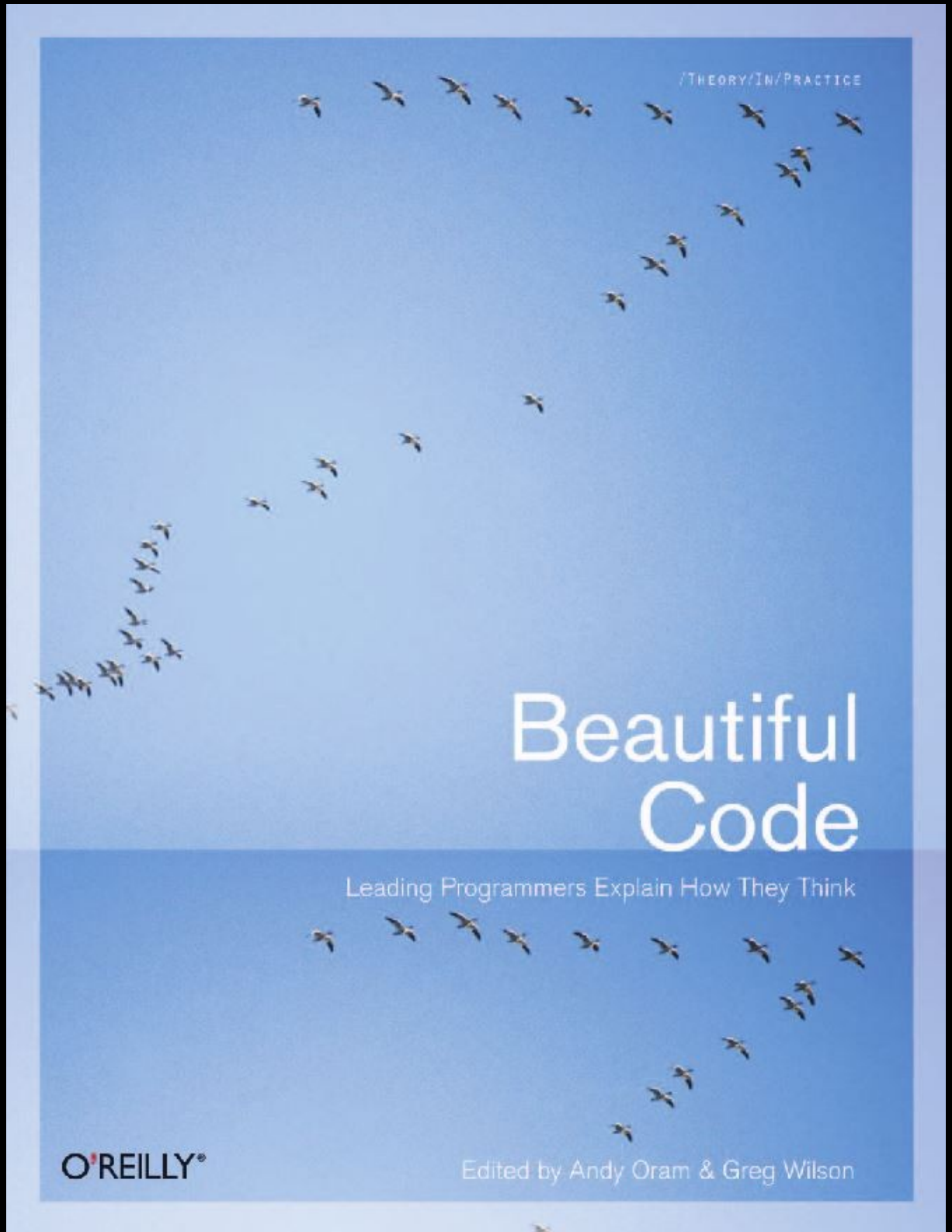
Objects in Linux

Trust us, we know what we are doing...

Greg Kroah-Hartman
gregkh@linuxfoundation.org


/Theory/In/Practice
Beautiful Code
Leading Programmers Explain How They Think
O'REILLY®
Edited by Andy Oram & Greg Wilson

Early 2000's
“Unify all Linux devices”

Pat Mochel – OSDL
Greg K-H - IBM

```
struct device {
        struct list_head node;
        struct list_head children;
        struct device *parent;

        char name[DEVICE_NAME_SIZE];
        bus_id[BUS_ID_SIZE];

        spinlock_t lock;

        atomic_t refcount;

        struct driver_dir_entry *dir;
        struct device_driver *driver;

        void *driver_data;
        void *platform_data;
        u32 current_state;
        unsigned char *saved_state;
  };
```

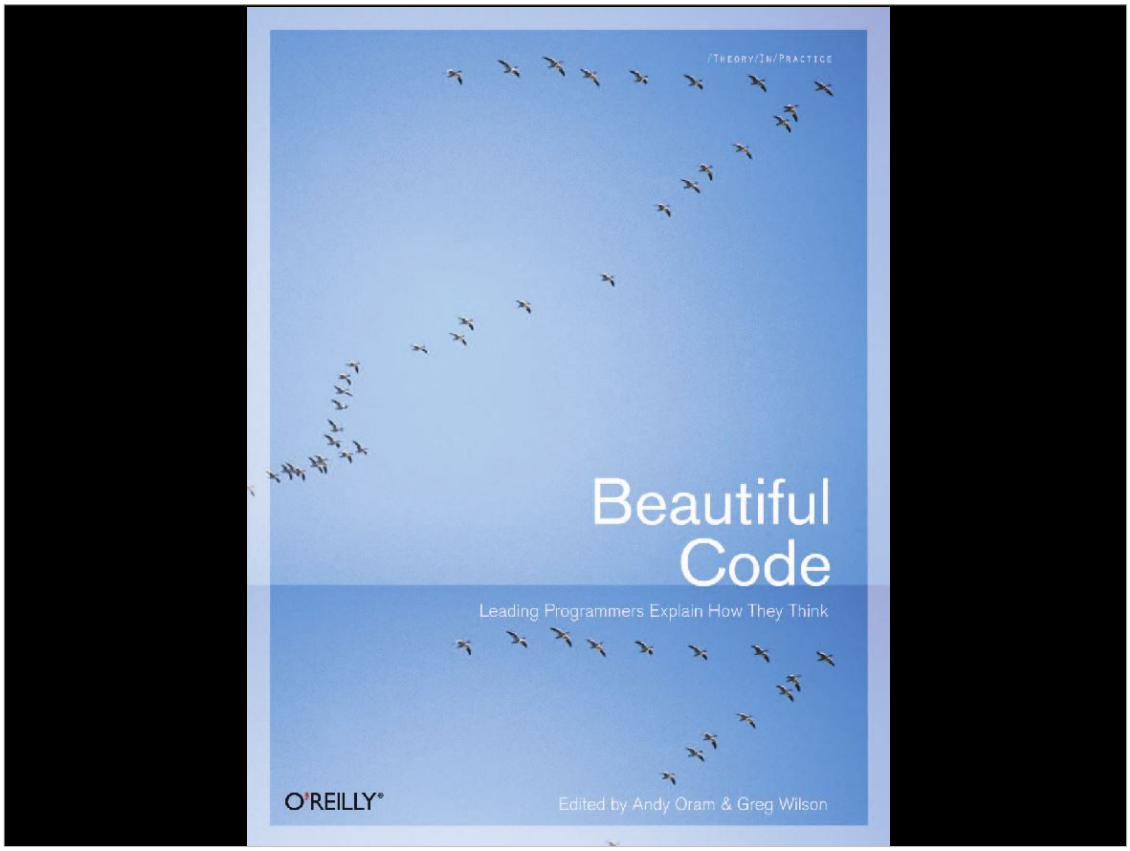
```
struct device {
        struct list_head node;
        struct list_head children;
        struct device *parent;

        char name[DEVICE_NAME_SIZE];
        bus_id[BUS_ID_SIZE];

        spinlock_t lock;

        atomic_t refcount;

        struct driver_dir_entry *dir;
        struct device_driver *driver;

        void *driver_data;
        void *platform_data;
        u32 current_state;
        unsigned char *saved_state;
  };
```

github.com/gregkh/presentation-kref

Objects in Linux

Trust us, we know what we are doing...

Greg Kroah-Hartman
gregkh@linuxfoundation.org



Talk based on a chapter I wrote for “Beautiful Code – Leading Programmers Explain How They Think”

Edited by Andy Oram and Greg Wilson
O'Reilly
2007

Early 2000's
“Unify all Linux devices”

Pat Mochel – OSDL
Greg K-H - IBM

Pat wanted this for power management and suspend/resume

I wanted this for persistant device naming.

All devices and subsystems were islands

Both tasks needed a way to see all devices in the system, suspend/resume wanted to know which device to suspend in which order.

Naming needed a way to assign a character or block device to a specific hardware device

```
struct device {
        struct list_head node;
        struct list_head children;
        struct device *parent;

        char name[DEVICE_NAME_SIZE];
        bus_id[BUS_ID_SIZE];

        spinlock_t lock;

        atomic_t refcount;

        struct driver_dir_entry *dir;
        struct device_driver *driver;

        void *driver_data;
        void *platform_data;
        u32 current_state;
        unsigned char *saved_state;
  };
```

We came up with 'struct device'

All busses in the kernel were changed to create a structure based on this one. It was passed to the new driver core, and the driver core created a hierarchy of everything in the kernel.

This can be seen in sysfs (which used to be called driverfs)

```
struct device {
        struct list_head node;
        struct list_head children;
        struct device *parent;

        char name[DEVICE_NAME_SIZE];
        bus_id[BUS_ID_SIZE];

        spinlock_t lock;

        atomic_t refcount;

        struct driver_dir_entry *dir;
        struct device_driver *driver;

        void *driver_data;
        void *platform_data;
        u32 current_state;
        unsigned char *saved_state;
  };
```

We came up with 'struct device'

All busses in the kernel were changed to create a structure based on this one. It was passed to the new driver core, and the driver core created a hierarchy of everything in the kernel.

This can be seen in sysfs (which used to be called driverfs)

Obligatory Penguin Picture