



به نام خدا
دانشکده‌ی مهندسی برق و کامپیوتر دانشکده فنی دانشگاه تهران
مبانی کامپیوتر و برنامه نویسی



اساتید:
دکترمرادی، دکتر هاشمی

عنوان:
کار با فایل در زبان C

نیمسال اول
1402-03

کار با فایل:

برای کار با فایل‌ها در زبان C باید ابتدا یک اشاره‌گر از نوع FILE بسازیم. سپس می‌توانیم با استفاده از تابع fopen یک فایل را از حافظه کامپیوتر باز کرده و به محتوای آن دسترسی پیدا کنیم. مقدار بازگشتی این تابع اشاره‌گر از نوع FILE است. این تابع به صورت زیر به کار می‌رود:

```
FILE *myfile = fopen("filename", "mode");
```

همانطور که مشاهده می‌کنید تابع fopen دو ورودی دریافت کرده که ورودی اول آدرس و نام فایل با فرمت char* و ورودی دوم نوع رفتار با فایل را مطابق جدول زیر تعیین می‌کند:

File mode	Meaning of mode	During inexistence of file
"r"	open text file for reading	If the file <u>does not</u> exist, fopen() returns NULL .
"w"	open text file for writing	If the file <u>exists</u> , its contents are overwritten . If the file <u>does not</u> exist, it will be created .
"a"	Open a text file in append mode for writing at the end of the file	If the file <u>does not</u> exist, it will be created .
"r+"	open text file for both reading and writing	If the file <u>does not</u> exist, fopen() returns NULL .
"w+"	open text file for both reading and writing	If the file <u>exists</u> , its contents are overwritten . If the file <u>does not</u> exist, it will be created .
"a+"	open text file for both reading and appending	If the file <u>does not</u> exist, it will be created .

توجه: در صورتی که بخواهیم خواندن و نوشتن در فایل به صورت باینری انجام شود، از عبارات زیر استفاده می‌کنیم:

File mode	Meaning of mode
"rb"	open binary file for reading
"wb"	open binary file for writing
"ab"	open for append. (data is added to the end of the file.)

- برای کسب اطلاعات بیشتر در مورد تابع fopen() پیشنهاد می‌شود مطالب [این لینک](#) را مطالعه کنید.

خواندن و نوشتن در فایل:

برای نوشتن در فایل می‌توان از توابع `fwrite` و `fprintf` و برای خواندن از فایل نیز می‌توان از توابع `fread` و `fscanf` استفاده کرد. در ادامه به بررسی اجمالی این توابع می‌پردازیم:

توابع `fread` و `fwrite` یک قطعه (block) از اطلاعات را در فایل می‌نویسند یا می‌خوانند. به این منظور این توابع به عنوان ورودی اول یک اشاره‌گر به ابتدای یک آرایه، ورودی دوم اندازه‌ی هر قسمت از `block`، ورودی سوم طول قطعه و ورودی چهارم اشاره‌گر از نوع `FILE` دریافت می‌کنند. سپس به اندازه‌ی (اندازه‌ی هر قسمت \times طول) از آدرس اشاره‌گر به آرایه آغاز کرده و در فایل می‌نویسند (یا می‌خوانند). به قطعه کد زیر توجه کنید:

```
#include <stdio.h>
int main() {
    FILE *myfile = fopen("out.txt", "wb");
    char *str = "Hello!?!";
    fwrite(str, sizeof(char), 5, myfile);
    fclose(myfile);
    return 0;
}
```

توجه:

- حتما باید در انتهای برنامه فایل‌های باز شده را با استفاده از دستور `fclose` ببندیم.
- در صورتی که مانند کد بالا در قسمت آدرس تنها اسم فایل را ذکر کنیم، مرجع آدرس فایل پوشه محل قرارگیری پروژه و برنامه است.
- پس از استفاده از توابع `fread` و `fwrite` پیمایش‌کننده‌ی فایل در محل جدیدی قرار می‌گیرد. این محل اولین محل پس از محتوای خوانده یا نوشته شده است. برای جلوگیری از خطا از تابع `fseek` استفاده می‌کنیم که در ادامه به بررسی آن می‌پردازیم.
- توابع `fscanf` و `fprintf` همانند توابع `scanf` و `printf` عمل می‌کنند با این تفاوت که ورودی اول آنها از نوع اشاره‌گر به فایلی است که قبلا باز کرده‌ایم.

توصیه: برای آشنایی بیشتر با عملکرد این توابع به [این لینک](#) مراجعه کنید!

1- انجام دهید!

اکنون قصد داریم برنامه‌ای بنویسیم که 10 حرف اول الفبای انگلیسی را در یک فایل متنی با حروف بزرگ و با فاصله از هم ذخیره نماید. خروجی باید یک فایل txt با محتوای زیر باشد:

A B C D E F G H I J

بدین منظور، برنامه‌ی زیر را تکمیل نمایید.

```
#include <stdio.h>
#include <stdlib.h>
#define SIZE_OF_ARRAY ...
#define OUTPUT_FILE_ADDRESS "MyArray.txt"
char* array_producer()
{
    char *array = (char *)calloc( SIZE_OF_ARRAY , SIZE_OF_ARRAY * sizeof(char));
    for(int i=0;i<10;i++){
        array[...]=(...);
    }
    return array;
}
int main()
{
    char* the_array = array_producer();
    FILE *output;
    output = fopen(... , "...");
    fwrite(... , ... , SIZE_OF_ARRAY , ... );
    fclose(output);
    return 0;
}
```

راهنمایی: برای مقداردهی آرایه، از کد اسکی حروف استفاده کنید. (A=65)


نتیجه را به دستیاران آموزشی نشان دهید. (قسمت 1) ✓

✚ در مورد مشکلاتی که در صورت عدم استفاده از fclose ممکن است اتفاق بیفتد، در اینترنت تحقیق کنید.

2- انجام دهید!

حال هدف نوشتن برنامه‌ای است تا فایل‌ی را که در قسمت قبل ساختیم بخواند و متن داخل آن را به صورت معکوس در فایل دیگری به نام `output.txt` بنویسد. به این منظور قطعه کدهای زیر را کامل کنید:

```
#include <stdio.h>
#include <stdlib.h>
#define SIZE_OF_ARRAY ...
#define INPUT_FILE_ADDRESS "MyArray.txt"
#define OUTPUT_FILE_ADDRESS "output.txt"
char *read_input_file()
{
    char *in_order_array = (char *)malloc(SIZE_OF_ARRAY * sizeof(char));
    FILE *input = fopen(INPUT_FILE_ADDRESS, ...);
    fread(...);
    ... ;
    return in_order_array;
}
char *reverse_array(char *in_order_array)
{
    char *reversed_array = (char *)malloc(SIZE_OF_ARRAY * sizeof(char));
    for (int i = 0, j = SIZE_OF_ARRAY-1 ; i < SIZE_OF_ARRAY; i++, j--)
    {
        *(reversed_array + i) = *(in_order_array + j);
    }
    return reversed_array;
}
void write_reversed_array_in_file(char *in_order_array)
{
    char *reversed_array = reverse_array(in_order_array);
    FILE *output;
    output = fopen(OUTPUT_FILE_ADDRESS, ... );
    fwrite(...);
    ... ;
}
int main() {
    char *in_order_array = read_input_file();
    write_reversed_array_in_file(in_order_array);
    return 0;
}
```

نتیجه را به دستیاران آموزشی نشان داده و نحوه کار برنامه را به طور کامل توضیح دهید.(قسمت 2) 

تابع feof و getc :

در کار با فایل‌ها، انتهای فایل با مقدار ثابتی (یک کاراکتر) به نام EOF معرفی می‌شود. همواره می‌توان با بررسی برابری آخرین کاراکتر دریافت شده و ثابت EOF رسیدن به انتهای فایل را بررسی کرد. همچنین تابع feof() که ورودی آن اشاره‌گر به فایل مورد نظر است، برای بررسی اینکه آیا نشانگر فایل به انتهای فایل اشاره می‌کند یا خیر استفاده می‌شود. در صورت رسیدن به پایان، مقدار غیر صفر برمی‌گرداند، در غیر این صورت، مقدار 0 را برمی‌گرداند.

از تابع getc() که ورودی آن نیز اشاره‌گر به فایل مورد نظر است، برای خواندن تک کاراکتر از ورودی استفاده می‌شود. همانند تابع fread ، پس از استفاده از این تابع نیز پیمایش کننده‌ی فایل در محل جدیدی قرار می‌گیرد. این محل اولین محل پس از محتوای خوانده یا نوشته شده است.

➤ [Useful link!](#)

3- انجام دهید!

در این قسمت قصد داریم برنامه‌ای را که در قسمت قبل نوشتیم، بهینه سازی کنیم! فرض کنید اطلاعاتی از طول فایل ورودی نداشته باشیم. در این صورت چگونه می‌توان از برنامه قسمت قبل استفاده کرد؟
✚ ابتدا می‌خواهیم برنامه‌ای بنویسیم که فایل Input.txt را (که همراه این فایل در اختیارتان گذاشته شده است) به عنوان ورودی دریافت کرده و طول آن را برمی‌گرداند. به برنامه زیر توجه کنید:

```
#include <stdio.h>
#define INPUT_FILE_ADDRESS "Input.txt"
int main() {
    FILE *input = fopen(INPUT_FILE_ADDRESS, "r");
    char temp;
    int length=0;
    while (1)
    {
        temp = getc(input);
        if (feof(input))
        {
            break;
        }
        length++;
    }
    fclose(input);
    printf("%d\n", length);
}
```

این برنامه را اجرا کرده و نحوه کار آن را به طور کامل برای دستیاران آموزشی شرح دهید. دلیل بکارگیری سطر هایلایت شده چیست؟ (قسمت 3)

✚ حال با استفاده از قطعه کد قسمت قبل، برنامه انجام دهید 2 را به گونه‌ای تغییر دهید که فایل Input.txt به عنوان ورودی دریافت کرده و متن داخل آن را به صورت معکوس در فایل دیگری به نام output.txt ذخیره کند.

ورودی:

```
Hello World!! We are learning to work with FILES in c!
```

خروجی:

```
!c ni SELIF htiw krow ot gninrael era eW !!dlroW olleH
```

نتیجه را به دستیاران آموزشی نشان دهید. (قسمت 4) ✓

✚ (امتیازی!) برنامه قسمت قبل را به گونه‌ای تغییر دهید که علاوه بر معکوس سازی، حروف کوچک به حروف بزرگ و حروف بزرگ به حروف کوچک تغییر داده شوند. فایل خروجی به صورت زیر خواهد بود.

```
!C NI selif HTIW KROW OT GNINRAEL ERA Ew !!DLROw OLLEh
```

🔙 4- انجام دهید! (دیباگ!)

حال می‌خواهیم یک عبارت دیگر را به انتهای فایل خود اضافه کنیم. کد زیر بدین منظور نوشته شده‌است. این کد را در یک پروژه‌ی جدید اجرا کنید. آیا به درستی عمل می‌کند؟

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#define INPUT_FILE_ADDRESS "Input.txt"
int main() {
    FILE *input = fopen(INPUT_FILE_ADDRESS, "w");
    char *text = " we love coding!! ";
    fwrite(text, 1, 1, input);
    return 0;
}
```

✚ ایرادات این برنامه رفع کنید.

✚ حال ورودی را به حالت اولیه برگردانده و به جای تابع fwrite از تابع fprintf استفاده کنید. ولی تلاش کنید تا خروجی به صورت زیر درآید:

```
Hello World!! We are learning to work with FILES in c!
we love coding!!
```

نتیجه را به دستیاران آموزشی نشان دهید. (قسمت 5) ✓

تابع fseek :

همانطور که ذکر شد برای کار با فایل‌ها یک اشاره‌گر از نوع FILE که به فایل مورد نظر اشاره می‌کند تعریف می‌کنیم. برای تغییر محل پیمایش‌کننده‌ی فایل، می‌توانیم از تابع fseek استفاده کنیم. این تابع به صورت زیر بکار می‌رود:

```
int fseek(FILE *pointer, long int offset, int position);
```

ورودی اول این تابع (pointer) اشاره‌گر به فایل مورد نظر، ورودی دوم (offset) مقدار تغییر مکان پیمایش‌کننده و ورودی سوم (position) مرجع تغییر است. Position سه مقدار زیر را می‌تواند اختیار کند:

- 1) SEEK_END : انتهای فایل را مشخص می‌کند.
- 2) SEEK_SET : شروع فایل را مشخص می‌کند.
- 3) SEEK_CUR : به مکان فعلی پیمایش‌کننده اشاره می‌کند.

▪ برای اطلاعات بیشتر می‌توانید به [این لینک](#) مراجعه کنید.

5- انجام دهید!

ابتدا فایل ورودی را به حالت اولیه برگردانید. سپس برنامه زیر را به نحوی تکمیل کنید که رشته text از خانه‌ی 14 ام فایل ورودی نوشته شود. نتیجه به صورت زیر خواهد بود:

```
Hello World!! we love coding!! to work with FILES in c!
```

```
#include <stdio.h>
#include <string.h>
#define INPUT_FILE_ADDRESS "Input.txt"
int main() {
    FILE *input = fopen(INPUT_FILE_ADDRESS, "r+");
    char *text = "we love coding!!";
    fseek(... , ... , ...);
    fwrite(... , ... , ... , ...);
    return 0;
}
```

نتیجه را به دستیاران آموزشی نشان دهید. (قسمت 6) ✓

توجه: همانطور که مشاهده می‌شود، در این برنامه از عملگر r+ در تابع fopen استفاده کرده‌ایم. اگر معنی این عملگر را فراموش کرده‌اید، به توضیحات ابتدای این آزمایش مراجعه کنید!

✚ حال به این برنامه قسمتی را اضافه کنید تا خروجی به شکل زیر باشد:

```
Hello World!! we love coding!! to work with FILES in c!  
Now we know how to use files in c :)
```

راهنمایی: برای رفتن به خط بعدی لازم است تا پیمایش کننده را به انتهای فایل برده و سپس با استفاده از تابع `fprintf ... !!!!!`

نتیجه را به دستیاران آموزشی نشان دهید.(قسمت 7) ✓

موفق باشید.

تهیه و تنظیم: امیرمرتضی رضائی