ALSA Jack Software Injection

Simple Introduction On Jack Injection

Here jack injection means users could inject plugin or plugout events to the audio jacks through debugfs interface, it is helpful to validate ALSA userspace changes. For example, we change the audio profile switching code in the pulseaudio, and we want to verify if the change works as expected and if the change introduce the regression, in this case, we could inject plugin or plugout events to an audio jack or to some audio jacks, we don't need to physically access the machine and plug/unplug physical devices to the audio jack.

In this design, an audio jack doesn't equal to a physical audio jack. Sometimes a physical audio jack contains multi functions, and the ALSA driver creates multi `jack_kctl` for a `snd_jack`, here the `snd_jack` represents a physical audio jack and the `jack_kctl` represents a function, for example a physical jack has two functions: headphone and mic_in, the ALSA ASoC driver will build 2 `jack_kctl` for this jack. The jack injection is implemented based on the `jack_kctl` instead of `snd_jack`.

To inject events to audio jacks, we need to enable the jack injection via `sw_inject_enable` first, once it is enabled, this jack will not change the state by hardware events anymore, we could inject plugin or plugout events via `jackin_inject` and check the jack state via `status`, after we finish our test, we need to disable the jack injection via `sw_inject_enable` too, once it is disabled, the jack state will be restored according to the last reported hardware events and will change by future hardware events.

The Layout of Jack Injection Interface

If users enable the SND_JACK_INJECTION_DEBUG in the kernel, the audio jack injection interface will be created as below:

```
$debugfs_mount_dir/sound
|-- card0
|-- |-- HDMI_DP_pcm_10_Jack
|-- |-- |-- jackin_inject
|-- |-- |-- kctl_id
|-- |-- |-- mask_bits
|-- |-- |-- status
|-- |-- |-- sw_inject_enable
|-- |-- |-- type
...
|-- |-- HDMI_DP_pcm_9_Jack
|--     |-- jackin_inject
|--     |-- kctl_id
|--     |-- mask_bits
|--     |-- status
|--     |-- sw_inject_enable
|--     |-- type
|-- card1
    |-- HDMI_DP_pcm_5_Jack
    |-- |-- jackin_inject
    |-- |-- kctl_id
    |-- |-- mask_bits
    |-- |-- status
    |-- |-- sw_inject_enable
    |-- |-- type
    ...
    |-- Headphone_Jack
    |-- |-- jackin_inject
    |-- |-- kctl_id
    |-- |-- mask_bits
    |-- |-- status
    |-- |-- sw_inject_enable
    |-- |-- type
    |-- Headset_Mic_Jack
        |-- jackin_inject
        |-- kctl_id
        |-- mask_bits
        |-- status
        |-- sw_inject_enable
        |-- type
```

The Explanation Of The Nodes

kctl_id

read-only, get jack_kctl->kctl's id

```
sound/card1/Headphone_Jack# cat kctl_id
Headphone Jack
```

mask_bits

read-only, get jack_kctl's supported events mask_bits

```
sound/card1/Headphone_Jack# cat mask_bits
0x0001 HEADPHONE(0x0001)
```

status

read-only, get jack_kctl's current status

- headphone unplugged:

```
sound/card1/Headphone_Jack# cat status
Unplugged
```

- headphone plugged:

```
sound/card1/Headphone_Jack# cat status
Plugged
```

type

read-only, get snd_jack's supported events from type (all supported events on the physical audio jack)

```
sound/card1/Headphone_Jack# cat type
0x7803 HEADPHONE(0x0001) MICROPHONE(0x0002) BTN_3(0x0800) BTN_2(0x1000) BTN_1(0x2000) BTN_0(0x4000
```

sw_inject_enable

read-write, enable or disable injection

- injection disabled:

```
sound/card1/Headphone_Jack# cat sw_inject_enable
Jack: Headphone Jack               Inject Enabled: 0
```

- injection enabled:

```
sound/card1/Headphone_Jack# cat sw_inject_enable
Jack: Headphone Jack               Inject Enabled: 1
```

- to enable jack injection:

```
sound/card1/Headphone_Jack# echo 1 > sw_inject_enable
```

- to disable jack injection:

```
sound/card1/Headphone_Jack# echo 0 > sw_inject_enable
```

jackin_inject

write-only, inject plugin or plugout

- to inject plugin:

```
sound/card1/Headphone_Jack# echo 1 > jackin_inject
```

- to inject plugout:

```
sound/card1/Headphone_Jack# echo 0 > jackin_inject
```