NAME

BUS_BIND_INTR, **bus_bind_intr** - bind an interrupt resource to a specific CPU

SYNOPSIS

#include <sys/param.h>
#include <sys/bus.h>

int
BUS_BIND_INTR(*device_t dev, device_t child, struct resource *irq, int cpu*);

int
bus_bind_intr(*device_t dev, struct resource *irq, int cpu*);

DESCRIPTION

The **BUS_BIND_INTR**() method allows an interrupt resource to be pinned to a specific CPU. The interrupt resource must have an interrupt handler attached via BUS_SETUP_INTR(9). The *cpu* parameter corresponds to the ID of a valid CPU in the system. Binding an interrupt restricts the cpuset(2) of any associated interrupt threads to only include the specified CPU. It may also direct the low-level interrupt handling of the interrupt to the specified CPU as well, but this behavior is platform-dependent. If the value NOCPU is used for *cpu*, then the interrupt will be "unbound" which restores any associated interrupt threads back to the default cpuset.

Non-sleepable locks such as mutexes should not be held across calls to these functions.

The **bus_bind_intr**() function is a simple wrapper around **BUS_BIND_INTR**().

Note that currently there is no attempt made to arbitrate between multiple bind requests for the same interrupt from either the same device or multiple devices. There is also no arbitration between interrupt binding requests submitted by userland via cpuset(2) and **BUS_BIND_INTR**(). The most recent binding request is the one that will be in effect.

RETURN VALUES

Zero is returned on success, otherwise an appropriate error is returned.

SEE ALSO

cpuset(2), BUS_SETUP_INTR(9), device(9)

HISTORY

The **BUS_BIND_INTR**() method and **bus_bind_intr**() functions first appeared in FreeBSD 7.2.