NAME

cfree − free allocated memory

LIBRARY

Standard C library (*libc*, *−lc*)

SYNOPSIS

#include <stdlib.h>

/* In SunOS 4 */
int cfree(void *_ptr_);

/* In glibc or FreeBSD libcompat */
void cfree(void *_ptr_);

/* In SCO OpenServer */
void cfree(char *ptr***[.***size* *** .***num***], unsigned int** *num***, unsigned int** *size***);**

/* In Solaris watchmalloc.so.1 */
void cfree(void *ptr***[.***elsize* *** .***nelem***], size_t** *nelem***, size_t** *elsize***);**

Feature Test Macro Requirements for glibc (see **feature_test_macros**(7)):

cfree():
Since glibc 2.19:
_DEFAULT_SOURCE
glibc 2.19 and earlier:
_BSD_SOURCE || _SVID_SOURCE

DESCRIPTION

This function should never be used. Use **free**(3) instead. Starting with glibc 2.26, it has been removed from glibc.

1-arg cfree

In glibc, the function **cfree**() is a synonym for **free**(3), "added for compatibility with SunOS".

Other systems have other functions with this name. The declaration is sometimes in<*stdlib .h*> and sometimes in<*malloc.h*>.

3-arg cfree

Some SCO and Solaris versions have malloc libraries with a 3-argument **cfree**(), apparently as an analog to **calloc**(3).

If you need it while porting something, add

```
#define cfree(p, n, s) free((p))
```

to your file.

A frequently asked question is "Can I use **free**(3) to free memory allocated with **calloc**(3), or do I need **cfree**()?" Answer: use**fr ee**(3).

An SCO manual writes: "The cfree routine is provided for compliance to the iBCSe2 standard and simply calls free. The num and size arguments to cfree are not used."

RETURN VALUE

The SunOS version of **cfree**() (which is a synonym for **free**(3)) returns 1 on success and 0 on failure. In case of error, *errno* is set to **EINVAL**: the value of *ptr* was not a pointer to a block previously allocated by one of the routines in the **malloc**(3) family.

VERSIONS

The **cfree**() function was removed in glibc 2.26.

ATTRIBUTES

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

Interface	Attribute	Value
cfree()	Thread safety	MT-Safe /* In glibc */

STANDARDS

The 3-argument version of **cfree**() as used by SCO conforms to the iBCSe2 standard: Intel386 Binary Compatibility Specification, Edition 2.

SEE ALSO

malloc(3)