

**NAME**

**libbuf** - minimal dynamic string library

**LIBRARY**

libbuf (libbuf, -lbuf)

**SYNOPSIS**

**#include** <buf.h>

*void*

**buf\_clear**(*struct buf \*b*);

*bool*

**buf\_dup**(*struct buf \*b, const struct buf \*src*);

*void*

**buf\_erase**(*struct buf \*b, size\_t pos, size\_t count*);

*void*

**buf\_finish**(*struct buf \*b*);

*bool*

**buf\_init**(*struct buf \*b*);

*bool*

**buf\_printf**(*struct buf \*b, const char \*fmt, ...*);

*bool*

**buf\_putc**(*struct buf \*b, char c*);

*bool*

**buf\_puts**(*struct buf \*b, const char \*s*);

*bool*

**buf\_reserve**(*struct buf \*b, size\_t desired*);

*bool*

**buf\_resize**(*struct buf \*b, size\_t desired, char c*);

*bool*

**buf\_shrink**(*struct buf \*b*);

*bool*

**buf\_sub**(*struct buf \*b, const struct buf \*src, size\_t pos, size\_t count*);

*bool*

**buf\_vprintf**(*struct buf \*b, const char \*fmt, va\_list ap*);

## DESCRIPTION

The **libbuf** library is a general purpose string library for C.

It automatically expands storage as required. It will first try to allocate twice as the current storage space until enough room is available or in case it would exceed maximum storage it will grow with minimal required storage.

Every function expects as first argument a buffer to work with which can never be NULL.

The *struct buf* is a publicly exposed structure with the following fields:

*data* (*char \**) The current NUL-terminated C string, maybe NULL if the buffer isn't initialized.

*length* (*size\_t*) The data length.

*capacity* (*size\_t*) The real capacity available for writing characters (not including NUL). A capacity of 5 means you can write 5 characters plus the NUL terminator without reallocating.

## SEE ALSO

buf\_clear(3), buf\_dup(3), buf\_erase(3), buf\_finish(3), buf\_init(3), buf\_printf(3), buf\_putc(3), buf\_puts(3), buf\_reserve(3), buf\_resize(3), buf\_shrink(3), buf\_sub(3), buf\_vprintf(3)

## AUTHORS

The **libbuf** library was written by David Demelier <markand@malikania.fr>