C LIBRARY FUNCTION - SETVBUF

Description

The C library function **int setvbuf***FILE ∗ stream, char ∗ buffer, intmode, size*$_t$*size* defines how a stream should be buffered.

Declaration

Following is the declaration for setvbuf function.

```
int setvbuf(FILE *stream, char *buffer, int mode, size_t size)
```

Parameters

- **stream** − This is the pointer to a FILE object that identifies an open stream.
- **buffer** − This is the user allocated buffer. If set to NULL, the function automatically allocates a buffer of the specified size.
- **mode** − This specifies a mode for file buffering −

mode	Description
_IOFBF	**Full buffering** − On output, data is written once the buffer is full. On Input the buffer is filled when an input operation is requested and the buffer is empty.
_IOLBF	**Line buffering** − On output, data is written when a newline character is inserted into the stream or when the buffer is full, what so ever happens first. On Input, the buffer is filled till the next newline character when an input operation is requested and buffer is empty.
_IONBF	**No buffering** − No buffer is used. Each I/O operation is written as soon as possible. The buffer and size parameters are ignored.

- **size** − This is the buffer size in bytes

Return Value

This function returns zero on success else, non-zero value is returned.

Example

The following example shows the usage of setvbuf function.

```
#include <stdio.h>

int main()
{
   char buff[1024];

   memset( buff, '\0', sizeof( buff ));

   fprintf(stdout, "Going to set full buffering on\n");
   setvbuf(stdout, buff, _IOFBF, 1024);

   fprintf(stdout, "This is tutorialspoint.com\n");
   fprintf(stdout, "This output will go into buff\n");
   fflush( stdout );
```

```
   fprintf(stdout, "and this will appear when programm\n");
   fprintf(stdout, "will come after sleeping 5 seconds\n");

   sleep(5);

   return(0);
}
```

Let us compile and run the above program to produce the following result. Here program keeps buffering the output into **buff** until it faces first call to fflush, after which it again starts buffering the output and finally sleeps for 5 seconds. It sends remaining output to the STDOUT before the program comes out.

```
Going to set full buffering on
This is tutorialspoint.com
This output will go into buff
and this will appear when programm
will come after sleeping 5 seconds
```