

# File Handling in C

CSE 1320

# File Handling in C

Part of the structure defined in the `typedef FILE` is a value that tracks the current position in the file.

We will refer to that as the ***file pointer***.

The file pointer moves as reads and writes are done.

# File Handling in C

## Two Types of Access

### **Sequential Access**

When a file is opened, reading (or writing) starts at the beginning of the file and proceeds through the file in a sequential manner.

Whenever a read is done, the file pointer moves to point to the next element in the file to be read.

### **Random Access**

Allows the reading of the records in any order.

# Random Access in Files

Two library functions in the standard C library help with random access of files

```
fseek(fp, offset, start);
```

<code>fp</code>	file handle ( <code>FILE *</code> ) - associated with an open file
<code>offset</code>	variable of type <code>long</code> that represents the byte offset or number of bytes that the pointer is to be moved
<code>start</code>	indicates the beginning position for the file pointer must be 0, 1 or 2

```
ftell(fp);
```

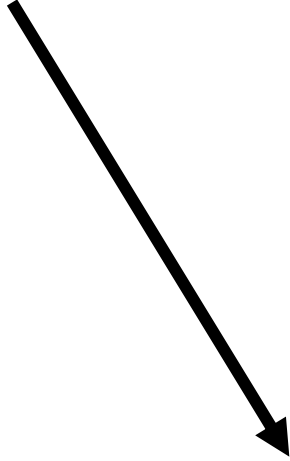
<code>fp</code>	file handle ( <code>FILE *</code> ) – associated with an open file returns the current byte offset from the beginning of the file
-----------------	--

```

for (i = 0; i < 5; i++)
{
    printf("Enter string %d ", i);
    fgets(buffer, sizeof(buffer), stdin);
    fprintf(MyFile, "%s", buffer);
}


```

Enter string 0	Hello
Enter string 1	there.
Enter string 2	How
Enter string 3	are
Enter string 4	you?



										1											2					
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	
H	e	l	l	o	\n	t	h	e	r	e	.	\n	H	o	w	\n	a	r	e	\n	y	o	u	?	\n	





										1										2									
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6			
H	e	l	l	o	\n	t	h	e	r	e	.	\n	H	o	w	\n	a	r	e	\n	y	o	u	?	\n				

```
fseek(MyFile, 0, 0);
```

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

```
{
```

```
    printf("ftell() = %d\t", ftell(MyFile));
```

```
    fscanf(MyFile, "%s", &buffer);
```

```
    printf("Printing string %d from file : %s\t", i, buffer);
```

```
    printf("ftell() = %d\n", ftell(MyFile));
```

```
}
```

*fscanf() stops at \n*

ftell() = 0	Printing string 0 from file : Hello	ftell() = 5
ftell() = 5	Printing string 1 from file : there.	ftell() = 12
ftell() = 12	Printing string 2 from file : How	ftell() = 16
ftell() = 16	Printing string 3 from file : are	ftell() = 20
ftell() = 20	Printing string 4 from file : you?	ftell() = 25







# Random Access in Files

Defines from `stdio.h` that can be used with file access

```
#define SEEK_SET          0          /* Seek from beginning of file.  */
#define SEEK_CUR          1          /* Seek from current position.  */
#define SEEK_END          2          /* Seek from end of file.  */
```