

به نام خدا

دستور کار آزمایشگاه شماره 9

کار با فایل در C

در این جلسه شما ابتدا به نحوه‌ی تبادل اطلاعات با فایل‌ها در زبان C آشنا می‌شوید.

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 رو 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 می‌کند 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 fscanf و fread 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

۱. انجام دهید!



- هدف نوشتن برنامه‌ای است تا یک فایل را بخواند و متن داخل آن را به صورت معکوس در فایل دیگری بنویسد. به این منظور:
- (۱) فایل قرار داده شده همراه دستور کار (`input.txt`) را در پوشه‌ی پروژه قرار دهید.
 - (۲) آرایه‌ای از نوع `char` ساخته و با استفاده از دستور `fread` اطلاعات داخل فایل `input.txt` را در آن قرار دهید. (راهنمایی: تعداد کاراکترهای درون فایل 51 است.)
 - (۳) آرایه‌ی دیگری ساخته و اطلاعات آرایه‌ی قبلی را به صورت معکوس درون آن بنویسد.
 - (۴) فایل جدیدی به نام `out.txt` بسازید و اطلاعات آرایه‌ی جدید را در آن بنویسد.
- نتیجه را به دستیاران آموزشی نشان دهید.

۲. انجام دهید!



- (۱) در قسمت قبل فایل `input.txt` را از پوشه‌ی محل پروژه حذف کنید 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؟
- نتایج را با دستیاران آموزشی در میان بگذارید.

← 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.

(۲) طول رشته‌ی درون فایل input.txt را تغییر دهید و برنامه را اجرا کنید.

نتیجه را به دستیاران آموزشی نشان دهید.

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.

← ۴. انجام دهید!

هدف تغییر کد قسمت اول به طریقی است که علاوه بر معکوس‌سازی متن ورودی حروف یکی در میان حذف شوند. (در این مثال خطوط تیره باید حذف شوند). برای این کار :

(۱) از کد قسمت سوم استفاده کنید تا کاراکترهای 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 پیمایش‌کننده را به محل بعد از خط تیره انتقال دهید.

نتیجه را به دستیاران آموزشی نشان دهید.

← ۵. انجام دهید!

همانطور که در قسمت اول نیز ذکر شد حالت‌های مختلفی در خواندن و نوشتن فایل‌های ورودی و خروجی قرار دارد. یکی از این حالات append کردن است. در این حالت وقتی می‌خواهیم 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) برنامه‌ای بنویسید که اطلاعات درون فایل `input.txt` را به انتهای فایل `out.txt` که از قسمت قبل به دست آمده است اضافه کند.

نتیجه را به دستیاران آموزشی نشان دهید.

← ۶. انجام دهید! (امتیازی)

یکی از حالات ذکر شده در نوشتن و خواندن فایل‌ها حالت باینری است. از این حالت برای خواندن و نوشتن فایل‌های غیر متنی استفاده می‌کنیم. در این قسمت می‌خواهیم یک فایل تصویری را باز کرده، تغییراتی در آن بدهیم 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 می‌کند. همچنین در فایل از نوع `bmp` که در این قسمت با آن کار می‌کنیم هر پیکسل از تصویر توسط یک مقدار ۸ بیتی که نمایانگر مقادیر `RGB` هستند نشان داده می‌شوند.

- ۱) فایل `input2.bmp` را با استفاده از دستور `fopen` و در حالت `"rb"` باز کنید.
- ۲) آرایه‌ای از کاراکتر به طول ۱۵۴ بسازید و به همین اندازه از ابتدای فایل خوانده و در آرایه ذخیره کنید.
- ۳) آرایه‌ای سه بعدی به ابعاد `[50][50][3]` بسازید (طول*عرض*سه مقدار `RGB`) و با استفاده از دو حلقه بلوک‌های ۳ تایی از فایل خوانده و در این آرایه ذخیره کنید.
- ۴) تمامی مقادیر آرایه‌ی 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 آرایه‌ی سه بعدی به طول ۱۵۴ را در فایل ذخیره کنید.
- ۷) مقادیر آرایه‌ی سه بعدی را به همان روش خواندن در فایل جدید ذخیره کنید.
- ۸) فایل جدید ایجاد شده را مشاهده کنید.

نتیجه را به دستیاران آموزشی نشان دهید.

موفق باشید