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

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

$ man realloc.3p

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

realloc ? memory reallocator

SYNOPSIS

```
#include <stdlib.h>
void *realloc(void *ptr, size_t size);
```

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 realloc() function shall deallocate the old object pointed to by ptr and return a pointer to a new object that has the size specified by size. The contents of the new object shall be the same as that of the old object prior to deallocation, up to the lesser of the new and old sizes. Any bytes in the new object beyond the size of the old object have indeterminate values. If the size of the space requested is zero, the behavior shall be implementation-defined: either a null pointer is

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. If the space cannot be allocated, the object shall remain unchanged.

If ptr is a null pointer, realloc() shall be equivalent to malloc() for the specified size.

If ptr does not match a pointer returned earlier by calloc(), malloc(), or realloc() or if the space has previously been deallocated by a call to free() or realloc(), the behavior is undefined.

The order and contiguity of storage allocated by successive calls to realloc() is unspecified. The pointer returned if the allocation suc? ceeds 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 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.

RETURN VALUE

Upon successful completion, realloc() shall return a pointer to the (possibly moved) allocated space. If size is 0, either:

* A null pointer shall be returned and, if ptr is not a null pointer, errno shall be set to an implementation-defined value.
* A pointer to the allocated space shall be returned, and the memory object pointed to by ptr shall be freed. The application shall en? sure that the pointer is not used to access an object.

If there is not enough available memory, realloc() shall return a null pointer and set errno to [ENOMEM]. If realloc() returns a null pointer and errno has been set to [ENOMEM], the memory referenced by ptr shall not be changed.

ERRORS

The realloc() function shall fail if:

ENOMEM Insufficient memory is available.

The following sections are informative.

EXAMPLES

None.

APPLICATION USAGE

The description of realloc() has been modified from previous versions of this standard to align with the ISO/IEC 9899:1999 standard. Previous versions explicitly permitted a call to realloc(p, 0) to free the space pointed to by p and return a null pointer. While this behavior could be interpreted as permitted by this version of the standard, the C lan? guage committee have indicated that this interpretation is incorrect. Applications should assume that if realloc() returns a null pointer, the space pointed to by p has not been freed. Since this could lead to double-frees, implementations should also set errno if a null pointer actually indicates a failure, and applications should only free the space if errno was changed.

RATIONALE

None.

FUTURE DIRECTIONS

This standard defers to the ISO C standard. While that standard cur? rently has language that might permit realloc(p, 0), where p is not a null pointer, to free p while still returning a null pointer, the com? mittee responsible for that standard is considering clarifying the lan? guage to explicitly prohibit that alternative.

SEE ALSO

calloc(), free(), malloc()

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

COPYRIGHT

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.ker?nel.org/doc/man-pages/reporting_bugs.html .