The Linux Kernel API

The Linux Kernel API

Table of Contents

Chapter 1. Driver Basics

1.1. Driver Entry and Exit points

module_init

Name

`module_init` — driver initialization entry point

Synopsis

```
module_init  ( x);
```

Arguments

x

function to be run at kernel boot time or module insertion

Description

`module_init` will add the driver initialization routine in the "__initcall.int" code segment if the driver is checked as "y" or static, or else it will wrap the driver initialization routine with `init_module` which is used by insmod and modprobe when the driver is used as a module.

module_exit

Name

`module_exit` — driver exit entry point

Synopsis

```
module_exit  ( x);
```

Arguments

x

function to be run when driver is removed

Description

`module_exit` will wrap the driver clean-up code with `cleanup_module` when used with rmmod when the driver is a module. If the driver is statically compiled into the kernel, `module_exit` has no effect.

1.2. Atomic and pointer manipulation

atomic_read

Name

`atomic_read` — read atomic variable

Synopsis

```
atomic_read  ( v);
```

Arguments

v

pointer of type atomic_t

Description

Atomically reads the value of *v*. Note that the guaranteed useful range of an atomic_t is only 24 bits.

atomic_set

Name

`atomic_set` — set atomic variable

Synopsis

```
atomic_set  ( v);  i);
```

Arguments

v

pointer of type atomic_t

i

required value

Description

Atomically sets the value of *v* to *i*. Note that the guaranteed useful range of an atomic_t is only 24 bits.

atomic_add

Name

`atomic_add` — add integer to atomic variable

Synopsis

```
void atomic_add  (int i); atomic_t * v);
```

Arguments

i

integer value to add

v

pointer of type atomic_t

Description

Atomically adds *i* to *v*. Note that the guaranteed useful range of an atomic_t is only 24 bits.

atomic_sub

Name

`atomic_sub` — subtract the atomic variable

Synopsis

```
void atomic_sub  (int i); atomic_t * v);
```

Arguments

i