



به نام خدا

دانشکده مهندسی برق و کامپیوتر دانشکده فنی دانشگاه تهران

مبانی کامپیوتر و برنامه نویسی



استاد: دکتر مرادی و دکتر توسلی پور

آزمایشگاه ششم (کار با فایل در C)

نیمسال اول

۱۴۰۰

کار با فایل :

برای کار با فایل‌ها در زبان C باید ابتدا یک اشاره‌گر از نوع FILE بسازیم. پس از آن با استفاده از دستور fopen می‌توانیم یک فایل از حافظه‌ی کامپیوتر را باز کرده و به محتوای آن دسترسی پیدا کنیم. مقدار بازگشتی این تابع اشاره‌گر از نوع FILE است. به مثال زیر توجه کنید:

```
FILE *myfile = fopen("out.txt", "wb");
```

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:

"r"	خواندن از فایل متنی
"w"	نوشتن در فایل متنی
"a"	اضافه کردن به انتهای فایل متنی
"rb"	خواندن از فایل به صورت باینری
"wb"	نوشتن در فایل به صورت باینری
"ab"	اضافه کردن به انتهای فایل به صورت باینری

برای نوشتن در فایل توابعی مانند fprintf و fwrite و برای خواندن توابعی مانند fread و fscanf وجود دارند. توابع fprintf و fscanf همانند توابع printf و scanf عمل می‌کنند با این تفاوت که ورودی اول آن‌ها از نوع FILE* است. از این رو به بررسی توابع fread و fwrite می‌پردازیم. این توابع 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 * طول از آدرس اشاره‌گر به آرایه آغاز کرده و در فایل می‌نویسند (یا می‌خوانند). به قطعه کد زیر توجه کنید.

```
FILE *myfile = fopen("out.txt", "wb");  
char *str = "Hello!?";  
fwrite(str, sizeof(char), 5, myfile);  
fclose(myfile);
```

نکته: حتما باید در انتهای برنامه فایل‌های باز شده را با استفاده از دستور fclose ببندیم.

نکته: 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.

نکته : در صورتی که مانند کد بالا در قسمت آدرس تنها اسم فایل را ذکر کنیم، مرجع آدرس فایل آدرس زیر است :
Documents -> Visual Studio 201x -> Projects -> project name -> project name

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. ورودی اول این تابع اشاره‌گر به فایل مورد نظر، ورودی دوم مقدار تغییر مکان پیمایش‌کننده و ورودی سوم مرجع تغییر است؛ که با استفاده از `SEEK_SET` به ابتدای فایل و با استفاده از `SEEK_CUR` به مکان فعلی پیمایش‌کننده اشاره می‌کند.

← **۰. مشاهده کنید! (اختیاری)**

در این قسمت شما می‌بایست قطعه کدی که زیر آورده شده است را دنبال کنید کد تحلیل کنید و نتیجه را دریابید. برای امتحان درستی فرضیه خود فایل `Q0_code` که در کنار شرح آزمایش قرار داده شده را `Run` کنید، دقت داشته باشید که `input_Q0` را در کنار این فایل قرار داده باشید.

```
#include <stdio.h>
int main()
{
    FILE *inputFile;
    char charInPos;
    int i,numOfChars;
    inputFile=fopen("input_Q0.txt","r"); //File ready to be read from.
    if(inputFile==NULL) {
        printf("File does not exist..");
    }
    else{
        fseek(inputFile,0,SEEK_END);
        numOfChars =ftell(inputFile); //Num of File's characters
        while(i<numOfChars)
        {
            //Method goes in reverse
            i++;
            fseek(inputFile,-i,SEEK_END);
            charInPos=fgetc(inputFile);
            printf("%c",charInPos);
        }
        fclose(inputFile); //In the end of its use, we have to close it
        return 0;
    }
}
```

۱. انجام دهید!



هدف نوشتن برنامه‌ای است تا یک فایل را بخواند و متن داخل آن را به صورت معکوس در فایل دیگری بنویسید. به این منظور، قطعه کدهای زیر را کامل کنید:

```
#define ZERO 0
#define ONE 1
#define READ_CHAR_SIZE 78
#define WRITE_CHAR_SIZE 78
#define INPUT_TXT_ADDRESS "input.txt"
#define OUTPUT_FILE_ADDRESS "output.txt"

char* read_input_file() {
    char* in_order_array = (char*)malloc(READ_CHAR_SIZE *
sizeof(char));
    FILE* input = fopen(INPUT_TXT_ADDRESS, /* read from a .txt
file... */ );
    fread(/* Complete this part with cogent reasoning */);

    /*Fill if it's needed! */
    return in_order_array;
}

char* reverse_array(char* in_order_array) {
    char *reversed_array = (char*)malloc(READ_CHAR_SIZE *
sizeof(char))

    for (int i = ZERO; i < READ_CHAR_SIZE; i++){
        //Reverse it
    }
    return reversed_array;
}

void write_reversed_array_in_file(char* in_order_array) {
    char *reversed_array = reverse_array(in_order_array);
    FILE* output = fopen(OUTPUT_FILE_ADDRESS, "w");
    fwrite(/* Complete this part with cogent reasoning */);
    /*Fill if it's needed! */
}

int main() {
    char* in_order_array = read_input_file();
    write_reversed_array_in_file(in_order_array);
    return 0;
}
```

قسمت ۱: قطعه کدهای داده شده را کامل کرده و آن‌ها را به همراه نتایج به دست آمده تشریح کنید.

← ۲. انجام دهید!

- (۱) در قسمت قبل فایل `input.txt` را از پوشه‌ی محل پروژه حذف کرده و سپس دوباره برنامه را اجرا کنید. چه اتفاقی می‌افتد؟
- (۲) فایل `input.txt` را دوباره در محل پروژه قرار دهید.
- (۳) در مورد مشکلاتی که در صورت عدم استفاده از `fclose` ممکن است اتفاق بیفتد، در اینترنت تحقیق کنید.
- (۴) قبل از دستورات `fclose` در کد قسمت قبل `breakpoint` بگذارید و برنامه را در حالت `debug` اجرا کنید. پس از توقف برنامه در محل `breakpoint` سعی کنید فایل‌های `input.txt` یا `out.txt` را از محل پروژه حذف کنید. چه اتفاقی می‌افتد؟

قسمت ۲: موارد خواسته شده را انجام دهید. نتایج به دست آمده و همچنین پاسخ سوالات را تشریح کنید.

← ۳. انجام دهید! (EOF)

در کار با فایل‌ها، انتهای فایل با مقدار ثابتی به نام EOF معرفی می‌شود. همواره می‌توان با بررسی برابر بودن آخرین کاراکتر دریافت شده و ثابت EOF و یا با استفاده از تابع `feof`، که ورودی آن اشاره‌گر به فایل مورد نظر است، رسیدن به انتهای فایل را بررسی کرد.

- (۱) برنامه‌ی قسمت اول را به گونه‌ای تغییر دهید تا با متغیر بودن طول رشته‌ی درون فایل `input.txt` عملیات معکوس‌سازی را همانند قبل انجام دهد. برای سادگی، فرض کنید که طول رشته ورودی خوانده شده هیچ‌گاه از ۱۰۰ عبور نخواهد کرد.
- (۲) طول رشته‌ی درون فایل `input.txt` را تغییر دهید و برنامه را اجرا کنید.

قسمت ۳: موارد خواسته شده را انجام دهید. نتایج به دست آمده را تشریح کنید.

← ۴. انجام دهید!

هدف تغییر کد قسمت اول به طریقی است که علاوه بر معکوس سازی متن ورودی، حروف یکی در میان حذف شوند برای این کار :

- (۱) از کد قسمت سوم استفاده کنید تا کاراکتر های ورودی را تک تک دریافت کنید.
- (۲) پس از دریافت هر کاراکتر (با استفاده از دستور `fread` یا `fgetc`) با استفاده از دستور `fseek` مکان پیمایش کننده را به محل بعدی انتقال دهید.

قسمت ۴: روند ذکر شده را طی کرده و نتیجه به دست آمده را در کادر زیر بنویسید.

← ۵. انجام دهید!

همانطور که در قسمت اول نیز ذکر شد، حالت های مختلفی در خواندن و نوشتن فایل های ورودی و خروجی قرار دارد. یکی از این حالات `append` کردن است. در این حالت وقتی می خواهیم اطلاعاتی را در فایل بنویسیم این اطلاعات را به انتهای یک فایل که موجود است اضافه کنیم. برای این کار باید حالت باز کردن فایل را 'a' قرار دهیم.

- (۱) برنامه ای بنویسید که اطلاعات درون فایل `input.txt` را به انتهای فایل `out.txt` که از قسمت قبل به دست آمده است اضافه کند.

قسمت ۵: نتیجه به دست آمده را تشریح کنید.

← ۶. انجام دهید! (امتیازی)

یکی از حالات ذکر شده در نوشتن و خواندن فایل ها حالت دودویی است. از این حالت برای خواندن و نوشتن فایل های غیر متنی استفاده می کنیم. در این قسمت می خواهیم یک فایل تصویری را باز کرده، تغییراتی در آن داده و ذخیره کنیم. پیش از آن به این نکته توجه کنید که هر فایل در ابتدای خود دارای تعاریفی است که نوع و اطلاعات فایل را را تعیین می کند. همچنین در فایل از نوع `bmp` که در این قسمت با آن کار می کنیم هر پیکسل از تصویر توسط سه مقدار ۸ بیتی که نمایانگر مقادیر `RGB` هستند نشان داده می شوند.

- (۱) فایل `input2.bmp` را با استفاده از دستور `fopen` و در حالت "rb" باز کنید.
- (۲) آرایه ای از کاراکتر به طول ۱۵۴ بسازید و به همین اندازه از ابتدای فایل خوانده و در آرایه ذخیره کنید.
- (۳) آرایه ای سه بعدی به ابعاد [50][50][3] بسازید (طول*عرض*سه مقدار `RGB`) و با استفاده از دو حلقه بلوک های ۳ تایی از فایل خوانده و در این آرایه ذخیره کنید.
- (۴) تمامی مقادیر آرایه ای سه بعدی را ۱۰۰ عدد اضافه کنید.
- (۵) فایل جدیدی به نام `out.bmp` ایجاد کرده و نوع باز کردن آن را "wb" انتخاب کنید.
- (۶) ابتدا مقادیر آرایه ای به طول ۱۵۴ را در فایل ذخیره کنید.
- (۷) مقادیر آرایه ای سه بعدی را به همان روش خواندن در فایل جدید ذخیره کنید.

۸) فایل جدید ایجاد شده را مشاهده کنید.

قسمت ۶: موارد خواسته شده را انجام دهید و نتایج به دست آمده را دریابید.

موفق باشید