

Міністерство освіти і науки України
Національний університет “Львівська Політехніка”

Лабораторна робота №19
З дисципліни
“Програмування частина 2”

Виконав:
Студент групи АП-11
ЧУМА ТАРАС

Прийняла

Гордійчук – Бублівська О. В.

Львів-2024

Тема роботи: Дослідження способів організації потокового уведення/виведення в мові програмування C.

Мета роботи: Дослідження способів створення, оновлення та оброблення файлів потокового уведення/виведення даних у мові C.

Теоретичні
відомості

Тема роботи: Дослідження способів організації потокового уведення/виведення в мові програмування C.

Зберігання даних у змінних і масивах є тимчасовим; всі ці дані втрачаються при завершенні роботи програми. Для постійного зберігання великих

Тема роботи: Дослідження способів організації потокового об'ємів уведення/виведення в мові програмування C.

дані 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.

Стандартна бібліотека підтримує численні функції читання даних з файлів і запису даних у файли. Функція `fgetc`, подібно `getchar`, прочитує з файлу один символ. Функція `fgetc` отримує як аргумент вказівник на `FILE` для файлу, з якого прочитуватиметься символ. Виклик `fgetc(stdin)` читає один символ з `stdin` – стандартного уведення. Такий виклик еквівалентний виклику `getchar()`. Функція `fputc`, подібно `putchar`, записує один символ у файл. Як аргумент функція отримує символ, який повинен бути записаний. Виклик функції – `fputc('a', stdout)` записує символ 'a' в `stdout` – стандартний вивід. Такий виклик цієї функції еквівалентний `putchar('a')`.

Отже мова C розглядає файл як структуру з ім'ям шаблону (тегом) `FILE`, що вказує на ім'я файлу, його статус та текучу позицію. Режими функції `fopen`

– знаходяться в `stdio.h`:

“r” – відкрити для читання (файл має існувати);

“w” – створити для запису (якщо файл існує, вміст губиться);

“a” – відкрити для додавання в існуючий файл (файл створюється, якщо не існує);

“rb” – відкрити двійковий файл для читання;

“wb” – відкрити двійковий файл для запису;

“r+” – відкрити файл для читання і запису (файл має існувати);

“w+” – створити файл для читання і запису (якщо файл існує, вміст губиться).

Приклад 1

```
1  #include <stdio.h>
2
3  int main() {
4      FILE *in;
5      int ch;
6
7      if ((in = fopen("proba", "r")) != NULL) {
8          while ((ch = getc(in)) != EOF) {
9              putc(ch, stdout);
10             }
11             fclose(in);
12         } else {
13             printf("Файл proba не відкривається\n");
14         }
15
16         return 0;
17     }
```

Файл proba не відкривається

```
...Program finished with exit code 0
Press ENTER to exit console.
```

main.c

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

```
...Program finished with exit code 0
Press ENTER to exit console.
```

return

//B

```
1  #include <stdio.h>
2  #define LINE 80
3
4  int main() {
5      FILE *ff;
6      char string[LINE];
7
8      ff = fopen("opus", "r");
9      while (fgets(string, LINE, ff) != NULL) {
10         puts(string);
11     }
12     fclose(ff);
13
14     return 0;
15 }
```

...Program finished with exit code 0
Press ENTER to exit console.

//r

```

1  #include <stdlib.h>
2  #include <stdio.h>
3
4  int main() {
5      int f1, f2, f3, f4, f5;
6      FILE *fp;
7
8      fp = fopen("C:\\temp\\sample.txt", "r");
9      if (fp == NULL) {
10         printf("Помилка відкриття файлу\n");
11         return 1;
12     }
13
14     fscanf(fp, "%d\n%d\n%d\n%d\n%d\n", &f1, &f2, &f3, &f4, &f5);
15     printf("Значення: %d, %d, %d, %d, %d\n", f1, f2, f3, f4, f5);
16
17     fclose(fp);
18
19     return 0;
20 }
21

```

Помилка відкриття файлу

Приклад 2

Функції форматного обміну з файлами `fprintf()` і `fscanf()` у мові C дозволяють записувати та зчитувати дані з файлів у відповідності з вказаними форматами, аналогічно функціям `printf()` і `scanf()`, які працюють зі стандартним введенням та виведенням.

Отже, ось їх відмінності:

`fprintf()` та `fscanf()`:

`fprintf()`: Ця функція використовується для запису даних у файл з використанням заданих форматів. Вона приймає файловий вказівник та список аргументів, аналогічно `printf()`.

`fscanf()`: Ця функція використовується для зчитування даних з файлу за заданими форматами. Вона приймає файловий вказівник та список аргументів, аналогічно `scanf()`.

Відмінності від printf() та scanf():

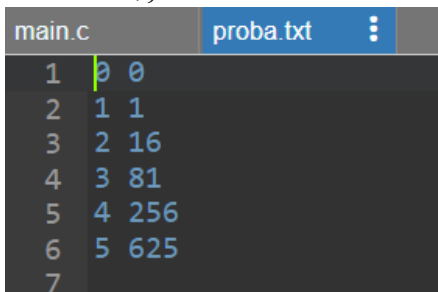
printf() та scanf(): Ці функції працюють зі стандартним введенням та виведенням (консоль).

fprintf() та fscanf(): Ці функції працюють з файлами, що дозволяє записувати та зчитувати дані з файлів.

Приклад 3

```
// Запис файлу
#include <stdio.h>
int main() {
    FILE *pf; // Вказівник на файл
    int k;
    // Відкриття файлу для запису
    if ((pf = fopen("proba.txt", "w")) == NULL) { // Перевірка чи відкриття файлу
        пройшло успішно
        perror("proba.txt"); // Виведення повідомлення про помилку
        return 1; // Повернення коду помилки
    }
    // Запис даних у файл
    for (k = 0; k <= 5; k++) {
        fprintf(pf, "%d %d\n", k, k*k*k*k); // Запис значень у файл
    }
    fclose(pf); // Закриття файлу
    return 0;}

```



```
// Читання даних із файлу proba.txt
#include <stdio.h>
int main() {
    FILE *pf; // Вказівник на файл
    int n, nn, l;
    // Відкриття файлу для читання
    if ((pf = fopen("proba.txt", "r")) == NULL) { // Перевірка чи відкриття файлу
        пройшло успішно
        perror("proba.txt"); // Виведення повідомлення про помилку
        return 1; } // Повернення коду помилки
    }
    // Читання даних з файлу
    for (l = 0; l <= 5; l++) {

```

```
fscanf(pf, "%d %d\n", &n, &nn); } // Зчитування значень з файла  
fclose(pf); // Закриття файлу  
return 0;}
```

```
Read values: 0 0  
Read values: 1 1  
Read values: 2 16  
Read values: 3 81  
Read values: 4 256  
Read values: 5 625
```

```
1  #include <stdio.h>  
2  #include <stdlib.h>  
3  
4  int main() {  
5  
6      int array[13] = {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12};  
7      FILE *file;  
8  
9  
10     file = fopen("numbers.bin", "wb");  
11     if (file == NULL) {  
12         printf("Помилка відкриття файлу для запису.\n");  
13         return 1;  
14     }  
15     fwrite(array, sizeof(int), 13, file);  
16     fclose(file);  
17  
18  
19     int a, b, c, d, e, f, g, h, i, j, k, l, m;  
20  
21  
22     file = fopen("numbers.bin", "rb");  
23     if (file == NULL) {  
24         printf("Помилка відкриття файлу для читання.\n");  
25         return 1;  
26     }  
27     fread(&a, sizeof(int), 1, file);  
28     fread(&b, sizeof(int), 1, file);  
29     fread(&c, sizeof(int), 1, file);  
30     fread(&d, sizeof(int), 1, file);  
31     fread(&e, sizeof(int), 1, file);  
32     fread(&f, sizeof(int), 1, file);  
33     fread(&g, sizeof(int), 1, file);
```



```
34 fread(&h, sizeof(int), 1, file);
35 fread(&i, sizeof(int), 1, file);
36 fread(&j, sizeof(int), 1, file);
37 fread(&k, sizeof(int), 1, file);
38 fread(&l, sizeof(int), 1, file);
39 fread(&m, sizeof(int), 1, file);
40 fclose(file);
41
42
43 printf("%d %d %d %d %d %d %d %d %d %d %d %d %d\n", a, b, c, d, e, f, g, h, i, j, k, l, m);
44
45 return 0;
46 }
47
```

input

0 1 2 3 4 5 6 7 8 9 10 11 12

..Program finished with exit code 0
Press ENTER to exit console.

Відповіді на контрольні запитання

1) Суть поняття файлу в мові програмування C:

У мові програмування C файл - це послідовність даних, яка зберігається на зовнішньому носії, такому як жорсткий диск, флеш-накопичувач або інше зберігання. Файли використовуються для зберігання та обміну даними між програмами та користувачами.

2) Основні режими відкриття файлу в мові C:

В мові C можна відкрити файл у таких режимах:

"r": Відкриття для читання. Файл має існувати.

"w": Відкриття для запису. Якщо файл існує, його вміст буде видалено; якщо файл не існує, він буде створено.

"a": Відкриття для додавання. Дані додаються до кінця файлу. Файл, якщо він існує, не буде видалено.

"r+": Відкриття для читання та запису. Файл має існувати.

"w+": Відкриття для читання та запису. Якщо файл існує, його вміст буде видалено; якщо файл не існує, він буде створено.

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

Основні функції при роботі з файлами в мові C включають такі:

fopen(): Відкриття файлу.

fclose(): Закриття файлу.

fread(): Читання з файлу.

fwrite(): Запис до файлу.

fseek(): Пошук позиції у файлі.

ftell(): Отримання поточної позиції у файлі.

feof(): Перевірка кінця файлу.

4) Способи позиціювання в файлі в мові C:

У мові С для позиціювання в файлі можна використовувати функцію `fseek()`, яка дозволяє встановлювати позицію вказівника файлу у відповідності з вказаними параметрами. Параметри `fseek()` включають:

Початок файлу (`SEEK_SET`).

Поточна позиція (`SEEK_CUR`).

Кінець файлу (`SEEK_END`).