

NAME

dev_refthread, **devvn_refthread**, **dev_relthread** - safely access device methods

SYNOPSIS

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

```
#include <sys/conf.h>
```

```
struct cdevsw *
```

```
dev_refthread(struct cdev *dev, int *ref);
```

```
struct cdevsw *
```

```
devvn_refthread(struct vnode *vp, struct cdev **devp, int *ref);
```

```
void
```

```
dev_relthread(struct cdev *dev, int ref);
```

DESCRIPTION

The **dev_refthread()** (or **devvn_refthread()**) and **dev_relthread()** routines provide a safe way to access devfs(5) devices that may be concurrently destroyed by **destroy_dev()** (e.g., removable media).

If successful, **dev_refthread()** and **devvn_refthread()** acquire a "thread reference" to the associated *struct cdev* and return a non-NULL pointer to the cdev's *struct cdevsw* method table. For the duration of that reference, the cdev's associated private data and method table object are valid. Destruction of the cdev sleeps until the thread reference is released.

A reference cannot prevent media removal. It is an implementation detail of individual drivers how method calls from callers with **dev_refthread()** references are handled when the device is pending destruction. A common behavior for disk devices is to return the ENXIO status, but that is not required by this KPI.

The **devvn_refthread()** variant of **dev_refthread()** extracts the *struct cdev* pointer out of the VCHR vnode(9) automatically before performing the same actions as **dev_refthread()**. Additionally, a pointer to the *struct cdev* is returned to the caller via **devp*. **devvn_refthread()** correctly handles possible parallel reclamation of the vnode.

dev_relthread() is used to release a reference to a *struct cdev*. **dev_relthread()** **must** only be invoked when the associated invocation of **dev_refthread()** or **devvn_refthread()** returned a non-NULL *struct cdevsw* *.

CONTEXT

struct cdev objects have two reference counts, *si_refcount* and *si_threadcount*. The **dev_refthread()**, **devvn_refthread()**, and **dev_relthread()** functions manipulate the *si_threadcount*. The *si_threadcount* reference guarantees the liveness of the *struct cdev* object. The other *si_refcount* reference provides only the weaker guarantee that the memory backing the *struct cdev* has not been freed.

RETURN VALUES

If **dev_refthread()** or **devvn_refthread()** are unsuccessful, they return NULL. *If these routines are unsuccessful, they do not increment the struct cdev si_threadcount and do not initialize the value pointed to by the *ref parameter in any way.*

SEE ALSO

devfs(5), destroy_dev(9)

CAVEATS

Do not invoke **dev_relthread()** unless the matching refthread routine succeeded!