Full credit is given to the above companies including the OS that this PDF file was generated!

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

$ man malloc_get_state.3

malloc_get_state(3) Library Functions Manual malloc_get_state(3)

NAME

malloc_get_state, malloc_set_state - record and restore state of malloc implementation

LIBRARY

Standard C library (libc, -lc)

SYNOPSIS

```
#include <malloc.h>

void *malloc_get_state(void);

int malloc_set_state(void *state);
```

DESCRIPTION

Note: these functions are removed in glibc 2.25.

The malloc_get_state() function records the current state of all malloc(3) internal book? keeping variables (but not the actual contents of the heap or the state of malloc_hook(3) functions pointers). The state is recorded in a system-dependent opaque data structure dy? namically allocated via malloc(3), and a pointer to that data structure is returned as the function result. (It is the caller's responsibility to free(3) this memory.)

The malloc_set_state() function restores the state of all malloc(3) internal bookkeeping variables to the values recorded in the opaque data structure pointed to by state.

RETURN VALUE

On success, malloc_get_state() returns a pointer to a newly allocated opaque data structure.

On error (for example, memory could not be allocated for the data structure), mal? loc_get_state() returns NULL.

On success, malloc_set_state() returns 0. If the implementation detects that state does not

point to a correctly formed data structure, malloc_set_state() returns -1. If the implemen?
tation detects that the version of the data structure referred to by state is a more recent
version than this implementation knows about, malloc_set_state() returns -2.

ATTRIBUTES

For an explanation of the terms used in this section, see attributes(7).

Interface	Attribute	Value
malloc_get_state(), malloc_set_state()	Thread safety	MT-Safe

STANDARDS

GNU.

NOTES

These functions are useful when using this malloc(3) implementation as part of a shared li?
brary, and the heap contents are saved/restored via some other method. This technique is
used by GNU Emacs to implement its "dumping" function.

Hook function pointers are never saved or restored by these functions, with two exceptions:
if malloc checking (see mallopt(3)) was in use when malloc_get_state() was called, then mal?
loc_set_state() resets malloc checking hooks if possible; if malloc checking was not in use
in the recorded state, but the caller has requested malloc checking, then the hooks are re?
set to 0.

SEE ALSO

malloc(3), mallopt(3)