ARYATECHNO

C File Handling

Topics : C
Written on April 13, 2023

File handling is an important aspect of programming, allowing programs to read from and write to files on a computer's file system. In C programming language, file handling is accomplished using a set of functions that are defined in the stdio.h library. Here are some of the commonly used functions for file handling in C:

1. `fopen()`: opens a file and returns a pointer to a FILE structure, which can be used to access the file.

```
FILE *fopen(const char *filename, const char *mode);
```

2. `fclose()`: closes a file that was previously opened with `fopen()`.

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

3. `fread()`: reads a block of data from a file into a buffer.

```
size_t fread(void *ptr, size_t size, size_t count, FILE *stream);
```

4. `fwrite()`: writes a block of data from a buffer to a file.

```
size_t fwrite(const void *ptr, size_t size, size_t count, FILE *stream);
```

5. `fprintf()`: writes a formatted string to a file.

```
int fprintf(FILE *stream, const char *format, ...);
```

6. `fscanf()`: reads formatted input from a file.

```
int fscanf(FILE *stream, const char *format, ...);
```

7. `fgets()`: reads a line of text from a file into a buffer.

```
char *fgets(char *str, int n, FILE *stream);
```

8. `fputs()`: writes a string to a file.

```
int fputs(const char *str, FILE *stream);
```

To use these functions, you first need to open the file using `fopen()`, which returns a pointer to a FILE structure. You can then use other functions like `fread()`, `fwrite()`, and `fprintf()` to read from or write to the file, depending on your needs. Once you are done using the file, you should close it using `fclose()`.

Here's an example of how to read and write from a file in C:

```c
#include <stdio.h>

int main() {
    // Open the file for writing
    FILE *file = fopen("data.txt", "w");
    if (file == NULL) {
        printf("Error opening file.\n");
        return 1;
    }
    // Write some data to the file
    fprintf(file, "Hello, world!\n");
    fprintf(file, "This is a test.\n");
    // Close the file
    fclose(file);
    // Open the file for reading
    file = fopen("data.txt", "r");
    if (file == NULL) {
        printf("Error opening file.\n");
        return 1;
    }
    // Read the data from the file and print it to the console
    char buffer[1024];
    while (fgets(buffer, 1024, file)) {
        printf("%s", buffer);
    }
    // Close the file
    fclose(file);
    return 0;
}
```

In this example, the program first opens a file named `data.txt` for writing using `fopen()` with the `"w"` mode. It then writes some data to the file using `fprintf()`. Once it is finished writing to the file, it closes the file using `fclose()`.

The program then opens the same file for reading using `fopen()` with the `"r"` mode. It reads the data from the file using `fgets()` and prints it to the console using `printf()`. Once it is finished reading from the file, it closes the file using `fclose()`.

This program would output:

```
Hello, world!
This is a test.
```