

# Sound API Ubuntu Linux

Asked 12 years, 8 months ago   Modified 12 years, 8 months ago

Viewed 10k times



8



I want to write some native C/C++ code that takes some uncompressed PCM audio data and plays it out through my speakers on Ubuntu Linux 12.04. Can you tell me what are the different default audio APIs/libraries that are installed? What is their low-level architecture and how do they interact with the kernel? (which syscalls)

(I've heard different terms like ALSA, PulseAudio, OSS, and so on, but I don't really have a clear picture of which does what and how they fit together.)

c++

c

linux

audio

ubuntu

Share

Improve this question

Follow

asked Apr 17, 2012 at 4:13



[Andrew Tomazos](#)

68.4k ● 43 ● 202 ● 343

3 Answers

Sorted by:

Highest score (default)





6



On Linux, you have **LOTS** of Linux audio APIs to choose from. [Here is a graph I made a few years ago demonstrating the relationships among the various Linux audio APIs.](#)

However, if you are only targeting Ubuntu Linux and using C/C++, and you don't need anything too fancy (just taking raw PCM data and shoving it out to the speakers), I would recommend using PulseAudio and sticking to the "simple" API. [A trivial example can be found here](#). It's the path of least resistance to getting a proof of concept working.

Share Improve this answer

answered Apr 17, 2012 at 7:04

Follow



Multimedia Mike

13.2k ● 5 ● 48 ● 62



5



In very simple terms, ALSA provides the low level infrastructure for audio in Linux, and Pulse provides the higher (more desktop-friendly) level. As far as most mainstream Linux distributions are concerned (Ubuntu included), OSS is obsolete.

You ask about syscalls. You don't use audio in Linux via syscalls. If you choose to use ALSA directly (not a good idea for simple audio playback), you'd employ its userspace library libasound2 (even OSS would be used via /dev/ files rather than syscalls).

If you are targeting modern Linux distros only, consider using the Pulse API. If you want more flexibility as to what distro you are targeting and the potential for cross-platform support, you could try a library which abstracts away the OS-specific audio API - for example, PortAudio (<http://www.portaudio.com/>).

If you've never done any audio development, you may find a library such as PortAudio easier to work with than addressing PulseAudio directly.

Share Improve this answer

answered Apr 17, 2012 at 4:38

Follow



[Rich Drummond](#)

3,509 ● 1 ● 16 ● 16



0



[FMOD](#) should be great for you. It's fairly easy to integrate to any C++ project (and many more languages) and runs under Windows, **Linux (32bit and 64bit)**, Mac, Android, PS3, Xbox, etc. Also [FMOD Ex](#) provides both a low-level API and data-driven API.



It is free to use if you are not planning on making money with your application. Otherwise you might have to pay for a license starting at about 100\$.

Share Improve this answer

answered Apr 17, 2012 at 4:30

Follow



[Chris911](#)

4,199 ● 1 ● 25 ● 33