

alphasort(3p) — Linux manual page

[ПРОЛОГ](#) | [Имя](#) | [СИНОПСИС](#) | [Описание](#) | [ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ](#) | [ОШИБКИ](#) | [ПРИМЕРЫ](#) |
[ИСПОЛЬЗОВАНИЕ ПРИЛОЖЕНИЯ](#) | [ОБОСНОВАНИЕ](#) | [БУДУЩИЕ НАПРАВЛЕНИЯ](#) | [СМ . ТАКЖЕ](#) | [Авторские права](#)

ALPHASORT(3P) Руководство программиста POSIX ALPHASORT (3P)

PROLOG [top](#)

Эта страница руководства является частью Руководства программиста POSIX. Реализация этого интерфейса в Linux может отличаться (обратитесь к соответствующей странице руководства Linux для получения подробной информации о поведении Linux), или интерфейс может быть не реализован в Linux.

НАЗВАНИЕ [top](#)

alphasort, scandir – сканирование каталога

СИНОПСИС [top](#)

#включить <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 **));
```

ОПИСАНИЕ [вверху](#)

Функция `alphasort()` может использоваться в качестве функции сравнения для функции `scandir()` для сортировки записей каталога `d1` и `d2` в алфавитном порядке. Сортировка происходит так, как если бы вызывалась функция `strcoll()` для элемента `d_name` структур `dirent`, передаваемых в качестве двух параметров. Если функция `strcoll()` терпит неудачу, возвращаемое значение `alphasort()` не определено.

Функция `alphasort()` не должна изменять настройку `errno` в случае успеха. Поскольку возвращаемое значение не зарезервировано для указания ошибки, приложение, желающее проверить наличие ошибок, должно установить `errno` равным 0, затем вызвать `alphasort()`, а затем проверить `errno`.

Функция `scandir()` сканирует каталог `dir`, вызывая функцию, на которую ссылается `sel` в каждой записи каталога. Записи, для которых функция, на которую ссылается `sel`, возвращает ненулевые значения, должны храниться в строках, выделенных как вызов `malloc()`, и сортироваться как вызов `qsort()` с помощью функции сравнения.

`compar`, за исключением того, что `compar` не нужно предоставлять общий заказ. Строки собираются в массив `namelist`, который должен быть выделен как бы вызовом `malloc()`. Если `sel` является нулевым указателем, все записи должны быть выбраны. Если функция сравнения `compar` не обеспечивает полный порядок, порядок, в котором хранятся записи каталога, не указан.

ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ [top](#)

После успешного завершения функция `alphasort()` возвращает целое число, большее, равное или меньшее 0, в зависимости от того, является ли имя записи каталога, на которую указывает `d1`

, лексически больше, равно или меньше каталога, на который указывает *d2*, когда оба они интерпретируются соответствующим образом текущей локали. Возвращаемое значение не зарезервировано для указания на ошибку.

После успешного завершения *скандир* функция *()* возвращает количество записей в массиве и указатель на массив через *список имен параметров*. В противном случае функция *scandir()* вернет значение *-1*.

ОШИБКИ [top](#)

Функция *scandir()* завершится ошибкой, если:

EACCES отказано в разрешении поиска для компонента префикса пути *dir* или отказано в разрешении чтения для *dir*.

ELOOP Цикл существует в символических ссылках, встречающихся при разрешении аргумента *dir*.

ENAMETOOLONG

Длина компонента пути больше, чем {NAME_MAX}.

ENOENT Компонент *dir* не называет существующий каталог или *dir* является пустой строкой.

ENOMEM Недостаточно места для хранения.

ENOTDIR

Компонент *dir* называет существующий файл, который не является ни каталогом, ни символической ссылкой на каталог.

E_OVERFLOW

Одно из значений, возвращаемых или передаваемых

функции обратного вызова, не может быть представлено правильно.

Функция *scandir()* может выйти из строя, если:

При разрешении аргумента `dir` было обнаружено более `{SYMLoop_MAX}` символических ссылок.

EMFILE Все файловые дескрипторы, доступные процессу, в данный момент открыты.

ENAMETOOLONG

Длина пути превышает `{PATH_MAX}`, или разрешение пути символической ссылки дает промежуточный результат с длиной, превышающей `{PATH_MAX}`.

ENFILE В настоящее время в системе открыто слишком много файлов.

Следующие разделы носят информативный характер.

ПРИМЕРЫ [сверху](#)

Пример печати файлов в текущем каталоге:

```
#включить <dirent.h>
#включить <stdio.h>
#включить <stdlib.h>
...
struct dirent ** список имен;
int i,n;

n = scandir(".", &namelist, 0, alphasort);
if (n < 0)
perror("скандир");
else {
for (i = 0; i < n; i++) {
```

```
printf("%s\n", namelist[i]->d_name);  
бесплатно(namelist[i]);  
}  
}  
бесплатно (список имен);  
...
```

ИСПОЛЬЗОВАНИЕ ПРИЛОЖЕНИЯ [top](#)

Если *dir* содержит имена файлов, которые не образуют символьные строки, или которые содержат символы вне домена последовательности сортировки текущей локали, функция *alphasort()* не должна предоставлять полный порядок. Это условие невозможно, если все имена файлов в каталоге состоят только из символов из переносимого набора символов *filename*.

Функция *scandir()* может выделять динамическое хранилище во время своей работы. Если *scandir()* принудительно завершается, например, *longjmp()* или *siglongjmp()* выполняется функцией, на которую указывает *sel* или *compar*, или подпрограммой прерывания, *scandir()* не имеет возможности освободить это хранилище, поэтому оно остается постоянно выделенным. Безопасный способ обработки прерываний – сохранить факт возникновения прерывания, а затем подождать, пока *scandir()* не вернется, чтобы воздействовать на прерывание.

Для функций, выделяющих память как бы с помощью *malloc()*, приложение должно освободить такую память, когда ее больше нет. требуется вызов *бесплатно()*. Для *scandir()* это *namelist* (включая все отдельные строки в *списке* имен).

ОБОСНОВАНИЕ [сверху](#)

Нет.

FUTURE DIRECTIONS [top](#)

Нет.

СМ . ТАКЖЕ [top](#)

[qsort\(3p\)](#), [strcoll\(3p\)](#)

Объем базовых определений POSIX.1–2017, [dirent.h\(0p\)](#)

COPYRIGHT [top](#)

Части этого текста перепечатаны и воспроизведены в электронном виде по стандарту IEEE Std 1003.1–2017, Стандарт для информации
Технология -- Portable Operating System Interface (POSIX), The
Open Group Base Specifications Issue 7, 2018 Edition, Copyright
© 2018 г. Институтом электротехники и электроники

Engineers, Inc и Open Group. В случае любого
расхождения между этой версией и исходным стандартом IEEE и
Open Group исходный стандарт IEEE и Open Group

Стандарт – это документ рефери. Оригинальный стандарт можно
получить онлайн по адресу <http://www.opengroup.org/unix/online.html> .

Любые типографские ошибки или ошибки форматирования, которые появляются на этой странице
, скорее всего, были допущены во время преобразования
исходных файлов в формат man-страницы. Чтобы сообщить о таких ошибках, см.
https://www.kernel.org/doc/man-pages/reporting_bugs.html .

IEEE/ The Open Group 2017 ALPHASORT (3P)

Страницы, которые ссылаются на эту страницу: [dirent.h\(0p\)](#), [qsort\(3p\)](#), [scandir\(3p\)](#),
[strcoll\(3p\)](#)

HTML-рендеринг создан 2021-08-27 Майклом Керриском, автором *интерфейса программирования Linux*, сопровождающим проекта *Linux man-pages*.

Для получения подробной информации о углубленных **курсах обучения системному программированию Linux / UNIX**, которые я преподаю, посмотрите [здесь](#).

Хостинг от [jambit GmbH](#).

