MANAV
RACHNA
||vidyanatariksha||

Programming for Problem Solving

(BCS-101A)

UNIT 6

BCS-101.6: File Handling

Sameeksha Khare
Assistant Professor
Deptt. of CSE

Console oriented Input/Output

- Console oriented – use terminal (keyboard/screen)
- scanf(“%d”,&i) – read data from keyboard
- printf(“%d”,i) – print data to monitor
- Suitable for small volumes of data
- Data lost when program terminated

File Handling

- A file is a collection of data stored on a secondary storage device like hard disk.
- It represents a sequence of bytes on the disk where a group of related data is stored.
- It is created for permanent storage of data.
- It is a readymade structure.
- Declaration

```
FILE *fp;
```

File Operations

- Opening a file.
- Naming a file.
- Reading / writing a file.
- Closing a file.

Opening a File

- A file must be “opened” before it can be used.

```
        FILE   *fp;
    :
    fp = fopen (filename, mode);
```

  - fp is declared as a pointer to the data type FILE.
  - filename is a string - specifies the name of the file.
  - fopen returns a pointer to the file which is used in all subsequent file operations.
  - mode is a string which specifies the purpose of opening the file:

“r” :: open the file for reading only

“w” :: open the file for writing only

“a” :: open the file for appending data to it

File Opening modes

- **“r”** **-** open file for reading only
- **“w”** **-** open file for writing only
- **“a”** **-** open file for appending (adding) data
- **“r+”** - same as r except both for reading/writing
- **“w+”** - same as w except both for reading and writing
- **“a+”** - same as a except both for reading and writing

MODES

- ***r*** **- open a file in read-mode, set the pointer to the beginning of the file.**
- ***w*** **- open a file in write-mode, set the pointer to the beginning of the file.**
- ***a*** **- open a file in write-mode, set the pointer to the end of the file.**
- ***rb*** **- open a binary-file in read-mode, set the pointer to the beginning of the file.**
- ***wb*** **- open a binary-file in write-mode, set the pointer to the beginning of the file.**
- ***ab*** **- open a binary-file in write-mode, set the pointer to the end of the file.**
- ***r+*** **- open a file in read/write-mode, if the file does not exist, it will not be created.**
- ***w+*** **- open a file in read/write-mode, set the pointer to the beginning of the file.**
- ***a+*** **- open a file in read/append mode.**
- ***r+b*** **- open a binary-file in read/write-mode, if the file does not exist, it will not be created.**
- ***w+b*** **- open a binary-file in read/write-mode, set the pointer to the beginning of the file.**
- ***a+b*** **- open a binary-file in read/append mode.**

Contd.

- Points to note:
  - Several files may be opened at the same time.
  - For the “w” and “a” modes, if the named file does not exist, it is automatically created.
  - For the “w” mode, if the named file exists, its contents will be overwritten.

File Handling Functions

Function	description
fopen()	create a new file or open a existing file
fclose()	closes a file
getc()	reads a character from a file
putc()	writes a character to a file
fscanf()	reads a set of data from a file
fprintf()	writes a set of data to a file
getw()	reads an integer from a file
putw()	writes an integer to a file
fseek()	set the file pointer to desired location
ftell()	gives current position in the file
rewind()	set the file pointer to the beginning of the file

Opening a file

```
FILE *fp;
fp = fopen(“file-name”, “mode”);
```

Closing a file

```
fclose(fp);
```

getc() and putc()

- ch = getc(fp);
- putc(ch,fp);

getw() and putw()

- num = getw(fp);
- putw(num,fp);

fscanf() and fprintf()

```
fprintf(f1, “%d %f\n”, i, f);
fprintf(stdout, “%f \n”, f);  /*stdout refers to screen */
fscanf(f2, “%d %f”, &i, &f);
```

- fscanf() returns EOF when end-of-file reached.

1. Program to print a file on screen

```
#include <stdio.h>
void main()
{          FILE *f1;
           char ch;
           f1= fopen(“INPUT”, “w”);
           while((ch=getchar()) != EOF)
                      putc(ch,f1);
           fclose(f1);
           f1=fopen(“INPUT”, “r”);

           while((ch=getc(f1))!=EOF)
                      printf(“%c”, ch);
           fclose(f1);
}
```

EOF (end of file)

- EOF is a macro defined in stdio.h as an int with a negative value.
- It is normally returned by functions that perform read operations to denote either an error or end of input.

2. Program to copy the contents of one file to another file.

```
#include<stdio.h>
#include<conio.h>
void main( )
{
FILE *f1, *f2;
char ch;
printf(“Enter the Data”);
f1=fopen(“DATA”, “w”);
while((ch=getchar())!=EOF)
{
        putc(ch,f1);
}
fclose(f1);
f1=fopen(“DATA”,”r”);
f2=fopen(“COPY”,”w”);
while((ch=getc(f1))!=EOF)
{
        putc(ch,f2);
}
fclose(f1);
fclose(f2);
getch();
}
```

3. Write a program to create a file named INPUT which contain a series of integers. Now read these integers from INPUT file and write all even numbers into a file to be called EVEN and all odd numbers into a file to be called ODD.

```
#include<stdio.h>
#include<conio.h>
void main( )
{
FILE *f1, *f2, *f3;
int n,i;
printf(“Enter the Data”);
f1=fopen(“INPUT” , “w”);
for(i=1;i<=30;i++)
{
    scanf(“%d”,&n);
    putw(n,f1);
}
fclose(f1);
```

```
f1=fopen(“INPUT” , ”r”);
f2=fopen(“EVEN” , ”w”);
f3=fopen(“ODD” , ”w”);
while((n=getw(f1))!=EOF)
{
    if(n%2==0)
            putw(n,f2);
    else
            putw(n,f3);
}
fclose(f1);
fclose(f2);
fclose(f3);
getch();
}
```

4. Program to check whether the contents of two files are same.

```
#include<stdio.h>
#include<conio.h>
void main( )
{
FILE *f1, *f2;
char ch1,ch2;
f1=fopen(“DATA1”, “r”);
f2=fopen(“DATA2”, “r”);
do
{
		ch1=getc(f1);
		ch2=getc(f2);
}while((ch1==ch2)&&(ch1!=EOF)&&(ch2!=EOF)
);
```

```
if((ch1==EOF)&&(ch2==EOF))
   printf(“Contents are equal”);
else
 printf(“Contents are not equal.”);
fclose(f1);
fclose(f2);
getch();
}
```

5. Program to count the number of occurrence of a word in the given file.

```
#include<stdio.h>
#include<conio.h>
void main( )
{
FILE *f1;
char ch1,str1[10],str2[10];
int  count =0;
printf(“Enter the word \n”);
gets(str1);
f1=fopen(“INPUT”, “r”);
while(!feof(f1))
{
            fscanf(f1, “%s”,str2);
            if(!strcmp(str1,str2))
                        count++;
}
if(count==0)
 printf(“%s does not exist”,str1);
else
printf(“%s exists with %d occurence.”,str1,count);
fclose(f1);
getch();
}
```

feof()

- It is the function that returns zero when file pointer is not at the end of file. It returns nonzero value when file pointer is at the end of file.
- Declaration:

```
int feof(FILE *stream);
```

QUIZ

1. ___fopen()__________ is used to open a file.

[UPTU 2008-09 (Odd) Marks-1]

2. ___fclose()_________ is used to close a file.

[UPTU 2008-09 (Odd) Marks-1]

3. Mode ________“r”__________ opens existing file for read operations only. [UPTU 2010-11 (Odd) Marks-1]

FAQ’s

1. What is the full form and significance of EOF ? [MTU 2012-13 (Odd) Marks-2]
2. Discuss basic file handling functions. [GBTU 2011-12 (Even) Marks-2]
3. Write a program in C to copy the content of a given file say "a.txt" to another file say "b.txt“. [UPTU 2003-10 (Even) Marks-5]
4. What is the use of header files? [UPTU 2009-10 (Even) Marks-5]
5. A file named DATA contains a series of integer numbers. Write a program to read these numbers and then write all "odd" numbers to a file to be called ODD and all "even" numbers to a file to be called EVEN. [MTU 2011-12 (Odd) Marks-10]