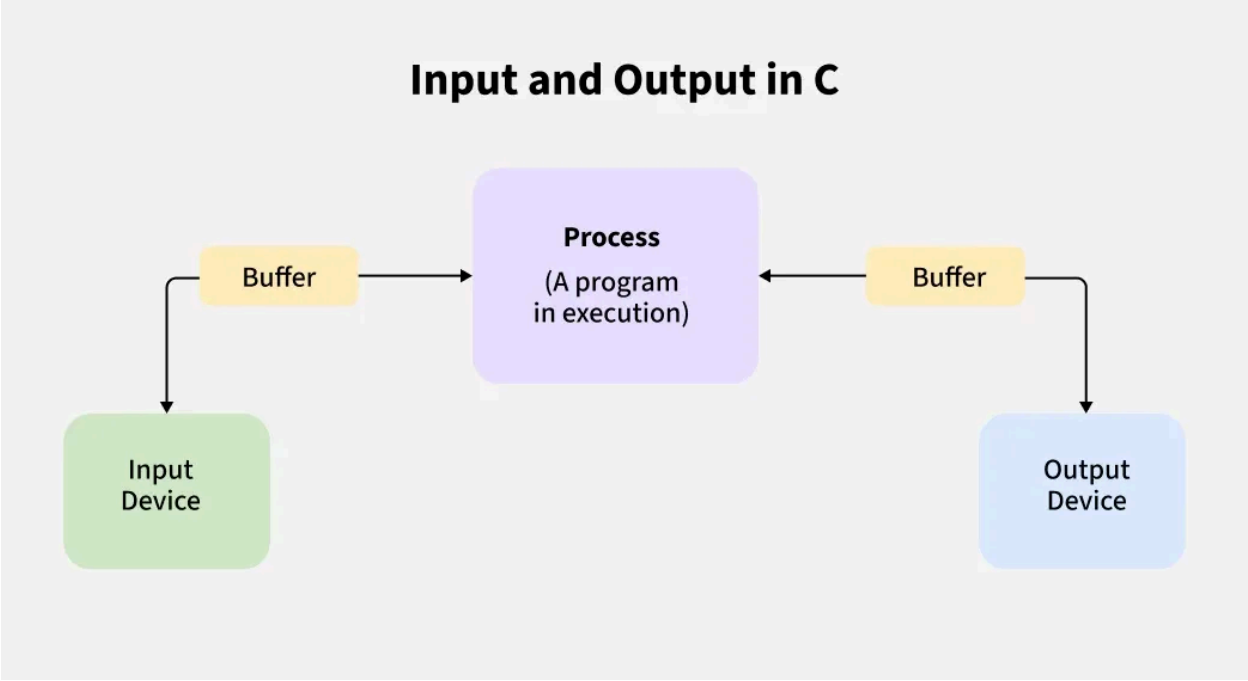
Input and Output in C

Input and output are fundamental operations in C programming that enable interaction between the program and the external world. These operations allow a program to take user input (via the keyboard) and display results or messages (on the monitor or console).

Input and Output Streams

- **Input Stream**: Data flows from the input device (keyboard) to the program.
- **Output Stream**: Data flows from the program to the output device (monitor).

The input device (keyboard) and output device (monitor) are relatively slow compared to the CPU, necessitating the use of **buffers** to manage the flow of data efficiently, avoiding character-by-character processing.



Streams in C

When a C program starts running, it interacts with three predefined streams:

- **stdin (Standard Input Stream)**: For input operations (keyboard).
- **stdout (Standard Output Stream)**: For output operations (monitor).
- **stderr (Standard Error Stream)**: For error messages.

These streams are open and ready for use by default. The program writes or reads data in the form of characters or bytes, depending on the function being used.

Common Input and Output Functions in C

C provides a set of functions to take input and print output as a part of its standard features inside <**stdio.h**> header file.

1. scanf

scanf is used for reading data from the standard input device. It is a general-purpose function that can handle various data types like integers, floats, characters, and strings. But it by default does not read strings with spaces.

2. printf

printf is used for writing data to the standard output device. It can be used for displaying integers, floats, characters, and strings.

3. fgets

fgets reads strings, including those with spaces, from the standard input or file streams. Originally designed for file input but can also be used for standard input.

4. gets (Deprecated)

gets was previously used to read strings from standard input but is unsafe and no longer recommended due to security issues.

5. puts

puts is used to write strings to the standard output device. It is simpler to use than printf for string outputs.

6. getchar and putchar

- **getchar**: Reads a single character from the standard input.
- **putchar**: Writes a single character to the standard output.

7. getc and putc

Extended versions of getchar and putchar that can operate on any stream (e.g., files or stdin/stdout).

File Streams vs. Standard Streams

C uses the same syntax for reading from and writing to files as it does for standard input/output operations. By providing a file stream instead of stdin or stdout, functions like fgets, fputs, getc, and putc can seamlessly work with files.

Error Stream – stderr

The stderr stream is used for error messages. It is unbuffered, meaning messages are displayed immediately without waiting for the buffer to fill.