

オープンソース PDF ライブラリの作成

ライセンスフリーな高性能 PDF 出力エンジンの開発

1. 背景

PDF (Portable Document Format) は、アドビシステムズによって開発されたファイルフォーマットで、インターネット上での文書交換のためのファイル形式として広く普及している。

PDF は仕様が完全に公開されたファイル形式のため、PDF を出力するソフトウェアライブラリは非常に多くのものが存在している。しかし、これらの多くは商用であり仕様分野が限られている。オープンソースの実装も複数存在しているが Java や PHP 等特定の言語向けのものしか存在していない。

2. 目的

本プロジェクトでは、開発者が数年前から開発をおこなっている C++ による PDF 出力ライブラリである Haru Free PDF Library (以下 libharu) をベースにして、複数の開発言語から利用可能な完成度の高い PDF 作成ライブラリを作成し、利用しやすいライセンスで公開することで、よりユーザー層を広げることを目的としている。

3. 開発の内容

旧バージョンの libharu は、C++ で実装されていたが、新バージョンでは、より移植性を高めるため、C 言語で作り直している。このため、ソースコードはほぼ全面的に書き換えられている。

3 - 1 基本的な PDF 作成機能の開発

PDF は、オブジェクトの集合体として構成されている。これらのオブジェクトを効率的に管理し、メモリリークを防ぐために、オブジェクトの生成・破棄を管理するメカニズムを実装した。

この機能により、エラーが発生した際にも、HPDF_Free() という関数を呼び出すことで、確実にリソースの解放がおこなえるようになり、安全性を高めるとともに、全体的なコード量を抑えることに成功した。

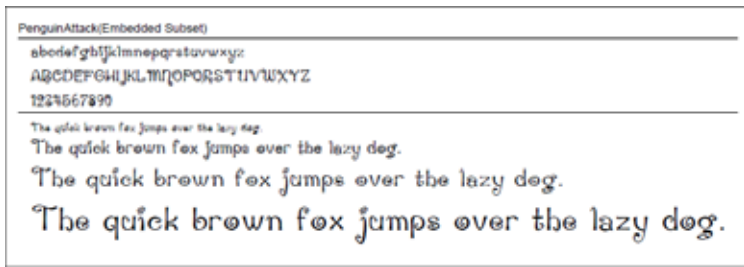
また、公開関数と非公開関数を明確に分け、公開関数では引数の入念なチェックにより、例えばユーザープログラムの不具合のより無効なポインタを渡されたようなケースでもセグメンテーション違反のような致命的なエラーが発生しにくいような実装をおこなった。具体的には、各オブジェクト (実際は構造体) の先頭 4 バイトにシグネチャを埋め込み、この値が正しいかをチェックしている。また、オブジェクトを破棄する際にこのシグネチャをゼロ埋めすることでダブルフリー問題への対策もおこなっている。

3 - 2 テキスト出力エンジンの開発

歴史的な理由により、PDF のテキスト処理は非常に複雑なものになっている。これをうまく処理するために、フォント定義体 (以下 FontDef) と、エンコーダー (以下 Encoder) という 2 つのクラスを作成し、これを組み合わせてフォントオブジェクト (以下 Font オブジェクト) を作成する仕組みを作成した。

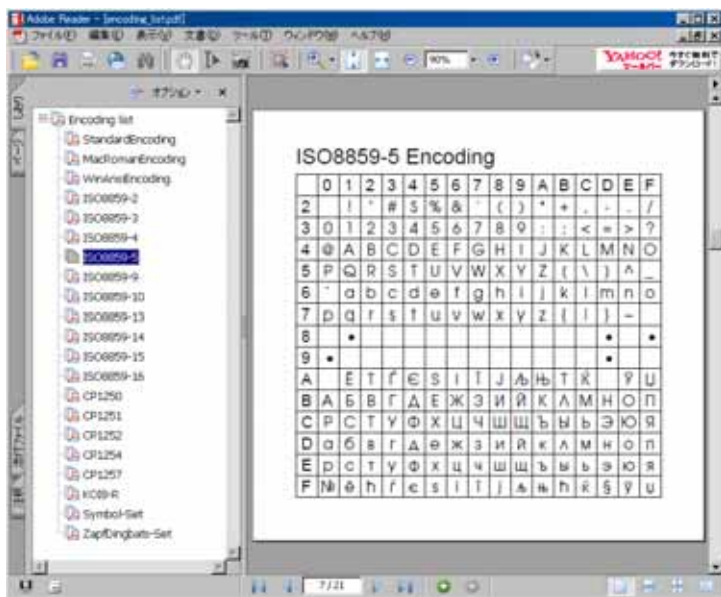
フォントについては、PDF 組み込みの 14 書体以外に、外部の Type1 フォント、TrueType フォントも使用可能であり、フォントファイルの埋め込みも実装した。

- ・ フォントを埋め込んだドキュメントの例

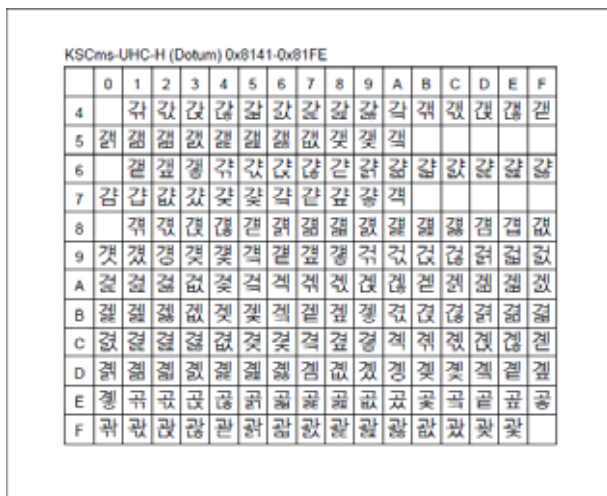


エンコーディングについては、PDF 組み込みの3種類のエンコーディング以外に、ISO8859-2 ~ 16、Microsoft コードページ 1250 ~ 1257、日中韓国語をサポートした。

- ・ 各種欧米文字セットを使用した PDF ドキュメントの例



- ・ 韓国語の PDF ドキュメントの例



これらの機能以外にも、可変幅の2バイトフォントでも、文字幅を正しく算出する機能(海外の製品では、2バイト文字は固定幅とみなしているものも多い)、文字を回転・変形させた場合でも、文字の出力位置を正しく追跡できるといった細かい機能も実装されている。

3 - 3 拡張機能の開発

拡張機能として、イメージ出力、ファイル圧縮、暗号化等の機能を開発した。

イメージ出力については、Jpeg および PNG イメージに加えて、別プログラムで作成した PDF のイメージ格納形式のデータを直接インポートする機能を新たに作成した。

この機能を使用すると、イメージのフォーマット変換のオーバーヘッドを無くすることができるため、更なるパフォーマンスの向上を実現することが可能である。

また、Jpeg イメージについては、これまで Jpeglib を使用していたが、独自の Jpeg データ解析ルーチンを自作し、Jpeglib を不要とした。

暗号化については、これまでサポートしていた 40bit の低レベル暗号化に加え、40bit ~ 128bit の可変キー長の高レベル暗号化をサポートした。

また、暗号化のコードについても、流用していた BSD ライセンスのコードを全廃し、自作コードとパブリック・ドメインのコードを組み合わせる形に変更した。

これにより、使用している全てのコードが ZLIB/PNG ライセンスで統一されることになり、使用・配布におけるライセンスへの考慮が楽になった。

3 - 4 他の開発言語のためのインターフェース機能

C/C++以外の言語からの利用を可能にするため、共有ライブラリとしても使用できるように実装をおこなった。共有ライブラリ化は、以前から要望のあった機能で、PowerBasic という Basic 亜種のユーザが独自に DLL 化をおこない、配布をおこなっていたが、今回はこの成果物を取り込み、正式に共有ライブラリをサポートすることとした。

共有ライブラリを他の開発言語から使用するためのサンプル実装として、Delphi/FreePascal 用の IF を作成した。

また、これとは別のサンプル実装として、Ruby 用の拡張ライブラリを作成した。

Ruby においては、現状 PDF を作成するための環境があまり整っておらず、特に日本語 PDF の作成については、貧弱な状態であるため、多くのユーザに利用されることが期待できる。

4 . 従来の技術(または機能)との相違

libharu は革新的な技術を売りにしたソフトウェアでは無いが、完成度の高さ、実行速度の速さ、ライセンスフリーである という3つの点で、他の競合製品に対して十分なアドバンテージを持っている。

具体的には、については機能的な部分だけでなく、高負荷が予想されるサーバ用途でも十分実用に耐えるだけの信頼性を確保している。については、商用製品にも引けを取らないパフォーマンスを持ち、Java や PHP 等で作られた他のオープンソースの実装よりはるかに高速である。また に関しては、ZLIB / PNG ライセンスという制約の緩いライセンスを採用しているため、商用・非商用に関わらず自由に使用可能である。

5 . 期待される効果

オープンソースのソフトウェア、クローズドソースの商用製品のどちらにも制限無く使用できるという利点のため、旧バージョンの libharu は、PDF 出力エンジンとしてソフトウェアに組み込まれた形で使用されるケースが多い。

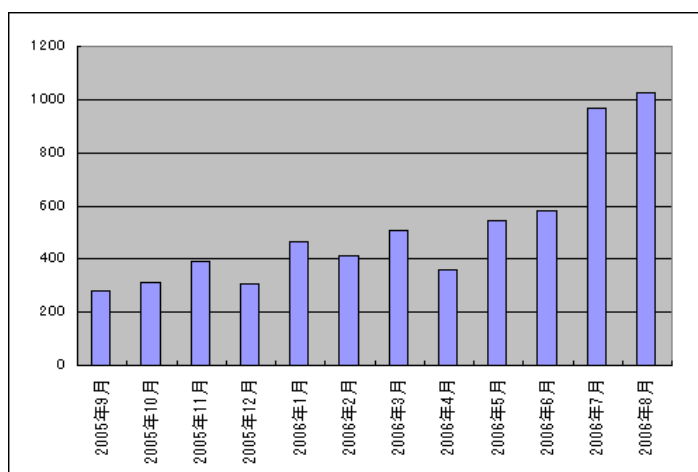
今回の成果で、さらなる高機能・高性能化と他の開発言語で使用するためのインターフェースを用意したことでより多くのソフトウェアで使用されることを期待している。

6. 普及(または活用)の見通し

開発をおこなったソフトウェアを、2006年7月23日に、libharu2としてSourceforge.netに公開をおこなった。

共有ライブラリをサポートしたことで、対象ユーザが広がったため、ダウンロード数は、3倍以上になっている。

過去1年間の月別ダウンロード数(2006年8月分は19日までの途中経過)



今後も開発を継続的におこなうことで、さらなるユーザ数の増加を実現する予定である。

7. 開発者名(所属)

菅野 健(オリンパスシステムズ株式会社)

(参考)開発者URL

- Haru Free Library(libharu)プロジェクトサイト
<http://sourceforge.net/projects/libharu/>