

**NAME**

**backtrace** - fill in the backtrace of the currently executing thread

**LIBRARY**

Backtrace Information Library (libexecinfo, -lexecinfo)

**SYNOPSIS**

```
#include <execinfo.h>
```

*size\_t*

```
backtrace(void **addrlist, size_t len);
```

*char \*\**

```
backtrace_symbols(void * const *addrlist, size_t len);
```

*int*

```
backtrace_symbols_fd(void * const *addrlist, size_t len, int fd);
```

*char \*\**

```
backtrace_symbols_fmt(void * const *addrlist, size_t len, const char *fmt);
```

*int*

```
backtrace_symbols_fd_fmt(void * const *addrlist, size_t len, int fd, const char *fmt);
```

**DESCRIPTION**

The **backtrace()** function places into the array pointed by *addrlist* the array of the values of the program counter for each frame called up to *len* frames. The number of frames found (which can be fewer than *len*) is returned.

The **backtrace\_symbols\_fmt()** function takes an array of previously filled addresses from **backtrace()** in *addrlist* of *len* elements, and uses *fmt* to format them. The formatting characters available are:

a The numeric address of each element as would be printed using %p.

n

The name of the nearest function symbol (smaller than the address element) as determined by **dladdr(3)** if the symbol was dynamic, or looked up in the executable if static and the `/proc` filesystem is available to determine the executable path.

d

The difference of the symbol address and the address element printed using `0x%tx`.

#### D

The difference of the symbol address and the address element printed using `+0x%tx` if non-zero, or nothing if zero.

f The filename of the symbol as determined by `dladdr(3)`.

The array of formatted strings is returned as a contiguous memory address which can be freed by a single `free(3)`.

The `backtrace_symbols()` function is equivalent of calling `backtrace_symbols_fmt()` with a format argument of `%a <%n%D> at %f`

The `backtrace_symbols_fd()` and `backtrace_symbols_fd_fmt()` are similar to the non `_fd` named functions, only instead of returning an array or strings, they print a new-line separated array of strings in `fd`, and return 0 on success and -1 on failure.

## RETURN VALUES

The `backtrace()` function returns the number of elements that were filled in the backtrace. The `backtrace_symbols()` and `backtrace_symbols_fmt()` return a string array on success, and NULL on failure, setting `errno`. Diagnostic output may also be produced by the ELF symbol lookup functions.

## SEE ALSO

`dladdr(3)`, `elf(3)`

## HISTORY

The `backtrace()` library of functions first appeared in NetBSD 7.0 and FreeBSD 10.0.

## BUGS

1. Errors should not be printed but communicated to the caller differently.
2. Because these functions use `elf(3)` this is a separate library instead of being part of `libc/libutil` so that no library dependencies are introduced.
3. The Linux versions of the functions (there are no `_fmt` variants) use `int` instead of `size_t` arguments.