NAME

ibv_fork_init - initialize libibverbs to support fork()

SYNOPSIS

#include <infiniband/verbs.h>

int ibv_fork_init(void);

DESCRIPTION

ibv_fork_init() initializes libibverbs's data structures to handle **fork()** function calls correctly and avoid data corruption, whether **fork()** is called explicitly or implicitly (such as in **system()**).

It is not necessary to use this function if all parent process threads are always blocked until all child processes end or change address spaces via an **exec()** operation.

RETURN VALUE

ibv_fork_init() returns 0 on success, or the value of errno on failure (which indicates the failure reason).

NOTES

ibv_fork_init() works on Linux kernels supporting the **MADV_DONTFORK** flag for **madvise()** (2.6.17 and higher).

Setting the environment variable **RDMAV_FORK_SAFE** or **IBV_FORK_SAFE** has the same effect as calling **ibv_fork_init()**.

Setting the environment variable **RDMAV_HUGEPAGES_SAFE** tells the library to check the underlying page size used by the kernel for memory regions. This is required if an application uses huge pages either directly or indirectly via a library such as libhugetlbfs.

Calling **ibv_fork_init()** will reduce performance due to an extra system call for every memory registration, and the additional memory allocated to track memory regions. The precise performance impact depends on the workload and usually will not be significant.

Setting **RDMAV_HUGEPAGES_SAFE** adds further overhead to all memory registrations.

SEE ALSO

fork(2), **wait**(2), **system**(3), **exec**(3), **ibv_get_device_list**(3)

AUTHORS

Dotan Barak <dotanba@gmail.com>