NAME

stralloc – dynamically allocated strings

SYNTAX

#include <stralloc.h>

int **stralloc_ready**(&*sa*,*len*);
int **stralloc_readyplus**(&*sa*,*len*);

int **stralloc_copy**(&*sa*,&*sa2*);
int **stralloc_copys**(&*sa*,*buf*);
int **stralloc_copyb**(&*sa*,*buf*,*len*);

int **stralloc_cat**(&*sa*,&*sa2*);
int **stralloc_cats**(&*sa*,*buf*);
int **stralloc_catb**(&*sa*,*buf*,*len*);

int **stralloc_append**(&*sa*,*buf*);
int **stralloc_0**(&*sa*);

int **stralloc_starts**(&*sa*,*buf*);

stralloc *sa* = {0};
stralloc *sa2* = {0};
unsigned int *len*;
char **buf*;

DESCRIPTION

A **stralloc** variable holds a string in dynamically allocated space. String length is limited only by memory. String contents are unrestricted.

The **stralloc** structure has three components: *sa***.s** is a pointer to the string, or 0 if it is not allocated; *sa***.len** is the number of bytes in the string, if it is allocated; *sa***.a** is the number of bytes allocated for the string, if it is allocated. A **stralloc** variable should be initialized to {0}, meaning unallocated.

stralloc_ready makes sure that *sa* has enough space allocated for *len* characters. It allocates extra space if necessary.

stralloc_readyplus makes sure that *sa* has enough space allocated for *len* characters more than its current length. If *sa* is unallocated, **stralloc_readyplus** is the same as **stralloc_ready**.

stralloc_copy copies *sa2* to *sa*, allocating space if necessary. Here *sa2* is an allocated **stralloc** variable.

stralloc_copys copies a 0-terminated string, *buf*, to *sa*, without the 0.

stralloc_copyb copies *len* characters from *buf* to *sa*.

stralloc_cat appends *sa2* to *sa*, allocating space if necessary. If *sa* is unallocated, **stralloc_cat** is the same as **stralloc_copy**.

stralloc_cats and **stralloc_catb** are analogous to **stralloc_copys** and **stralloc_copyb**.

stralloc_append adds a single character, **buf*, to *sa*, allocating space if necessary.

stralloc_0 adds a single 0 character to *sa*.

stralloc_starts returns 1 if the 0-terminated string *buf*, without the 0, is a prefix of *sa*.

ERROR HANDLING

If a **stralloc** routine runs out of memory, it leaves *sa* alone and returns 0, setting **errno** appropriately. On success it returns 1; this guarantees that *sa* is allocated.

SEE ALSO

alloc(3), error(3)