

**NAME**

getunwind – copy the unwind data to caller's buffer

**SYNOPSIS**

```
#include <syscall.h>
```

```
#include <linux/unwind.h>
```

```
long getunwind(void *buf, size_t buf_size);
```

*Note:* There is no glibc wrapper for this system call; see NOTES.

**DESCRIPTION**

*Note: this function is obsolete.*

The IA-64-specific **getunwind()** system call copies the kernel's call frame unwind data into the buffer pointed to by *buf* and returns the size of the unwind data; this data describes the gate page (kernel code that is mapped into user space).

The size of the buffer *buf* is specified in *buf\_size*. The data is copied only if *buf\_size* is greater than or equal to the size of the unwind data and *buf* is not NULL; otherwise, no data is copied, and the call succeeds, returning the size that would be needed to store the unwind data.

The first part of the unwind data contains an unwind table. The rest contains the associated unwind information, in no particular order. The unwind table contains entries of the following form:

|            |                                       |
|------------|---------------------------------------|
| u64 start; | (64-bit address of start of function) |
| u64 end;   | (64-bit address of end of function)   |
| u64 info;  | (BUF-relative offset to unwind info)  |

An entry whose *start* value is zero indicates the end of the table. For more information about the format, see the *IA-64 Software Conventions and Runtime Architecture* manual.

**RETURN VALUE**

On success, **getunwind()** returns the size of the unwind data. On error, -1 is returned and *errno* is set to indicate the error.

**ERRORS**

**getunwind()** fails with the error **EFAULT** if the unwind info can't be stored in the space specified by *buf*.

**VERSIONS**

This system call is available since Linux 2.4.

**CONFORMING TO**

This system call is Linux-specific, and is available only on the IA-64 architecture.

**NOTES**

This system call has been deprecated. The modern way to obtain the kernel's unwind data is via the **vdso(7)**.

Glibc does not provide a wrapper for this system call; in the unlikely event that you want to call it, use **syscall(2)**.

**SEE ALSO**

**getauxval(3)**

**COLOPHON**

This page is part of release 5.02 of the Linux *man-pages* project. A description of the project, information about reporting bugs, and the latest version of this page, can be found at <https://www.kernel.org/doc/man-pages/>.