

**NAME**

`unw_step` -- advance to next stack frame

**SYNOPSIS**

```
#include <libunwind.h>
```

```
int unw_step(unw_cursor_t *cp);
```

**DESCRIPTION**

The `unw_step()` routine advances the unwind cursor `cp` to the next older, less deeply nested stack frame.

**RETURN VALUE**

On successful completion, `unw_step()` returns a positive value if the updated cursor refers to a valid stack frame, or 0 if the previous stack frame was the last frame in the chain. On error, the negative value of one of the error-codes below is returned.

**THREAD AND SIGNAL SAFETY**

`unw_step()` is thread-safe. If cursor `cp` is in the local address-space, this routine is also safe to use from a signal handler.

**ERRORS**

`UNW_EUNSPEC`

An unspecified error occurred.

`UNW_ENOINFO`

Libunwind was unable to locate the unwind-info needed to complete the operation.

`UNW_EBADVERSION`

The unwind-info needed to complete the operation has a version or a format that is not understood by libunwind.

`UNW_EINVALIDIP`

The instruction-pointer (“program-counter”) of the next stack frame is invalid (e.g., not properly aligned).

`UNW_EBADFRAME`

The next stack frame is invalid.

`UNW_ESTOPUNWIND`

Returned if a call to `find_proc_info()` returned `-UNW_ESTOPUNWIND`.

In addition, `unw_step()` may return any error returned by the `find_proc_info()`, `get_dyn_info_list_addr()`, `access_mem()`, `access_reg()`, or `access_fpreg()` call-backs (see `unw_create_addr_space(3)`).

#### **SEE ALSO**

`libunwind(3)`, `unw_create_addr_space(3)`

#### **AUTHOR**

David Mosberger-Tang

Email: [dmosberger@gmail.com](mailto:dmosberger@gmail.com)

WWW: <http://www.nongnu.org/libunwind/>.