

**NAME**

rdma\_reg\_write - register data buffer(s) for remote RDMA write access.

**SYNOPSIS**

```
#include <rdma/rdma_verbs.h>
```

```
struct ibv_mr * rdma_reg_write (struct rdma_cm_id *id, void *addr, size_t length);
```

**ARGUMENTS**

id            A reference to a communication identifier where the message buffer(s) will be used.

addr         The address of the memory buffer(s) to register.

length       The total length of the memory to register.

**DESCRIPTION**

Registers a memory buffer that will be accessed by a remote RDMA write operation. Memory buffers registered using `rdma_reg_write` may be targeted in an RDMA write request, allowing the buffer to be specified on the remote side of an RDMA connection as the `remote_addr` of `rdma_post_write`, or similar call.

**RETURN VALUE**

Returns a reference to the registered memory region on success, or `NULL` on error. If an error occurs, `errno` will be set to indicate the failure reason.

**NOTES**

`rdma_reg_write` is used to register a data buffer that will be the target of an RDMA write operation on a queue pair associated with an `rdma_cm_id`. The memory buffer is registered with the protection domain associated with the identifier. The start of the data buffer is specified through the `addr` parameter, and the total size of the buffer is given by `length`.

All data buffers should be registered before being posted as a work request. Users must deregister all registered memory by calling `rdma_dereg_mr`.

**SEE ALSO**

`rdma_cm(7)`, `rdma_create_id(3)`, `rdma_create_ep(3)`, `rdma_reg_msgs(3)`, `rdma_reg_read(3)`, `ibv_reg_mr(3)`, `ibv_dereg_mr(3)`, `rdma_post_write(3)`