

**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.