alphasort(3p) — Linux manual page

PROLOG | NAME | SYNOPSIS | DESCRIPTION | RETURN VALUE | ERRORS | EXAMPLES | APPLICATION USAGE | RATIONALE | FUTURE DIRECTIONS | SEE ALSO | COPYRIGHT

PROLOG top

This manual page is part of the POSIX Programmer's Manual. The Linux implementation of this interface may differ (consult the corresponding Linux manual page for details of Linux behavior), or the interface may not be implemented on Linux.

NAME top

alphasort, scandir — scan a directory

SYNOPSIS top

```
#include <dirent.h>

int alphasort(const struct dirent **d1, const struct dirent **d2);
int scandir(const char *dir, struct dirent ***namelist,
    int (*sel)(const struct dirent *),
    int (*compar)(const struct dirent **, const struct dirent **));
```

DESCRIPTION top

The *alphasort*() function can be used as the comparison function for the *scandir*() function to sort the directory entries, *d1* and *d2*, into alphabetical order. Sorting happens as if by calling the *strcoll*() function on the *d_name* element of the **dirent** structures passed as the two parameters. If the *strcoll*() function fails, the return value of *alphasort*() is unspecified.

The *alphasort*() function shall not change the setting of *errno* if successful. Since no return value is reserved to indicate an error, an application wishing to check for error situations should set *errno* to 0, then call *alphasort*(), then check *errno*.

The *scandir*() function shall scan the directory *dir*, calling the function referenced by *sel* on each directory entry. Entries for which the function referenced by *sel* returns non-zero shall be stored in strings allocated as if by a call to *malloc*(), and sorted as if by a call to *qsort*() with the comparison function *compar*, except that *compar* need not provide total ordering. The strings are collected in array *namelist* which shall be allocated as if by a call to *malloc*(). If *sel* is a null pointer, all entries shall be selected. If the comparison function *compar* does not provide total ordering, the order in which the directory entries are stored is unspecified.

RETURN VALUE top

Upon successful completion, the *alphasort*() function shall return an integer greater than, equal to, or less than 0, according to whether the name of the directory entry pointed to by *d1* is lexically greater than, equal to, or less than the directory pointed to by *d2* when both are interpreted as appropriate to the current locale. There is no return value reserved to indicate an

error.

Upon successful completion, the *scandir*() function shall return the number of entries in the array and a pointer to the array through the parameter *namelist*. Otherwise, the *scandir*() function shall return -1.

ERRORS top

The *scandir*() function shall fail if:

EACCES Search permission is denied for the component of the path prefix of *dir* or read permission is denied for *dir*.

ELOOP A loop exists in symbolic links encountered during resolution of the *dir* argument.

ENAMETOOLONG
The length of a component of a pathname is longer than {NAME_MAX}.

ENOENT A component of *dir* does not name an existing directory or *dir* is an empty string.

ENOMEM Insufficient storage space is available.

ENOTDIR
A component of *dir* names an existing file that is neither a directory nor a symbolic link to a directory.

EOVERFLOW
One of the values to be returned or passed to a callback function cannot be represented correctly.

The *scandir*() function may fail if:

ELOOP More than {SYMLOOP_MAX} symbolic links were encountered during resolution of the *dir* argument.

EMFILE All file descriptors available to the process are currently open.

ENAMETOOLONG
The length of a pathname exceeds {PATH_MAX}, or pathname resolution of a symbolic link produced an intermediate result with a length that exceeds {PATH_MAX}.

ENFILE Too many files are currently open in the system.

The following sections are informative.

EXAMPLES top

An example to print the files in the current directory:

```
    #include <dirent.h>
    #include <stdio.h>
    #include <stdlib.h>
    ...
    struct dirent **namelist;
    int i,n;

        n = scandir(".", &namelist, 0, alphasort);
        if (n < 0)
            perror("scandir");
        else {
            for (i = 0; i < n; i++) {
                printf("%s\n", namelist[i]->d_name);
                free(namelist[i]);
                }
```

```
                }
            free(namelist);
        ...
```

APPLICATION USAGE top

If *dir* contains filenames that do not form character strings, or which contain characters outside the domain of the collating sequence of the current locale, the *alphasort*() function need not provide a total ordering. This condition is not possible if all filenames within the directory consist only of characters from the portable filename character set.

The *scandir*() function may allocate dynamic storage during its operation. If *scandir*() is forcibly terminated, such as by *longjmp*() or *siglongjmp*() being executed by the function pointed to by *sel* or *compar*, or by an interrupt routine, *scandir*() does not have a chance to free that storage, so it remains permanently allocated. A safe way to handle interrupts is to store the fact that an interrupt has occurred, then wait until *scandir*() returns to act on the interrupt.

For functions that allocate memory as if by *malloc*(), the application should release such memory when it is no longer required by a call to *free*(). For *scandir*(), this is *namelist* (including all of the individual strings in *namelist*).

RATIONALE top

None.

FUTURE DIRECTIONS top

None.

SEE ALSO top

qsort(3p), strcoll(3p)

The Base Definitions volume of POSIX.1-2017, dirent.h(0p)

COPYRIGHT top

Pages that refer to this page: dirent.h(0p), qsort(3p), scandir(3p), strcoll(3p)

HTML rendering created 2021-08-27 by Michael Kerrisk, author of *The Linux Programming Interface*, maintainer of the Linux *man-pages* project.