

Linux Driver Model

“web woven by a spider on drugs”

Greg Kroah-Hartman
gregkh@linuxfoundation.org

github.com/gregkh/presentation-driver-model

You don't need to know this.

struct kref

- Reference counting
- No locks
- Release function required

struct kobjects

- Base object type
- sysfs representation
- Data structure glue
- Hotplug event handling

struct attribute

- sysfs files for kobjects
- 1 text value per file
- Binary files possible
- Never manage individually

struct kset

- Groups kobjects together

kobj_type

- release()
- sysfs functions for kobject
- Namespace handling

struct device

- Universal structure
- Belongs to a bus or “class”

struct device_type

- Same bus, different type

struct device_driver

- controls a device
- probe/remove
- shutdown/suspend/resume
- Default attributes

struct bus_type

- binds devices and drivers
- matching
- uevents
- shutdown

bus responsibilities

- register bus
- create devices
- register drivers
- suspend/resume

Create a device

- set the bus type
- set parent pointer
- set attribute groups
- `device_initialize()`
- ..other stuff..
- `device_add()`

Register a driver

- set the bus type
- set up probe/release
- set module ownership
- `driver_register()`

struct class

- user visable devices
- yes, it's a bus...
- suspend/resume
- release
- default attributes

class responsibilities

- class_create/class_register
- reserve major/minor
- use in device_create
- device_destroy
- free major/minor
- suspend/resume if wanted

shutdown

```
if device→class
    if class→shutdown
        device→class→shutdown(device)
```

```
if device→bus
    if bus→shutdown
        device→bus→shutdown(device)
    shutdown must call
        driver→shutdown(device)
```

Driver writer hints

- attribute groups only
- never call `sysfs_*`()

Class writer hints

- attribute groups only
- never call `sysfs_*`()

Bus writer hints

- My sincere appologies

“raw” sysfs/kobjects?

- Do not do it
- Really, no.
- Use a class or a bus
- Please no.
- Read the documentation
- Read it again
- Expect tough review cycle



github.com/gregkh/presentation-driver-model

Linux Driver Model

“web woven by a spider on drugs”

Greg Kroah-Hartman
gregkh@linuxfoundation.org

github.com/gregkh/presentation-driver-model



Subtitle is LWN quote

In 2.4 all subsystems were isolated and did things their own way. After adding hotplug support to the second subsystem, I figured this needed to change

Pat Mochel wanted to get suspend/resume to work, I wanted persistent device naming. By the time 2.6 came out, we had naming solved, still working on suspend/resume...

You don't need to know this.

No driver write should have to mess with the driver core at all.

Well, minor things, see my last slide.

It's a complex mess, is a whole chapter in LDD3, and there are parts of the kernel code that I keep forgetting and having to relearn at times.

Messy stuff.

struct kref

- Reference counting
- No locks
- Release function required

Don't ever do your own reference counting

Kref is “proven” correct

Use it

You need a lock outside of the object, it must be there

If no release function, why are you doing reference counting?

struct kobjects

- Base object type
- sysfs representation
- Data structure glue
- Hotplug event handling

Handles all of the basic housekeeping for kernel objects

Handles all of the links to other kobjects, the hierarchy and other good stuff. Solid functions that are known good.

Created out of the development process by Al Viro during the driver core development and was originally used in char devices

NEVER touch the kobject in a char device structure, it doesn't do what you think it does.

struct attribute

- sysfs files for kobjects
- 1 text value per file
- Binary files possible
- Never manage individually

You will be yelled at if you don't follow these rules.

Bad examples are a histogram graph plot by a cpufreq driver

Hopefully all fixed.

Always document them in Documentation/ABI

struct kset

- Groups kobjects together

A “set” of kobjects that belong to the same type of “subsystem”. Don’t have to be the same type of object.

How you group a kobject together

kobj_type

- `release()`
- sysfs functions for kobject
- Namespace handling

The functions to call for your kobject

A kobject always has a `kset` and a `kobj_type`
Both are needed.

Namespace stuff is for only networking
kobjects, messy stuff.

struct device

- Universal structure
- Belongs to a bus or “class”

Basic features of all types of devices in the kernel

Name, platform data, driver data, dma information, MSI information, CMA, firmware info, iommu, and so on

Always belongs to something, never have a “blank” device.

NEVER ABUSE PLATFORM DEVICES!

struct device_type

- Same bus, different type

Devices that are of the same bus, but do different things.

USB example – device, interface, endpoint, port

Everything is handled by the same bus, but sometimes you need to do minor different things based on the type.

struct device_driver

- controls a device
- probe/remove
- shutdown/suspend/resume
- Default attributes

Note

struct bus_type

- binds devices and drivers
- matching
- uevents
- shutdown

Note

bus responsibilities

- register bus
- create devices
- register drivers
- suspend/resume

Note

Create a device

- set the bus type
- set parent pointer
- set attribute groups
- `device_initialize()`
- ..other stuff..
- `device_add()`

Note

Register a driver

- set the bus type
- set up probe/release
- set module ownership
- `driver_register()`

Note

struct class

- user visible devices
- yes, it's a bus...
- suspend/resume
- release
- default attributes

Note

class responsibilities

- `class_create/class_register`
- `reserve major/minor`
- `use in device_create`
- `device_destroy`
- `free major/minor`
- `suspend/resume if wanted`

Note

shutdown

```
if device→class
    if class→shutdown
        device→class→shutdown(device)

if device→bus
    if bus→shutdown
        device→bus→shutdown(device)
    shutdown must call
        driver→shutdown(device)
```

Note

Driver writer hints

- attribute groups only
- never call `sysfs_*`()

Note

Class writer hints

- attribute groups only
- never call `sysfs_*`()

Note

Bus writer hints

- My sincere appologies

Note

“raw” sysfs/kobjects?

- Do not do it
- Really, no.
- Use a class or a bus
- Please no.
- Read the documentation
- Read it again
- Expect tough review cycle

Note



github.com/gregkh/presentation-driver-model

Obligatory Penguin Picture

