Course
ELEC-A7100
Course materials
Your points
Microsoft Teams
Code Vault

This course has already ended.

« 1 I/O - streams Course materials 3 Read and Write operations »
ELEC-A7100 / 9. I/O Streams / 2 I/O-stream open and close

Ninth round tasks as a `ZIP file.`

I/O-stream open and close¶

I/O stream can be accessed using the `FILE*` data type. It is an abstract data type for accessing the stream using the specified functions in the C library. A new FILE reference can be obtained using the **fopen** function that opens a named I/O stream (e.g. a file), and returns a FILE* pointer that allows accessing the I/O stream. After a successful fopen operation, the stream can be accessed by using the I/O functions. When the stream is not needed anymore, it should be closed with **fclose** function that disassociates the FILE variable from the I/O stream, and releases the resources used by stream management. I/O streams can be associated with any kind of I/O resource, but in the following we mostly work with files.

Here are the basic functions to use a stream. These are defined in the **stdio.h** header. More information about the function can be found from the man pages.

- `FILE *fopen(const char *path, const char *mode)` opens the given file for input and/or output. The **path** parameter is a string that contains filename (possibly including path) of the file to be opened. mode is also a string that consists of combination of letters that define the mode in which file is opened: "r" means the file is opened for reading; "w" means the file is opened for writing (erasing the previous content of file); "r+" opens the file for both reading and writing. In these cases, the file operation starts from the beginning of file, but with "a" mode, file can be opened for writing starting from the end of the file. It is used for example for appending new content on top of previous one. "a+" does the same, but allows also file reading. The function returns pointer to a new FILE object that can be used by the following functions to access the file, or NULL if opening the file did not succeed. Specifier "b" means that the content will be handled as binary data.
- `int fclose(FILE *fp)` closes the given file, after which it cannot be accessed anymore. The return value is 0 if closing was successful or constant EOF (that equals to -1) on failure.

Try to copy the below program to your own machine and execute it.

```
#include <stdio.h>

int main()
{
    FILE* file_ptr = NULL;
    char* file_name = "test_file.txt";

    /* Open file */
    file_ptr = fopen(file_name, "r");
    if (NULL != file_ptr)
    {
        printf("File opened successfully\n");
        fclose(file_ptr);
    }
    else
    {
        printf("File open error\n");
    }
    return 0;
}
```

If you get "File open error", find out the reason. Execute the below program and find out why it does not throw the same error.

```
#include <stdio.h>

int main()
{
    FILE* file_ptr = NULL;
    char* file_name = "test_file.txt";

    /* Open file */
    file_ptr = fopen(file_name, "w");
    if (NULL != file_ptr)
    {
        printf("File opened successfully\n");
        fclose(file_ptr);
    }
    else
    {
        printf("File open error\n");
    }
    return 0;
}
```