Linuxサウンドサブシステムのモバイルプラットフォーム向け低消費電力戦略

坂本 貴史

June 26, 2016

今回の話題

- Linux のサウンドサブシステムについて語ります
- 文字ばかりです
- 独自研究に基づいています
- 諸事情 (後述) により、今回の発表内容を英文レポートの形にまとめ、ALSA のコミュニティに公開するつもりです
- 不明点等は気軽に質問してください

自己紹介

- 坂本貴史
- Ubuntu Japanese Team メンバー (2010 年から)
  - 主にフォーラム管理
  - たまに原稿書き
- ALSA のアップストリームのコミッター
  - 一部のデバイスドライバー群のメンテナー
  - カーネル側のコア機能からユーザーランドのライブラリ、ツールまで手広く
- ミラクル・リナックス株式会社所属 (2014 年から)
  - 組み込みソフトウェア部門で働いてる

今回の内容

以下のタイムテーブルを予定しています

1. モバイルプラットフォームの要件 (10 分)
2. サウンドデバイス (5 分)
3. Linux のサウンドサブシステム (10 分)
4. アプリケーション (5 分)
5. プログラミングモデルの流行り廃り (10 分)

モバイルプラットフォームの要件

とにかく消費電力を下げる

- 二次電池の放電時間を可能な限り長くしたい
- 発熱を下げたい

消費電力を下げるには

- CPU をなるべく使わないよう、タスクを処理する

CPU 使用ケース

- プロセスによる計算
- カーネルによる計算
- ハードウェア割り込み
- ソフトウェア割り込み
- メモリーアドレス間コピー

処理

プロセスでの処理

- 普段皆さんが実行しているあれやこれやの処理

カーネルでの処理

- ユーザーランドに対し、ハードウェアを抽象して見せるあれやこれやの処理
  - メモリー管理
  - スケジューラー
  - デバイスドライバー
  - タイマーサービス

割り込みサービスルーチン

ハードウェア割り込みサービスルーチン

- ハードウェアが状態を変更した時に呼ばれるコード
- プロセスやカーネルにとっては、突然起こる事象
- デバイスドライバ開発の文脈では、top half とか呼ばれる

ソフトウェア割り込みサービスルーチン

- 大抵はハードウェア割り込みサービスルーチンからスケジュールされる処理
- デバイスドライバ開発の文脈では、bottom half とか呼ばれる

メモリーアドレス間コピー

- メモリーアクセスは CPU 時間を使う

CPU をなるべく使わない

- 動作周波数を減らす
- プログラムの特性に従い、別々なプロセッサにスケジュールする
  - Big.LITTLE 戦略
- ハードウェア割り込みを減らす
  - Dynamic Tick
- Direct Memory Access(DMA)
  - メモリーアドレス間コピー処理を DMA コントローラーにオフロード
  - 大量のデータを処理する際に有効

サウンドデバイス

音声を扱うためのデバイス

- 通話
- 音楽プレイヤー
- 携帯レコーダー

サウンドデバイスの作り

- アナログ回路
- コーデック
- シリアルデータバス
- コントローラー
- ペリフェラルバス

Intel High Definition Audio(HDA) の場合

- アナログ回路
- HDA コーデック
- HDA シリアルバス
- HDA コントローラー
  - PCH 集積
  - SoC 集積
- PCI-Express バス

HDA じゃない場合 (一例)

- アナログ回路
- Inter IC Sound(I2S) コーデック
- I2S シリアルバス
- I2S コントローラー
  - SoC 集積
- 何かのデータバス

サウンドデバイスを使う方法

- ペリフェラルバス越しにコントローラーを操作する
- ペリフェラルバス越しにデータを送受信する
  - 秒間 44.1k の PCM サンプルフレームとか
  - 44100 * 2 * 2 byte = 176400 bytes / sec

データの送受信

- Direct Media Access を利用すると、CPU を使わずにペリフェラルにデータを送れる
- 典型的な制御例
  - DMA コントローラーに処理を依頼
  - DMA コントローラーは処理を終えるとハードウェア割り込みを上げる
  - ハードウェア割り込みハンドラーから先で、再度処理を依頼
  - 大抵はソフトウェア割り込みハンドラー内

サウンドデバイスドライバーの役割

- ペリフェラルバスドライバーを使い、コントローラーを操作する
- DMA ドライバーフレームワークを使い、データを送受信する
- それらを、ユーザースペースの要求に応じて行う

Linuxのサウンドサブシステム

Advanced Linux Sound Architecture (ALSA)

- ハードウェアベンダからソフトウェアハウスまで、色々な人が開発に参加
  - Cirrus Logic, Realtek
  - TI, Marvel, NXP, Renesas
  - Intel, DELL
  - RedHat, SUSE, Google
  - 市井の普通のひと
- オープンソースソフトウェア
  - カーネルランドとユーザーランドの実装を持つ

サウンドデバイスドライバーフレームワーク

- open/close
  - データ構造を持つメモリを割り当てる、開放する
- hw_param/hw_free
  - ハードウェアのセットアップを行う、止める
- prepare
  - データ送受信の準備を行う
- trigger
  - データ送受信を始める、止める

データ送受信処理

- DMA するためのメモリーページを確保
- 2 つのコンテキストから扱う
- アプリケーションのコンテキスト
  - プロセスの仮想メモリ空間にマップしとく
  - そのページを読み書きする
  - 読み書きしたらそのフレーム数を ioctl(2) でドライバーに通知
- 割り込みコンテキスト
  - ハードウェア割り込みをハンドル
  - 次の転送指示を出す
- データ転送量とアプリケーション読み書きタイミングの同期が必要

タイミング同期方法

- DMA 用メモリーページ内の PCM フレーム数を、「PCM バッファ」というアイディアで管理
- PCM バッファ
  - バッファを複数の断片に分割（period）
  - 1 回の DMA 転送で、1 period 相当の PCM フレームを送る
  - 転送を終えたフレーム位置 (hw_ptr) と、アプリケーションが処理をしたフレーム位置 (sw_ptr) とで管理
  - この 2 つの位置が期待通りでない場合、XRUN という状態になったと判断し、転送を強制終了

アプリケーション

サウンドサーバー

- アプリケーションの音声出力を集約し、サウンドデバイスに送る
- サウンドデバイスからの音声を、各アプリケーションに配送する
- アプリケーションとのデータのやり取りは、共有メモリやプロセス間通信を使用

サウンドサーバーの一例

普通の Linux デスクトップ環境

- PulseAudio
  - I/O ライブラリは alsa-lib
- Jack Audio Connection Kit の jack サーバ
  - 設計が古い（後述）

Chrome OS

- Chromium OS Audio Server (CRAS)
- I/O ライブラリは alsa-lib

Android

- 忘れた
- I/O ライブラリは tinyalsa

プログラミングモデルの流行り廃り

旧来のモデル

- 2005 年あたりまで
- 1 period 相当の PCM フレーム転送終了を待つようなモデル
  - Open Sound System 由来のモデル
- レイテンシー対策
  - 秒間のハードウェア割り込み回数を増やす
  - CPU 時間の無駄遣い

最近のモデル

- 2006 年あたりから
- 省電力の達成
  - ハードウェア割り込み数は増やさない
- レイテンシー対策
  - ハードウェアによってはドライバが、転送済みフレーム数をほぼリアルタイムに返すことができる
  - PCM バッファの rewind を使い、sw_ptr を period 内で hw_ptr ギリギリまで巻き戻すことで達成
- Time scheduling
  - period 無視
  - hw_ptr/sw_ptr 管理によるプロセスのブロックを行わない
  - プロセス側の都合がよいタイミングで読み書きを行う

Rewind の制約

- アプリケーションプロセスとハードウェア割り込みの実行は非同期
- rewind した直後のハードウェア割り込み発生により、hw_ptr が動いて XRUN を起こす可能性
- すぐに XRUN 起こさない程度に rewind するような機構が必要
  - DMA の 1 回の転送単位を把握しておく必要あり

DMA の転送単位

- DMA コントローラーや転送指示を出すドライバーによってまちまち
- 単位となる値は、DMAEngine Slave API が整理して提供
  - drivers/dma 以下を参照
- この情報を共有するための ALSA の kernel/userspace インターフェイスが不足している?

Rewind safeguard

- PulseAudio の ALSA モジュールが内部的に持つ値
- rewind の補正値を、当てずっぽうで指定

このモデルへの対応

- 対応しているドライバがそんなに多くない
  - TI 社の OMAP series のサウンドインターフェイス用ドライバ
  - TI 社の AMxxxx series のサウンドインターフェイス用ドライバ
  - Marvel 社の PXA series のサウンドインターフェイス用ドライバ
  - Marvel 社の Kirkwood series のサウンドインターフェイス用ドライバ
  - Intel 社の HDA 用ドライバ
  - C-Media 社の CMI878x (Oxygen HD Audio) 用ドライバ

このモデルの難点

- デバイスドライバの改良が必要
- パケット志向なドライバに対しては手法が一般化されていない
  - USB Audio Device Class ドライバ (USB)
  - IEC 61883-1/6 ドライバ (IEEE 1394 バス、TSN)
    - 私の悩みどころ