**CHAPTER 8 :**

**FILE HANDLING**

**Old Questions Solutions**

### What are the file opening modes? Write a program to open a new file, read name, address and telephone number of 10 employees from a user and write to a file.[2013Fall]

Ans: The file opening modes specifies the way in which a file should be opened (i.e. for reading, writing or both, appending at the end of the file, overwriting the file, etc. I other words, it specifies the purpose of opening of file. There are mainly six modes.

|  |  |  |
| --- | --- | --- |
| **Mode** | **Meaning of mode** |  |
| **During Inexistence of file** |
| r | Open for reading. | If the file does not exist, fopen() returns NULL. |
| w | Open for writing | If the file exists, its contents are overwritten. If the file does not exist, it will be created. |
| a | Open for append.  Data is added to the end of the file. | If the file does not exist, it will be created. |
| r+ | Open for both reading and writing. | New data is written at the beginning overwriting existing data. If the file does not  exist, fopen() returns NULL. |
| w+ | Open for both reading and writing. | If the file exists, its contents are overwritten. If the file does not exist, it will be created. |
| a+ | Open for both reading and appending. | New data is appended at the end of file. If the file does not exist, it will be created. |

#include <stdio.h> int main()

{

char name[50],address[20]; int num;

long tel;

printf("Enter number of students: "); scanf("%d", &num);

FILE \*fptr;

fptr = (fopen("D:\\student.txt", "w")); if(fptr == NULL)

{

printf("Error!"); exit(1);

}

for(int i=0; i <num;i++ )

{

printf("For Employee%d \n",i+1); printf("Enter name:"); scanf("%s",&name);

printf("Enter address:"); scanf("%s",&address); printf("Enter telephone:"); scanf("%l",&tel);

fprintf(fptr,"\n Name: %s \nAddress:%s\n Telephone:

%l",name,address,tel);

}

fclose(fptr); return 0;

}

### Why do we need file handling? Describe the different handling modes.[2013Spring]

Ans: There is a time when the output generated by compiling and running the program does not serve the purpose. If we want to check the output of the program several times, it becomes a tedious task to compile and run the same program multiple times. This is where file handling comes into play. Here are some of the following reasons behind the popularity of file handling:

**Reusability**: It helps in preserving the data or information generated after running the program.

**Large storage capacity:** Using files, you need not worry about the problem of storing data in bulk.

Saves time: There are certain programs that require a lot of input from the user. You can easily access any part of the code with the help of certain commands.

**Portability:** You can easily transfer the contents of a file from one computer system to another without having to worry about the loss of data.

Second part[2013Fall] Qno.1

### What is the significance of file pointer in file handling? Consider the following structure.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Roll.no. | Name | Address | faculty | Date of  birth |  |  |
|  |  |  |  | Mm | dd | yy |

Write a program to create “student.txt” file to store the above records for 100 students. Also display these records of students who are not from Pokhara.[2014Fall]



File Pointer:

A file pointer is a pointer to a structure, which contains information about the file, including its name, current position of the file, whether the file is being read or written, and whether errors or end of the file have occurred. The user does not need to know the details, because the definitions obtained from stdio.h include a structure declaration called FILE. The only declaration needed for a file pointer is symbolized by

*FILE \*fp;*

This says that fp is the file pointer that points to a FILE structure. Opening a file:

The fopen() function opens a stream for use and links a file with that stream. A file pointer associated with that file is then returned by the fopen() function. Most often the file is a disk file.

*FILE \*fp; fp=fopen("try.c","r");*

fopen() opens the file “try.c” in read mode. When a file is opened in read mode, three important tasks are performed by fopen().

1. A search is carried out on the disk for the file to be opened.
2. If the file is found, it is loaded into memory from the disk. In case the file is too large, then the file is loaded part wise. In case the file is not found, a NULL() is returned by fopen(). “stdio.h” contains a macro defined as NULL, which indicates that the attempt to open the file failed.
3. fopen() then open sets up a character pointer. The character pointer is a part of FILE structure and points to the first character in memory where the file is loaded.

Closing a File: As said earlier, there is usually a limit on the number of files that can be opened at one time, and so it is important to close the file once it has been used. This ensures that various system resources will be free and reduces the risk of overshooting the set limit. The fclose() function closes a stream that was opened by a call to fopen(). It writes any data still remaining in the disk buffer to the file. The prototype for fclose() is

*Fclose (FILE \*fp);*

Where fp is the file pointer. The function fclose() returns an integer value 0 for successful closure, any other value indicates an error. The fclose() generally fails when a disk has been prematurely removed from the drive or there is no more space on the disk.

#include<stdio.h> #include<stdlib.h> int main()

{

FILE \*fptr;

char Name[20], Address[20],Faculty[20]; int Month, Day,Year,Roll; fptr=fopen("D:\\student.txt","w"); if(fptr==NULL)

{

printf("\n File cannot be created"); exit(1);

}

fprintf(fptr, " Roll number\tName\tAddress\tFaculty\tDate of birth- Month Day Year");

for(int i=0;i<100;i++)

{

printf("For Student%d",i); printf("\nRoll\n"); scanf("%d",&Roll); printf("Name\n"); scanf("%s",&Name); printf("Address\n"); scanf("%s",&Address); printf("Faculty\n"); scanf("%s",&Faculty);

printf("Date of Birth in Month-Day-Year\n"); scanf("%d%d%d",&Month,&Day,&Year);

fprintf(fptr,"\n %d\t%s\t%s\t%s\t%d-%d-%d" , Roll,Name,Address,Faculty,Month,Day,Year);

}

fclose(fptr); return(0);

}

### What is the significance of file pointer? Describe the different file opening modes in C.[2014Spring]

Ans: page [2014Fall] Qno.3 Second part[2013Fall] Qno.1

### Why header files in C is included in program? Give reasons. Also list-out different header files that you know. Illustrate the program showing the use of header file.[2015Fall]

Ans: A header file is a file containing C declarations and macro definitions to be shared between several source files. We request the use of a header file in the program by *i*ncluding it, with the C preprocessing directive ‘#include’.

Header files serve two purposes.

System header files declare the interfaces to parts of the operating system. We include them in your program to supply the definitions and declarations we need to invoke system calls and libraries.

We own header files contain declarations for interfaces between the source files of our program. Each time we have a group of related declarations and macro definitions all or most of which are needed in several different source files, it is a good idea to create a header file for them.

Including a header file produces the same results as copying the header file into each source file that needs it. Such copying would be time-consuming and error-prone. With a header file, the related declarations appear in only one place. If they need to be changed, they can be changed in one place,

and programs that include the header file will automatically use the new version when next recompiled. The header file eliminates the labor of finding and changing all the copies as well as the

risk that a failure to find one copy will result in inconsistencies within a program.

C Header File

1. #include<stdio.h> (Standard input-output header)

Used to perform input and output operations in C like scanf() and printf().

1. #include<string.h> (String header)

Perform string manipulation operations like strlen and strcpy.

1. #include<conio.h> (Console input-output header)

Perform console input and console output operations like clrscr() to clear the screen and getch() to get the character from the keyboard.

1. #include<stdlib.h> (Standard library header)

Perform standard utility functions like dynamic memory allocation, using functions such as malloc() and calloc().

1. #include<math.h> (Math header )

Perform mathematical operations like sqrt() and pow(). To obtain the square root and the power of a number respectively.

1. #include<ctype.h>(Character type header)

Perform character type functions like isaplha() and isdigit(). To find whether the given character is an alphabet or a digit respectively.

1. #include<time.h>(Time header)

Perform functions related to date and time like setdate() and getdate(). To modify the system date and get the CPU time respectively.

Example using your own header file

// C program to use the above created header file #include <stdio.h> //built-in header file #include "myhead.h" //user defined header file

int main()

{

add(4, 6);

/\*This calls add function written in myhead.h and therefore no compilation error.\*/

multiply(5, 5);

// Same for the multiply function in myhead.h printf("BYE!See you Soon");

return 0;

}

Output:

Added value:10 Multiplied value:25 BYE!See you Soon

**NOTE :** The above code compiles successfully and prints the above output only if you have created the header file and saved it in the same folder the above c file is saved.

* 1. Write a program to read the name, author, and the price of 500 books in a library from the file ‘library.dat’. Now print the book name and price of those books whose price is above Rs.300.

Ans:

#include<stdlib.h> #include <stdio.h> int main()

{

char BookName[20], Author[20];

float Price;

FILE \*fptr;

fptr=fopen("D:\\library.dat","r");

if(fptr==NULL)

{

printf("Error!");

exit(1);

}

for(int i=0;i<2;++i)

{

sprintf("Enter Book Name"); scanf("%s",&BookName); printf("Enter Author Name"); scanf("%s",&Author);

printf("Enter Book Price"); scanf("%f",&Price); if(Price>300)

{

fprintf(fptr,"\n %s \t %f ", BookName,Price);

}

}

fclose(fptr); return 0;

}

### What are the different file opening modes in C? Write a program to input name, address, registration no, faculty and academic year of admission in university of ‘n’ number of students of ‘Pokhara University’ and append them on a data file called ‘STUDENT.DAT’ then display the records of those students by reading the records from ‘STUDENT.DAT’ data file who got admission in 2016.

Ans: First part solution Qno.1

#include <stdio.h> void main()

{

FILE \*fptr;

char name[20], address[20],faculty[20]; int reg , acd\_year, n;

printf("Enter number of students:"); scanf("%d",&n);

fptr=fopen("D://STUDENT.DAT","a");

if (fptr == NULL)

{

printf("File does not exists \n");

return;

}

for(int i=0;i<n;i++)

{

printf("Pokhara University \n"); printf("Enter the name \n");

scanf("%s",&name);

printf("Enter Address\n"); scanf("%s",&address);

printf("Enter the registration number\n");

scanf("%i", &reg);

printf("Faculty\n");

scanf("%s",&faculty);

printf("Academic Year \n");

scanf("%i",&acd\_year);

if(acd\_year=2016)

{

fprintf(fptr,"\nStudents Admitted in 2016 \n Name\t\tAddress\t\tReg Number\t\tFaculty\n %s\t\t%s\t\t%i\t\t%s\t ",name,address,reg,faculty);

}

}

fclose(fptr);

}

### Write short notes on file opening in C.[2015Fall]

Ans: Qno.1

### What are the different modes of opening a file? Write a program to create a file “hello.txt” , write data into the file and finally read the data from the user.[2017Fall]

Ans: First part …[2013Fall] Qno.1 #include <stdio.h>

int main()

{

int DATA\_SIZE=1000;

char data[DATA\_SIZE]; FILE \* fPtr;

fPtr = fopen("D://hello.txt","w"); if(fPtr == NULL)

{

printf("Unable to create file.\n"); exit(1);

}

printf("Enter contents to store in file : \n"); fgets(data, DATA\_SIZE, stdin);

fputs(data, fPtr); fclose(fPtr);

printf("File created and saved successfully. \n"); return 0;

}

### Why is file handling necessary in C programming? Write a program to input a name , address, faculty, program and GPA (in maximum 4.0) of 500 students and store them in 'RESULT.DAT' data file and display the records of those students whose faculty is 'Engineering" and GPA 3.5.[2016Spring]

Ans:[2013Spring] Qno.2

#include <stdio.h> #include<stdlib.h> int main()

{

char name[50],address[20],faculty[20],program[20]; int num,i;

float GPA;

printf("Enter number of students: "); scanf("%d", &num);

FILE \*fptr;

fptr = (fopen("D:\\Result.DAT", "w")); if(fptr == NULL)

{

printf("Error!"); exit(1);

}

fprintf(fptr,"\nName\t\tAddress\t\tFaculty\t\tProgram\t\tGPA\n"); for( i=0; i <num;i++ )

{

printf("For student %d \n",i+1); printf("Enter name:"); scanf("%s",&name); printf("Enter address:"); scanf("%s",&address); printf("Enter faculty:"); scanf("%s",&faculty); printf("Enter program"); scanf("%s", &program); printf("Enter GPA");

scanf("%f",&GPA); if(faculty=='engineering'||'Engineering' &&GPA==3.5)

{

fprintf(fptr,"%s\t\t%s\t\t%s\t\t%s\t\t%f\n", name,address,faculty,program,GPA);

}

}

fclose(fptr); return 0;

}