fgets() in C

The **fgets()** function in C is a reliable way to read strings, especially when dealing with inputs containing spaces. Unlike scanf(), which stops reading at spaces, fgets() reads the entire string, including spaces, up to a specified length.

Syntax

> **fgets**(name, len, stream);

where,

- **name**: Name of the string variable where the input string will be stored.
- **len**: Length upto which the string is to be read.
- **stream**: Input stream, often **stdin** (standard input) for user input.

In C, strings are null terminated, meaning an additional character (\0) is added to the end of the string. So, **len** parameter should always be 1 + the desired size and it includes the space for the null terminator (\0).

fgets() for User Input

Originally, fgets() was designed to read input from files. However, by providing stdin as the input stream, it can read from standard input, like user input from the keyboard. This makes it versatile for various use cases.

Example

C

```c
#include <stdio.h>

int main() {
    char name[100];
    printf("Enter Your Name: \n");
    fgets(name, 100, stdin);
    printf("Hi %s,\n", name);
    printf("Welcome to GfG");
    return 0;
}
```

Output

```
Enter Your Name:
Sandeep Jain (entered by user)
Hi Sandeep Jain,
Welcome to GfG
```

In the above code, if the user inputs a name shorter than 99 characters, the input is read as-is and stored in the name variable. If the input exceeds the specified limit, only the first 99 characters are stored, and the rest is ignored.

Why Use fgets()?

In the past, the gets() function was commonly used to read strings, but it was considered unsafe. It lacked a mechanism to limit the number of characters being read, which could lead to buffer overflows and make programs vulnerable to attacks. For this reason, gets() has been deprecated in modern versions of C.

The fgets() function addresses these issues by allowing programmers to specify the maximum number of characters to read, including the null terminator (\0). This safety feature makes it a preferred choice for reading strings.