Re: [2.6 patch] schedule obsolete OSS drivers for removal

Source: http://linux.derkeiler.com/Mailing−Lists/Kernel/2006−01/msg02842.html

- *From*: Hannu Savolainen <hannu@xxxxxxxxxxxxx>
- *Date*: Sun, 8 Jan 2006 02:21:36 +0200 (EET)

On Sat, 7 Jan 2006, Takashi Iwai wrote:

> > There are two very opposite approaches to do a sound subsystem. The ALSA
> > way is to expose every single detail of the hardware to the applications
> > and to allow (or force) application developers to deal with them. The OSS
> > approach is to provide maximum device abstraction in the API level (by
> > isolating the apps from the hardware as much as practically possible).
>
> Agreed, it's a good point.
>
> Note that for long time, I've commented that I myself do _not_
> recommend to use ALSA API directly with apps. Rather I've recommended
> to use other portable libraries with ALSA/other backends. Writing an
> app with alsa−lib is just like to write a graphical program with X11
> lowlevel library without any toolkits...

Takashi, I knew you are smart enough to realize this. What is needed at the kernel level is a driver API that is strong enough to provide all the functionality needed to implement good user land libraries (including alsa−lib). At the same time the kernel API itself should be suitable to be used in mainline applications. At this moment Jack is already used by most high end Linux sound apps which is good approach.

> > Both ways have their good and bad sides. During past years the ALSA
> > advocates have been dictating that the ALSA approach is the only available
> > way to go (all resistance is futile).
>
> Heh, it's natural that developers think their own things work better
> :)

This is natural and acceptable. What is not acceptable is that 1000's of existing applications are forced to be converted to use some new API because the original one is seen as deprecated by the developers of the new one. Or to be forced to access the devices through some LD_PRELOAD hack and redundant layers of library code that provide very little if any added value.

> > But after all what is the authority
> > who makes the final decision? Is it the ALSA team (who would like to think
> > in this way)? Or do the Linux/Unix users and audio application developers

> > have any word to say?
>
> I, at least, have never thought that the OSS _API_ would die. Since
> they have existed, they will exist. The question on LKML should be
> rather the implementation (for apps it doesn't matter how the sound
> system is implemented as long as it keeps API).
This is OK as far as long as the approach taken doesn't make it impossible to develop the OSS API as a pure kernel API as it has been designed to be. Development of OSS is still continuing as a stand alone project. In the long term it will be open sourced and possibly given to some Xorg like consortium. How soon (if ever) this is going to happen is mostly a funding issue.

The ALSA kernel API is not documented and it's not intended to be used directly by the apps. If OSS is moved behind some user land wrappers then there is no kernel level API available for (embedded) software developers. Also it's very hard to believe that Linux maintainers love to have a kernel API that is only known to bunch of current ALSA developers and requires use of given library implementation (directly or indirectly).

A challenge will be finding a way how both OSS and ALSA APIs can coexist without disturbing each other. There are many ways (better than artifically forcing OSS API to be emulated in user land). I'm confident that something can be worked out.

> In the implementation of OSS API, there is a clear bottleneck: you
> have to implement everthing in the kernel level because of its
> definition.
I very well know this limitation. I have never claimed that everything can or should be done in the kernel. There is lot of stuff that can be done better in user land library/app code. OSS itself doesn't contain any library code because we think there are other developers that can do libraries better than we.

> Remember that the original thread started from the
> reduction of the kernel codes. Putting more stuff which could be done
> better in user−space is a major drawback, IMO.
Completely agree. There are many things such as effect plugins that cannot be done in kernel space (or technically they can be done but they should not be done).

However there are things like endianess conversions that can be done equally well in kernel space (while in an ideal world they belong to userland). Such simple operations don't take longer than nanoseconds to execute and they make implementing the user land code simplier.

And surprise surprise there are things that can be done much better in kernel space than in userland. If some processing is done in the interrupt handler there are no scheduling latencies involved. Interrupt handlers fire in microseconds so the device can be programmed to interrupt after each sample. This in totally cannibalises the system by raising interrupt

overhead significantly (up to 5−20%). It is known that kernel developers don't like this kind of core porno at all. However if zero latency audio processing is the primary purpose of the system then this is the way to go.

Avoiding kernel code that can be handled in userland is the rule. But as you may see even this rule has some exceptions.

Best regards,

Hannu
−−−−−
Hannu Savolainen (hannu@xxxxxxxxxxxxx)
http://www.opensound.com (Open Sound System (OSS))
http://www.compusonic.fi (Finnish OSS pages)
OH2GLH QTH: Karkkila, Finland LOC: KP20CM
−

- ***References***:
  - ***Re: [2.6 patch] schedule obsolete OSS drivers for removal***
    - *From:* Marcin Dalecki
  - ***Re: [2.6 patch] schedule obsolete OSS drivers for removal***
    - *From:* Jan Engelhardt
  - ***Re: [2.6 patch] schedule obsolete OSS drivers for removal***
    - *From:* Marcin Dalecki
  - ***Re: [2.6 patch] schedule obsolete OSS drivers for removal***
    - *From:* Lee Revell
  - ***Re: [2.6 patch] schedule obsolete OSS drivers for removal***
    - *From:* Marcin Dalecki
  - ***Re: [2.6 patch] schedule obsolete OSS drivers for removal***
    - *From:* Lee Revell
  - ***Re: [2.6 patch] schedule obsolete OSS drivers for removal***
    - *From:* Marcin Dalecki
  - ***Re: [2.6 patch] schedule obsolete OSS drivers for removal***
    - *From:* Lee Revell
  - ***Re: [2.6 patch] schedule obsolete OSS drivers for removal***
    - *From:* Marcin Dalecki
  - ***Re: [2.6 patch] schedule obsolete OSS drivers for removal***
    - *From:* Jesper Juhl
  - ***Re: [2.6 patch] schedule obsolete OSS drivers for removal***
    - *From:* Marcin Dalecki
  - ***Re: [2.6 patch] schedule obsolete OSS drivers for removal***
    - *From:* Hannu Savolainen
  - ***Re: [2.6 patch] schedule obsolete OSS drivers for removal***
    - *From:* Takashi Iwai

- Prev by Date: ***Re: [2.6 patch] no longer mark MTD_OBSOLETE_CHIPS as BROKEN and remove broken MTD_OBSOLETE_CHIPS drivers***
- Next by Date: ***[PATCH −mm] fixup *at syscalls additions (alpha, sparc64)***
- Previous by thread: ***Re: [2.6 patch] schedule obsolete OSS drivers for removal***
- Next by thread: ***Re: [2.6 patch] schedule obsolete OSS drivers for removal***
- Index(es):
    - ***Date***
    - ***Thread***