

**NAME**

**scandir**, **scandirat**, **scandir\_b**, **alphasort**, **versionsort** - scan a directory

**LIBRARY**

Standard C Library (libc, -lc)

**SYNOPSIS**

```
#include <dirent.h>
```

*int*

```
scandir(const char *dirname, struct dirent ***namelist, int (*select)(const struct dirent *),  
int (*compar)(const struct dirent **, const struct dirent **));
```

*int*

```
scandirat(int dirfd, const char *dirname, struct dirent ***namelist, int (*select)(const struct dirent *),  
int (*compar)(const struct dirent **, const struct dirent **));
```

*int*

```
scandir_b(const char *dirname, struct dirent ***namelist, int (^select)(const struct dirent *),  
int (^compar)(const struct dirent **, const struct dirent **));
```

*int*

```
alphasort(const struct dirent **d1, const struct dirent **d2);
```

*int*

```
versionsort(const struct dirent **d1, const struct dirent **d2);
```

**DESCRIPTION**

The **scandir**() function reads the directory *dirname* and builds an array of pointers to directory entries using `malloc(3)`. It returns the number of entries in the array. A pointer to the array of directory entries is stored in the location referenced by *namelist*.

The *select* argument is a pointer to a user supplied subroutine which is called by **scandir**() to select which entries are to be included in the array. The select routine is passed a pointer to a directory entry and should return a non-zero value if the directory entry is to be included in the array. If *select* is null, then all the directory entries will be included.

The *compar* argument is a pointer to a user supplied subroutine which is passed to `qsort(3)` to sort the completed array. If this pointer is null, the array is not sorted.

The **alphasort()** function is a routine which can be used for the *compar* argument to sort the array alphabetically using `strcoll(3)`.

The **versionsort()** function is a routine which can be used for the *compar* argument to sort the array naturally using `strverscmp(3)`.

The memory allocated for the array can be deallocated with `free(3)`, by freeing each pointer in the array and then the array itself.

The **scandirat()** function is similar to **scandir()**, but takes an additional *dirfd* argument. If the supplied *dirname* is absolute, the function's behavior is identical to that of **scandir()**, the *dirfd* argument is unused. If *dirname* is relative, *dirfd* must be a valid file descriptor referencing a directory, in which case the *dirname* lookup is performed relative to the directory referenced by *dirfd*. If *dirfd* has the special value `AT_FDCWD`, then the current process directory is used as the base for relative lookups. See `openat(2)` for additional details.

The **scandir\_b()** function behaves in the same way as **scandir()**, but takes blocks as arguments instead of function pointers and calls **qsort\_b()** rather than **qsort()**.

## DIAGNOSTICS

Returns -1 if the directory cannot be opened for reading or if `malloc(3)` cannot allocate enough memory to hold all the data structures.

## SEE ALSO

`openat(2)`, `directory(3)`, `malloc(3)`, `qsort(3)`, `strcoll(3)`, `strverscmp(3)`, `dir(5)`

## STANDARDS

The **versionsort()** function is a GNU extension and conforms to no standard.

## HISTORY

The **scandir()** and **alphasort()** functions appeared in 4.2BSD. The **scandirat()** and **versionsort()** functions were added in FreeBSD 13.2.