

●Linux オーディオのメリット

Windows でも Mac OS X でもなく、Linux でオーディオを扱うメリットは何でしょうか。筆者は仕事の必要性から Windows, Mac, Linux の 3 種類の OS を使い分けています。それぞれの OS 向けに再生アプリケーションも作って無料で配布しています。この中で、筆者がハイレゾオーディオを再生するために使用しているのは主に Linux です。Linux には以下に示すメリットがあります。

- Windows の標準ドライバでは再生できない USB オーディオクラス 2 が接続できること
- 低スペックの PC でも動作すること
- 無料であること
- サウンドドライバのソースコードが入手できるため、不具合が見つかってでもメーカーの対応を待たずに自分で修正できること

それでは、実際に Linux でオーディオの再生を実行してみます。

●今回の目標（達成不可能 ミッション・インポッシブル？）

今回は、編集部から次に示す具体的な目標を一つ与えられました。

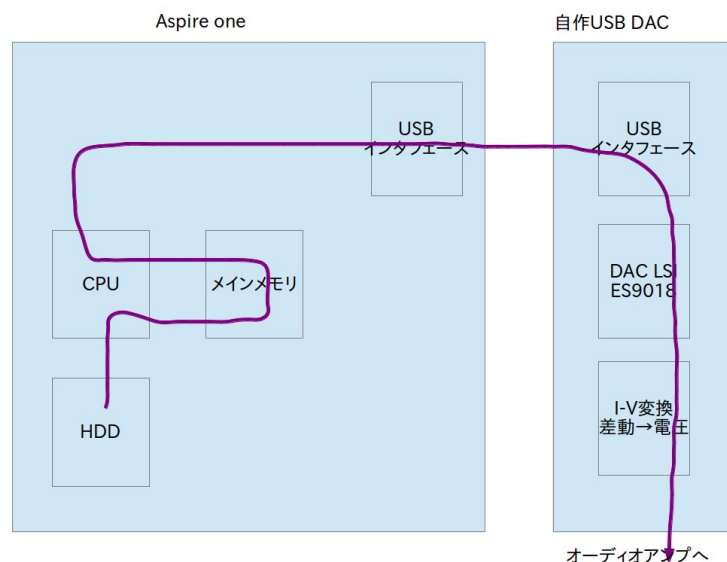
「デスクトップの PC と Linux を使用して、USB オーディオの転送速度の上限に挑戦せよ」

ところが結論を言うと達成不可能な目標でした。というのも、デスクトップのメインマシンを持ち出すまでもなく、atom N450 プロセッサを搭載した旧式ノートパソコンで、自宅にある最高転送速度（PCM 24bit 384kHz 2ch）の USB DAC を再生するのに成功してしまったのです。

Linux の性能限界以前に、USB DAC が性能上限を迎えてしまったわけで、あらためて Linux の I/O 性能の高さを確認することができました。

■実験環境の説明

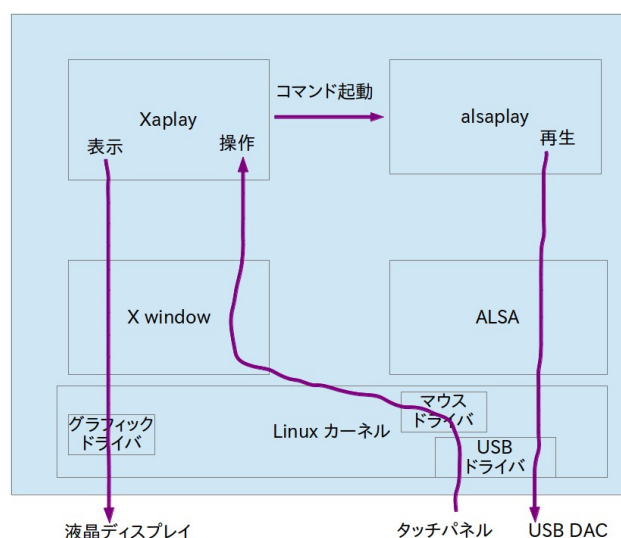
実験には、図 1 に示す以下の機材と図 2 のソフトウェアを用意しました。



<図 1>

パソコン	acer 社 ASPIRE one CPU atom N450 クロック 1.66GHz メモリ 2GB に増量
USB DAC	自作の USB DAC DAC LSI は ES9018 USB インタフェースは自作 プロトコルは USB オーディオクラス 2 準拠 24bit 伝送で 384kHz までの PCM を再生可能

表 1 今回使用したハードウェア

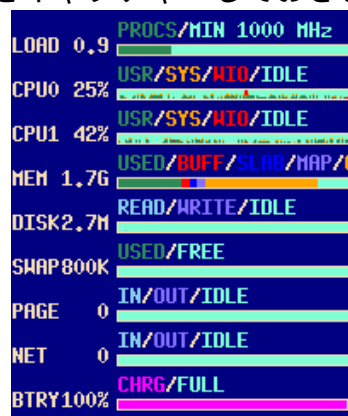


<図 2>

OS	Ubuntu 14.04 32bit
アプリケーション	自作の再生アプリケーション Xaplay + alsaplay(udaplay)

表 2 今回使用したソフトウェア

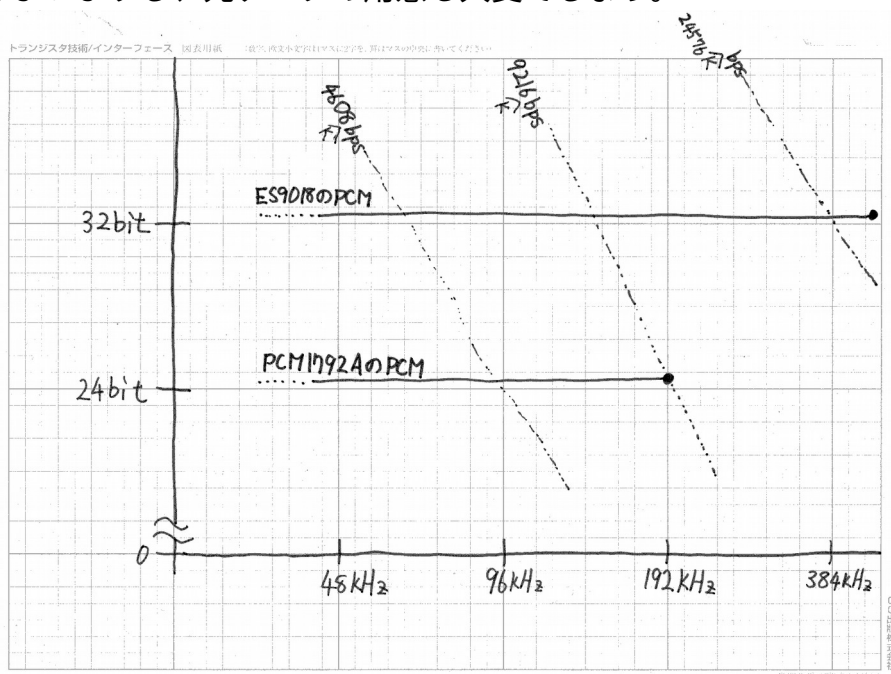
再生時には xosview という負荷監視ツールでシステムの負荷を監視しました。図 3 に再生しているタイミングでの負荷をキャプチャーしておきます。



<図 3>

■ハイレゾサウンド転送レートの現状上限

ハイレゾサウンドのデジタルデータをアナログに変換するのは、DAC LSI です。現状のハイレゾデータ上限を決めているのは、DAC LSI が受け付けられるデータと言ってもよいでしょう(図4)。よく使われる BurrBrown の PCM179X シリーズは、上限が 192kHz です。その上まで対応している LSI は滅多にありません。今回使用した ES9018 は、384kHz に対応している珍しい方です。さらに上の 768kHz を再生できる LSI はまだ作られていないようです。仮に 768kHz の再生が可能だとしても、データ量が多くて HDD がすぐに満タンになってしまいますし、元データの用意も大変でしょう。

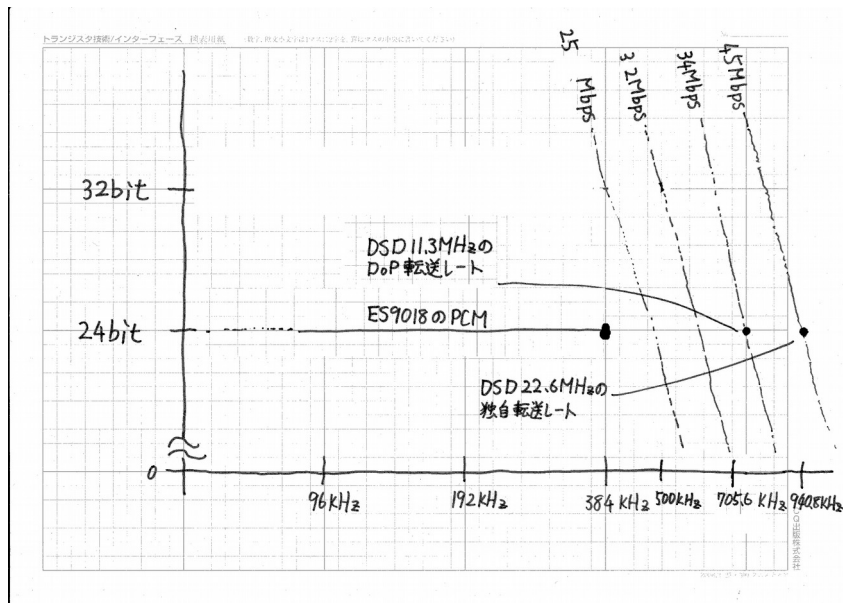


<図4>

現在のパソコン性能や、ネット販売のハイレゾサウンドを購入する時のインターネット転送速度を考えると、24bit192kHz というのは、現実的な妥協点なのかもしれません。

■ウラワザでもっと高速の転送を試してみる

筆者の自作 USB DAC は PCM 上限が 24bit384kHz2ch です。ところが、PCM ではなくて DSD だともっと高レート(図5参照)の転送ができるのです。もちろんこちらも atom のノートパソコンで試してみました。

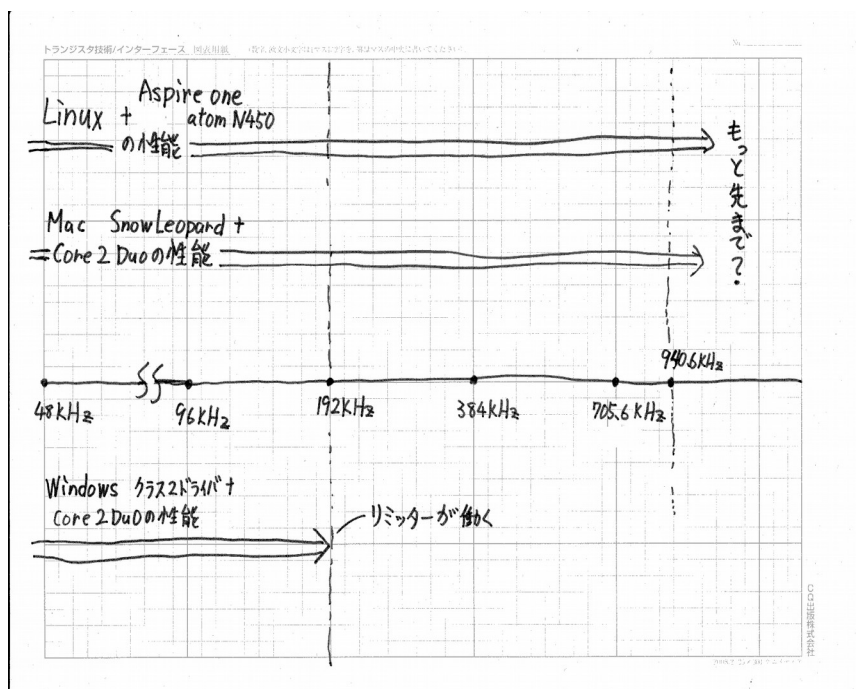


<図 5>

11.2896MHz2ch の DoP (PCM 換算で 24bit705.6kHz 相当)を転送できました。さらにもう一つ上があって、独自方式で DSD22.5792MHz2ch も転送できます。こちらは独自方式なので PCM に換算すると 24bit940.8kHz 相当の転送レートになります。ビットレートに直すとステレオ 2 チャンネルなので、45Mbps になります。地デジの動画 (最大 24Mbps) より高レートですね。

■Mac OS X と比較してみる

USB オーディオクラス 2 を OS 標準添付のデバイスドライバで再生するとすると、Linux の他に Mac OS X も忘れてはいけません。こちらも筆者手持ちのマシンで実験してみました。結果を図 6 に示します。OS バージョン 10.6.8 の Snow Leopard を搭載した Mac Book で、Linux と同様の再生が可能でした。CPU は、2.1GHz の Core 2 Duo です。



<図 6>

ところが、思わぬところでトラブルが発生しました。Snow Leopard のサポート期間が Windows XP のように終わっていたのです。Windows XP のサポート終了は大騒ぎになりましたが、Snow Leopard の方は、ひっそりアップデートが止まっています。しかも最新 OS Mavericks に入れ替えようとしたら、マシンが古くて性能が低いから OS 入れ替えが不可能と表示されてしまいます。

片や Linux では atom N450 でオーディオ再生ができていたのに、片や Mac OS X では Core 2 Duo マシンのサポートが終わってしまっています。

■Windows と比較してみる

それでは、Windows で再生してみたらどうでしょうか。自作 USB DAC のデバイス ID を某社のオーディオと同じにして、オーディオクラス 2 の転送を試みました。ちなみに、メーカー製デバイスドライバはライセンス条項があって、本来ならば自作オーディオのドライバとして使うことができません。たまたま筆者は過去に某社と商談したことがあって、デバイスドライバの使用許可をもらってあるのです。

結果は、Core 2 Duo のマシンを使用して 192kHz 以下の再生は正常にできましたが、それ以上の再生ができませんでした。リミッターがデバイスドライバにあるのか、Windows 側にあるのかは未確認です。

■ハイレゾ再生時のタブー

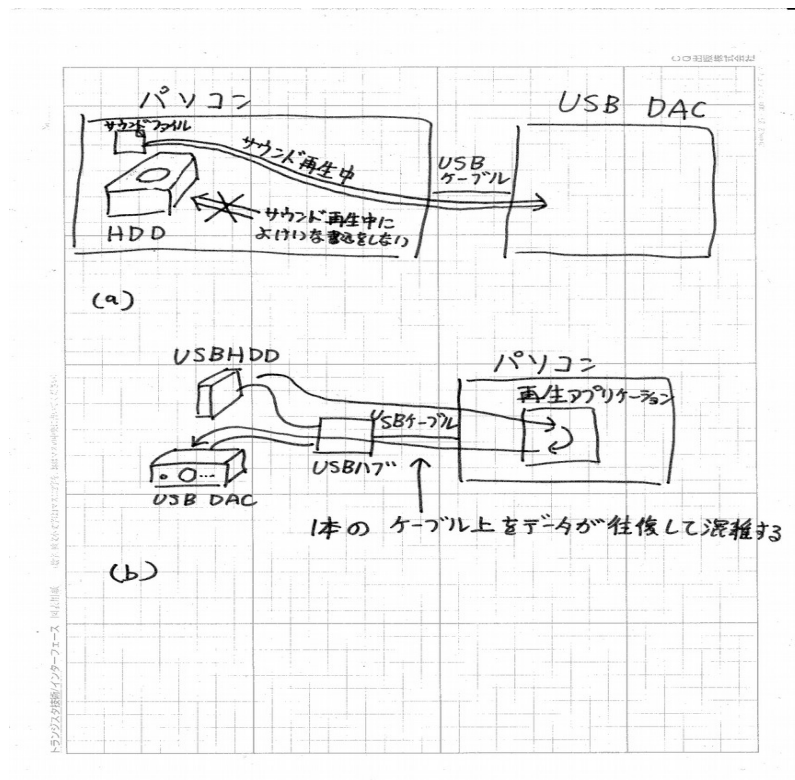
ハイレゾ再生時には、高速でデータが流れます。HDD から連続して大量のデータを読み出しますし、USB にも大量のデータが流れます。そこで安定したサウンド再生を実行するために、どの OS を使っていても避けなければならないことがあります。

タブー 1 他のアプリケーションで HDD に同時に大量アクセスすることは禁止

図 7(a)のように他のアプリケーションが HDD を読み書きすると、サウンド再生のための読み出しが邪魔されます。特に音楽再生時に OS のアップデートをかけると、音が途切れて当たり前です。

タブー 2 USB ハブにオーディオと HDD を同時に接続することは禁止

サウンドデータを USB HDD に保存している人もいるでしょう。この USB HDD と再生用の DAC を同じ USB ハブに接続してはいけません。ハブと PC の間のケーブルは共通なので、図 7(b)に示すように HDD の読み出しデータと、DAC への書き込みデータが共存して大混雑となります。見掛け上ハブがないからと言って安心してはいけません。ノートパソコンでは、USB インタフェース LSI を節約するために、1 個のインタフェースと PC 内部に隠れた USB ハブでポート数を増やしているケースがあります。サウンドデータはできるだけ内蔵 HDD に保存しましょう。



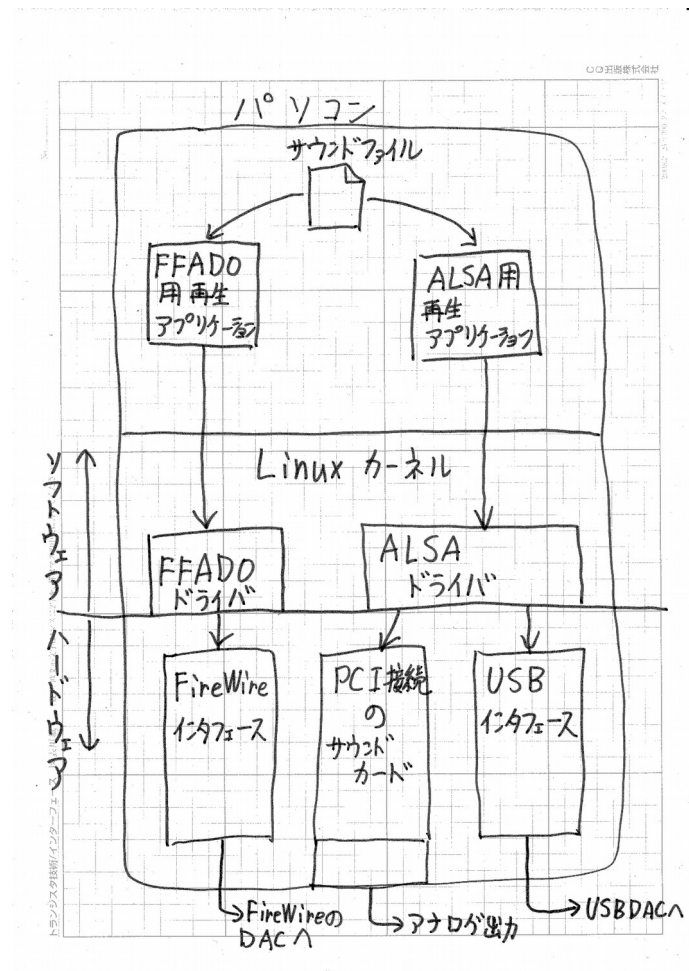
<図 7>

●Linux でオーディオ再生の基礎

Linux でオーディオ再生するために必要となる技術の紹介をしておきます。詳細は、後の章で述べられるため、簡単な説明にとどめます。

■デバイスの接続バスとドライバ

PC につながるオーディオデバイスは、現在は USB 接続が主流です。その前は、図 8 に示すように PCI バスにつながるものや、IEEE1394 (FireWire) バスも使用されていました。FireWire バスのサウンドデバイスは FFADO というドライバを使用します。その他ほとんどの PCI バスデバイスと USB 接続のサウンドドライバは ALSA を使用します。



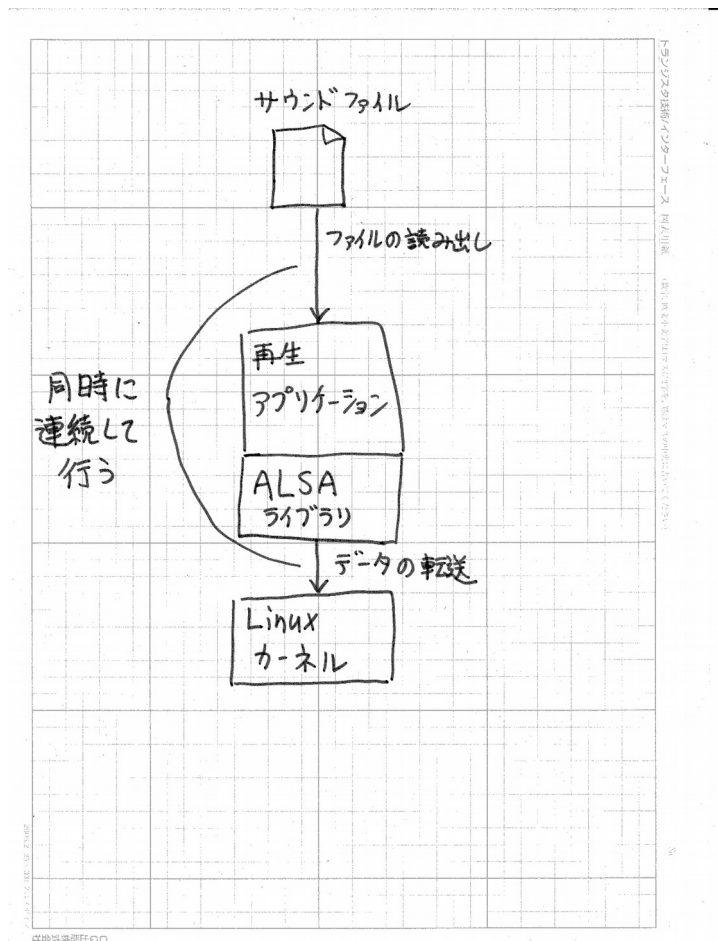
<図 8>

RaspberryPi には FireWire と PCI バスが存在しないので、USB か I2S で接続することになります。

■アプリケーション

アプリケーションがなくては、サウンドの再生はできません。

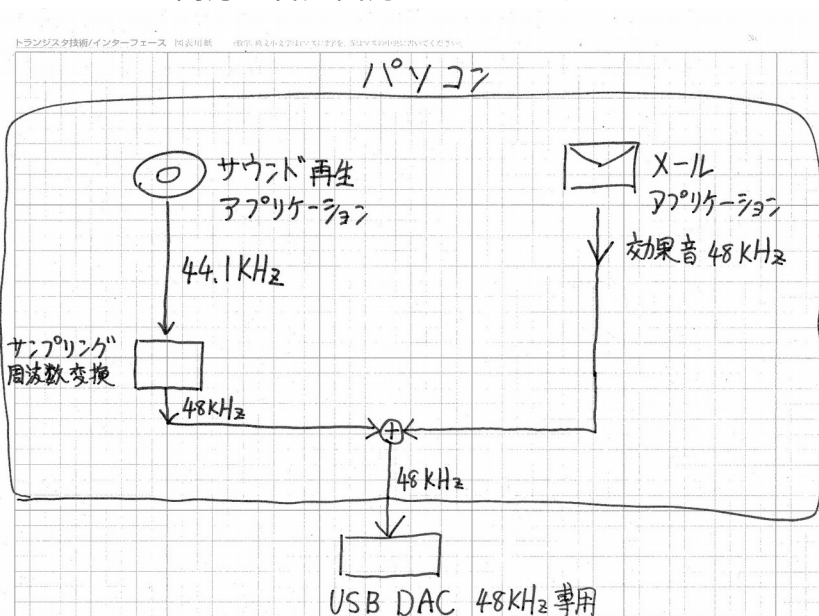
USB DAC から音を出すためには、図 9 に示すように ALSA の API を呼び出してサウンドデータを渡すことになります。再生時には、一方でファイルからデータを読み出し、もう一方で ALSA にデータを渡すという作業を同時に行う必要があります。そのため、マルチスレッドでプログラミングするのが一般的です。



<図 9>

■ビットパーフェクト

パソコンのサウンドドライバは、図4に示すようにサウンドデータのサンプリング周波数変換機能やミキシング機能を持っています。例えば48kHzのサンプリング周波数しか再生できないデバイスにCDの44.1kHzサンプリングのデータを送り込むためにはサンプリング周波数の変換が必要です。ゲームで遊んでいるときにメール到着の効果音を鳴らすためには、ゲームとメール両方の音声出力をミキシングしなければなりません。

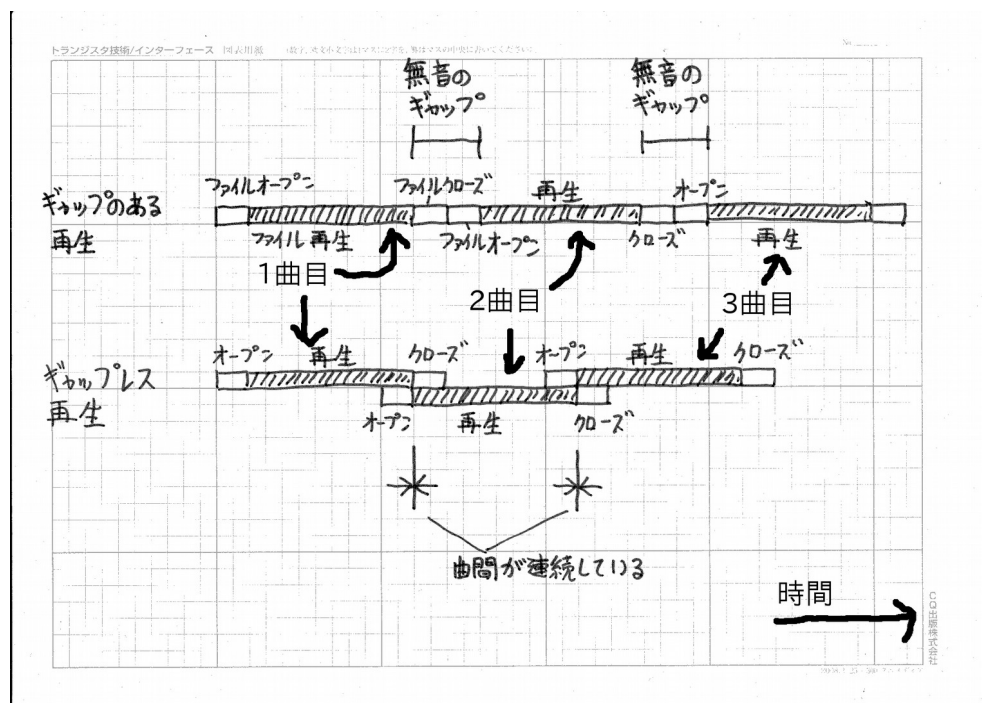


<図 10>

Hi-Fi オーディオの世界では、原音再生にこだわるので余計な変換、ミキシング機能を嫌います。サウンド再生専用デバイスに独占して、変化のかからないサウンド再生をするルートが求められます。そのようなルートでの再生をビットパーフェクトと呼びます。ALSA も、ビットパーフェクトの音声再生をサポートしています。

■ギャップレス再生

アプリケーションをシンプルに作成すると、1 曲の再生が終わってから次の曲を再生し始めるまでに、ファイルのクローズ処理、オープン処理、その他の初期化処理などで時間がかかります。するとライブ版の CD など前後の曲で音がつながっている場合、図 11 のように余計な無音が挟まってしまい興ざめです。

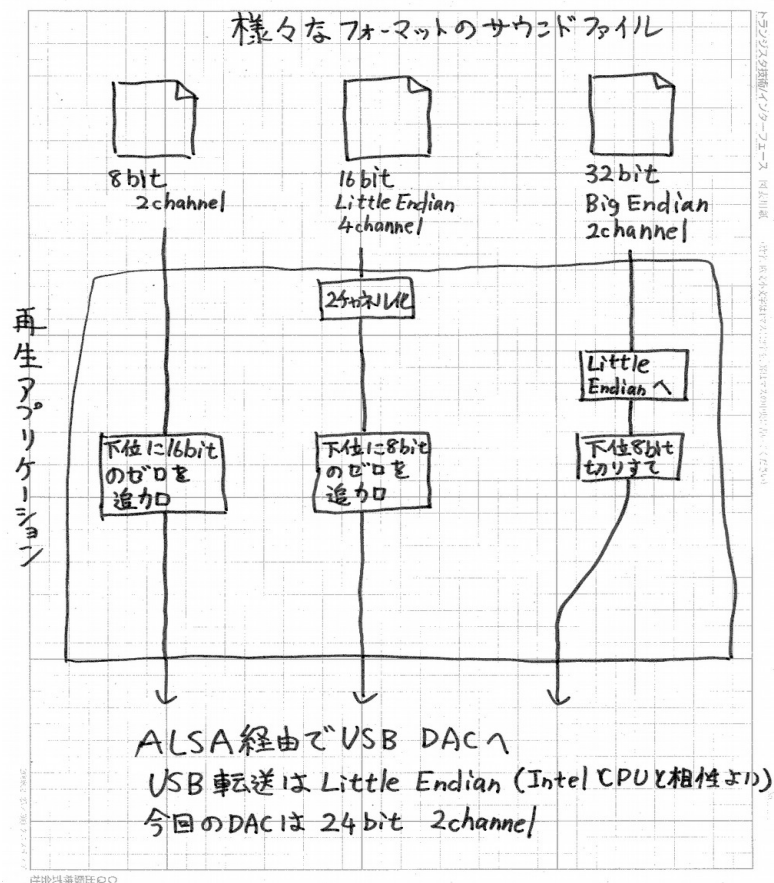


<図 11>

曲間を途切れなく再生するためには、アプリケーション側で工夫が必要になります。そのような工夫をしたアプリケーションによる再生を、ギャップレス再生と呼びます。

■フォーマット変換

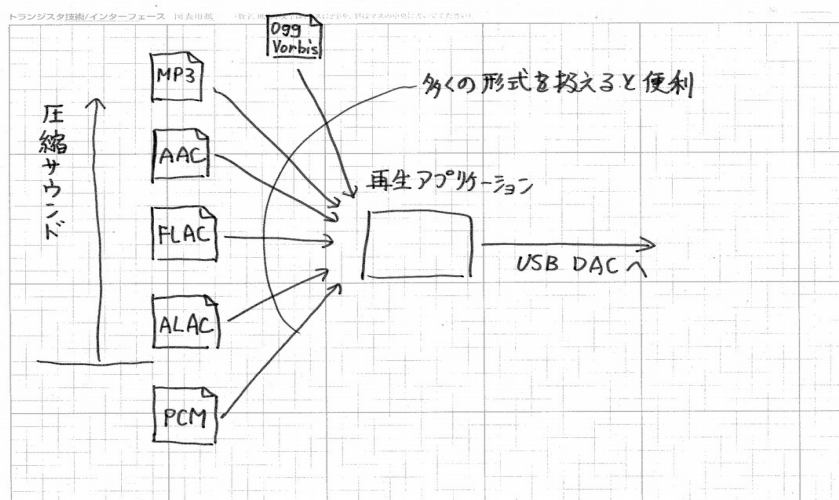
ファイルから読み出した PCM データをそのまま USB DAC に送ることができるとは限りません。PCM データもデータ長 8,16,24,32bit、エンディアンビッグ/リトル、チャンネル数など様々なバリエーションがあります。ファイルから読み出したデータを USB DAC が受け付ける形式に変換するのも、図 12 のようにアプリケーションの役目です。



<図 12>

■圧縮音声

Hi-Fi オーディオの世界では、MP3 などのロッシー圧縮は圧縮の前後でデータが変わってしまうために好まれません。それでも何らかの理由で MP3 を再生したい場合もあるでしょう。また、圧縮前のデータが完全に復活できる FLAC、ALAC などのロスレス圧縮は、頻繁に使用されます。図 13 のようにアプリケーション側でこれらのデータの展開機能を持つと、使用する場面が広がります。



<図 13>

コラム 今回性能評価に使用した Linux 向け再生アプリケーション

筆者は数年前から、Linux 向けのビットパーフェクト再生アプリケーションを、無料で配布しています。

今回性能評価に使用したアプリケーションも、配布しているものと同じです。再生時のフォーマット変換、圧縮音声の展開、曲間のギャップレスに対応しています。

このアプリケーションは、筆者の WEB サイト

<http://www.nakata-jp.org/computer/freesoft/aplayex/>

で公開しています。ただしソースコードは非公開です。

●USB オーディオクラス 2

今回実験に使用した USB DAC と PC の間のプロトコルを説明します。

USB の規格には、オーディオやビデオ、HID (Human Input Device)などの種類別にクラスがあります。それぞれのクラス毎にプロトコルが決められています。クラスプロトコル毎にすべての取り決めにしたがうようパソコンのドライバを作れば、規格に準拠したデバイスは追加のドライバがなくても接続できます。

オーディオのクラスは、最初に決めたクラス 1 が機能面、性能面で不足してきたので、拡張する意味でクラス 2 が策定されました。Linux と Mac OS X は、両方のオーディオクラスに対応しているのですが、Windows はなぜかクラス 2 に標準対応しません。Windows にクラス 2 デバイスを接続する時は、デバイスメーカーがデバイスドライバも用意します。

■ディスクリプタ

USB ターゲットデバイス（今回は DAC のこと）はホスト（今回は PC のこと）に接続した直後に、ホストの要求に応じて自己紹介データすなわちディスクリプタを送信します。USB DAC のディスクリプタには、デバイスの ID、名称、データ bit 長、チャネル数などを含んでいて、ホスト側から可能な操作、受け付けられるデータが分かるようになっています。オーディオクラス 1 とクラス 2 のコンフィグレーションディスクリプタは、一部共通のパートを使いますが、基本的に別です。

■SOF

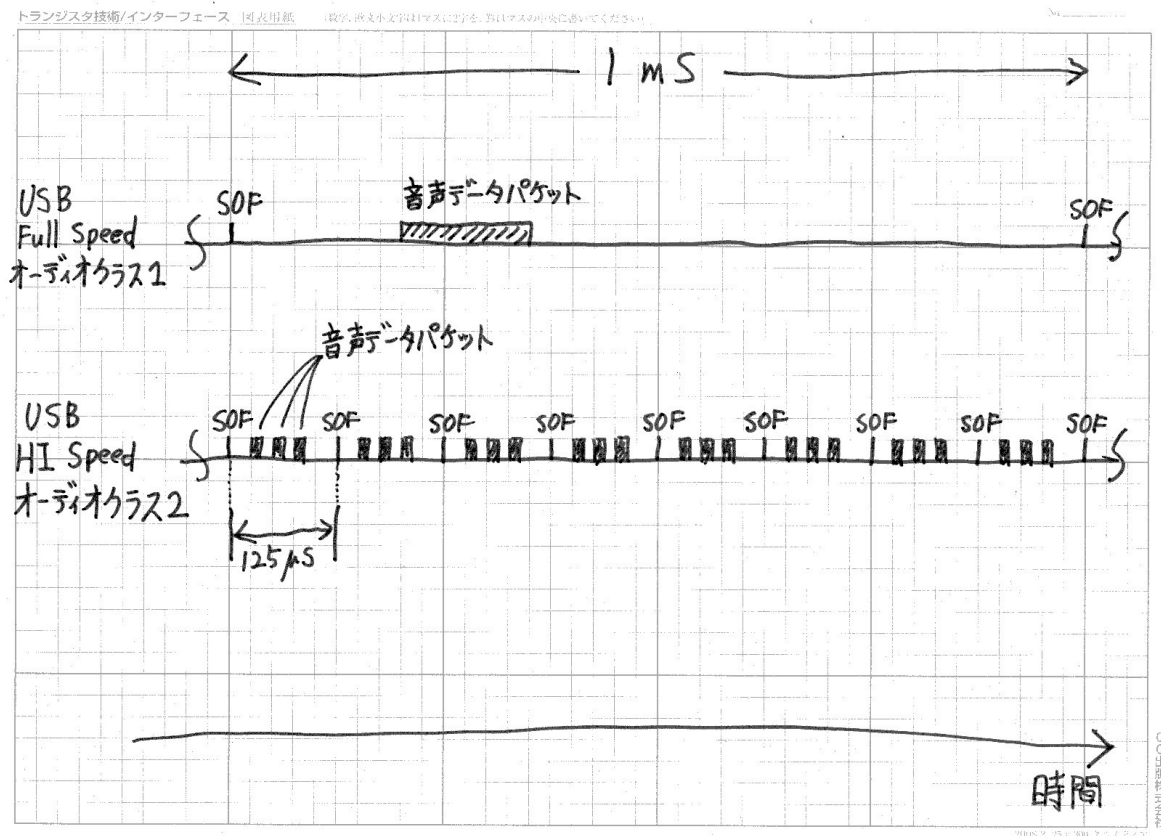
USB のホストは、ターゲットがどんなデバイスでも USB 転送のタイミングを測るため、定期的に SOF (Start Of Frame) パケットを送ります。その間隔は Full Speed(12Mbps) で 1ms、Hi Speed(480Mbps)で 125 μ s です。

データの転送は SOF と SOF の間に行います。

■アイソクロナス転送

USB オーディオは、再生サンプリングレートを切り替えない限り、一定速度でデータを受信します（図 5 参照）。USB 上を一定速度でデータ送受信するために、アイソクロナス転送が定められています。オーディオクラス 1 では、Full Speed を用い SOF と SOF の間に 1 個ずつ、最大 1023Byte のパケットを 1 個送ります。そのため、オーディオクラス 1 の最大転送レートは約 1MByte/s に限られます。このレートでは 24bit192kHz2ch のデータを送ることができません。

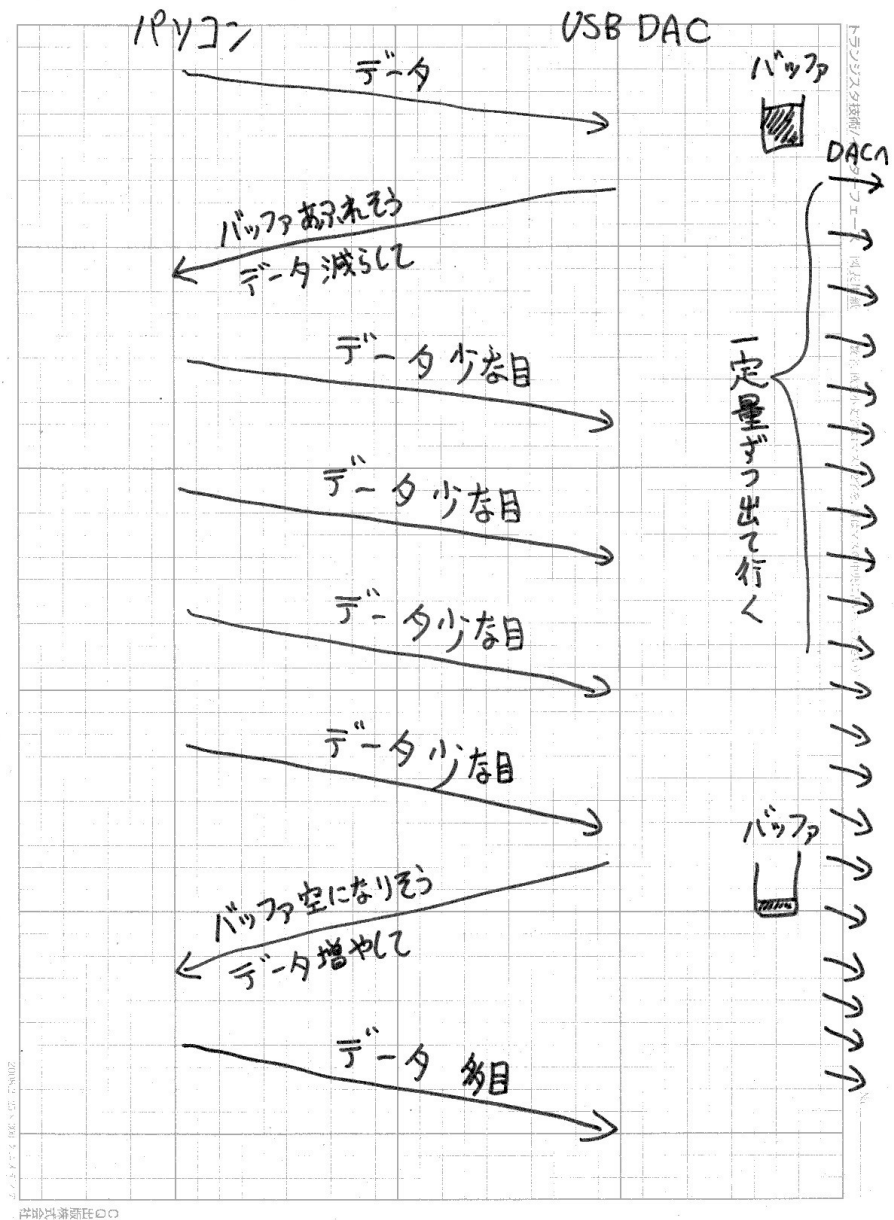
オーディオクラス2では、HI Speed を用い SOF と SOF の間に最大 1024Byte のパケットを 3 個まで送れます。最大転送レートは約 24MByte/s になります。bps に直すと、192Mbps なので、規格上は今回の USB DAC の 4 倍速までいけそうですね。



<図 14>

■アシンクロナス転送

アイソクロナス転送は、さらに細かく分けられます。今回の USB DAC は、アシンクロナス転送を採用しました。アシンクロナス転送は、USB DAC の中にクロック発振器を持ち、ホストのクロックと非同期でデータを再生します。ホストのクロックと非同期なので、ホストから一定の速度でデータを送ると、データバッファがあふれたり（ホストのクロックの方がターゲットと比較して早い方へずれている場合）データバッファが再生中に空になったり（ホストのクロックが遅い方へずれている場合）します。その問題を避けるため、図 15 に示すフィードバックという手順でターゲット、ホスト間のフローコントロールを行います。



<図 15>

■ALSA が例外扱いする USB DAC

ALSA の USB ドライバまわりのソースコードを読むと、特定のデバイス ID の時だけ USB オーディオクラスの標準処理と違うことを実行するパターンが散見されます。これは、実際に販売されている USB DAC の中に、オーディオクラスの規格から外れた動作をするものがあって、ドライバ側で違いを吸収しているものと思われます。

●まとめ

Linux マシンで、ハイレゾの USB DAC ヘデータを送る上限速度を調べてみました。Atom N450 を搭載したノートパソコンで、24bit384kHz2ch を伝送できただけでなく、24bit940.8kHz2ch 相当の伝送速度でも動作することを確認しました。Linux のサウンドドライバは、高スループット、高信頼であるといえるでしょう。

●結論

Linux は、標準で USB オーディオクラス 2 の再生に対応しており、旧式のノートパソコンを使用しても 45Mbps の高速転送レートでサウンドを送ることができました。

市販のハイレゾ USB DAC

市販のハイレゾ USB DAC から USB 接続で 24bit192kHz2ch 以上（※）の再生が可能なものをリストアップします。

※USB DAC の中には、USB と S/PDIF の両方の入力が可能で、S/PDIF だけ 192kHz に対応しているものもありますが、今回は除外しました。

USB オーディオクラス 2 準拠をうたっているものは、おそらく Linux に接続できると期待しています。しかし表中には独自伝送方式のものが混ざっているかもしれません。購入前には慎重にご確認ください。

また、24bit192kHz を再生可能な DAC でも、24bit176.4kHz を再生可能なものと不可能なものがあります。

●オーディオクラス 2 と WEB サイトに明記されていた市販 USB DAC(2014 年 6 月調べ)

メーカー	PCM	DSD	入力	出力
FOSTEX HP - A4	44.1kHz/48kHz/ 88.2kHz/96kHz/ 176.4kHz/ 192kHz	2.8MHz/ 5.6MHz	USB (Type B) 光デジタル入力 microSD カード	ヘッドフォン出力 (6.3mm ステレオ標準ジャック) アナログ(RCA) 出力 光デジタル出力
RATOC RAL-24192UT1	44.1kHz/48kHz/ 88.2kHz/96kHz/ 176.4kHz/ 192kHz	対応せず	USB (Type B)	同軸デジタル出力 アナログ (RCA) 出力 ヘッドホン (6.3mm ステレオ標準ジャック) 出力
DENON DA-300USB	32/44.1/48/64/88.2/96/176.4/192kHz	2.8224/5.6448MHz	USB (Type B) 光デジタル入力×2 同軸デジタル入力	アナログ (RCA) 出力 ヘッドホン出力 (6.3mm ステレオ標準ジャック)

Marantz SA8005	32/44.1/48/64/88 .2/96/176.4/192k Hz	2.8224/5.64 48MHz	USB (Type A) USB (Type B) 光デジタル入力 同軸デジタル入 力 CD/SACD	光デジタル出力 同軸デジタル出 力 アナログ(RCA) 出力 ヘッドホン出力 (6.3mm ステ レオ標準ジャッ ク)
-------------------	--	----------------------	--	---

●オーディオクラス2かどうか確認できなかった USB DAC

メーカー	型番	備考
CambridgeAudio	DAC magic 100	
TEAC	UD-H01	
ASUSTek	XONAR ESSENCE STU	
SONY	PHA-2	
ONKYO	DAC-1000	
PIONEER	N-50	ネットワークオーディオ機能 も内蔵

参考文献

- ・ 誠文堂新光社 河合一著 デジタル・オーディオの基本と応用
- ・ USB 規格書 Universal Serial Bus Specification Release 2.0
- ・ USB 規格書 Universal Serial Bus Device Class Definition for Audio Devices Release 2.0
- ・ BurrBrown PCM1792A データシート
- ・ ESS 社 ES9018 データシート
- ・ CQ 出版社 TECHI USB ハード & ソフト開発のすべて
- ・ インターフェース 2014 年 3 月号特集 Appendix5
- ・ トランジスタ技術 2012 年 2 月号特集第 4 章

図

図1

タイトル ハードウェア上を流れるサウンド再生データ

キャプション HDD から読み出されたデータは一旦メモリにたくわえられ、圧縮の展開やフォーマット変換などを経た後 USB インタフェースから出て行きます。USB DAC の中では、インタフェースが DAC LSI 向けの信号に変換し、最後にアナログになってデ

バイスの外へ出て行きます。

図 2

タイトル ソフトウェア間のつながり

キャプション Xaplay は GUI インタフェースであって、X window からマウスイベントを受け取り、描画を行います。alsaplay は、CUI の再生コマンドですが、Xaplay の子プロセスとして起動され、ALSA インタフェースを経由してサウンドを再生します。Xaplay と alsaplay が連携することで、一つの再生アプリケーションとして扱えます。

図 3

タイトル xosview のスクリーンキャプチャー

キャプション DoP で 11MHz の DSD を再生している時の xosview です。Linux の負荷をグラフ表示しています。LOAD アベレージが 0.9 なのは、ほぼ常に udaplay (alsaplay の姉妹コマンドで DSD 再生用) が動作しているからです。atomN450 は HyperThreading 機能があるので、CPU 負荷のグラフが 2 本あります。両方を合わせても最大瞬間負荷 70%位なので、フルの 200%までまだまだ余裕があります。バッファの読み書きは間欠的に行うので、負荷が一定せずグラフがでこぼこしています。サウンドを連続再生したあとなので、メモリの半分以上をキャッシュに使われています。ディスク入力が 2.7MBytes/s なのは、大量の DSD (11Mbps×2ch) を連続して読み出しているからです。SWAP アウトしたプロセスが 800kBytes ありますが、サウンド再生には関係ないプロセスのようです。その証拠に、再生中にページのスワップイン/アウトは発生していません。ネットワーク通信量も 0 です。

図 4

タイトル DAC LSI が受け付ける PCM の上限

キャプション BurrBrown の DAC チップ PCM1792A は 24bit192kHz2ch までを入力することが出来ます。ES9018 は、32bit500kHz2ch までを入力することが出来ます。同じ bps のラインを点線で示してみました。サンプリング周波数が同じならば、ビット長の長い方が転送速度は上ですし、同じ転送速度で並べれば 32bit の方がサンプリング周波数は低くなります。

USB の転送も LSI の入力も、半端なサンプリング周波数を受け付けられますが、現実に使われるサンプリング周波数は、44.1kHz× (2 のべき乗: 1,2,4,8,...)と 48kHz× (2 のべき乗) がほとんどです。

図 5

タイトル 今回の USB DAC が受け付ける転送速度

キャプション 今回の USB DAC では USB 上を 24bit の PCM データが流れます。

DAC LSI の ES9018 は 32bit までの PCM を受け付ける能力があるのですが、設計した筆者が『そこまでの能力は必要ないだろう』と考えて、24bit で抑えています。24bit の PCM 転送ラインの延長線上に、DoP と独自方式 DSD の転送速度があります。

図 6

タイトル OS 別の USB オーディオ転送速度比較

キャプション Linux と Mac OS X は、標準添付のドライバで USB オーディオクラス 2 が使用できます。

今回用意した筆者手持ちの中でも最高転送速度を誇る USB DAC を相手に、簡単に上限までの再生ができてしまいました。

Windows は、追加のデバイスドライバを使用してもなぜかリミッターがかかってしまい、24bit192kHz2ch の再生が上限でした。

図 7

タイトル ハイレゾ再生時のタブー

キャプション (a)ハイレゾ再生時には、HDD とカーネルのディスクドライバ、ファイルシステムに高負荷がかかります。ハイレゾ再生と同時に HDD をアクセスすると、HDD の読み出しが遅くなって、サウンド再生が途切れることがあります。

(b)1 台の USB ハブにサウンドの入った USB HDD と、再生用の USB DAC を同時につなぐと、パソコン<->ハブ間のケーブル上にサウンド読み出しと再生のデータが同時に流れます。こちらもサウンド再生が途切れる原因になるので避けましょう。

図 8

タイトル Linux のサウンドドライバは ALSA だけではない

キャプション 今でこそサウンド再生には USB DAC が主流ですが、その前には FireWire バスや PCI バスに接続するサウンドカードがありました。FireWire デバイスは FFADO ドライバを使用します。PCI バスに接続するカードはほとんどが、USB DAC と同様に ALSA の担当となります。

図 9

タイトル ALSA 経由のサウンド再生

キャプション サウンドファイルの中身を USB DAC に送るためには、ファイルを読み出して ALSA へ送り込むアプリケーションが必要になります。ファイルの読み出しと、ALSA へのデータ書き込みは同時進行するので、マルチスレッドプログラミングが一般的です。

図 10

タイトル パソコン内のサウンドの流れ（ビットパーフェクトではない一般の場合）

キャプション パソコン内では、当たり前のようにサウンドのミキシングやサンプリング周波数変換が行われています。ビットパーフェクト再生を行うと、ミキサーやサンプリング周波数変換器をバイパスしてデバイスへ直接データを届けることができます。

図 11

タイトル 再生曲間のギャップ

キャプション 複数のサウンドファイルを再生するとき、ループの中で初期化、ファイルオープン、再生、ファイルクローズ、後始末を繰り返すと、曲と曲の間に再生されな

い無音部分ができてしまいます。無音部分ができない連続再生をギャップレス再生と呼びますが、オープン、クローズなどを、サウンド再生しながらバックグラウンドで実行します。

図 12

タイトル PCM のフォーマット変換

キャプション PCM にも様々な形式があります。ビット長、エンディアン、チャンネル数などのバリエーションです。読み出した PCM は、無変更で USB DAC に送ることができるケースはまれで、アプリケーションが加工する必要があります。

図 13

タイトル 圧縮サウンドの対応

キャプション サウンドデータを圧縮して保存している人もいます。圧縮されたサウンドを再生時にアプリケーションで自動展開できると、アプリケーションの使い勝手がよくなります。

図 14

タイトル USB ケーブルを流れるサウンドデータと SOF

キャプション SOF は、Full スピードで 1msec おきに、HI スピードで 125μsec おきに流れます。

サウンドデータは、SOF の間に入ります。Full スピードでオーディオクラス 1 の時は、SOF の間に最大一つずつ、HI スピードでオーディオクラス 2 の時は、最大三つ入ります。データ量が少ない時は、間引くことも可能です。

図 15

タイトル USB アイソクロナス、アシンクロナス転送のフロー制御

キャプション データの送信間隔は、図 5 のように決まっていますが、アシンクロナス転送では、一度に転送するデータ量を増減させます。フロー制御のデータは、ホストが定期的にターゲットへ問い合わせます。ターゲットは尋ねられる度に、『データを 3 サンプル多めに』とか『1.25 サンプル少なめに』とか定量的に指示します。図中では、煩雑になるのを避けるため、データの ACK がホストに返るところやフロー制御の問い合わせパケットを省略しています。