



*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 'posix\_memalign.3p' command**

### **\$ man posix\_memalign.3p**

POSIX\_MEMALIGN(3P)      POSIX Programmer's Manual      POSIX\_MEMALIGN(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

posix\_memalign ? aligned memory allocation (ADVANCED REALTIME)

#### SYNOPSIS

```
#include <stdlib.h>

int posix_memalign(void **memptr, size_t alignment, size_t size);
```

#### DESCRIPTION

The posix\_memalign() function shall allocate size bytes aligned on a boundary specified by alignment, and shall return a pointer to the allocated memory in memptr. The value of alignment shall be a power of two multiple of sizeof(void \*).

Upon successful completion, the value pointed to by memptr shall be a multiple of alignment.

If the size of the space requested is 0, the behavior is implementation-defined: either a null pointer shall be returned in memptr, or the behavior shall be as if the size were some non-zero value, except that the behavior is undefined if the value returned in memptr is used to access an object.

The free() function shall deallocate memory that has previously been allocated by posix\_memalign().

## RETURN VALUE

Upon successful completion, posix\_memalign() shall return zero; otherwise, an error number shall be returned to indicate the error and the contents of memptr shall either be left unmodified or be set to a null pointer.

If size is 0, either:

- \* posix\_memalign() shall not attempt to allocate any space, in which case either an implementation-defined error number shall be returned, or zero shall be returned with a null pointer returned in memptr, or
- \* posix\_memalign() shall attempt to allocate some space and, if the allocation succeeds, zero shall be returned and a pointer to the allocated space shall be returned in memptr. The application shall ensure that the pointer is not used to access an object.

## ERRORS

The posix\_memalign() function shall fail if:

**EINVAL** The value of the alignment parameter is not a multiple of sizeof(void \*).

**ENOMEM** There is insufficient memory available with the requested alignment.

The following sections are informative.

## EXAMPLES

The following example shows how applications can obtain consistent behavior on error by setting \*memptr to be a null pointer before calling posix\_memalign().

```
void *ptr = NULL;
...
//do some work, which might goto error
if (posix_memalign(&ptr, align, size))
    goto error;
//do some more work, which might goto error
```

...

error:

```
free(ptr);
```

```
//more cleanup;
```

## APPLICATION USAGE

The `posix_memalign()` function is part of the Advisory Information option and need not be provided on all implementations.

## RATIONALE

None.

## FUTURE DIRECTIONS

None.

## SEE ALSO

`free()`, `malloc()`

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](https://www.kernel.org/doc/man-pages/reporting_bugs.html).