

Sistemsko programiranje – The Linux Device Model

Jernej Vičič

1.1.2012

Vsebina

- 1 Uvod
- 2 The Linux Device Model
 - Kobjects, ksets, subsystems
 - Vodilo – Bus
 - Naprava – Device
 - Gonilnik – Driver
 - Razred – Class
 - Sestavimo skupaj

Uvod

- unified device model for the kernel,
- 2.6 device model ponuja to abstrakcijo:
- dela, ki jih podpira jedro:
 - Power management and system shutdown
 - Communications with user space
 - Hotpluggable devices
 - Device classes
 - Object lifecycles

The Linux Device Model

- kompleksna podatkovna struktura,
- le del strukture za miško USB,
- slika na naslednji prosojnici:

Del strukture za miško USB

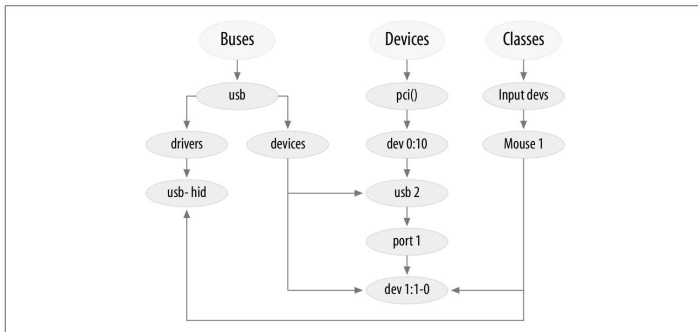


Figure: Del strukture za miško USB.

Opis slike

- devices – naprave,
 - kako je miška povezana v sistem,
- bus – vodilo,
 - kako je miška povezan v posamezno vodilo,
- classes – razredi,
 - kaj ponuja naprava *miška*.

Kako si bomo ogledali model

- bottom-up presentation of the device model,
- od konkretnih konstruktov k abstraktnim,
- kratek pregled,
- poglavje 14 nudi manjkajoče dele.

Kobjects, Ksets, in Subsystems

- Kobjects – osnovne strukture, ki podpirajo model,
- Ksets – ta struktura združuje več Kobjects,
- Subsystems – predstavitev kot celota za višje nivoje jedra.

Kobjects

- osnovne strukture, ki podpirajo model,
- *struct kobject*,
- štetje referenc na predmete – reference counting of objects,
- predstavitev v sysfs – sysfs representation,
- zbirka podatkovnih struktur - Data structure glue,
- sistem za obveščanje o novi strojni opremi - hotplug event handling.

Subsystem

- subsystem – podsistem,
- predstavitev kot celota za višje nivoje jedra,
- ponavadi (ne vedno) je predstavljen na vrhu hierarhije sysfs,
- primeri:
 - *block_subsys* – */sys/block* – bločne naprave,
 - *devices_subsys* – */sys/devices* – osnovna hierarhija naprav.

Subsystem

```
struct subsystem {  
    struct kset kset;  
    struct rw_semaphore rwsem;  
};
```

- predstavljen je s strukturo subsystem,
- je ovojnica okoli kseta,
- dodan je še semafor.

Vodilo – Bus

- *vodilo* – kanal med procesorjem in napravami,
- za predstavitev modela:
- vse naprave so povezane na vodilo,
- lahko je to "interno" vodilo,
- vodila so lahko priključena na druga vodila,
- kontroler USB je ponavadi naprava PCI,
- vodilo je v modelu predstavljeno kot struktura.

Vodilo – Bus

```
struct bus_type {
    char *name;
    struct subsystem subsys;
    struct kset drivers;
    struct kset devices;
    int (*match)(struct device *dev, struct device_driver *drv);
    struct device *(*add)(struct device *parent, char *bus_name,
                          struct device_driver *drv);
    int (*hotplug) (struct device *dev, char **envp,
                   int num_envp, char *buffer, int buffer_size);
    /* ... manjkajoča polja ... */
};
```

Vodilo – Bus

- *name* – ime vodila,
- v `sysfs` je predstavljen v posebnem razdelku *subsystem*,
- dva kseta:
 - poznani gonilniki za to vodilo,
 - naprave, ki so priključene na to vodilo,
- množica metod.

Naprava – Device

```
struct device {
    struct device *parent;
    struct kobject kobj;
    char bus_id[BUS_ID_SIZE];
    struct bus_type *bus;
    struct device_driver *driver;
    void *driver_data;
    void (*release)(struct device *dev);
    /* ... manjkajoča polja ... */
};
```

- predstavitev naprtave na najnižjem nivoju,

Naprava – Device

- *struct device *parent* – naprava je priključena na starša napravo,
 - ponavadi je to vodilo,
- *struct kobject kobj* – kobject, ki predstavlja to napravo,
- *char bus_id[BUS_ID_SIZE]* – niz, ki identificira napravo na vodilu,
- *struct bus_type *bus* – katera vrsta naprave,
- *struct device_driver *driver* – kazalec na strukturo gonilnika,
- *void (*release)(struct device *dev)* – metoda, ki se pokliče, ko se umakne zadnja referenca na napravo.

Gonilnik – Driver

- model vodi listo vseh gonilnikov v sistemu,
- tako lahko za novo napravo izbere pravi gonilnik,
- struktura:

Gonilnik – Driver

```
struct device_driver {
    char *name;
    struct bus_type *bus;
    struct kobject kobj;
    struct list_head devices;
    int (*probe)(struct device *dev);
    int (*remove)(struct device *dev);
    void (*shutdown) (struct device *dev);
    /* ... manjkajoča polja ... */
};
```

Razred – Class

- visokonivojski pogled na naprave,
- abstrahira nizkonivojske podrobnosti,
- primer:
 - gonilnik vidi disk SCSI ali disk ATA
 - class nivo oba diska predstavi kot "disk",
- razredi omogočajo uporabniškemu prostoru delo z napravami glede na njihovo uporabo,
- predstavljeni v sysfs pod `/sys/class`.

Sestavimo skupaj

- kako podsistem PCI komunicira z modelom gonilnikov (driver model),
- osnovni koncepti kako se dodaja/briše gonilnik.

Proces izdelave gonilnika

