بسم الله الرحمن الرحیم

دستورات کار با فایل های متنی

در زبان C

استاد دلاکی

از توضیحات ، تعاریف و مباحث نظری صرف نظر شده است

- **ایجاد فایل متنی در زبان C**

برای ایجاد فایل یا باز کردن فایل موجود، ابتدا باید اشاره گری از نوع فایل به شکل زیر تعریف شود:

```
FILE  *f;
```

برای باز کردن یا ایجاد یک فایل جدید، از فرمت زیر استفاده می شود:

```
f=fopen("d:\\book.txt","r+t");
```

توضیح:

- f : همان اشاره گری است که در ابتدا تعریف شده است و قرار است در اینج برای دستیابی به فایل مورد نظرمان از این اشاره گر استفاده کنیم.
- fopen() : دستوری است برای باز کردن فایل مورد نظر ما .
- "d:\\book.txt" : آدرس، نام و پسوند فایل فایلی است که قرار است باز یا ایجاد کنیم. حتما از دوتا \\ برای آدرس استفاده شود
- "r+t" نوع باز کردن فایل را مشخص می سازد که در ادامه توضیح داده می شود.

- **اطمینان از باز شدن یا ایجاد فایل**

برای اطمینان از اینکه فایل مورد نظر با موفقیت باز شده است، می توان از کدی مانند کد زیر استفاده کرد. این کد باعث می شود تا در صورت عدم باز شدن فایل، پیغام خطا نمایش داده شود:

```
if ((fp=fopen("A:\\data.txt","w"))==NULL)
{ printf("can not open file “);
}
```

از آنجایی که اگر نتوان فایل موردنظر را باز کرد، NULL برگشت داده می شود، می توان با مقایسه دستور fopen با NULL مطمئن شد که فایل باز شده است و در غیر اینصورت پیغام خطا نمایش داده شود.

➢ انواع مختلف باز کردن فایل

فایل را می توان به نوعی باز کرد که فقط بتوان اطلاعات آن را مشاهده کرد یا فقط بتوانیم اطلاعاتی را به آن اضافه کرد. یا فایل را در صورت وجود باز و در غیر اینصورت فایل جدید ایجاد کند.حالات مختلف در زیر آمده است.هر یک از این حالات درون پرانتز fopen در قسمت آخر قرار می گیرد:

MODE	مفهوم
r(rt)	فایلی از نوع text را به عنوان ورودی باز می کند
w(wt)	فایلی از نوع text را به عنوان خروجی باز می کنند
a(at)	فایل را طوری باز می کند که بتوان اطلاعاتی را به آن اضافه کرد.
rb	فایلی از نوع باینری را به عنوان ورودی باز می کند.
wb	فایلی از نوع باینری را به عنوان خروجی باز می کند.
ab	فایل موجود از نوع باینری را طوری باز می کند که بتوان اطلاعات را به انتهای آن اضافه نمود.
r+(r+t)	فایل موجود از نوع text را به عنوان ورودی و خروجی باز می کند
w+(w+t)	فایلی از نوع text را به عنوان ورودی و خروجی باز می کند
a+(a+t)	فایل موجود از نوع text را به عنوان ورودی و خروجی باز می کند
r+b	فایل موجود از نوع باینری را به عنوان ورودی و خروجی باز می کند
w+b	فایل موجود از نوع باینری را به عنوان ورودی و خروجی باز می کند
a+b	فایل موجود از نوع باینری را به عنوان ورودی و خروجی باز می کند (این فایل از قبل می تواند وجود داشته باشد).

- **بستن فایل**

باید توجه کرد تا زمانیکه فایل بسته نشود ، تغییرات اعمال شده ذخیره نمی شوند.برای بستن فایل از دستورات زیر استفاده می شود:

- برای بستن یک فایل. در اینجا فایل **f** بسته می شود fclose(**f**);
- برای بستن تمامی فایل هایی که در برنامه باز کرده ایم fcloseall();

- **نوشتن در فایل**

- **Putc یا fputc :** از این دو تابع برای نوشتن یک حرف درون فایل استفاده می شود:

در هر دودستور putc (فرمت fputc نیز مشابه همین است) ، حرف a درون **Char ch;**

فایل **f** نوشته می شود**Ch='a';**

- ✓ **putc(ch, f);**

یا

- ✓ **putc('a',f);**

- **fputs() :** برای نوشتن یک رشته درون فایل کاربرد دارد**:**

```
Char  str[]="Ali";
```

- ✓ **fputc(str, f);**

یا

- ✓ **futs("Ali",f);**

در هر دو فرمت بالا، کلمه **Ali** درون فایلی که f به آن اشاره می کند، نوشته می شود.

- **fprintf()**: برای نوشتن داده با فرمت خاص درون یک فایل استفاده می شود:

```
int  a=12;
```

- ✓ **fprintf(f,"Num = %d",a);**

دستور فوق متن زیر را درون فایل**f**می نویسد:

Num = 12

: fwrite

مانند fprintf میباشدامابااینتفاوتکهدارایسرعتبیشترینسبتبه ()fprintfمیباشد .

fwrite(void *buffer, intnum_byte,int count,FILE *fp)

مثال:

```
Char table[20];
fwrite(table,sizeof(char),200,fp);
```

تا اینجا دستورات مختلفی را درباره بازکردن فایل،بستن فایل و همچنین نوشتن در فایل بیان کردیم.

کد زیر عبارت"IN THE NAME OF GOD"وسپس حاصل جمع دو عدد را درون فایلی با نام test در درایو C می نویسد:

```
int main(){
FILE  *f;
inta,b;
a=3; b=4;
char str[]="IN THE NAME OF GOD";
f=fopen("c:\\test.txt","at");
fputs(str,f);
a=a*b;
```

```
fprintf(f,"%d",a);
fclose(f);
return 0;
}
```

توجه داشته باشید که نوع باز شدن فایل در خط ششم کد، "at" انتخاب شده است. زیرا این نوع، اگر فایلی در مسیر مورد نظر وجود نداشته باشد، یک فایل جدید ایجاد می کند و مطالب را درون آن می نویسد. اگر هم فایل از قبل وجود داشت، آن را باز و سپس مطالب جدید را در انتهای آن فایل اضافه می کند. در حالی که مثلا نوع "w" فایل جدید را ایجاد می کند و مطالب را به ابتدای آن اضافه می کند. یا نوع "r+t" فایلی را که موجود باشد به عنوان ورودی و خروجی باز و مطالب را به ابتدای آن اضافه می کند. حالت های مختلف را می توان با امتحان کردن به راحتی آزمایش کرد.

دستور بستن فایل حتما باید در پایان کارمان با فایل بنویسیم.چون در غیر اینصورت تغییرات ما در فایل ثبت نمی شود.

پس از بستن فایل، حتی اگر از برنامه هم خارج نشده باشیم، تغییرات در فایل ثبت شده است و می توان با باز کردن فایل متنی، تغییرات آن را مشاهده کرد.

- **خواندن از فایل**

دستورات خواندن از فایل نیز مانند دستورات نوشتن است و در واقع قرینه هم هستند

- getc() یا fputc() :

از این دو دستور برای خواندن تنها یک حرف از درون فایل،(با توجه به مکان موقعیت سنج) به کار برده می شود.موقعیت سنج در ادامه توضیح داده خواهد شد.شکل کلی استفاده بدین صورت است:

```
char  ch;
ch=fgetc(f);
```

با توجه به دستور بالا، حرفی از فایل f که قبلا باز شده و موقعیت سنج روی آن قرار دارد، درون متغیر ch **کپی** می شود. و سپس موقعیت سنج یک حرف جلو می شود.

- fgets() : این دستور یک رشته را از فایل می خواند:

fget(نام رشته ،طول رشته ،اشاره گر فایل);

مثال : fgets(str,5,f);

در مثال فوق، پنج حرف ازفایلی که f به آن اشاره می کند و محلی که موقعیت سنج در آنجاست، خوانده می شود و درون رشته str قرار می گیرد

- fscanf() : خواندن داده با فرمت خاص از درون فایل:

fscanf(اشاره گر فایل,"%c یا %d یا...",&(ها)متغیر)

مثال : fscanf(f,"%d",&num);

- fread() : مانند scanf است اما fread دارای سرعت بیشتری است:

```
Char arr[10];
fread(arr,sizeof(char),10,fp);
```

- **موقعیت سنج**

منظور از موقعیت سنج مکانی است که اگر ما حرفی را در فایل درج کنیم ،آن حرف در آنجا جا درج می شود و یا اگر حرفی را از فایل بخوانیم، آن حرف به خوانده می شود که البته با درج و یا خواندن هر حرف یا رشته، به تعداد حروف وارد شدهیا خوانده شده، موقعیت سنج جلو می رود. مثلا هنگامی که فایلی به صورت "at" باز شود، موقعیت سنج در انتهای فایل قرار می گیرد و اگر حرفی در این لحظه درون فایل درج شود، چون موقعیت سنج انتهای فایل قرار دارد، حرف ورودی نیز در انتهای فایل درج می شود. توسط دستورات زیر می توان موقعیت سنج را جابجا کرد:

- rewind() : توسط این دستور موقعیت سنج از هر جایی از فایل که باشد به ابتدای فایل منتقل می شود:

انتقال موقعیت سنج به ابتدای فایل ;(f rewind(f

- fseek : توسط این دستور، می توانیم موقعیت سنج را به هر جایی از فایل که بخواهیم جابجا نماییم:

fseek (f, num_byte, origin)

f : اشاره گر به فایل ماست

num_byte : تعداد حروفی که می خواهیم موقعیت سنج جلو و یا عقب برده شود

origin : محلی از فایل است که می خواهیم موعیت سنج را نسبت به آن محل جابجا کنیم.

origin تنها می تواند یکی از این سه موقعیت باشد:

۱. SEKK_SET : نسبت به انتهای فایل
۲. SEEK_CUR : نسبت به محل کنونی موقعیت سنج
۳. SEEK_END: نسبت به انتهای فایل

✓ به عنوان مثال می خواهیم، کاراکتر بیستم از فایلی که f به آن اشاره می کند را بخوانیم:

- بدون توجه به اینکه موقعیت سنج در چه مکانی قرار دارد، از دستور زیر استفاده می کنیم:

```
char  ch;
fseek( f , 20, SEEK_SET);
 ch=fgetc(f);
```

✓ خواندن کاراکتری که دو کاراکتر قبل از پایان فایل قرار دارد:

```
fseek( f , -2 , SEEK_END);
ch=fgetc(f);
```

- SEEK_END نشان دهنده اینست که ما می خواهیم موقعیت سنج را نسبت به آخر فایل جابجا کنیم
- عدد به صورت منفی وارد شده، چون می خواهیم موقعیت سنج را به عقب منتقل کنیم

- ✓ نوشتن کاراکتری، سه کاراکتر قبل از محلی که اکنون موقعیت سنج در آن قرار دارد:

```
char ch;
ch='a';
fseek( f , -3 , SEEK_CUR);
fputc(ch,f);
```

- ❖ توجه داشته باشید هرگونه درج و نوشتن در فایل، به غیر از انتهای فایل ، روی کاراکتر قبلی خواهد بود. به عنوان مثال اگر فرض شود،حرفی که موقعیت سنج روی آن قرار دارد،حرف 'a' باشد و ما بخواهیم حرف 'b' را نیز درج کنیم، حرف 'b' جایگزین حرف 'a' می شود و به شکل 'ab' نخواهد بود.

- ❖ پس از رسیدن به انتهای فایل ، در صورتی که کاراکتری خوانده شود، توابع fgetc و getc علامت |EOF| را برمی گردانند. یعنی اگر حرفی از فایل خوانده شود و آن حرف برابر با EOF باشد، می توان نتیجه گرفت که به انتهای فایل رسیده ایم:

```
charch;
ch=fgetc(f);
if(ch==EOF)
{ printf("payane file.");
  }
```

روش دیگر برای تشخیص رسیدن به انتهای فایل، استفاده از تابع feof() می باشد. این تابع اگر به انتهای فایل مورد نظر رسیده باشیم ارزش درستی و در غیر اینصورت ارزش نادرستی را باز می گرداند:

```
If(  feof( f ) )
printf(“payane file.”);
```

یا

```
If(  feof( f ) ==1)
    printf(“payane file.”);
```

دستورات بیان شده در فوق، دستورات اصلی کار با فایل ها در زبان C می باشد که در ترکیب با دستورات آموخته شده در دروس برنامه نویسی مانند حلقه های for و while و دستورات شرطی if و ... ، می توانند تا بالای ۹۵ درصد از نیاز یک دانشجو برای کار با فایل را فراهم آورند.

یادگیری سایر مطالب بیشتر به عهده خود دانشجو گذاشته می شود.

پایان