

# Обмін даними з файлами

У компілятор мови С не включено спецзасобів для введення/виведення даних, тому обмін даними реалізовано через бібліотечні функції.

Бібліотеки більшості систем програмування мови С підтримують функції для введення/виведення даних на трьох рівнях:

- високорівневе, т.зв. потокоорієнтоване введення/ виведення - використовує однаковий підхід у програмуванні обміну даними з файлами та зовнішніми пристроями і єдиний інтерфейс. Всі файли та дані з пристроїв розглядаються як неструктуровані набори байтів – *потоки*.

Прототипи функцій записані в заголовному файлі `<stdio.h>`

- функції введення/виведення низького рівня базуються на засобах обміну даними, що властиві конкретній операційній системі. Дані функції не виконують форматування даних і не застосовують буферизації. Функції блокоорієнтовані і забезпечують вигоду у швидкодії тоді, коли обсяг блоку даних, що передається за одну операцію, кратний ємкості сектора диска (512 б). Прототипи даних функцій зберігаються в *<io.h>*.
- функції консольного введення/виведення доповнюють можливості високорівневих функцій щодо введення з клавіатури і керування текстовим режимом виведення інформації. Прототипи оголошені в заголовному файлі *<conio.h>*.

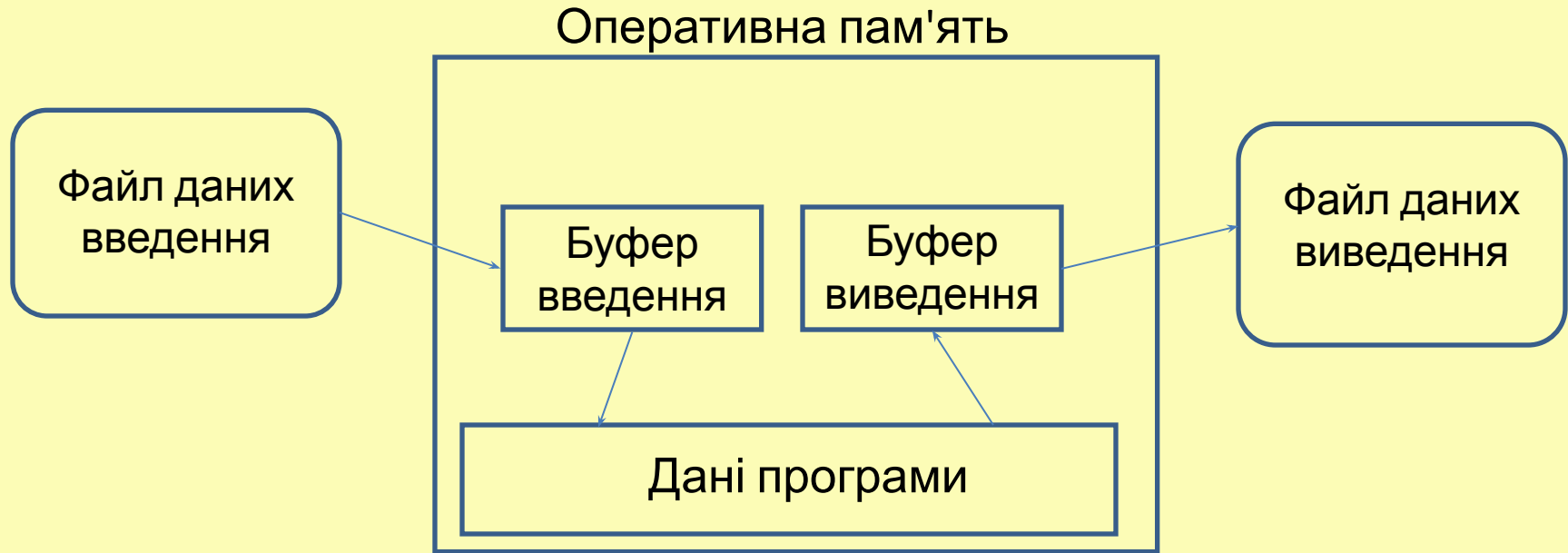
*Файлом* вважається іменована сукупність даних, розташованих на зовнішньому носії, а також термінальні пристрої (клавіатура, принтер ...).

Для уніфікації процесів файлового обміну даними у функціях високого рівня використовують поняття *поток*.

*Потік (stream)* – послідовність байтів, що надходять від певного логічного пристрою (файлу) або передаються у цей файл (пристрій).

Для узагальнення обміну даними у процесах введення/виведення здійснюється проміжна буферизація даних. Для кожного відкритого файлу в оперативній пам'яті створюється буфер обміну заданої ємності.

Буферизація мінімізує кількість звертань до фізичних пристроїв, які найбільше гальмують процеси введення/виведення даних.



Високорівневий потоковий обмін даними – це обмін байтами з фізичними файлами і логічними пристроями через систему буферизації, що дає змогу опрацьовувати дані різних форматів і розмірів.

# Відкриття/закриття потоків

Перед введенням даних з конкретного файлу або записом даних у файл, необхідно створити потік, пов'язаний з цим фізичним файлом. Створення потоку реалізує функція відкриття файлу, прототип якої оголошено так:

***FILE \*fopen (char \*file\_name, char \*fmode);***

Параметри функції є вказівниками на символльні рядки.

*file\_name* – ім'я файлу, *fmode* – режим обміну даними.

За умови успішного відкриття потоку створюється спеціальна структура зі шаблоном *FILE* і функція повертає її адресу. Якщо ж потік відкрити

Основні режими відкриття файла (другий параметр *fmode* функції *fopen ( )* ):

- "r" – (від англ. read). Файл тільки для читання;
- "w" - (від англ. write). Файл тільки для запису;
- "a" – (від англ. append). Файл тільки для доповнення;
- "r+" - файл для читання з можливістю запису в нього;
- "w+" - файл для запису з можливістю читання з нього;
- "a+" - файл для доповнення з можливістю читання.

Додатково в параметрі *fmode* можна задавати текстовий (t ) чи бінарний (b) режим відкриття потоку за замовчуванням встановлюється

Приклад:

```
FILE *p1, *p2;
```

```
p1=fopen ("ishodn.dan", "r");
```

```
p2=fopen ("rezult.dan", "w");
```

Для закриття потоків використовується функція

```
int fclose (FILE *fp);
```

*fp* - вказівник на потік, який треба закрити. При успішному використанні функції вона повертає значення 0.

Приклад:

```
fclose(f);
```

# Стандартні потоки, перескерування потоків

На початку виконання кожної С-програми відкриваються стандартні потоки:

***stdin*** – потік введення, який пов'язується з клавіатурою;

***stdout*** – потік виведення даних на екран;

***stderr*** – потік повідомлень про помилки, скеровуються на екран.

Стандартні потоки можна перескерувувати (перепризначати), пов'язувати їх із заданим файлом чи пристроєм.

Перескерування потоків виконує функція

***FILE \*freopen (char \*fname, char \*fmode, FILE \*fp);***

Функція пов'язує потік *fp* з файлом *fname*. Режим



Значення та форми завдання параметрів *fname* і *fmode* такі ж, як для функції *foopen* (). За умови успішного виконання функція повертає вказівник на створений потік, а в разі помилки – NULL.

*Приклад: перше звертання до puts() викликає виведення повідомлення на екран, а друге – у файл.*

```
#include <stdio.h>
```

```
void main()
```

```
{
```

```
    puts("Виведення на екран");
```

```
    freopen("example.xz", "w", stdout);
```

```
    puts("Виведення в файл");
```