10. File management

Dr. Ami Tusharkant Choksi
Associate Professor,
Computer Engineering Department,
C.K.Pithawala College of Engineering & Technology, Surat.
2019
ami.choksi@ckpcet.ac.in

8. Structure

Introduction to file management and its functions

Total Hours: 04

Module Weightage: 9%

Table of contents

Introduction to file management and its functions

- To store information on the hard disk permanently, we use file.
- i.e.collection of information is accessed as a filename e.g.filename.txt

Operations on a file

- Creation of a new file (fopen with attributes as “a” or “a+” or “w” or “w++”)
- Opening an existing file (fopen)
- Reading from file (fscanf or fgetc, fgets)
- Writing to a file (fputs, fwrite)
- Moving to a specific location in a file (fseek, rewind)
- Closing a file (fclose)

fopen()

- To open a file fopen function is used.
- FILE *fopen(const char *filename, const char *mode) opens the filename pointed to, by filename using the given mode.
- filename is e.g.myfamily.txt
- mode is e.g. read mode, write mode, append mode.

fopen()

- Modes are:
  - “r” – Searches file. If the file is opened successfully fopen( ) loads it into memory and sets up a pointer which points to the first character in it. If the file cannot be opened fopen( ) returns NULL.
  - “w” – Searches file. If the file exists, its contents are overwritten. If the file doesn’t exist, a new file is created. Returns NULL, if unable to open file.

fopen()

- “a” – Searches file. If the file is opened successfully fopen( ) loads it into memory and sets up a pointer that points to the last character in it. If the file doesn’t exist, a new file is created. Returns NULL, if unable to open file.
- “r+” – Searches file. If is opened successfully fopen( ) loads it into memory and sets up a pointer which points to the first character in it. Returns NULL, if unable to open the file.

fopen()

- “w+” – Searches file. If the file exists, its contents are overwritten. If the file doesn’t exist a new file is created. Returns NULL, if unable to open file.
- “a+” – Searches file. If the file is opened successfully fopen( ) loads it into memory and sets up a pointer which points to the last character in it. If the file doesn’t exist, a new file is created. Returns NULL, if unable to open file.

fopen()-syntax

- `FILE *filePointer;`
- So, the file can be opened as
- `filePointer = fopen("fileName.txt", "w")`
- Read Mode
- `filePointer = fopen("fileName.txt", "r")`
- Append Mode
- `filePointer = fopen("fileName.txt", "a")`

File Write Program

```
#include <stdio.h>
#include <string.h>
int main(){
FILE *fptr;
char data[50]="Writing file for the first time.";
fptr = fopen("FileW.txt","w");
if(fptr==NULL){
printf("file open for write failed");}
```

File Write Program

```
else{
if(strlen(data)){
fputs(data,fptr);
fputs("\n",fptr);}
fclose(fptr);
printf("Data successfullly written");
}
return 0;}
```

Output

File name “FileW.txt” is created and contents are written in it.

We can see the content of file in Ubuntu terminal as, follows

$ cat FileW.txt

Writing file for the first time.

File Read Program

```
#include <stdio.h>
#include <string.h>
int main(){
FILE *fptr;
char data[50];
fptr = fopen("FileW.txt","r");
if(fptr==NULL){
printf("file open for write failed");
}
```

File Read Program

```
else{
while(fgets(data,50,fptr)!=NULL){
printf("%s",data);
}
fclose(fptr);
printf("Data successfully read");
}
return 0;
}
```

File Read Program: Ouptut

Writing file for the first time.

Data successfully read

File Append Program

```
#include <stdio.h>
#include <string.h>
int main(){
FILE *fptr;
char data[50]="Appending file.";
fptr = fopen("FileW.txt","a");
if(fptr==NULL){
printf("file open for append failed");}
```

File Append Program

```
else{
if(strlen(data)){
fputs(data,fptr);
fputs("\n",fptr);}
fclose(fptr);
printf("Data successfully appended");
}
return 0;}
```

Output

- Output:

Data successfully appended

- $ cat FileW.txt

Writing file for the first time.

Appending file.

Output:explanation

- Contents are appended to “FileW.txt”, if file is already exists.
- File name “FileW.txt” is created if not already available and contents are written in it.
- We can see the content of file in Ubuntu terminal as, follows
- $ cat FileW.txt
- Writing file for the first time.

Read file characterwise

```
#include <stdio.h>
int main () {
    FILE *fp;
    int c;
    fp = fopen("FileW.txt","r");
```

Read file characterwise

```
    while(1) {
        c = fgetc(fp);
        if( feof(fp) ) {
            break ;         }
        printf("%c", c);
    }
    fclose(fp);
    return(0);
}
```

Read file characterwise: Output

Writing file for the first time.

Appending file.

Moving to a specific location in a file

- int fseek(FILE *stream, long int offset, int whence)
- fseek() sets the file pointer to a specific location in file. The can be
- SEEK_CUR, SEEK_END, and SEEK_SET
- These macros are used in the fseek function to locate different positions in a file.

fseek() write program

```
#include <stdio.h>
int main () {
   FILE *fp;
   fp = fopen("FileW.txt","w+");
   fseek( fp, 7, SEEK_SET );
   fputs(" C Programming Language",
fp);
   fclose(fp);
   return(0);}
```

fseek() program: Output

```
This is C Programming Language
```

–set the write pointer at 7th position from the beginning and used puts() statement which over-write the file with the above content.

References

1. File Management in C, https://www.geeksforgeeks.org/basics-file-handling-c/
2. File I/O, https://www.tutorialspoint.com/c_standard_library/c_function_fopen.htm
3. fseek() and ftell(), https://www.geeksforgeeks.org/fseek-in-c-with-example/
4. fseek, https://fresh2refresh.com/c-programming/c-file-handling/fseek-seek_set-seek_cur-seek_end-functions-c/
5. fseek, https://www.javatpoint.com/fseek-in-c