NAME

device - an abstract representation of a device

SYNOPSIS

*typedef struct _device *device_t*;

DESCRIPTION

The device object represents a piece of hardware attached to the system such as an expansion card, the bus which that card is plugged into, disk drives attached to the expansion card etc. The system defines one device, *root_bus* and all other devices are created dynamically during autoconfiguration. Normally devices representing top-level buses in the system (ISA, PCI etc.) will be attached directly to *root_bus* and other devices will be added as children of their relevant bus.

The devices in a system form a tree. All devices except *root_bus* have a parent (see device_get_parent(9)). In addition, any device can have children attached to it (see device_add_child(9), device_add_child_ordered(9), device_find_child(9), device_get_children(9), and device_delete_child(9)).

A device which has been successfully probed and attached to the system will also have a driver (see device_get_driver(9) and driver(9)) and a devclass (see device_get_devclass(9) and devclass(9)). Various other attributes of the device include a unit number (see device_get_unit(9)), verbose description (normally supplied by the driver, see device_set_desc(9) and device_get_desc(9)), a set of bus-specific variables (see device_get_ivars(9)) and a set of driver-specific variables (see device_get_softc(9)).

Devices can be in one of several states:

DS_NOTPRESENT the device has not been probed for existence or the probe failed

DS_ALIVE the device probe succeeded but not yet attached

DS_ATTACHED the device has been successfully attached

DS_BUSY the device is currently open

The current state of the device can be determined by calling device_get_state(9).

SEE ALSO

devclass(9), driver(9)

AUTHORS

This manual page was written by Doug Rabson.