

**NAME**

`unw_set_reg` -- set register contents

**SYNOPSIS**

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

```
int unw_set_reg(unw_cursor_t *cp, unw_regnum_t reg, unw_word_t val);
```

**DESCRIPTION**

The `unw_set_reg()` routine sets the value of register `reg` in the stack frame identified by cursor `cp` to the value passed in `val`.

The register numbering is target dependent and described in separate manual pages (e.g., `libunwind-ia64(3libunwind)` for the IA-64 target). Furthermore, the exact set of accessible registers may depend on the type of frame that `cp` is referring to. For ordinary stack frames, it is normally possible to access only the preserved (“callee-saved”) registers and frame-related registers (such as the stack pointer). However, for signal frames (see `unw_is_signal_frame(3libunwind)`), it is usually possible to access all registers.

Note that `unw_set_reg()` can only write the contents of registers whose values fit in a single word. See `unw_set_fpreg(3libunwind)` for a way to write registers which do not fit this constraint.

**RETURN VALUE**

On successful completion, `unw_set_reg()` returns 0. Otherwise the negative value of one of the error codes below is returned.

**THREAD AND SIGNAL SAFETY**

`unw_set_reg()` is thread-safe as well as safe to use from a signal handler.

**ERRORS**

`UNW_EUNSPEC`

An unspecified error occurred.

`UNW_EBADREG`

An attempt was made to write a register that is either invalid or not accessible in the current frame.

`UNW_EREADONLY`

An attempt was made to write to a read-only register.

In addition, `unw_set_reg()` may return any error returned by the `access_mem()`, `access_reg()`, and

access\_fpreg() callbacks (see unw\_create\_addr\_space(3libunwind)).

**SEE ALSO**

libunwind(3libunwind), libunwind-ia64(3libunwind), unw\_get\_reg(3libunwind),  
unw\_is\_signal\_frame(3libunwind), unw\_set\_fpreg(3libunwind)

**AUTHOR**

David Mosberger-Tang

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

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