

Міністерство освіти і науки України
Національний університет «Львівська політехніка»
Інститут телекомунікацій, радіоелектроніки та електронної техніки
кафедра «Радіоелектронні пристрої та системи»



Звіт з лабораторної роботи №19

з дисципліни «Програмування»

Підготував:
ст. групи АП-11
Фостик Віталій

Прийняла:
Гордійчук-Бублівська О.В.

Тема:

Дослідження способів організації потокового введення/виведення в мові програмування C.

Мета:

Дослідження способів створення, оновлення та оброблення файлів потокового введення/виведення даних у мові C.

Теоретичні відомості:

Зберігання даних у змінних і масивах є тимчасовим; всі ці дані втрачаються при завершенні роботи програми. Для постійного зберігання великих об'ємів даних використовуються файли. Комп'ютери зберігають дані на пристроях вторинної пам'яті, головним чином дискових пристроях. Комп'ютер обробляє елементи даних в двійковому вигляді, тобто у вигляді комбінацій нулів і одиниць.

Для програмістів обтяжливо працювати з даними низького рівня, якими є біти. Замість цього програмісти вважають за краще працювати з даними у вигляді десяткових цифр, букв, і спеціальних знаків, які називаються символами.

Подібно до того, як символи складаються з бітів, поля складаються з символів. Поле є групою символів, які передають значення.

Оброблювані комп'ютерами елементи утворюють ієрархію даних, в якій елементи даних стають більші за розміром і складніші за структурою в міру просування від бітів до символів (байтів), полів і так далі.

Існує багато способів організації записів у файлі. Найбільш популярний з них називається послідовним файлом, в якому записи, як правило, зберігаються в порядку, що визначається за певним правилом. У файлі нарахування заробітної плати, наприклад, записи зберігалися б впорядкованими за табельним номером.

Хід роботи:

1. Дослідити та дати пояснення прикладів, викладених нижче.
2. Розглянути функції форматного обміну з файлами `fprintf()`, `fscanf()` пояснити їх відмінності від функцій `printf()`, `scanf()`.
3. Виконати програму, що створює файл `proba.txt` і записує в нього символічні зображення чисел від 0 до 5 і їх кубів. Наступною програмою прочитати дані із файлу `proba.txt`. У звіті дати детальне пояснення роботи програм.
4. Виконати завдання згідно варіанта

7.	Задати 11 змінних цілого типу, записати їх в файл на диску, прочитати їх з файлу в масив розмірністю 11, масив вивести на екран.
----	--

Виконання роботи:

The image displays a C++ development environment with two code snippets and their corresponding program outputs.

Code Snippet 1:

```
#include <stdio.h>

int main() {
    FILE *in;
    int ch;

    if ((in = fopen("E://text.txt", "r")) != NULL) {
        while ((ch = getc(in)) != EOF) {
            putc(ch, stdout);
        }
        fclose(in);
    } else {
        printf("FILE test NOT OPEN\n");
    }

    return 0;
}
```

Execution Result 1:

```
C:\Users\Andrey\Desktop\Untitled1.exe
Hello World!
-----
Process exited after 0.0262 seconds with return value 0
Для продолжения нажмите любую клавишу . . .
```

Code Snippet 2:

```
1 #include <stdio.h>
2
3 int main() {
4     FILE *ff;
5     int base;
6
7     ff = fopen("E://sam.txt", "r");
8     fscanf(ff, "%d", &base);
9     fclose(ff);
10
11     ff = fopen("E://data.txt", "a");
12     fprintf(ff, "sam is %d.\n", base);
13     fclose(ff);
14
15     return 0;
16 }
17
```

Execution Result 2:

```
data.txt - Блокнот
Файл  Правка  Формат  Вид  Справка
sam is 0.
```

Compilation Results:

```
Курси  Журнал компіляції  Налашт
Compilation results...
-----
- Errors: 0
- Warnings: 0
- Output Filename: C:\Users\Andrey
- Output Size: 322,955078125 KiB
- Compilation Time: 0,16s
```

```

#include <stdio.h>
#define LINE 80

int main() {
    FILE *ff;
    char string[LINE];

    ff = fopen("E://opus.txt", "r");
    if (ff == NULL) {
        perror("ERROR OPEN FILE");
        return 1;
    }

    while (fgets(string, LINE, ff) != NULL) {
        puts(string);
    }

    fclose(ff);
    return 0;
}

```

C:\Users\Andrey\Desktop\Untitled1.exe

```

-----
Process exited after 0.02501 seconds with return value 0
Для продолжения нажмите любую клавишу . . .

```

```

#include <stdlib.h>
#include <stdio.h>

int main() {
    int f1, f2, f3, f4, f5;
    FILE *fp;

    fp = fopen("E:\\text.txt", "r");
    if (fp == NULL) {
        perror("ERROR OPEN FILE");
        return 1;
    }

    if (fscanf(fp, "%d%d%d%d%d", &f1, &f2, &f3, &f4, &f5) != 5) {
        fprintf(stderr, "ERROR READ FILE\n");
        fclose(fp);
        return 1;
    }

    printf("The values are %d, %d, %d, %d, %d.\n", f1, f2, f3, f4, f5);
    fclose(fp);

    return 0;
}

```

C:\Users\Andrey\Desktop\Untitled1.exe

```

The values are 10, 20, 30, 40, 50.

-----
Process exited after 0.02486 seconds with return value 0
Для продолжения нажмите любую клавишу . . .

```

```

1  #include <stdio.h>
2
3  int main() {
4      FILE *pf;
5      int n, nn, l;
6
7      if ((pf = fopen("E://proba.txt", "r")) == NULL) {
8          perror("proba.txt");
9          return 1;
10     }
11
12     for (l = 0; l <= 5; l++) {
13         fscanf(pf, "%d %d\n", &n, &nn);
14         printf("chuslo: %d, kyb: %d\n", n, nn);
15     }
16
17     fclose(pf);
18     return 0;
19 }
20

```

C:\Users\Andrey\Desktop\Untitled1.exe

```

chuslo: 0, kyb: 0
chuslo: 1, kyb: 1
chuslo: 2, kyb: 8
chuslo: 3, kyb: 27
chuslo: 4, kyb: 64
chuslo: 5, kyb: 125

```

```

-----
Process exited after 0.02603 seconds with return value 0
Для продолжения нажмите любую клавишу . . .

```

Завдання 7 варіант:

```

1  #include <stdio.h>
2
3  #define SIZE 11
4
5  int main() {
6      int num1 = 98, num2 = 11, num3 = 14, num4 = 99, num5 = 156, num6 = 74, num7 = 54, num8 = 88, num9 = 90, num10 = 100, num11 = 110;
7      FILE *file;
8      file = fopen("E://numbers.txt", "w");
9      if (file == NULL) {
10         perror("ERROR OPEN FILE");
11         return 1;
12     }
13     fprintf(file, "%d %d %d %d %d %d %d %d %d %d %d", num1, num2, num3, num4, num5, num6, num7, num8, num9, num10, num11);
14     fclose(file);
15     int numbers[SIZE];
16     file = fopen("E://numbers.txt", "r");
17     if (file == NULL) {
18         perror("ERROR OPEN FILE");
19         return 1;
20     }
21     printf("massyv:\n");
22     for (int i = 0; i < SIZE; i++) {
23         fscanf(file, "%d", &numbers[i]);
24         printf("%d\n", numbers[i]);
25     }
26     fclose(file);
27     return 0;
28 }
29

```

C:\Users\Andrey\Desktop\Untitled1.exe

```

massyv:
98
11
14
99
156
74
54
88
90
100
110

```

Журнал компіляції Налаштування Ресурси

Compilation results...

Errors: 0

```

-----
Process exited after 0.03207 seconds with return value 0
Для продолжения нажмите любую клавишу . . .

```

Контрольні запитання:

1. У мові програмування C файл використовується для зберігання та обробки даних на зовнішньому носії (наприклад, на диску). Файл може містити будь-який тип даних, включаючи текст, числа, зображення тощо. У C файл представляється вказівником на структуру даних, що містить інформацію про файл, таку як ім'я, розмір, тип тощо.

2. Основні режими відкриття файлу в мові C:

- "r" - відкриття файлу для читання.
- "w" - відкриття файлу для запису. Якщо файл вже існує, він буде очищений; якщо файл не існує, він буде створений.
- "a" - відкриття файлу для додавання. Дані додаються в кінець файлу, а не очищуються.
- "r+" - відкриття файлу для читання та запису.
- "w+" - відкриття файлу для читання та запису. Якщо файл вже існує, він буде очищений; якщо файл не існує, він буде створений.
- "a+" - відкриття файлу для читання та запису. Дані додаються в кінець файлу, а не очищуються.

3. Основні функції при роботі з файлами в мові C:

- `fopen()` - відкриває файл у вказаному режимі.
- `fclose()` - закриває відкритий файл.
- `fread()` - зчитує дані з файлу.
- `fwrite()` - записує дані у файл.
- `fseek()` - переміщує позицію у файлі.
- `ftell()` - повертає поточну позицію у файлі.
- `rewind()` - переміщує позицію у файлі на початок.

4. Способи позиціювання в файлі в мові C:

- Пошукове позиціювання: використовується `fseek()` або `rewind()` для переміщення позиції у файлі на заданий зсув або на початок.
- Поточне позиціювання: поточна позиція у файлі визначається автоматично при читанні або записі. Функції `ftell()` та `fgetpos()` можуть використовуватися для отримання поточної позиції.