



Linux Ubuntu 26.04 Manual Pages on command 'setvbuf.3'

\$ man setvbuf.3

setbuf(3) Library Functions Manual setbuf(3)

NAME

setbuf, setbuffer, setlinebuf, setvbuf - stream buffering operations

LIBRARY

Standard C library (libc, -lc)

SYNOPSIS

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

```
int setvbuf(size_t size;
```

```
FILE *restrict stream, char buf[restrict size],
```

```
int mode, size_t size);
```

```
void setbuf(FILE *restrict stream, char *restrict buf);
```

```
void setbuffer(size_t size;
```

```
FILE *restrict stream, char buf[restrict size],
```

```
size_t size);
```

```
void setlinebuf(FILE *stream);
```

Feature Test Macro Requirements for glibc (see feature_test_macros(7)):

setbuffer(), setlinebuf():

Since glibc 2.19:

```
_DEFAULT_SOURCE
```

glibc 2.19 and earlier:

```
_BSD_SOURCE
```

DESCRIPTION

The three types of buffering available are unbuffered, block buffered, and line buffered.

Normally, all files are block buffered. If a stream refers to a terminal (as `stdout` normally does), it is line buffered. The standard error stream `stderr` is always unbuffered by default.

- _IONBF unbuffered
- _IOLBF line buffered
- IOFBF fully buffered

The other three calls are, in effect, simply aliases for calls to `setvbuf()`. The `setbuf()` function is exactly equivalent to the call

The `setbuffer()` function is the same, except that the size of the buffer is up to the caller, rather than being determined by the default `BUFSIZ`. The `setlinebuf()` function is exactly equivalent to the call:

The function `setvbuf()` returns 0 on success. It returns nonzero on failure (mode is invalid or the request cannot be honored). It may set `errno` on failure.

Page 2/3

