцій повинні передаватися їм як параметри. Використання глобальних змінних у функціях не допускається.

Варіант 4 .

1. Написати функцію, що динамічно виділяє пам'ять під масив структур (не менше шести елементів), заповнює його даними в режимі діалогу і записує масив у текстовий файл.
2. Написати функцію, що витягає дані про сканер з описаного вище текстового файлу в структуру типу scan info. Обов'язковий параметр - номер необхідного запису. Функція повинна повертати нульове значення, якщо зчитування пройшло успішно, і -1 в іншому випадку.
3. Привести приклад програми, що створює файл із даними про сканери, (дані вводяться з клавіатури), не менш восьми записів, і здійснює вивід на дисплей даних про необхідний запис.
4. Усі необхідні дані для функцій повинні передаватися їм як параметри. Використання глобальних змінних у функціях не допускається.

Варіант 5.

Запис може здійснюватися в будь яку позицію, при чому, якщо між записом, що вводиться, і останнім (чи початком файлу) існують пропуски, вони заповнюються нулями.

Написати функцію, що "ущільнює" описаний вище текстовий файл шляхом видалення з нього записів, що містять усі нулі.

Привести приклад програми, що створює файл із даними про сканери (дані вводяться з клавіатури) не менше шести записів, і здійснює його ущільнення.

Усі необхідні дані для функцій повинні передаватися їм у якості параметрів. Використання глобальних змінних у функціях не допускається.

Варіант 6.

1. Написати функцію, що динамічно виділяє пам'ять під масив структур (не менше шести елементів), заповнює його даними в режимі діалогу і записує масив у тестовий файл.
2. Написати функцію, що запитує дані про сканер у режимі діалогу і заміщає запис у текстовому файлі по заданому номеру. Обов'язковий параметр - номер запису, що заміщається. Функція повинна повертати нульове значення, якщо запис пройшов успішно, і -1 у іншому випадку.
3. Привести приклад програми, що створює файл із даними про сканери (дані вводяться з клавіатури) не менш восьми записів, і здійснює вставку нових даних про сканер.
4. Усі необхідні дані для функцій повинні передаватися їм як параметри. Використання глобальних змінних у функціях не допускається.

Варіант 7.

1. Написати функцію, що записує в текстовий файл дані про сканер із приведеної структури.

2. Написати функцію, що вводить дані про сканер із клавіатури в структуру типу `scan_info`, і якщо дані про цей сканер відсутні у файлі, розміщує вміст структури в кінець файлу; у протилежному випадку, видає відповідне повідомлення.

3. Привести приклад програми, що створює файл із даними про сканери (дані вводяться з текстового файлу) - 6-8 записів, і доповнює файл записами про 2-3 сканери, що вводяться з клавіатури.

4. Усі необхідні дані для функцій повинні передаватися їм як параметри. Використання глобальних змінних у функціях не допускається.

Варіант 8.

1. Написати функцію, що записує в текстовий файл дані про сканер з приведеної структури.

2. Написати функцію, що вводить дані про сканер із клавіатури в структуру типу `scan_info` і дописує у кінець файлу ці дані.

3. Привести приклад програми, що створює файл із даними про сканери (дані вводяться з текстового файлу) - 6-8 записів, і доповнює цей файл 1-2 новими записами, що вводяться з клавіатури.

4. Усі необхідні дані для функцій повинні передаватися їм як параметри. Використання глобальних змінних у функціях не допускається.

Варіант 9.

1. Написати функцію, що запитує кількість сканерів, інформація про які буде вводитися, динамічно виділяє пам'ять під масив структур відповідного розміру і заповнює його даними в режимі діалогу (з клавіатури).

2. Написати функцію, що записує даний масив у створюваний текстовий файл. Якщо ціна сканера менше 200, то дані про цей сканер у файл не записуються. Інформація про інші сканери міститься в бінарному файлі, при чому спочатку пишуться дані про всі сканери, ім'я яких починається з заголовної букви, а потім – із прописної.

3. Привести приклад програми, що створює файл із даними про сканери і здійснює вивід на дисплей даних про необхідний запис (або всіх, або по номеру).

4. Усі необхідні дані для функцій повинні передаватися їм як параметри. Використання глобальних змінних в функціях не допускається.

Підсумок

Після виконання лабораторної роботи студент повинен навчитися створювати програми, у яких здійснюється робота з файлами (зчитування/запис) на мові програмування C.

Контрольні питання

1. Що таке файл ?
2. Для чого призначений файл і його використання ?
3. Що таке символи ?
4. Що таке поле ?
5. Що називається найбільш популярний тип організації записів в файлі ?
6. Що називають групою зв'язаних між собою файлів ?
7. Назвіть функції мови C які використовуються для роботи з файлами ?

Контрольні вправи

1. В файлі операційної системи “*Test1.in*” є 5 стрічок, кожна з яких має довжину сторін прямокутників (значення довжини розділені двома пробілами).

Написати:

- визначення масиву структур для зберігання вказаних довжин сторін прямокутника, їхню площу і периметр;
- фрагмент програми для читання довжини сторони прямокутника із файла операційної системи “*Test1.in*”;
- фрагмент програми, яка обчислює і друкує площі і периметри прямокутників у файл операційної системи “*Test1.out*”.

2. В текстовому файлі “*Test2.in*” є інформація про квартиру, яка має наступний вигляд:

01	Комната	15
.....		
05	Кухня	5

Кожна стрічка має відомості по одній кімнаті: перші дві позиції – порядковий номер кімнати, третя позиція – “пробіл”, з поз. 4 починається назва кімнати довжиною не більше 15 символів, з поз. 21 – метраж кімнати.

Написати:

- визначення масиву структур для зберігання вказаних даних і фрагмент програми для читання даних про квартиру із файлу “*Test2.in*”;
- фрагмент програми для знаходження і друку загального метражу даної квартири в файл “*Test2.out*”

3. Що буде виведено у файл наступним фрагментом програми:

```
float    r;

int      i = 17;

r = 1.5f * 2.0elf;

fprintf ( stdout, "*r=%5.2e^%s^*i=%-+d\n*%-3s\n", r,
"_" , i, "*" );
```

4. Що буде виведено у файл наступним фрагментом програми:

```
float    r;

int      i = 17;

r = 1.5f * 2.0;

fprintf ( stdout, "*r=%5.2f^%5s^*i=%-+10d\n*%-30s\n",
r, "_" , i, "*" );
```

Джерела інформації

1. http://void.net.ua/The_C_Programming_Language.html.
2. http://publications.gbdirect.co.uk/c_book/.
3. <http://www.scribd.com/doc/16306895/Draft-ANSI-C-Rationale>.
4. <http://www.cplusplus.com/doc/tutorial/>.