NAME

stringlist, **sl_init**, **sl_add**, **sl_free**, **sl_find** - stringlist manipulation functions

LIBRARY

Standard C Library (libc, -lc)

SYNOPSIS

#include <stringlist.h>

StringList *
sl_init();

int
sl_add(*StringList *sl*, *char *item*);

void
sl_free(*StringList *sl*, *int freeall*);

char *
sl_find(*StringList *sl*, *const char *item*);

DESCRIPTION

The **stringlist** functions manipulate stringlists, which are lists of strings that extend automatically if necessary.

The *StringList* structure has the following definition:

```
typedef struct _stringlist {
        char    **sl_str;
        size_t  sl_max;
        size_t  sl_cur;
} StringList;
```

> *sl_str* a pointer to the base of the array containing the list.

> *sl_max*
>   the size of *sl_str*.

> *sl_cur*
>   the offset in *sl_str* of the current element.

The following stringlist manipulation functions are available:

sl_init() Create a stringlist. Returns a pointer to a *StringList*, or NULL in case of failure.

sl_free()

> Releases memory occupied by *sl* and the *sl->sl_str* array. If *freeall* is non-zero, then each of the items within *sl->sl_str* is released as well.

sl_add()

> Add *item* to *sl->sl_str* at *sl->sl_cur*, extending the size of *sl->sl_str*. Returns zero upon success, -1 upon failure.

sl_find()

> Find *item* in *sl*, returning NULL if it is not found.

SEE ALSO

free(3), malloc(3)

HISTORY

The **stringlist** functions appeared in FreeBSD 2.2.6 and NetBSD 1.3.