



Red Hat Enterprise Linux Release 9.2 Manual Pages on 'calloc.3p' command

\$ man calloc.3p

CALLOC(3P) POSIX Programmer's Manual CALLOC(3P)

PROLOG

This manual page is part of the POSIX Programmer's Manual. The Linux implementation of this interface may differ (consult the corresponding Linux manual page for details of Linux behavior), or the interface may not be implemented on Linux.

NAME

calloc ? a memory allocator

SYNOPSIS

```
#include <stdlib.h>

void *calloc(size_t nelem, size_t elsize);
```

DESCRIPTION

The functionality described on this reference page is aligned with the ISO C standard. Any conflict between the requirements described here and the ISO C standard is unintentional. This volume of POSIX.1?2017 defers to the ISO C standard.

The calloc() function shall allocate unused space for an array of nelem elements each of whose size in bytes is elsize. The space shall be initialized to all bits 0.

The order and contiguity of storage allocated by successive calls to calloc() is unspecified. The pointer returned if the allocation succeeds shall be suitably aligned so that it may be assigned to a pointer to any type of object and then used to access such an object or an array.

ray of such objects in the space allocated (until the space is explicitly freed or reallocated). Each such allocation shall yield a pointer to an object disjoint from any other object. The pointer returned shall point to the start (lowest byte address) of the allocated space. If the space cannot be allocated, a null pointer shall be returned. If the size of the space requested is 0, the behavior is implementation-defined: either a null pointer shall be returned, or the behavior shall be as if the size were some non-zero value, except that the behavior is undefined if the returned pointer is used to access an object.

RETURN VALUE

Upon successful completion with both `nelem` and `elsize` non-zero, `calloc()` shall return a pointer to the allocated space. If either `nelem` or `elsize` is 0, then either:

- * A null pointer shall be returned and `errno` may be set to an implementation-defined value, or
- * A pointer to the allocated space shall be returned. The application shall ensure that the pointer is not used to access an object.

Otherwise, it shall return a null pointer and set `errno` to indicate the error.

ERRORS

The `calloc()` function shall fail if:

`ENOMEM` Insufficient memory is available.

The following sections are informative.

EXAMPLES

None.

APPLICATION USAGE

There is now no requirement for the implementation to support the inclusion of `<malloc.h>`.

RATIONALE

None.

FUTURE DIRECTIONS

None.

SEE ALSO

free(), malloc(), realloc()

The Base Definitions volume of POSIX.1-2017, <stdlib.h>

COPYRIGHT

Portions of this text are reprinted and reproduced in electronic form from IEEE Std 1003.1-2017, Standard for Information Technology -- Portable Operating System Interface (POSIX), The Open Group Base Specifications Issue 7, 2018 Edition, Copyright (C) 2018 by the Institute of Electrical and Electronics Engineers, Inc and The Open Group. In the event of any discrepancy between this version and the original IEEE and The Open Group Standard, the original IEEE and The Open Group Standard is the referee document. The original Standard can be obtained online at <http://www.opengroup.org/unix/online.html>.

Any typographical or formatting errors that appear in this page are most likely to have been introduced during the conversion of the source files to man page format. To report such errors, see https://www.kernel.org/doc/man-pages/reporting_bugs.html.

IEEE/The Open Group

2017

CALLOC(3P)