

Infraestrutura de hotplug do linux (usb)





Agenda

1. Visão geral
2. Dispositivos suportados por driver
3. Tabela de dispositivos
4. Detecção e propagação de eventos (run time)
5. udevd
6. Detecção e propagação de eventos (boot time)



Visão geral

- Utilizado nos dispositivos *hotpluggable*
- Load/Unload do driver somente quando necessário
- Combinação kernel space + user space



Visão geral

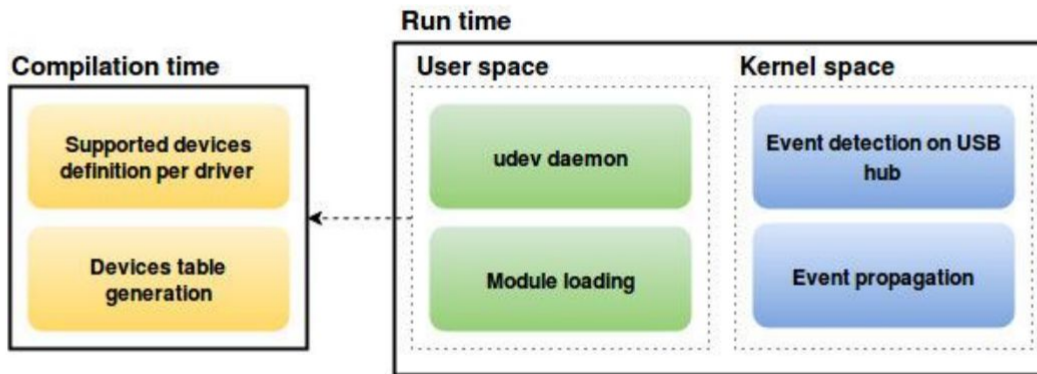


Fig 1. Infraestrutura de hotplug



Dispositivos suportados por driver

- Definidos através da macro `MODULE_DEVICE_TABLE()`

```
static struct usb_device_id vp7045_usb_table [] = {  
    { USB_DEVICE(USB_VID_VISIONPLUS, USB_PID_TWINHAN_VP7045_COLD) },  
    { USB_DEVICE(USB_VID_VISIONPLUS, USB_PID_TWINHAN_VP7045_WARM) },  
    { USB_DEVICE(USB_VID_VISIONPLUS, USB_PID_DNTV_TINYUSB2_COLD) },  
    { USB_DEVICE(USB_VID_VISIONPLUS, USB_PID_DNTV_TINYUSB2_WARM) },  
    { 0 },  
};  
MODULE_DEVICE_TABLE(usb, vp7045_usb_table);
```

Fig 2. Dispositivos suportados pelo driver `dvb_usb_vp7045`



Tabela de dispositivos

- Gerada no build do kernel
- O comando **depmod -a** gera o **module.alias**
- **module.alias**: relação dispositivo vs driver
- O comando **modprobe** utiliza este arquivo na carga dos drivers

```
andre@void:[~]$ cat /lib/modules/$(uname -r)/modules.alias | grep dvb_usb_vp7045
alias usb:v13D3p3224d*dc*dsc*dp*ic*isc*ip*in* dvb_usb_vp7045
alias usb:v13D3p3223d*dc*dsc*dp*ic*isc*ip*in* dvb_usb_vp7045
alias usb:v13D3p3206d*dc*dsc*dp*ic*isc*ip*in* dvb_usb_vp7045
alias usb:v13D3p3205d*dc*dsc*dp*ic*isc*ip*in* dvb_usb_vp7045
```

Fig 3. Dispositivo vs módulo

Detecção e propagação de eventos (run time)

- O driver do hub USB monitora as portas via polling
- Uma alteração no estado de uma porta dispara um evento
- O driver do hub registra os novos dispositivos no kernel
- O kernel notifica o user space via uevents

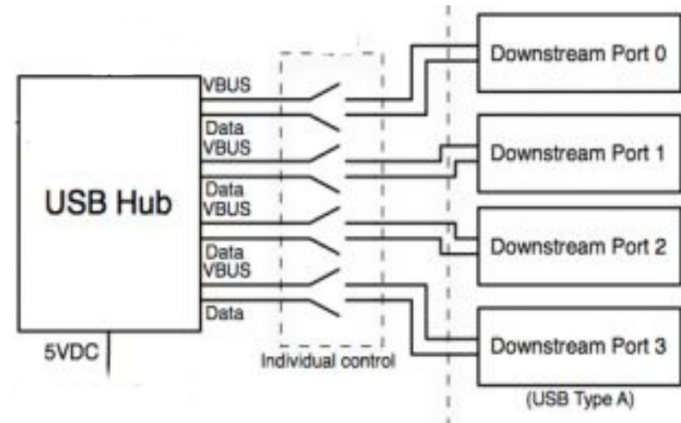


Fig 4. Conexão das portas ao hub usb



Detecção e propagação de eventos (run time)

- Retrieve das propriedades do dispositivo via mensagem de controle **USB_REQ_GET_DESCRIPTOR**
- Propriedades do dispositivo são codificadas na mensagem
- O parâmetro **MODALIAS** é utilizado na carga do módulo

```
if (add_uevent_var(env, "INTERFACE=%d/%d/%d",
    alt->desc.bInterfaceClass,
    alt->desc.bInterfaceSubClass,
    alt->desc.bInterfaceProtocol))
    return -ENOMEM;

if (add_uevent_var(env,
    "MODALIAS=usb:"
    "v%04Xp%04Xd%04Xdc%02Xdsc%02Xdp%02Xic%02Xisc%02Xip%02Xin%02X",
    le16_to_cpu(usb_dev->descriptor.idVendor),
    le16_to_cpu(usb_dev->descriptor.idProduct),
    le16_to_cpu(usb_dev->descriptor.bcdDevice),
    usb_dev->descriptor.bDeviceClass,
    usb_dev->descriptor.bDeviceSubClass,
    usb_dev->descriptor.bDeviceProtocol,
    alt->desc.bInterfaceClass,
    alt->desc.bInterfaceSubClass,
    alt->desc.bInterfaceProtocol,
    alt->desc.bInterfaceNumber))
```

Fig 5. Construção da mensagem



udev

- Notificado pelo kernel sobre eventos de hotplug.
- Gerencia as entradas no /dev.
- Carrega o módulo adequado com base no ID do dispositivo.
- Possui rules para carregamento de módulos.
- Faz o match com a rule adequada.

```
ACTION!="add", GOTO="nm_thunderbolt_end"

# Load the thunderbolt-net driver if we a device of type thunderbolt_xdomain
# is added.
SUBSYSTEM=="thunderbolt", ENV{DEVTYPE}=="thunderbolt_xdomain", RUN{builtin}+="kmod load thunderbolt-net"

# For all thunderbolt network devices, we want to enable link-local configuration
SUBSYSTEM=="net", ENV{ID_NET_DRIVER}=="thunderbolt-net", ENV{NM_AUTO_DEFAULT_LINK_LOCAL_ONLY}="1"

LABEL="nm_thunderbolt_end"
```

Fig 6. Rule para dispositivos thunderbolt_xdomain



Detecção e propagação de eventos (boot time)

- **Problema**

- O udev não pode tratar eventos de hotplug em tempo de boot
- Dispositivos já conectados não terão seus respectivos drivers carregados

- **Solução**

- Retrigger dos eventos de dispositivo emitidos em tempo de boot
- O comando udevadm trigger itera sobre os dispositivos no /sys requisitando tais eventos

```
andre@void:[devices]$ ls /sys/devices/pci0000:00/0000:00:00.0/
ari_enabled          firmware_node        power
broken_parity_status index                remove
class                iommu                rescan
config               iommu_group          resource
consistent_dma_mask_bits irq                  revision
d3cold_allowed       label                subsystem
device               local_cpulist        subsystem_device
dma_mask_bits         local_cpus            subsystem_vendor
driver                modalias              uevent
driver_override       msi_bus              vendor
enable                numa_node
```

Fig 7. O retrigger é feito através do arquivo uevent



Referências

- <https://www.kernel.org/doc/ols/2001/hotplug.pdf>
- <https://www.technovelty.org/linux/what-actually-happens-when-you-plug-in-a-usb-device.html>
- <https://doc.opensuse.org/documentation/leap/reference/html/book.opensuse.reference/cha.udev.html>
- <https://bootlin.com/doc/legacy/udev/udev.pdf>

Dúvidas?

alcm99@gmail.com

github.com/carvalhudo/presentations/tree/hotplug/usb-hotplug