

Lecture14

Opening and Closing Files. Formatted Input and Output in C Language

Opening a file.

If we want to store data in a file in the external memory, we must use the following general format for opening a file:

```
FILE *fp; fp=fopen("filename","mode"); if(fp==NULL){puts("file was not oppend"); return;}
```

The first statement declares the variable **fp** called file pointer to the structure data type **FILE** that is defined in the **stdio.h**. The second statement opens the file named **filename** of corresponding **mode** by using standard function **fopen()** and determines the value of file pointer **fp** for the given file. This pointer, which points to the data structure **FILE** that contains all the information about the file, is used as a communication link between the operating system and the program. The third statement verifies if function **fopen()** returned **NULL** pointer and if so it means that file was not opened. The **mode** of file can be of three main kinds:

r- the file for reading data ; **w**- the file for writing data; **a**- the file for appending data.

Consider the following statements:

```
FILE *fp1, *fp2; fp1=fopen("data.txt","r"); fp2=fopen("results.txt","w");
```

In these statements the **fp1** and **fp2** are created and assigned to open the files **data** and **results** respectively. The file **data** is opened for reading and **results** is opened for writing. In case the file **results** already exists, its contents are deleted and the file is opened as a new file. If file **data** does not exist **fp1** receives **NULL** pointer value and file is not opened.

Closing a file.

The input output library **stdio.h** supports the function to close a file of the following format: **fclose(fp);** A file must be closed as soon as all operations on it have been completed. The function **fclose()** closes the file associated with the file pointer **fp**.

Observe the following part of program:

```
FILE *fp1 *fp2; fp1=fopen ("data.txt","r"); fp2=fopen ("results.txt","w");
```

...

```
fclose(fp1); fclose(fp2);
```

The above program opens two files and closes them after all operations on them are completed. Once a file is closed its file pointer can be reversed on other file.

The **getc()** and **putc()** functions are analogous to **getchar()** and **putchar()** functions and handle one character at a time. The function call **putc(ch, fp1);** writes the character contained in character variable **ch** to the file associated with the pointer **fp1**; similarly the function **getc()** is used to read a character from a file that has been opened in read mode and then to assign it to character variable **ch = getc(fp2);**

The **fprintf()** and **fscanf()** formatted functions for writing in and reading from file.

The **fprintf()** and **fscanf()** formatted functions are identical to **printf()** and **scanf()** formatted functions except that they work on files. The first argument of these functions is a file pointer **fp** which specifies the file to be used. The general form of **fprintf()** function call is:

```
fprintf(fp, "control string", list);
```

Where **fp** is a file pointer associated with a file that has been opened for writing. The control string is file output specifications, list may include variable, constant and string. For example:

```
fprintf(fp, "%s %d %.2f\n", name, age, 7.5);
```

Here **name** is a character array, **age** is an integer variable and **7.5** is a float constant.

The general format of **fscanf()** function call is:

fscanf(fp, "control string", list);

This statement would cause the reading of items of list conform the control string. For example:

fscanf(fp, "%s%d", item, &quantity);

where **item** is a character array and **quantity** is an integer variable.

Practical exercise: Example of program for input and output information about students using files

```
#include<stdio.h>
#include<conio.h>
int main( )
{
char name[30], fname[20] ; int n, i, clas , year; float average;
FILE *fp;
clrscr( );
printf ("enter name of file: "); fflush(stdin); gets(fname);
fp = fopen(fname, "w"); if( fp==NULL) {puts("file was not open."); return 1;}
printf("enter number of students: "); scanf("%d", &n);
puts("\n\t enter info about students from keyboard and write info in file:\n");
for( i=0; i<n; i++)
{
printf("\n student %d :\n", i+1);
printf("name:   "); fflush(stdin); gets(name);
printf("year:    "); scanf("%d", &year);
printf("class:   "); scanf("%d", &clas);
printf("average: "); scanf("%f", &average);
fprintf(fp, "%s %d %d %.2f\n", name, year, clas, average);
}
fclose(fp);
fp = fopen(fname, "r"); if( fp== NULL) {puts("file was not open"); return 1;}
puts( "\n\t info about students reading from file:\n");
for(i=0; i<n; i++)
{
fscanf(fp, "%s%d%d%f", name, &year, &clas, &average);
printf("%d. name: %s year: %d class: %d average: %.2f\n ", i+1, name, year, clas, average);
}
fclose(fp);
getch( );
return 0;
}
```

Lecturer: Mihail Kulev, associate professor