

リアルタイムのマルチメディア処理環境としての BeOS

小林 茂

Beat Japan

kotobuki@b.email.ne.jp

1995年の登場以来、BeOSはマルチメディア処理に適したOSとして一部で高い評価を得てきた。しかし、残念ながら一般への普及度は低く、実際にどのようなことができるのかについてはあまり知られていないというのが実情である。BeOSはモダンOSの一つであり、よくデザインされたC++によるマルチメディア（ビデオ・オーディオ・MIDI）処理のためのフレームワーク・プリエンプティブなマルチタスキング・マルチスレッド・高性能な64bitジャーナリング機構付きファイルシステム、といった特徴を持っている。ここでは、リアルタイムのマルチメディア処理環境という視点からBeOSを紹介してみたい。

BeOS as a real-time multi media processing environment

Shigeru Kobayashi

Beat Japan

Since its debut in 1995, BeOS has been highly marked as well designed multi media oriented OS. Unfortunately, its possibility is not so well known in common compared with main stream OS like Windows, Mac OS and Linux. BeOS is a modern OS which features well designed C++ framework for multi media processing (video, audio and MIDI), preemptive multi tasking, multi thread, high performance 64 bit file system with journaling, and so on. I'd like to introduce BeOS as an OS for real-time multi media processing.

1. はじめに

最近になってパーソナルコンピュータ（以下 PC）のハードウェアの処理能力が著しく向上したことにより、従来であれば高価なワークステーションが必要とされたようなマルチメディア処理を十分に扱えるようになってきた。最近の PC 用 OS の多くがこうした能力を活かしてマルチメディアを扱う機能をサポートしているが、その多くは後付けで追加されたものである。このため、過去との互換性をサポートするために余計なオーバーヘッドがかかる・システム全体の規模が巨大なものになって動作が安定しない、といった問題を抱えている場合も多い。BeOS は後付けでこうした機能をサポートしたのではなく、最初からマルチメディア処理をターゲットとして開発された非常にユニークな OS である。以下にマルチメディア処理環境としての BeOS について紹介する。

2. BeOS の歴史

BeOS は Be, Inc. によって開発されたパーソナルコンピュータ向け OS であるが、当初は BeBox という PowerPC を 2 個搭載した独自アーキテクチャーのハードウェア専用の OS として 1995 年に登場した。その後、Apple の Macintosh 互換機をメインのターゲットに移すが、間もなく Apple の互換機政策の方針変更に伴って再びメインのターゲット変更を余儀なくされた。この結果、主力となるプラットフォームを x86 CPU を採用する AT 互換機（俗に言う PC）に移すこととなった。さらにその後 Be, Inc. はデスクトップ向け OS である BeOS からインターネットアプライアンスなどの組み込み市場をターゲットとした BeIA に主力を移すという方針転換を行った。このように BeOS は登場以来わずか数年間の間に何度も大きな変化を経験してきた OS である。現時点で最新のリリースは BeOS 5 で、個人・非商用利用に限り無償でダウンロードできる Personal Edition と、各国の販売代理店を経由して販売される Pro Edition の二種類がある [1]-[2]。

3. BeOS の特徴

BeOS の特徴を表す言葉としては「Media OS」という言葉が使われてきた。これは、マルチメディアの処理に適した OS という意味であるが、具体的な内容をピックアップすると次のようになる [3]-[6]。

- ・プリエンティブなマルチタスキング
- ・徹底したマルチスレッド化
- ・メモリプロテクション
- ・対称型マルチプロセッシング (SMP) 対応
- ・C++ で記述されたオブジェクト指向のフレームワーク
- ・ジャーナリング機構付の 64 bit ファイルシステム
- ・POSIX 準拠
- ・クロスプラットフォーム対応 (x86 と PPC)
- ・標準でビデオ・オーディオ・MIDI・OpenGL などの API を提供
- ・統合 GUI 開発環境を標準でバンドル
- ・アプリケーション間通信とスクリプティングをサポート

これらの特徴は Windows 2000・Mac OS X といった最近の OS と共通する部分も多い。また、フレームワークを記述する言語に C++ を採用していることなどからわかるように、全て新規に開発するのではなく、業界標準となっているものは積極的に採用しつつ、独自性を打ち出せるところに開発リソースを集中するという開発方針を採用してきている。以下に BeOS の特徴の中から何点かピックアップして具体的に紹介する。

3.1 スレッド

BeOS は OS そのものを徹底してマルチスレッド化しており、標準的な状態で 120 以上ものスレッドが起動されている。実際にアプリケーションを使用する際には、これにアプリケーションのスレッドが加わるため、通常の使用状態では約 150 以上ものスレッドが使用されていることになる。マルチスレッド自体は他の OS でも採用されているが、OS レベルでここまで徹底して採用している例は珍しい。BeOS の場合にはできるだけ細かい単位に処理を分割することで、CPU パワーのすべてを使い切るような状態（例えば重いソフトウェアコーデックを動かしてムービーを再生している状態）でもユーザーに対して素早くレスポンスを返すことができるようになってきている。また、複数の CPU が装備されている場合には、スレッド単位で処理を割り振ることにより高い効率で複数の CPU を利用することができる。

BeOS の場合にはそれぞれのスレッドに対して適切な優先度を設定できるようになっており、それぞれのスレッドが担当する処理に応じて CPU パワーを効率よく配分することができる。また、リアルタイム系のアプリケーションではスレッドをソフトウェアタイマーとして利用することも多い。BeOS の場合にはそうしたスレッドに高い優先度を割り当てることにより、アプリケーションレベルでも数十 μ 秒程度の実効精度を得ることができる。

3.2 Media Kit

次に、BeOS のフレームワークの中からマルチメディア処理に関連したものを二つ紹介する。最初に紹介する Media Kit (BeOS ではフレームワークを Kit という単位で呼ぶ) はビデオ・オーディオなどのストリーム系のデータを取り扱うための Kit である。Media Kit の特徴は次のようになっている。

- ・ μ 秒単位のタイミングをサポート
- ・ AVI・QuickTime・WAV・AIFF・MPEG-1 などの標準的なファイルフォーマットをサポート
- ・ Cinepak・MJPEG・MPEG-1・DV・MP3 などの標準的なコーデックをサポート
- ・ ストリーム専用のファイル入出力 API をサポート

Media Kit ではリアルタイム処理を行うそれぞれのモジュールを Node と呼び、Node を複数接続することでビデオやオーディオのストリームが流れるようになっている。fig. 1 は Cortex というアプリケーションで表示した、ある時点での Node 接続の様子である。この状態では、複数の Node からの出力が Audio Mixer を経由してサウンドカードの出力に接続されている。

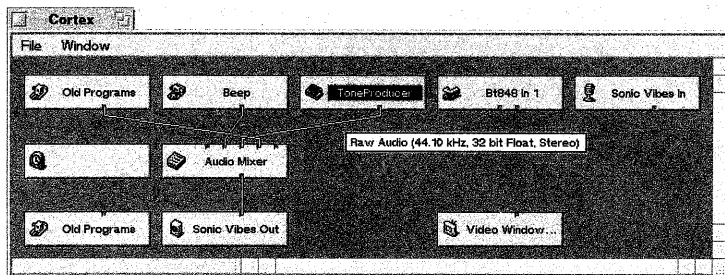


fig. 1: Cortex で表現した Node 間接続の様子

このアプリケーションをみると、音楽用のソフトウェア開発ツールとして広く用いられている Max/MSP や Pd に一見似ているように見えるかもしれないが、実際にはやや性格が異なるものである。Max/MSP や Pd はあくまで独立したアプリケーションであるのに対し、Cortex は Media Kit がシステムレベルで提供する機能を視覚化するフロントエンドの一つに過ぎない。

他の OS ではこうした機能は有力なサードパーティーが提供する 경우가多いが、BeOS の場合には OS ベンダー自身が提供している。これにより、例えば Media Kit に対する機能拡張モジュール (BeOS では Media add-ons と呼ぶ) を誰かがリリースすると、Media Kit を利用するアプリケーションの全てが恩恵を受けられるようになる。また、正確な同期機構をサポートするメディア用 API が OS レベルで提供されていることで、複数のアプリケーションを連携させて動かすことも可能になる。これらは BeOS の Media Kit の大きな特長である。

3.3 Midi Kit

Midi Kit は、その名の通り MIDI を取り扱うための Kit である。Midi Kit の特徴は次のようになっている。

- ・ μ 秒単位のタイミングをサポート
- ・ 基本的な MIDI の入出力をサポート
- ・ MIDI メッセージのパパーザーを標準で用意
- ・ アプリケーション間の通信をサポート

こうした機能が標準で用意されていることにより、MIDI について詳しく知らない場合でも比較的簡単に扱うことができる。また、タイミングの保証とアプリケーション間の通信がサポートされていることにより、ドライバレベルではなく全てアプリケーションレベルで開発を行うことが可能になる。これにより、開発期間が短縮できる・ソフトウェアの構造もシンプルになるためにソースコードも可読性の高いものになる、といったメリットが期待できる。

Midi Kit は BeOS の中でも最もシンプルな Kit である。すべてのクラスの継承関係を示したものが fig. 2 であるが、この図からわかるように、全てのクラスが BMidiEndpoint から派生しており、それぞれの役割もそのクラス名 (Producer/Consumer) から容易に想像できるものとなっている。

また、アプリケーション間通信がサポートされていることにより、いわゆる MIDI フィルターのような

なアプリケーションは非常に簡単に作成することができる。なお、アプリケーション間通信という latency が心配されるかもしれないが、BeOS の Midi Kit の場合には非常に高い優先度で MIDI 情報のやりとりを処理するため、数個のアプリケーションを経由した場合でもその間の latency は 1 msec. に満たない。

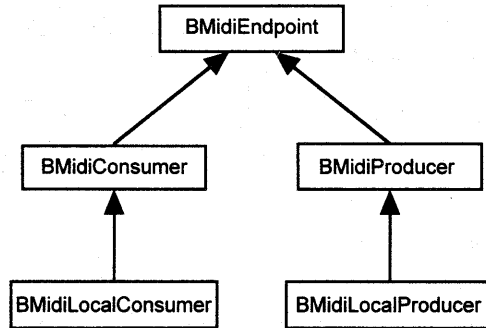


fig. 2: Midi Kit に含まれる全クラスの継承関係

4 BeOS の弱点

ここまでで BeOS の特長について紹介してきた。しかし、残念ながら BeOS にも弱点はある。以下の点は BeOS のユーザーやデベロッパーから BeOS の弱点として指摘されることの多いものである。

- ・最新のチップセットのサポートが弱い
- ・周辺機器のサポートが弱い（特に USB 接続の機器など）
- ・製品レベルのアプリケーションが少ない
- ・Be, Inc. のメインとなる商品ではなくなってしまった

これらの弱点を持つ OS が一般に普及するのは難しく、また使用目的によっては致命的な欠点となる。しかし、一度 BeOS が動作するハードウェアを組んではまえば安定したメディア処理環境を得ることができるし、必要に応じて Windows や Linux とマルチブートにしてそれぞれの OS の得意なところを組み合わせ使用して、といった利用方法も十分に現実的である。

5 BeOS をリアルタイムのパフォーマンスに使用した例

以下に昨年五月に京都国際交流会館で開催された BeaCon3^{註1} で発表した GeekPiano について簡単に紹介する。GeekPiano はデジタルピアノから出力された MIDI 信号をリアルタイムで映像に変換し、さらに演奏者の演奏中の様子をリアルタイムで取り込んで合成する、というシステムである。fig. 3 に GeekPiano の概念的なブロック構成を示す。

^{註1} <http://www.beatjapan.org/beacon3/> で当日のプログラムなどの公式な情報が公開されている。

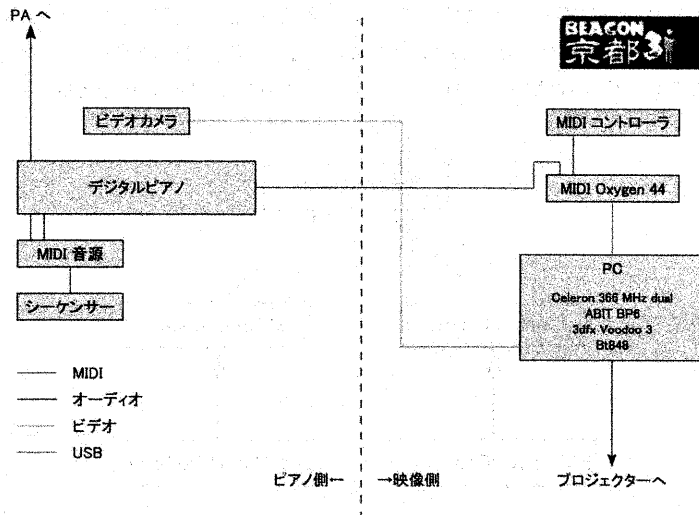


fig. 3: live://geekpiano2@beacon3.kyoto/ のシステム

こうした処理は従来は高価な画像処理用ワークステーションが必要となるもので、なかなかパーソナルなレベルで気軽に楽しめるものではなかった。しかし、BeOS の様にパーソナルコンピュータの性能を最大限に引き出してくれる OS があると、十分に安価なシステムで実現可能である。実際に、この発表で使用した PC は CPU に Celeron 366 MHz を二基搭載していたが、一基のみでも十分に機能することが確認できている。

6 おわりに

本稿では詳しく紹介できなかったが、GUI による使いやすい開発環境が標準で提供されるというのも BeOS を利用するメリットである。フレームワークが現時点の標準的なオブジェクト指向プログラミング言語である C++ で構成されていることにより、C++ を実践的に学びたい、あるいは自分のフレームワークを作る際の参考にしたい、といった場合にも有効に活用することができる。

BeOS のようなマイナーな OS は実際に動作している様子を目にすることも少なく、なかなかその良さが伝わらなかった。今回の発表が BeOS に触れる一つのきっかけになれば望外の喜びである。

参考文献

- [1] 柴田文彦: BeBox GuideBook, 1997
- [2] Scot Hacker, Henry Bortman, Chris Herboth: The BeOS BIBLE, 1999
- [3] The Be Development Team: Be Developer's Guide, 1997
- [4] 古賀信哉: Art of BeOS Programming - BeOSで楽しむプログラミングの世界, 1998
- [5] Dominic Giampaolo: Practical File System Design with the Be File System, 1998