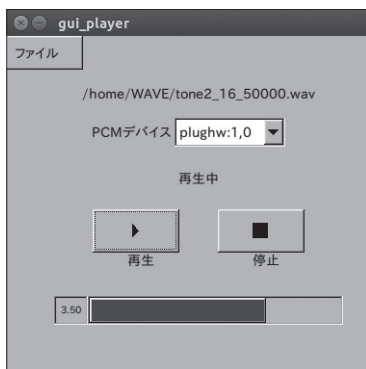


連載



データ入出力の仲介役ALSAライブラリの使い方からWAVE/FLAC/AIFFファイルの再生方法まで 高品位Linuxサウンド・アプリケーション・プログラミング超入門

第2回 サウンド・ファイルが再生されるまで 〈前編：PCMデバイスの設定〉

音羽 良 Ryou Otowa

ALSAライブラリを使ったサウンド再生プログラムの要点は、ALSAライブラリが提供しているAPI関数(以下、ALSA APIと呼ぶ)を適切に使いこなすことです。

まず、全体の処理の流れを解説します。ALSAライブラリはUSB-DACなどのオーディオ・デバイス・ハードウェアを「PCMデバイス」という論理デバイスと見なします。サウンドの再生/録音の前には特定のPCMデバイスに適合するように各種パラメータを設定する必要があります。今回はその設定に使うALSA APIを解説します。 〈編集部〉

PCMサウンド再生処理の流れ

● ALSAライブラリとアプリケーションの役割

ALSAライブラリを使用したPCMサウンド再生処理のフローを図1に示します。

図1で灰色のブロックの処理は、ALSAライブラリの提供するAPI(以降、ALSA APIと呼ぶ)が主体的に行います。一方、白いブロックの処理は、アプリケーションが行うか、もしくはアプリケーションがALSA以外の汎用ユーティリティ・プログラムを使用して行います。各処理の概要を以下に示します。

① サウンド・フォーマット情報の取得処理

WAVE形式、FLAC形式等のサウンド・ファイルのデータを解析し、サウンド・フォーマットに関する各種情報(標準化周波数、チャンネル数、量子化ビット数など)を取得します。

② PCMデバイスのオープン処理

ALSA APIにより、PCMデバイスをオープンします。

③ ハードウェア/ソフトウェア・パラメータの設定処理

ALSA APIにより、サンプル・フォーマットや標準化周波数などPCMデバイスの仕様に適合する

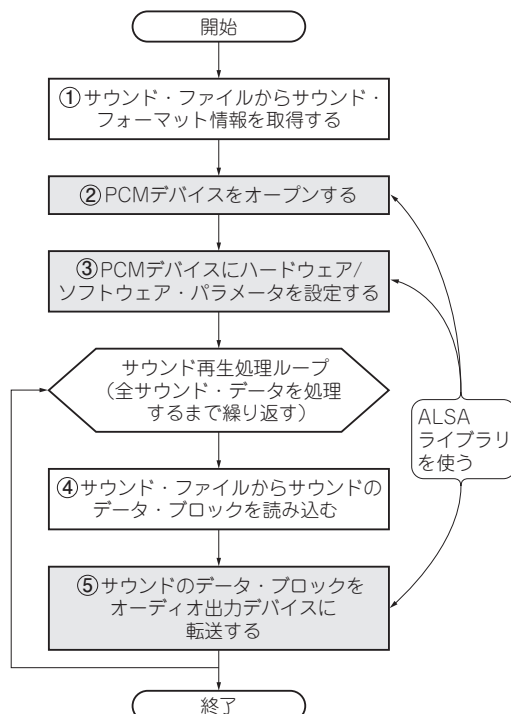


図1 PCMサウンド再生処理フロー

今回は②PCMデバイスのオープンと、③PCMデバイスへのパラメータ設定に使うAPI関数を解説する

ようにパラメータを設定します。

④ サウンド・データ読み込み処理

サウンド・ファイルからブロック単位でデータを読み込みます。このブロック単位のサウンド・データの塊を便宜上、データ・ブロックと呼ぶことにします。

⑤ サウンド・データ出力処理

ALSA APIにより、サウンドのデータ・ブロックを出力デバイスに転送します。

今回は、上記の処理のうち、②と③の手順と、その時に使うALSA APIを解説します。⑤に使うALSA APIは次回解説します。