

4X-1

日本語PostScriptインタプリタの試作

有賀 英雄 長谷部 幹夫 堤 義直
(株) 東芝 情報通信システム技術研究所

1. はじめに

PostScript^{*} は、Adobe Systems 社が開発したPDL(Page Description Language : ページ記述言語)であり、米国市場においてはプリンタインタフェースとして事実上の業界標準になってきている。国内でもDTP(Desk Top Publishing : 卓上出版)の普及に伴ない、PostScriptがPDLの業界標準仕様の地位を得ようとしている。そして、PostScriptの日本語化とその仕様を実現したプリンタの開発が待たれている。今回、東大PostScript研究会で検討された「日本語PostScript仕様 第一次案」に対応したインタプリタを試作したので報告する。

試作の目的としては、仕様案が実際のインタプリタの実現に際して矛盾のないこと、欧文版の仕様と整合すること、及びインタプリタの変更作業上の問題点のあらいだしである。今回の試作では、まず欧文用PostScriptインタプリタをJ3100GT上に実現した。そして、日本語仕様を実現する上で変更が必要となる部分を明確にした。次に、日本語仕様対応のユーザ定義フォントを扱えるようにインタプリタを変更した。試作したインタプリタの確認作業としては、日本語仕様に基づいたユーザ定義の漢字フォントプログラムを実行し、ディスプレイ上に漢字を表示して行った。

2. PostScript言語の特徴

PostScript言語の特徴としては、次の3つが挙げられる。

- 1) プリンタの物理的な解像度に依存しないプログラミングが可能である。
- 2) 印刷可能な文字でプログラムを記述できる。(エスケープ符号等の制御符号が不要)
- 3) 言語の文法が、後置記法である。スタック指向の言語であり、処理系がインタプリタとして実現される。

3. 日本語PostScript仕様の概要

日本語仕様の概要は以下のとおりである。

- 1) 欧文用PostScriptに対する上位互換性の保障。
- 2) 仕様の拡張は最小限に押え、新規オペレータの追加は行わない。

3) 漢字コードは、EUC(Extended UNIX Code)を使う。

4) 日本語の仕様範囲は制限しない。

日本語仕様では、フォント辞書に以下に示す幾つかのエントリの追加・拡張を行う。日本語フォントは、エントリが以下のように拡張されているだけで、操作はこれまでと同様である。

MaxCharBytes: 日本語フォント辞書の識別をするためのエントリである。値は、1文字を表すのに必要な最大バイト数で、日本語フォントでは整数2が入る。

EncodingProc: このエントリは、例えばシフトJISコードからEUCといったコード変換手続きを定義する。

Encoding: 日本語仕様では、2バイトコードからのEncodingを行うために、256の配列要素のうち後半の部分の要素を256の配列とする。

AllMetrics: フォント辞書内の全ての文字のメトリック情報を一括して変更する。

4. インタプリタの構成

インタプリタは、PP, PD, SC, CPの4つの部分から成る。構成を図1に示す。

PostScriptプログラム

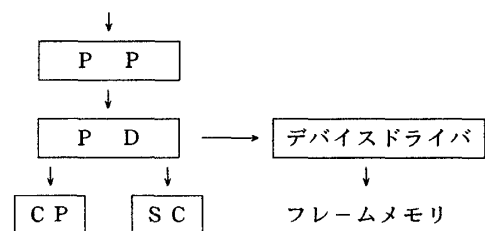


図1 インタプリタのソフトウェア構成

PP(PostScript Parser) は、インタプリタのパーサの部分である。描画処理はPD(Page Draw)で行なう。PPは、PDの提供する関数を呼び出して描画に関するオペレータを処理する。又、データI/O, PostScriptプログラムのヒープ(VM)の管理, フォント処理, フォントキャッシュ, PostSc

riptのオブジェクトの操作等である。PostScriptのオペレータは、必ずPPの関数でもしくは関数を通して処理される。

PDは、特にPostScript用の関数群というわけではなく、汎用のページ記述用のグラフィックルーチンである。PDは、ビットマップメモリ上に図形の要素を展開するためにSC(Scan Converter)とCP(Clip)の関数を呼び出す。SCは、図形をスキャンラインに分解するルーチンである。CPは、クリッピングを行なう。組み込みフォントの管理は、PDで行なう。描画結果は、デバイスの1頁分に相当するビットマップのフレームメモリに展開する。フレームメモリへの書き込みは、PDからデバイスドライバを通じて行われる。

5. フォント処理

PPのフォント処理部は、PostScriptのフォント機構及びキャッシュ機構の機能を合わせた部分である。フォント処理部は図2のように、文字の出力、文字のキャッシュ、その他のオペレータの処理の3つの部分に大きく分けられる。フォントの操作が必要なオペレータ及びキャッシュに関するオペレータがユーザのプログラムより発行されると、インタプリタ本体よりフォント処理部にある各オペレータに対応した関数が呼ばれる。

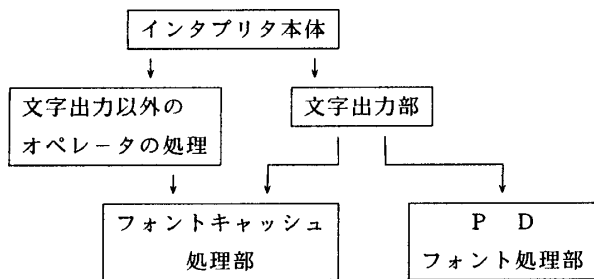


図2 フォント処理部の構成

文字出力部は、フォントキャッシュ部にキャッシュ処理の要求をして、展開しようとしている文字のキャッシュ処理を行う。組み込みフォントに対しては、PDのフォント処理部に内部コードを渡して文字を展開し、ユーザ定義フォントに対しては、ユーザの定義した手続きを実行して文字を展開する。

キャッシュ処理部は、文字のビットパターンの保存、検索、削除を行う。

文字を出力するオペレータ以外の処理では、オペランドスタック上のフォント辞書の操作やキャッシュの環境の設定などを行う。

6. インタプリタの拡張

日本語仕様対応のユーザ定義フォントを扱えるようにインタプリタの機能拡張を行なう。ユーザ定義フォントの文字出力と、内蔵フォントを含めたフォントのキャッシュは全てPPで処理しているため、組み込みフォントを除外した場合、PPの変更のみで対応できる。

試作においては、新しい関数の追加は行なわず、その機能を内部的に拡張した。拡張の内容を以下にまとめる。

1) 内部的なダミーオブジェクトの定義

日本語のフォント辞書と従来のフォント辞書を区別するのはMaxCharBytesエン트리である。その名前を辞書のkeyとして検索を行なえるようにインタプリタが既知のトークンとして登録しておく。

2) 文字出力部の変更

文字出力部の変更は、2バイトの文字コードを扱えるように拡張することである。show等のオペレータの処理関数が文字列から文字コードを取り出すときに、カレントフォント辞書のMaxCharBytesエントリの値が「2」ならば、符号系をEUCとして扱う。そして、文字出力部の中で2バイトコードが受け渡しできるようにする。

3) フォントキャッシュ処理部の変更

キャッシュ用データ構造の2バイト対応及び2バイト符号の文字をキャッシュするためのアルゴリズムの変更を行う。

7. まとめ

試作の結果、欧文用インタプリタのフォント処理部を変更するだけで日本語仕様を実現できることが判った。すなわち、インタプリタのホストからの入力部やトークンの解析部、或いはインタプリタの本体は欧文用と全く同じものが使用できる。

さらに、ユーザ定義フォントに関しては、特に日本語仕様外の機能やオペレータを付加することなく文字の処理が行なえ、仕様に矛盾のないことが確かめられた。

今後は、「日本語PostScript仕様 第二次案」対応のインタプリタの設計を組み込みフォントに関してより詳細に行なっていく。

参考文献

- 1 PostScript Language Reference Manual,
Adobe Systems Inc.
- 2 PostScript事始め, 石田晴久,
日経バイト1987年 4月～ 6月
- 3 日本語処理技術, 中原康, 電気学会雑誌, 昭61-12