NAME

strcasecmp, **strncasecmp** **-** compare strings, ignoring case

LIBRARY

Standard C Library (libc, -lc)

SYNOPSIS

#include <strings.h>

int
strcasecmp(*const char *s1*, *const char *s2*);

int
strncasecmp(*const char *s1*, *const char *s2*, *size_t len*);

#include <strings.h>
#include <xlocale.h>

int
strcasecmp_l(*const char *s1*, *const char *s2*, *locale_t loc*);

int
strncasecmp_l(*const char *s1*, *const char *s2*, *size_t len*, *locale_t loc*);

DESCRIPTION

The **strcasecmp**() and **strncasecmp**() functions compare the null-terminated strings *s1* and *s2*.

The **strncasecmp**() function compares at most *len* characters. The **strcasecmp_l**() and **strncasecmp_l**() functions do the same as their non-locale versions above, but take an explicit locale rather than using the current locale.

RETURN VALUES

The functions **strcasecmp**() and **strncasecmp**() return an integer greater than, equal to, or less than 0, depending on whether *s1* is lexicographically greater than, equal to, or less than *s2* after translation of each corresponding character to lower-case. The strings themselves are not modified. The comparison is done using unsigned characters, so that '\200' is greater than '\0'. The functions **strcasecmp_l**() and **strncasecmp_l**() do the same but take explicit locales.

SEE ALSO

bcmp(3), memcmp(3), strcmp(3), strcoll(3), strxfrm(3), tolower(3), wcscasecmp(3)

HISTORY

The **strcasecmp**() and **strncasecmp**() functions first appeared in 4.4BSD. Their prototypes existed previously in <*string.h*> before they were moved to <*strings.h*> for IEEE Std 1003.1-2001 ("POSIX.1") compliance.