

File Management in 'C'

The file management is a more flexible approach where data can be stored on the disks and read whenever necessary, without destroying the data. Appointments ~ Meetings

This method employs the concept of files to store data.

A file is a place on the disk where a group of related data is stored. Like most other languages, C supports a number of functions that have the ability to perform basic file operations, which include:

- (i) Naming a file.
- (ii) Opening a file.
- (iii) Reading data from a file.
- (iv) Writing data to a file.
- (v) Closing a file.

There are two distinct ways to perform file operations in C. The first one is known as the low-level I/O and uses UNIX system calls. The second method is referred to as the high-level I/O operation and uses functions in C's standard I/O library.

	M	T	W	T	F	S	S
	1	2	3
4	5	6	7	8	9	10	
11	12	13	14	15	16	17	
18	19	20	21	22	23	24	
25	26	27	28	29	30	.	.

31-03-2011

Thursday
090-275 • Week 13

MARCH

31

File Management.

1. What is File.

2. Operations on file

- (i) creation of a file
- (ii) opening a file
- (iii) Reading a file.
- (iv) Writing a file
- (v) closing a file.

3. Functions of file

- (i) fopen()
- (ii) fclose()
- (iii) fprintf()
- (iv) fscanf()
- (v)getc()
- (vi)putc()
- (vii) getw()
- (viii) putw()
- (ix) fseek()
- (x) ftell()
- (xi) rewind()

01

Friday
091-274 . Week 13

APRIL

01-04-2011

M	T	W	T	F	S	S
.	.	.	.	1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	.

4. Mode on File.

	<u>Mode</u>	<u>Meaning.</u>
(i)	r	Read
(ii)	w	Write
(iii)	a	append
(iv)	r+	Reading & writing from beginning.
(v)	w+	Reading & writing, Overwriting a file.
(vi)	a+	Reading & writing, appending to file.

5. How to create a File:

Syntax:

```
FILE *fptr;
fptr = fopen ("File_Name", "mode");
```

Example:

```
#include <stdio.h>
void main()
{
FILE *fptr;
fptr = fopen (" abc.txt ", " w ");
}
```

```
void main()
{
FILE *fptr;
fptr = fopen (" D: // abc.txt ", " w ");
}
```

When liberty becomes license, dictatorship is near

M	T	W	T	F	S	S
30	31	1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29

02-04-2011

Saturday
092-273 • Week 13

APRIL

02

6. How to close a File:

Syntax: `fclose (File_Name);`

Appointments Meetings

Example:

```
#include <stdio.h>
void main()
{
    FILE *fptr;
    fptr = fopen ( " abc.txt ", " r " );
    fclose ( fptr );
}
```

7. Writing to a File:

The stdio library offers the necessary functions to write a file.

- (i) `fputc (char, File_pointer);`
- (ii) `fputs (str, File_pointer);`
- (iii) `fprintf (File_pointer, str, variable_lists);`

Sunday 03

M	T	W	T	F	S	S
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31

191 Appointments Meetings

putc() Function :

```
#include <stdio.h>
```

```
#include <stdlib.h>
```

```
int main()
```

```
{
```

```
FILE *fptr;
```

```
char x;
```

```
int i;
```

```
fptr = fopen("abc.txt", "w");
```

```
printf("Enter five character");
```

```
for (i=0; i<5; i++)
```

```
{
```

```
scanf(" %c", &x);
```

```
putc(x, fptr);
```

```
fflush(stdin); // clear the stdin stream buffer.
```

```
}
```

```
fclose(fptr);
```

```
return 0;
```

```
}
```

	M	T	W	T	F	S	S
MAY	30	31	1
	2	3	4	5	6	7	8
	9	10	11	12	13	14	15
	16	17	18	19	20	21	22
	23	24	25	26	27	28	29

07-04-2011

Thursday
097-268 • Week 14

APRIL

07

9. fputc()

Appointments Meetings

```
#include <stdio.h>
int main ()
{
    FILE *fptr;
    int x;
    fptr = fopen("abc.txt", "w");

    for (x = 33; x <= 100; x++)
    {
        fputc(x, fptr);
    }
    fclose(fptr);

    return 0;
}
```

OUTPUT: ! " # \$ % ' ()

ABCD - ISK

M	T	W	T	F	S	S
30	31	*	*	*	*	1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29

05-04-2011

Tuesday
095-270 • Week 14
APRIL

05

9. fputs() function:

```
#include <stdio.h>
```

```
int main()
```

```
{
```

```
FILE * fptr;
```

```
fptr = fopen("abc.txt", "w+");
```

```
fputs("I am a good boy", fptr);
```

```
fputs("He is a doctor", fptr);
```

```
fputs("I am going to the market", fptr);
```

```
fclose(fptr);
```

```
return (0);
```

```
}
```

Appointments & Meetings

10.

fprintf() Function:

#include <stdio.h>

int main()

{

FILE *fptr;

fptr = fopen("abc.txt", "w");

fprintf(fptr, "I am a good boy");

fclose(fptr);

return 0;

}

	M	T	W	T	F	S	S
	30	31	1
MAY	2	3	4	5	6	7	8
	9	10	11	12	13	14	15
	16	17	18	19	20	21	22
	23	24	25	26	27	28	29

Reading Data from a File

There are three different functions Appointments and Meetings dedicated to reading data from a file.

- (i) `fgetc (File_pointer) ;`
- (ii) `fgets (Buffer, n, file_pointer) ;`
- (iii) `fscanf (file_pointer, Conversion_specifiers, Variable_addresses) ;`

M	T	W	T	F	S	S
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31

Appointments & Meetings

①

fgetc () Function:

D: Open, Read and close a file: Reading char by char.

#include <stdio.h>

void main()

{

FILE *fptr;

char c;

fptr = fopen ("abc.txt", "r");

while (1)

{

c = fgetc (fptr);

if (c == EOF)

break;

printf (" %c", c);

}

fclose (fptr);

}

1	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30
31	32	33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48	49	50
51	52	53	54	55	56	57	58	59	60
61	62	63	64	65	66	67	68	69	70
71	72	73	74	75	76	77	78	79	80
81	82	83	84	85	86	87	88	89	90
91	92	93	94	95	96	97	98	99	100

12-04-2011

APRIL

Appointments & Meetings

② fgets () Function

```
#include <stdio.h>
#include <stdlib.h>
void main()
{
```

```
char s[60];
FILE *fptr;
```

```
fptr = fopen("abc.txt", "r");
```

```
if (fptr == NULL)
{
```

```
printf("Error opening file");
exit(1);
}
```

```
while (fgets(s, 30, fptr) != NULL)
{
```

```
puts(s);
}
```

```
fclose(fptr);
```

```
}
```

Appointments or Meetings

③ fscanf() Function:

```
#include <stdio.h>
#include <stdlib.h>
```

```
int main()
{
```

```
FILE *fptr;
char name[50];
int roll;
float marks;
```

```
fptr = fopen("abc.txt", "r");
```

```
if (fptr == NULL)
```

```
{
    printf("Error opening file");
    exit(1);
}
```

```
while (fscanf(fptr, "%s %d %f", name, &roll, &marks))
{
    printf("%s %d %f", name, roll, marks);
}
```

```
fclose(fptr);
return 0;
```

```
}
```