

File Handling in 'C'

File handling in C allows programs to interact with files stored on a computer's permanent storage, enabling data persistence beyond the program's execution. This involves operations like creating, opening, reading, writing, and closing files.

Key Concepts and Functions:

- File Pointer (`FILE *`):

A pointer to a `FILE` object, declared in `stdio.h`, which represents an open file and keeps track of the current position within it.

- `fopen()`:

Opens a file. It takes two arguments: the file name (or path) and the mode.

- **Modes:**

- `"r"`: Read mode (file must exist).
- `"w"`: Write mode (creates a new file or truncates an existing one).
- `"a"`: Append mode (creates a new file or appends to an existing one).
- `"rb"`, `"wb"`, `"ab"`: Binary versions of the above modes.
- `"r+"`, `"w+"`, `"a+"`: Read and write modes.

`fclose()`:

Closes an opened file, ensuring data is saved and resources are released.

Writing to Files:

- `fprintf()`: Writes formatted output to a file (similar to `printf()`).
- `fputc()`: Writes a single character to a file.
- `fputs()`: Writes a string to a file.
- `fwrite()`: Writes blocks of data to a binary file.

Reading from Files:

- `fscanf()`: Reads formatted input from a file (similar to `scanf()`).
- `fgetc()`: Reads a single character from a file.
- `fgets()`: Reads a line (or a specified number of characters) from a file.
- `fread()`: Reads blocks of data from a binary file.

File Pointer Manipulation:

- `fseek()`: Sets the file pointer to a specific position.
- `ftell()`: Returns the current position of the file pointer.
- `rewind()`: Resets the file pointer to the beginning of the file.

Example (Writing and Reading a Text File):

C

```
#include <stdio.h>

int main() {
    FILE *fptr;
    char text[100];

    // Writing to a file
    fptr = fopen("example.txt", "w");
    if (fptr == NULL) {
        printf("Error opening file for writing!\n");
        return 1;
    }
    fprintf(fp, "Hello, C File Handling!\n");
    fputs("This is another line.\n", fp);
    fclose(fp);
    printf("Data written to example.txt\n");

    // Reading from a file
    fptr = fopen("example.txt", "r");
    if (fptr == NULL) {
        printf("Error opening file for reading!\n");
        return 1;
    }
    while (fgets(text, sizeof(text), fp) != NULL) {
        printf("%s", text);
    }
    fclose(fp);
    printf("Data read from example.txt\n");

    return 0;
}
```