NAME

strcat, strncat, strcmp, strncmp, strcasecmp, strncasecmp, strcpy, strncpy, strlen, index, rindex − string operations

SYNOPSIS

#include <strings.h>

char *strcat(s, append)
char *s, *append;

char *strncat(s, append, count)
char *s, *append;
int count;

strcmp(s1, s2)
char *s1, *s2;

strncmp(s1, s2, count)
char *s1, *s2;
int count;

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

strncasecmp(s1, s2, count)
char *s1, *s2;
int count;

char *strcpy(to, from)
char *to, *from;

char *strncpy(to, from, count)
char *to, *from;
int count;

strlen(s)
char *s;

char *index(s, c)
char *s, c;

char *rindex(s, c)
char *s, c;

DESCRIPTION

These functions operate on null-terminated strings. They do not check for overflow of any receiving string.

Strcat appends a copy of string *append* to the end of string *s*. *Strncat* copies at most *count* characters. Both return a pointer to the null-terminated result.

Strcmp compares its arguments and returns an integer greater than, equal to, or less than 0, according as *s1* is lexicographically greater than, equal to, or less than *s2*. *Strncmp* makes the same comparison but looks at at most *count* characters. *Strcasecmp* and *strncasecmp* are identical in function, but are case insensitive. The returned lexicographic difference reflects a conversion to lower-case.

Strcpy copies string *from* to *to*, stopping after the null character has been moved. *Strncpy* copies exactly *count* characters, appending nulls if *from* is less than *count* characters in length; the target may not be null-terminated if the length of *from* is *count* or more. Both return *to*.

Strlen returns the number of non-null characters in *s*.

Index (*rindex*) returns a pointer to the first (last) occurrence of character *c* in string *s* or zero if *c* does not occur in the string. Setting *c* to NULL works.