JAERI-M
93-110

Xウィンドウシステム上のCalcomp インターフェイスライブラリpiflibの開発

１９９３年５月

田辺　豪信*・横川三津夫・小沼　吉男

日本原子力研究所
Japan Atomic Energy Research Institute

JAERI-Mレポートは，日本原子力研究所が不定期に公刊している研究報告書です。
入手の問合わせは，日本原子力研究所技術情報部情報資料課（〒319－11茨城県那珂郡東海村）あて，お申しこしください。なお，このほかに財団法人原子力弘済会資料センター（〒319-11茨城県那珂郡東海村日本原子力研究所内）で複写による実費頒布をおこなっております。

JAERI-M reports are issued irregularly.
Inquiries about availability of the reports should be addressed to Information Division, Department of Technical Information, Japan Atomic Energy Research Institute, Tokai-mura, Naka-gun, Ibaraki-ken 319-11, Japan.

編集兼発行　日本原子力研究所
印　　刷　日立高速印刷株式会社

JAERI-M 93-110

Xウィンドウシステム上のCalcomp インターフェイスライブラリpiflibの開発

日本原子力研究所東海研究所情報システムセンター
田辺　豪信*・横川三津夫・小沼　吉男

（1993年4月14日受理）

原研の大型計算機における図形処理は，主にカルコンプ・ライブラリを利用したものが多くみられる。図形処理プログラムを，エンジニアリング・ワークステーション（EWS）上に移植し，利用して行くことは，ソフトウェア資産を引き継ぐ上で重要である。そこで，EWS上のウィンドウ環境として普及しているX Window Systemを利用したカルコンプ互換ライブラリを開発した。

本報告書では，カルコンプ互換ライブラリ“piflib”の内容と利用方法，及び大型計算機とEWSを併用した処理方法について述べる。

東海研究所：〒319－11 茨城県那珂郡東海村白方字白根2－4

＊原子力データセンター

Development of Calcomp Compatible Interface Library "piflib" on X Window System

Hidenobu TANABE*, Mitsuo YOKOKAWA and Yoshio ONUMA

Computing and Information Systems Center
Tokai Research Establishment
Japan Atomic Energy Research Institute
Tokai-mura, Naka-gun, Ibaraki-ken

(Received April 14, 1993)

Graphics processing at JAERI mainly has been executed on mainframe computers with Calcomp compatible graphics libraries. With spread of engineering workstations (EWS), it is important that those enormous graphics softwares be able to be carried out on EWS.

The Calcomp compatible interface library "piflib" has been developed on the X Window System, which is the most popular window environments on EWS. In this report, specifications of the library "piflib" and its usages are presented. The coorperative processing with mainframe computers is also described.

Keywords : Piflib, Xpif, Calcomp, EWS, Graphics, X Window System

---

* Nuclear Data Center

目　　次

Contens

1. はじめに

近年，高性能なエンジニアリング・ワークステーション (以下 EWS) の普及に伴いマルチウィンドウ環境下で仕事をする機会がふえつつある*. 原研所内の各研究室でも独自に EWS を導入し，利用するケースが増えている. しかし，従来大型計算機を利用してきたユーザにとって，EWS 上で図形処理を行うことは，EWS で用いられているウィンドウシステムとのインターフェイスを取らなければならず，大変な労力を必要とする. もちろん，各種フリーソフトウェアを利用することでユーザの負担は軽減できるが，原研所内でのニーズを考えた場合，大型計算機上で開発されてきたカルコンプ・ライブラリ対応の図形処理アプリケーションの資産は多く，また処理方法の継続性を考慮すれば EWS のウィンドウシステムから利用できた方が効率的に作業が行えるものと思われる.

また，EWS 上のウィンドウシステムについて考えると，Sun 上の Openwindows，HP 上の VUE，IBM RS6000 上の AIX-Windows 等，マサチューセッツ工科大学 (MIT) で開発された X Window System をベースにしたマルチウィンドウ環境を提供するケースが多く，X のライブラリを用いればマシンに依存しないアプリケーションを作ることができ，大型機の図形処理プログラムを容易に移植できるものと思われる. X Window System 上で動作するカルコンプ対応ライブラリの市販品はあるが，それらはバイナリ形式で提供されるため，ソフトウェアの機能強化などソースファイル・レベルの改良はできない.

このような背景から，カルコンプ・ライブラリと同等のインターフェイスを持ち，X Window System 上で処理を行なえるライブラリ “piflib” を開発した.

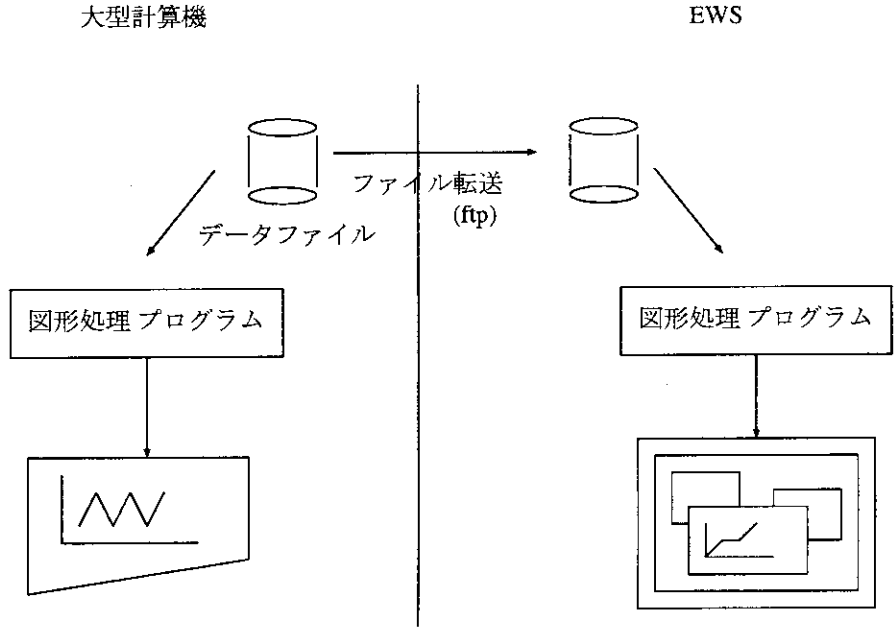

Fig. 1.1. Execution of Graphic Program on EWS

* 本報告書は，日本語 LaTeX，tgif+ を用いて書かれた.

Piflibでは，2つの処理形態を実現している．一つは，Fortranで書かれた大型機上のカルコンプ・ライブラリを用いた図形処理プログラムをEWS上でそのまま実行できるようにした．これは，大型機で動作しているカルコンプ・ライブラリと同等のFortran呼び出し形式を持つXとのインターフェイス・ライブラリを作成したことによる．ただし，図形処理に必要なデータ等はEWS上に転送しなければならない(Fig. 1.1).

もう一つは，従来通り大型機で図形処理を行うが，図形処理の結果を，大型機とEWS上で共通形式の中間ファイル(バイナリ形式)に書き出し，その中間ファイルをEWSに転送した後，ユーティリティ・プログラム“xpif”を用いて，EWS上で結果を表示させるようにした(Fig. 1.2). この中間ファイルは大型計算機でも利用可能である．

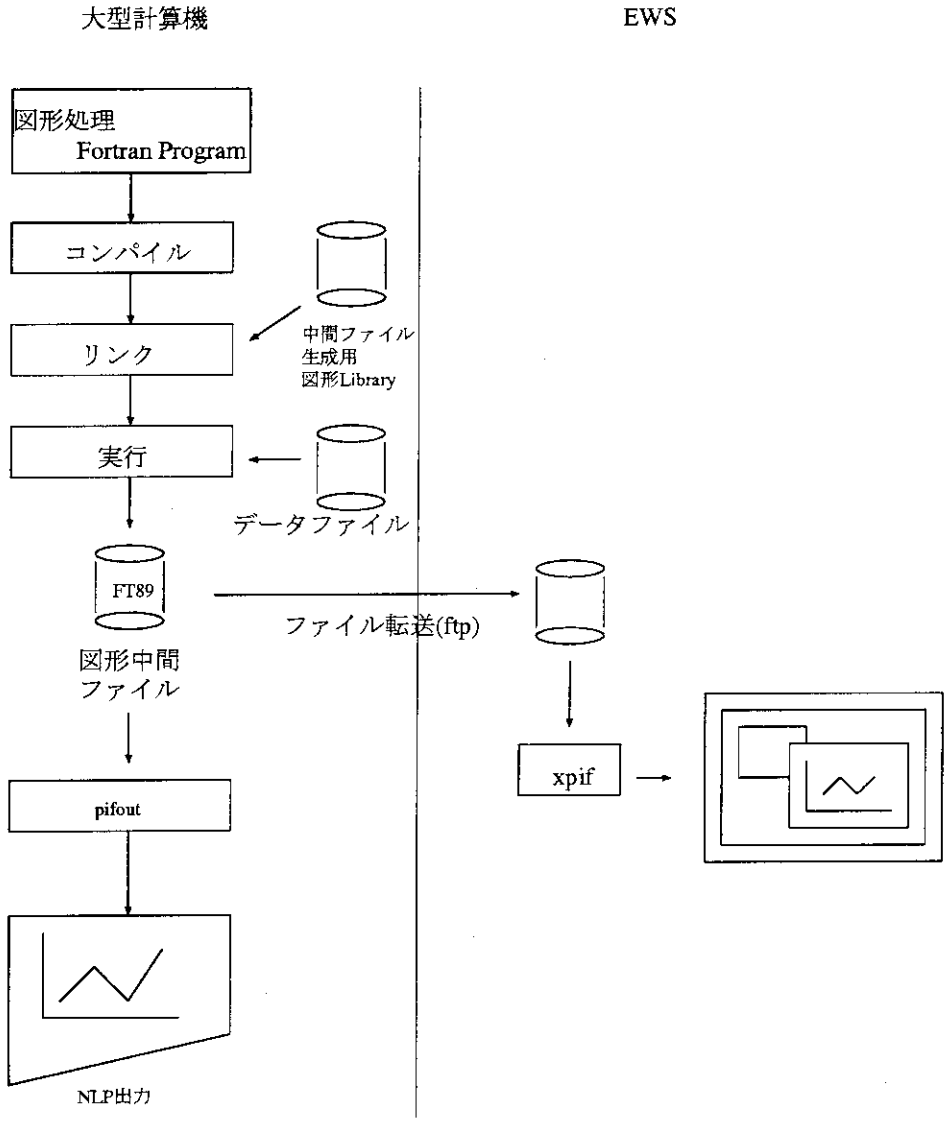

Fig. 1.2. Graphic Processing with Common Graphic Data File

また，EWS上での図形の出力先として，ウィンドウへの表示の他，PostScript形式のファイル出力をサポートした．この結果，プリント出力のデバイス依存性がなくなっている．さらに，TeXで作成した文章に図形を取り込めるよう，EPS(Encapsulated PostScript)ファイル

への出力も可能にしている.

以下では，カルコンプ互換ライブラリ piflib について，その内容と利用方法等について述べる．第2章「カルコンプ・インターフェイス・ライブラリ piflib」では，カルコンプ互換ライブラリ piflib の内部的な構造と，各ワークステーションへのインストール方法及び，ライブラリの使い方について述べる．第3章「図形出力ツール xpif」では，従来の方法と同じように，大型計算機上で図形中間ファイルを作成し，それをワークステーション上で処理するための手法と，そのために使用するユーティリティ・プログラムについて，その使い方と仕様について述べる．また，中間ファイルのデータ構造についても述べる．付録として，本ライブラリの詳細仕様を掲載した．本ライブラリを利用する上で，また本ライブラリを改良する上でのリファレンスとして利用して頂ければ幸いである.

2. カルコンプ・インターフェイス・ライブラリ piflib

Piflib （Plot Interface Library） Ver.1.0 は，日本原子力研究所情報システムセンターが作成した，X Window System 上で動作するカルコンプ・ライブラリ互換のプロット・ルーチンである．各ルーチンはすべて，大型計算機上で動作しているカルコンプ・ライブラリと同様の呼び出し形式で利用できる．このことは，従来大型計算機上で作成されたカルコンプ・ライブラリを用いたプログラムが，そのままワークステーション上で実行できるというメリットがある．これにより，ユーザは大型機の図形処理をワークステーションの環境へ容易に移行できるものと思われる.

図形の出力先として，ウィンドウへの出力のほか，PostScript 形式のファイルに出力することができる．このファイルをポストスクリプトプリンタに出力することで，図形のハードコピーが得られる (Fig. 2.1).

今回のバージョンでは，すべての基本ルーチン (HCBS) と一部のカラー・グラフィック・サブルーチン (CGSL) をサポートした．これ以外のルーチンについては，必要に応じて追加していく予定である.

EWS 上で，本ライブラリを利用するためには，

(1) X Window System が動作していること，
(2) FORTRAN コンパイラがインストールされていること，
(3) ライブラリ構築を行なうため，C コンパイラがインストールされていること，*().

が必要である.

2.1 X Window System について

X Window System は MIT の分散型教育環境開発プロジェクト「アテナ計画」の一環として開発されたもので，ネットワーク上の分散処理環境で使用することを前提としており，特定の

* Sun Solaris 2.1 から別ライセンスとなる.

への出力も可能にしている.

以下では，カルコンプ互換ライブラリ piflib について，その内容と利用方法等について述べる．第2章「カルコンプ・インターフェイス・ライブラリ piflib」では，カルコンプ互換ライブラリ piflib の内部的な構造と，各ワークステーションへのインストール方法及び，ライブラリの使い方について述べる．第3章「図形出力ツール xpif」では，従来の方法と同じように，大型計算機上で図形中間ファイルを作成し，それをワークステーション上で処理するための手法と，そのために使用するユーティリティ・プログラムについて，その使い方と仕様について述べる．また，中間ファイルのデータ構造についても述べる．付録として，本ライブラリの詳細仕様を掲載した．本ライブラリを利用する上で，また本ライブラリを改良する上でのリファレンスとして利用して頂ければ幸いである.

2. カルコンプ・インターフェイス・ライブラリ piflib

Piflib （Plot Interface Library） Ver.1.0 は，日本原子力研究所情報システムセンターが作成した，X Window System 上で動作するカルコンプ・ライブラリ互換のプロット・ルーチンである．各ルーチンはすべて，大型計算機上で動作しているカルコンプ・ライブラリと同様の呼び出し形式で利用できる．このことは，従来大型計算機上で作成されたカルコンプ・ライブラリを用いたプログラムが，そのままワークステーション上で実行できるというメリットがある．これにより，ユーザは大型機の図形処理をワークステーションの環境へ容易に移行できるものと思われる.

図形の出力先として，ウィンドウへの出力のほか，PostScript 形式のファイルに出力することができる．このファイルをポストスクリプトプリンタに出力することで，図形のハードコピーが得られる (Fig. 2.1).

今回のバージョンでは，すべての基本ルーチン (HCBS) と一部のカラー・グラフィック・サブルーチン (CGSL) をサポートした．これ以外のルーチンについては，必要に応じて追加していく予定である.

EWS 上で，本ライブラリを利用するためには，

(1) X Window System が動作していること，
(2) FORTRAN コンパイラがインストールされていること，
(3) ライブラリ構築を行なうため，C コンパイラがインストールされていること，*().

が必要である.

2.1 X Window System について

X Window System は MIT の分散型教育環境開発プロジェクト「アテナ計画」の一環として開発されたもので，ネットワーク上の分散処理環境で使用することを前提としており，特定の

* Sun Solaris 2.1 から別ライセンスとなる.

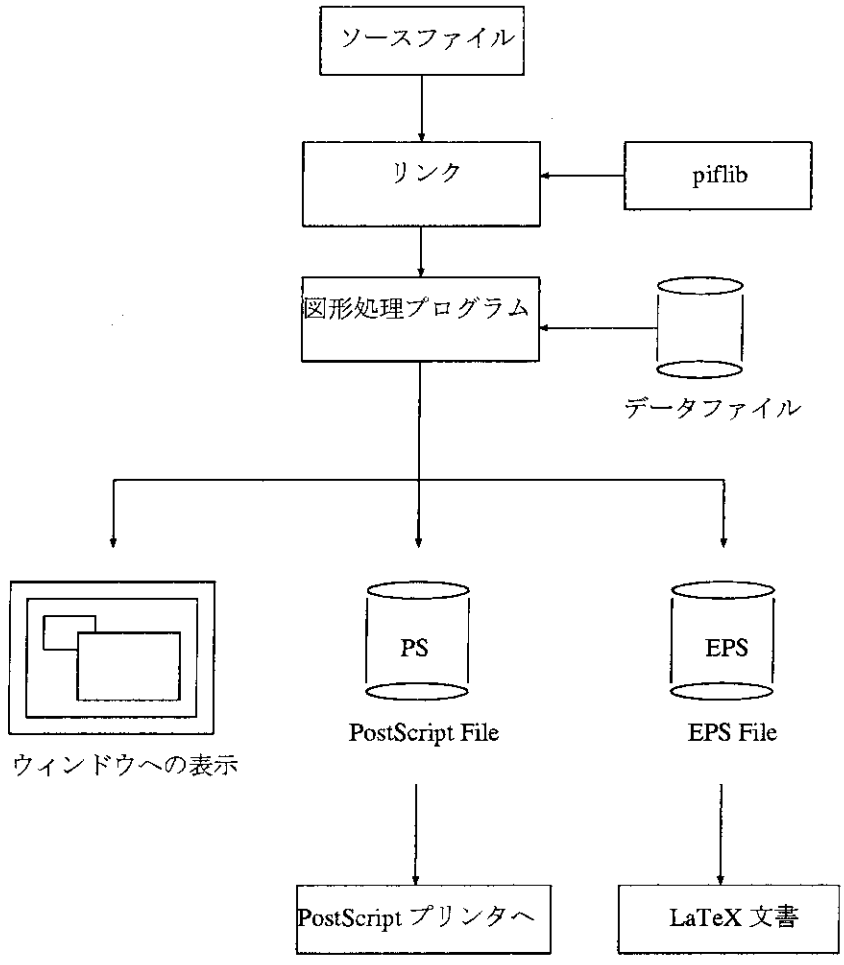

Fig. 2.1. Process with Piflib

ハードウェアに依存しない移植性の高さが特徴となっている．現在普及しているのは，X Window System Version 11 Release 4 (以後，X11R4 と略す．他のバージョン，リリースについても同様) であり，各社の EWS 上に移植されている．また，1991 年 12 月には X11R5 がリリースされた．

2.1.1 クライアント・サーバ・モデル

X Window System では，各処理を「クライアント」と「サーバ」と呼ばれる 2 つのプログラムによって行う．「サーバ」は，ウィンドウを生成し，スクリーン上に表示させたり，図形を出力させたりする．また，キーボードやマウスの入力を処理するのもサーバが行う．「クライアント」はウィンドウの生成を要求したり，図形の出力を要求したりするが，これらの要求は，X プロトコルというバイト・ストリームによってサーバに転送される．X プロトコルを受けたサーバは，クライアントの要求に従って，ディスプレイ上にウィンドウを表示したり，図形を出力したりする．一方，キーボード入力や，マウスボタンの動作は，サーバによって検知され，その情報は X プロトコル (イベント) として，クライアントに転送される．このように，クライアントは，直接デバイスとの入出力をせず，X プロトコルによって結合しているサーバを通し

て，間接的にデバイスとの入出力を制御する．

2.1.2 Xプロトコル

Xプロトコルとは，クライアントとサーバ間のコミュニケーションのための基本メッセージで，リクエスト，リプライ，イベント，エラーの4種類がある．

リクエスト は，ウィンドウの生成，ウィンドウの操作，図形の出力などの要求をクライアントがサーバに発行するときに使われる．また，各種情報をサーバに要求する時にも使用される．

リプライ は，クライアントから情報の要求をうけたサーバが，必要な情報をクライアントに返すため，サーバからクライアントに送るメッセージである．情報の要求をしたクライアントは，サーバからこのリプライが返るまで待ち状態に入る．ただし，ウィンドウの生成や図形の出力などの要求の場合には待たない．

イベント は，サーバ側で非同期に発生する事象をクライアントに知らせるためのメッセージである．事象発生の原因としては，キーボードのキーが押された，マウスのボタンが押されたなどの物理的原因によるものの他，ウィンドウの重なり状態が変わった，ウィンドウの大きさが変わったなどの論理的原因も含まれる．

エラー は，サーバ側で，エラーの発生が検知されたことをクライアントに知らせるメッセージである．

各メッセージの転送方向は決まっており，リクエストはクライアントからサーバへ，その他はサーバからクライアントへ転送される．

このXプロトコルを介してクライアント・サーバ間の通信がおこなわれるためクライアント・サーバ間でXプロトコルを遵守しなければならない．しかし，これを直接制御するのはかなり複雑な処理が必要となるので，アプリケーション開発の便宜をはかって「Xlib」というXプロトコルを扱うためのC言語から利用するライブラリが用意されている．

2.1.3 環境変数DISPLAY

X Window Systemは，ネットワーク上での分散処理環境で使用することを前提として開発されているので，サーバとクライアントが同一のマシン上にある必要はない．しかし，1つのネットワーク上に複数個のXサーバがある場合，ウィンドウをどのサーバに表示させるかをクライアントに指示しなければならない．このため，各ディスプレイ毎に名前が付けられており，この名前は，通常そのワークステーションの名前(ホスト名)とサーバ番号，スクリーン番号で以下のように表される．

ホスト名：サーバ番号．スクリーン番号

通常，この名前は，クライアント起動時に直接与えるか，あるいはDISPLAYという環境変数を通してクライアントに与える．環境変数を用いる場合には，自分が使用しているマシンがhost01というマシンである時に，Cシェルで次のように設定する．

```
setenv DISPLAY host01:0.0
```

この環境変数を設定すれば，動作させるクライアントは，全て host01 のディスプレイに表示されることになる

2.2 Piflib の内部仕様

カルコンプ・インターフェイス・ライブラリ piflib は，次の 2 種類のルーチン群から構成されている．

(1) カルコンプ互換のインターフェイス・ルーチン群 (Table 2.1)
(2) インターフェイス・ルーチン群から呼ばれ，X Window System 上で描画を行うルーチン群

この内，ユーザが実際に利用するのは，(1) のインターフェイス・ルーチン群である．これらのインターフェイス・ルーチン群は，標準カルコンプ・ライブラリと同様の呼び出し形式を保持している．また，(2) の描画処理を行うルーチン群は，次の 3 つの機能ルーチンに分けることができる．

(1) ウィンドウ描画ルーチン
(2) PostScript ファイルへの出力ルーチン
(3) EPS ファイル (TEX 用 PostScript ファイル) への出力ルーチン

Table 2.1. Interface Routines

サブルーチン	呼び出し形式	機能
PLOTS	CALL PLOTS(DUMMY,IDUMMY)	プロッタの初期化を行う
PLOT	CALL PLOT(X,Y,IPEN)	図形の描画及び，画面制御
SYMBOL	CALL SYMBOL(X,Y,H,STRING,TH,N)	シンボルを描く
FACTOR	CALL FACTOR(FCTR)	尺度因子を指定する
NEWPEN	CALL NEWPEN(IPEN)	ペン種を変更する
WHERE	CALL WHERE(X,Y,FCTR)	現在の座標値，尺度因子を返す
SETLTP	CALL SETLTP(LTP)	線種の属性を変更する
SETLWD	CALL SETLWD(LWD)	線幅の属性を変更する
SETCLR	CALL SETCLR(ICLR)	線の色を指定する
SETRGB	CALL SETRGB(IRED,IGREEN,IBLUE)	線の色を指定する
SYMFNT	CALL SYMFNT(X,Y,H,STRING,TH,N)	フォントを用いてシンボルを描く
DOT	CALL DOT(X,Y)	画面にドットを描く
CPFONT	CALL CPFONT(FNAME)	使用するフォントを指定する

2.2.1 命令の詳細

ここでは，ユーザがFORTRANプログラムから利用するpiflibルーチンについて，その機能と呼び出し形式及び，引数の型について述べる.

(1) PLOTS

機能： サーバとの接続を確立し，ウィンドウの生成，表示，属性の初期化等を行う．また，座標原点を，ウィンドウの左下隅から上に20.0mm，左に20.0mmに設定する.

呼び出し形式： `CALL PLOTS( DUMMY, IDUMMY)`

引数： どちらもダミー・パラメータ

(2) PLOT

機能： ペンを現在位置から指定された座標まで，直線的に移動する.

呼び出し形式： `CALL PLOT( X, Y, IPEN )`

引数：
- X, Y ： ペンを移動させる終点座標値. (単位:mm)
- IPEN ： ペンのアップ・ダウン及び，画面制御情報
  - $= 2$ ： ペンダウンで指定座標まで移動.
  - $= 3$ ： ペンアップで指定座標まで移動.
  - $= -2$ ： ペンダウンで指定座標まで移動後，その座標を原点とする.
  - $= -3$ ： ペンアップで指定座標まで移動後，その座標を原点とする.
  - $= 666$ ： 改ページ，原点のリセット及び，属性の初期化を行なう.
  - $= 777$ ： ウィンドウへの図形の表示を行う.
  - $= 888$ ： ウィンドウへの図形の表示を行う.
  - $= 999$ ： 処理の終了.

(3) SYMBOL(1)

機能： 指定された座標に，アルファベット，数字及び，特殊記号を描く.

呼び出し形式： `CALL SYMBOL( X, Y, H, STRING, TH, N )`

引数： X, Y ： 最初に描かれる文字の左下隅の座標値を指定する. (単位:mm)
H ： 文字を描く高さを指定する. (単位:mm)
STRING: 描きたい文字列.
TH ： 描きたい文字列の, X軸とのなす角度. (単位: 度)
N ： 文字数. (N ¿ 0)

ここで, X及び, Yを 999.0 と指定すると, 前回コールされた SYMBOL ルーチンで描かれた文字列の後ろに連続して描かれる.

(4) SYMBOL(2)

機能： 指定した座標を中心として, センター・シンボルを描く.

呼び出し形式： `CALL SYMBOL( X, Y, H, ITEM, TH, N )`

引数： X, Y ： センター・シンボルの中心座標. (単位:mm)
H ： 描きたいシンボルの高さ. (単位:mm)
ITEM ： センター・シンボルのコード番号. (0 ～ 15)
TH ： 描きたいシンボルのX軸とのなす角度 (単位: 度)
N ： = -1 と指定する

(5) FACTOR

機能： 図形全体の拡大, 縮小比率を指定する.

呼び出し形式： `CALL FACTOR( FCTR )`

引数： FCTR ： 尺度因子を指定する.

このルーチンの呼び出しによる FCTR 値の設定は, 次の変更があるまで有効である. このルーチンを使用しない場合は, FCTR 値は 1.0 である.

(6) NEWPEN

機能： ペン種を変更する.

呼び出し形式： `CALL NEWPEN( IPEN )`

引数： IPEN ： 1～3の整数を標準としてペン種を指定する.

	1	2	3
ウィンドウ	黒	赤	青
PostScript	細線	中線	太線

(7) WHERE

機能： 現在のペン位置の座標と, 尺度因子を返す.

呼び出し形式： CALL WHERE(X, Y, FCTR)

引数： X, Y ： ペンの現在位置の X, Y 座標をストアさせる.
FCTR ： 現在の尺度因子をストアさせる.

(8) SETLTP

機能： このルーチンの呼び出し以降, 描かれる線の線種を変更する.

呼び出し形式： CALL SETLTP(LTP)

引数： LTP ： 線種インデックス番号 (0 ～ 11). 標準値は 0(実線) である.

(9) SETLWD

機能： このルーチンの呼び出し以降, 描かれる線の線幅を変更する.

呼び出し形式： CALL SETLWD(LWD)

引数： LWD ： 線幅のインデックス番号 (1 ～ 5).

(10) SETCLR

機能： このルーチンの呼び出し以降, 描かれる線の色を指定する.

呼び出し形式： CALL SETCLR(ICLR)

引数： ICLR ： カラーインデックス番号 (0 ～ 7).

Index	0	1	2	3	4	5	6	7
表示色	黒	白	赤	緑	青	シアン	マゼンタ	黄

(11) SETRGB

機能： このルーチンの呼び出し以降, 描かれる線の色を RGB 値で指定する.

呼び出し形式： CALL SETRGB(IRED, IGREEN, IBLUE)

引数： IRED ： 赤色の成分値. (0 ～ 255)
IGREEN: 緑色の成分値. (0 ～ 255)
IBLUE ： 青色の成分値. (0 ～ 255)

(12) SYMFNT

機能： 指定した座標から, フォントを利用して文字列を描く.

呼び出し形式： CALL SYMFNT(X, Y, H, STRING, TH, N)

引数： X, Y ： 最初に描かれる文字の，左下隅の座標値．（単位:mm）
H ： 現在ダミーパラメータ．
STRING: 描きたい文字列．
TH ： 現在ダミーパラメータ．
N ： 文字数．

(13) DOT

機能： 指定した座標にドットを描く．

呼び出し形式： `CALL DOT( X, Y )`

引数： X, Y ： ドットを描く座標値．（単位:mm）

(14) CPFONT

機能： このルーチンの呼び出し以降，使用するフォントを指定する．

呼び出し形式： `CALL CPFONT( FNAME )`

引数： FNAME: X Window System で利用可能なフォント名．
指定したフォントが存在しない場合，エラーとなる．

2.2.2 イベント処理

Xlib を用いて図形処理を行う場合の手順は，次に示す処理を行う必要がある．

(1) サーバとの接続を確立
(2) ウィンドウの生成
(3) ウィンドウのマッピング
(4) イベント・ループの設定（ループ内で図形の描画）
(5) サーバとの接続を解除

piflib では，1～3 を PLOTS で処理し，4 を PLOT(x, y, 666 or 777)，5 を PLOT(x, y, 999) で処理している．

ここで，イベント・ループの設定とは，マウスやキーボードからの入力や，ウィンドウの状態の変化等で発生するイベントを処理するためのループを設けることである．通常，このループの中でウィンドウに図形を描画する処理を行う．イベント・ループを設定しない場合，図形を直接ウィンドウに描くと，ウィンドウのリサイズや，重ね合わせなどのイベント発生時に図形が消えてしまうことになる (Fig. 2.2)．

これを避けるためにイベント・ループを設け，ループ内で図形を描画することで，イベント発生時に自動的に再描画することになる．このイベント・ループの設定は，X のライブラリを利用する必要がある (Fig. 2.3)．

しかし，PLOT ルーチン等の描画ルーチンは，Fortran プログラムから呼ばれるものであり，また，従来のカルコンプ・ライブラリを利用したプログラムとの互換性を考えた場合，このようなイベント処理を行うことができない．そこで piflib では直接ウィンドウへの描画を行うのでは

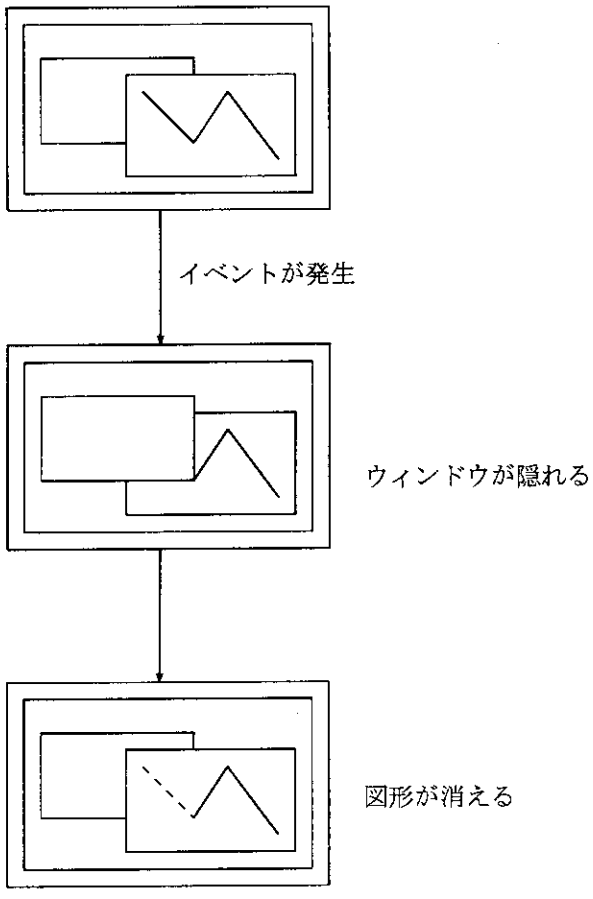

Fig. 2.2. No Preservation of contents of a Window

なく，すべての図形を一度ビックスマップ†上に描いておき，イベントが発生した場合にそこからウィンドウに図形をコピーすることで，イベントを処理している (Fig. 2.4).

2.2.3 PostScript 出力

Piflib で作成した図形は PostScript 形式のファイルに出力することができる. PostScript はインタープリタ型のページ記述言語であり，UNIX 上で動作するフリーウェア・プログラムの多くは，この形式のファイルを出力することができる. このファイルを PostScript 対応のプリンタに送ることによって図形のハードコピーを得ることができる.

また，TeX で書かれた文章に図形を取り込むための EPSF 形式のファイル出力にも対応している. EPSF(Encapsulated PostScript Files) は基本的には PostScript ファイルであるが，TeX 等の文書整形ツールや，プレゼンテーション・ツール等に図形を効果的に取り込むために利用される.

† ビックスマップとは，サーバー側のメモリ上に確保される描画可能な領域である. X ではウィンドウとビックスマップを総称して drawable と呼ぶ.

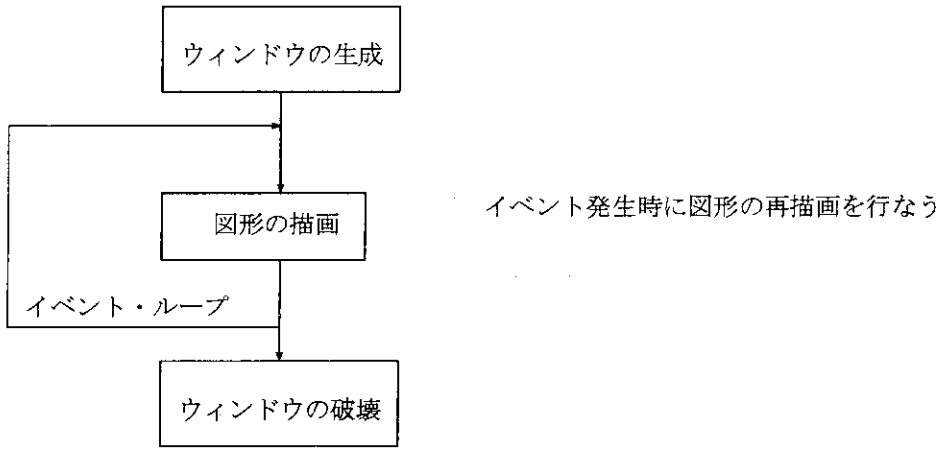

Fig. 2.3. Window Redrawing in the Event Loop

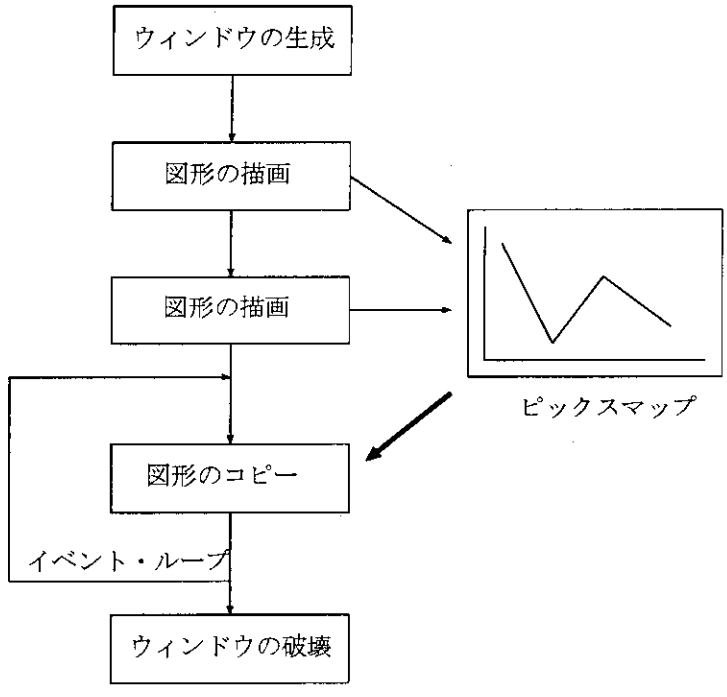

Fig. 2.4. Drawing by using Pixmap

2.2.4 図形の描画原点について

ウィンドウに図形を表示する場合，その原点はウィンドウの左下隅から横方向に 20mm，縦方向に 17mm である．これは，従来のカルコンプ・ルーチンとくらべて，左方向に 10mm ずれている．

PostScript 出力時の描画は，A4 用紙，ランドスケープモードでの出力となる．この時の描画原点は用紙の左下隅から横方向に 20mm，縦方向に 17mm である．なお，プリンタ固有のハード的な描画限界があるため，ウィンドウに表示された図形とプリントされる図形とが多少異なる場合があるので，その場合には図形プログラムの方で大きさを調整する必要がある．

2.3 使用方法

Piflib を EWS 上で使うには，まず，piflib ライブラリをそれぞれに EWS 上に構築し，Fortran プログラムをコンパイルする時にリンクするだけでよい．現在，X11R5 twm，Sun4 OpenWindows，HP9000 HP-VUE，IBM RS6000 AIX-Windows の各環境下で正常に動作することを確認した．

2.3.1 インストール方法

Piflib のインストールは，UNIX コマンドの make を使用することにより，簡単に行なうことができる．ここでは，piflib ライブラリに構築方法について述べる．このインストール作業では，後で述べるユーティリティ・プログラム “xpif” のインストールも同時に行なわれる．

(1) ソース・ファイルを入手する．

(2) ソース・ファイルをディスク上に展開する．

例)
```
% uncompress piflib.tar.Z
% tar xvf piflib.tar
% cd piflib
```

(3) 適当なエディタを用いて makefile を修正する．デフォルトのターゲットマシンは “SUN” である

(a) ライブラリをインストールするディレクトリを指定する．

例) `INSTDIR = /usr/local/lib`

(b) ユーティリティ・プログラムをインストールするディレクトリを指定する．

例) `BINDIR = /usr/local/bin`

(c) X のインクルード・ファイル及び，ライブラリが存在するディレクトリを指定する．

例)
```
INCLUDE = -I/usr/include
LIB_PATH = -L/usr/lib/X11
```

このとき，デフォルトで指定されているライブラリは，X11 ライブラリと，数学関数ライブラリである．

例) LIBS = -lX11 -lm

(d) FORTRAN から C で書かれたルーチンを呼び出す場合の関数名の付け方をマニュアルで調べ，関数名の後ろにアンダースコア ("_") が必要ならば CFLAGS に -D_NEED_UBAR を指定する.

例) CFLAGS = -D_NEED_UBAR

(e) 修正した makefile を保存する.

(4) ライブラリ，及びユーティリティの構築
シェルのコマンドラインから,

```
例)  % make
     % make install
```

と入力し，エラーがなければ終了. 必要のないオブジェクト・ファイルは,

```
例)  % make clean
```

で消去される.

ここで注意して欲しいのは，(3) のアンダースコアに関する設定である. SUN-FORTRAN 等の BSD 系 UNIX において，FORTRAN から C で書かれた関数をコールする場合，C の関数名の後ろにアンダースコアを付け加える必要がある（HP，IBM では System V 系 UNIX であるので必要ない）. ここで注意して欲しいのは，(3) の D_NEED_UBAR の設定である. 以下，Sun 上で，FORTRAN から "SAMPLE" という C で書かれた関数をコールする例を示す. C で書かれた副プログラムの方では "SAMPLE_" とアンダースコアが付いている.

FORTRAN :

```
REAL FOO, BOO
CALL SAMPLE( FOO, BOO )
```

C :

```
SAMPLE_( FOO, BOO )
   FLOAT FOO, BOO;
{
    ......

}
```

2.3.2 ユーザ・プログラムとのリンクの方法

ユーザ・プログラムと piflib ライブラリをリンクするためには，FORTRAN コンパイラのオプションで piflib と X のライブラリを指定するだけでよい. コンパイル・オプションの指定についてはそれぞれのマニュアルを参照して欲しい. 実際の作業の手順としては以下のようになる.

(1) FORTRAN でカルコンプ・ライブラリを用いたソース・プログラムを書く.

(2) リンクオプションにライブラリとそのディレクトリを指定し，コンパイルを行なう．

例えばSUNのマシン上でtest.fというFORTRANのソースファイルをコンパイルする場合には，

```
% f77 -o test -L/usr/local/lib -L/usr/lib/X11 -lpiflib -lX11 test.f
```

とすればよい．これで，testという実行ファイルが作成される．この例では，Xのライブラリが/usr/local/X11にあり，piflibライブラリが/usr/local/libにインストールされた場合のコンパイル方法を仮定している．各オプションの意味は，

-o 実行モジュール名を指定する．(デフォルトはa.out)

-L ライブラリが存在するディレクトリを指定する．

-l ライブラリ名を指定する．

である．通常，UNIX上でのライブラリはlib○○○.aというファイル名が付いている．-lオプションで指定する場合○○○の部分を指定すればよい．piflibは通常libpiflib.aというファイル名でインストールされるので，-lオプションを利用する場合-lpiflibとなる．同様に，XのライブラリはlibX11.aとなっているので-lX11である．

2.3.3 操作方法

Xウィンドウ環境下で，piflibライブラリをリンクしたプログラムを実行すると，最初に，以下のようなメニューが表示される．

```
*** Plot Interface Library Version 1.00 ***
  Enter one of the following numbers
      1 : X-Window
      2 : PostScript File
      3 : EPS File for TeX
  Select [ 1 - 3 ] :
```

ここで，1～3のいずれかを選択する．メニューの意味は以下の通りである．

(1) ウィンドウへの図形出力
(2) PostScriptファイルへの出力 (Fig. 2.5)
(3) EPSファイル(Encapsulated PostScript File)への出力 (Fig. 2.6)

(2)，又は(3)を選択した場合，出力するファイル名を聞いてくるので，適当なファイル名を入力する(デフォルトは'piflib.ps')．また，3を指定した場合は，1画面につき1ファイルが必要となるため，入力したファイル名に対してページ番号が自動的に付加される．

ウィンドウへの表示を選択した場合，改ページは，そのウィンドウにマウスカーソルを合わせて何かのキーを押すと行なえる．最終ページから抜けるときも同様である．

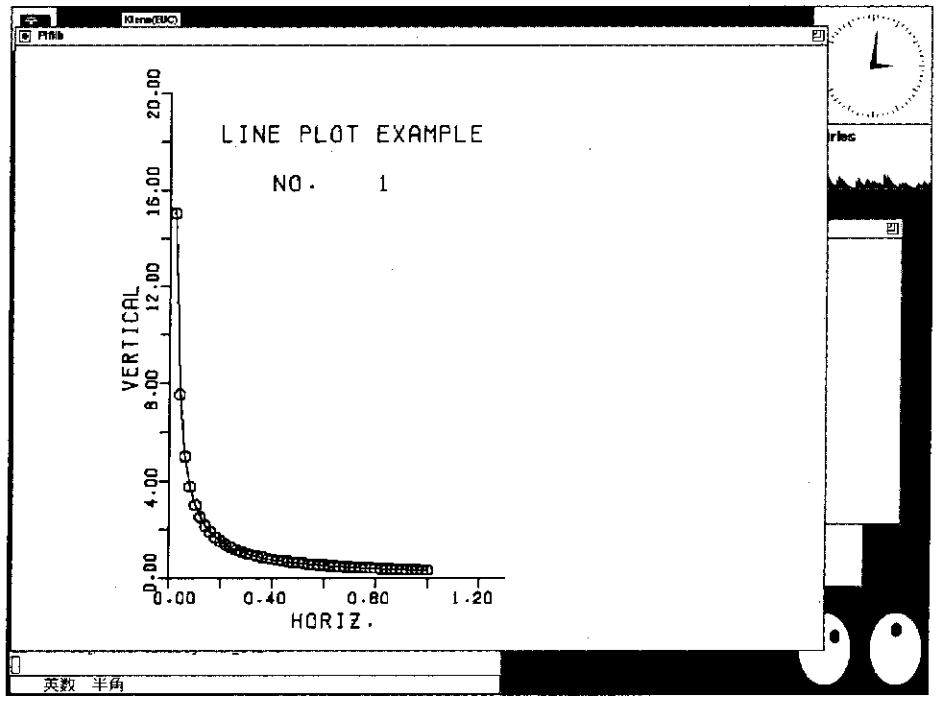

Fig. 2.5. Example Output on X Window

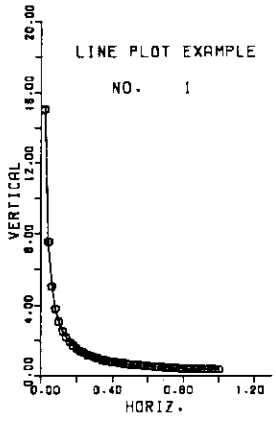

Fig. 2.6. Example Output to EPS File

2.3.4 今後の開発課題

カルコンプ仕様の基本ルーチンでは，フォントに関する指定はできないが，X や PostScript が持っているフォントを用いることにより美しい文字が得られる．今回のバージョンでは，ウィンドウ表示の時に文字フォントを設定することができるが，フォントを使用するルーチンが十分に整備されておらず，これを使用した場合にエラーとなることが多い．また，PostScript ファイル出力においてはフォント指定をサポートしていない．これらは今後の開発課題である．

バグの修正については，随時サポートしていく予定である．また，追加要求のあるルーチンについても対処していきたい．これらのバグレポート及び，必要なルーチンの追加等は，以下に示すメールアドレスにメールを送って頂きたい．

```
piflib-bug@sugar.tokai.jaeri.go.jp
```

このアドレスは，sugar（Sun4/2GX）上に設けられたメーリングアドレスであり，すべての piflib 開発者に自動的にメールが配送される．

3. 図形出力ツール xpif

大型計算機上で動作しているカルコンプ・ライブラリは，図形処理を行った後，その図形を NLP に出力せずに中間ファイルに出力することができる．今回の piflib 作成に合わせて，この中間ファイルの形式を改良し，大型計算機上のライブラリを整備し直すとともに，EWS 上で利用するためのユーティリティ・プログラムを開発した．

この章では，中間ファイルの内部仕様について述べるとともに，大型計算機と EWS を併用した図形処理の方法について述べる．

3.1 図形中間ファイル

大型計算機上の作画用の中間ファイル形式は既に仕様があるが，今回，大型計算機とワークステーションを連動した処理を行うに当たり，従来の中間ファイルのデータ構造を参考にいくつかの変更を行い，新たに中間ファイルの形式を定めた．また，文字コードの扱いについても統一をはかった．

3.1.1 図形中間ファイルの内容

大型計算機上の図形中間ファイルはレコード長 80 の順編成データセットである．各レコードは，先頭 1 バイトに命令部，その後にパラメータ部が続く．通常中間ファイルの第 1 レコードには，作画データの始まりを示す命令が，最終レコードには，作画データの終りを示す命令が格納される．また，各ページの区切りには，改ページを表す命令が格納される．

Table 3.1に，作画用中間ファイルで使用される命令と，対応する 16 進コードを示す．

2.3.4 今後の開発課題

カルコンプ仕様の基本ルーチンでは，フォントに関する指定はできないが，X や PostScript が持っているフォントを用いることにより美しい文字が得られる．今回のバージョンでは，ウィンドウ表示の時に文字フォントを設定することができるが，フォントを使用するルーチンが十分に整備されておらず，これを使用した場合にエラーとなることが多い．また，PostScript ファイル出力においてはフォント指定をサポートしていない．これらは今後の開発課題である．

バグの修正については，随時サポートしていく予定である．また，追加要求のあるルーチンについても対処していきたい．これらのバグレポート及び，必要なルーチンの追加等は，以下に示すメールアドレスにメールを送って頂きたい．

```
piflib-bug@sugar.tokai.jaeri.go.jp
```

このアドレスは，sugar（Sun4/2GX）上に設けられたメーリングアドレスであり，すべての piflib 開発者に自動的にメールが配送される．

3. 図形出力ツール xpif

大型計算機上で動作しているカルコンプ・ライブラリは，図形処理を行った後，その図形を NLP に出力せずに中間ファイルに出力することができる．今回の piflib 作成に合わせて，この中間ファイルの形式を改良し，大型計算機上のライブラリを整備し直すとともに，EWS 上で利用するためのユーティリティ・プログラムを開発した．

この章では，中間ファイルの内部仕様について述べるとともに，大型計算機と EWS を併用した図形処理の方法について述べる．

3.1 図形中間ファイル

大型計算機上の作画用の中間ファイル形式は既に仕様があるが，今回，大型計算機とワークステーションを連動した処理を行うに当たり，従来の中間ファイルのデータ構造を参考にいくつかの変更を行い，新たに中間ファイルの形式を定めた．また，文字コードの扱いについても統一をはかった．

3.1.1 図形中間ファイルの内容

大型計算機上の図形中間ファイルはレコード長 80 の順編成データセットである．各レコードは，先頭 1 バイトに命令部，その後にパラメータ部が続く．通常中間ファイルの第 1 レコードには，作画データの始まりを示す命令が，最終レコードには，作画データの終りを示す命令が格納される．また，各ページの区切りには，改ページを表す命令が格納される．

Table 3.1に，作画用中間ファイルで使用される命令と，対応する 16 進コードを示す．

Table 3.1. Instructions

命令部	16 進	パラメータ部
INIT	0x00	なし
FIN	0x01	なし
PLOT	0x02	X, Y, IP
SPLT	0x03	ICOUNT, dX_1, dY_1, dX_2, dY_2, ...
MPLT	0x04	ICOUNT, HX_1, LX_1, HY_1, LY_1, IP_1, ...
SYMB	0x05	X, Y, H, A, N, ICHAR1
SYMC	0x06	ICHAR2
CSYM	0x07	X, Y, H, ICODE, A, NC
NEWP	0x08	IPEN
FCTR	0x09	FACT
NEXT	0x0a	ITYPE
NOTE	0x0b	NOTED

3.1.2 データ構造の変更点

従来の中間ファイルでは，命令部分に 4 バイト使用していたが，各命令に対応する 16 進コードを設定することによって 1 バイトのデータに圧縮した．また，座標値については 0.01mm を単位とし，整数値として扱う．例えば，X 座標値が 174.38mm である場合，中間ファイル内では 17438 という整数値になる．これを 4 バイトの整数値として 16 進形式で出力する．各パラメータについても 1 バイト，もしくは 4 バイトの 16 進形式で出力する．

3.1.3 文字コードの統一について

文字コードについては，大型機上では EBCDIC コードを使用しているのに対して，ワークステーション上では ASCII コードが使用されている．従って，そのままでは中間ファイルを大型機から転送した際に文字が正しく表示されない．そこで，文字コードは ASCII コードに統一した．

これらの変更にともない，大型計算機から EWS への中間ファイルの転送はバイナリモードで行う．

3.1.4 命令の詳細

Table 3.1に示した命令の意味は，次の通りである．

(1) INIT (0x00)

作画データの始まりを示す．

(2) FIN (0x01)

作画データの最後を示す.

(3) PLOT (0x02)

現在のペン位置から，ペンアップ，または，ペンダウンで(X,Y)へペンを移動する.

(X,Y) : 移動先のX,Y座標の値. (0.01mm単位, 4バイト)
IP : = 2 ペンダウンで移動.
= 3 ペンアップで移動.
= −2 ペンダウンで移動後，原点リセット.
= −3 ペンアップで移動後，原点リセット.

ペンの情報は，4ビットのデータを使用して，対応するビットのON, OFFで表現する. ただし，扱いやすさを考慮して1バイトのデータを用意し，その下位4ビットを使用する (Fig. 3.1) .

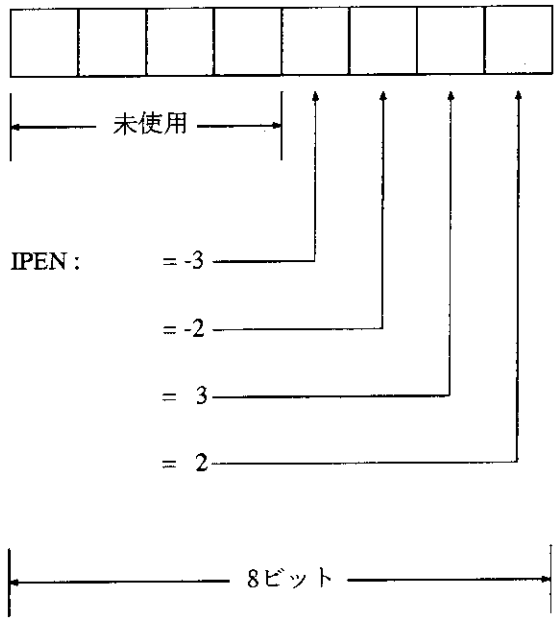

Fig. 3.1. Representation of Pen Status

(4) SPLT (0x03)

現在のペン位置に対して相対的に(dX,dY)だけペンを移動する. |dX|, |dY|は，-1.28mm ～ 1.27mmの小さな値の場合に使用され，それぞれ1バイトで表現する. ペンのアップ・ダウン情報は，以前に呼ばれたPLOT命令の値を継承する. このため，この命令が呼ばれる前に，必ずPLOT命令が実行されていなければならない.

ICOUNT: このレコード中にある (dX,dY) のペア数. (4バイト)

(5) MPLT (0x04)

現在のペン位置から，ペンアップ，または，ペンダウンで (X,Y) へペンを移動する. X,Y はそれぞれ，-327.67mm ～ 327.67mm までを 2 バイトで表現している.

IP : PLOT の IP に同じ. (4バイト)
ICOUNT : このレコード中にある (HX_i,LX_i,HY_i,LY_i ,IP_i) のペア数. (4バイト)

(6) SYMB (0x05)

指定の位置へ，大きさと角度を指定して文字列を描く.

X, Y : 文字列の書きだし位置の座標. (PLOT の X,Y に同じ)
H : 文字の大きさ (高さ).(単位:0.01mm)
A : 文字列の方向を，x 軸に対する反時計回りの角度で示す. (単位: 度)
N : 文字数
ICHAR1 : 描かれる文字列. (ASCII コード)

(7) SYMC (0x06)

SYMB 命令で文字列が溢れた場合，連続してこの命令が必要になる.

ICHAR2 : SYMB の ICHAR1，または先行する SYMC の ICHAR2 に引き続く文字列. (ASCII コード)

(8) CSYM (0x07)

指定の位置を中心に，大きさと角度を指定してセンタードシンボルを描く.

X, Y : シンボルの中心座標. (PLOT の X,Y に同じ)
H : シンボルの大きさ (高さ). (単位:0.01mm)
A : シンボルの X 軸に対する傾きを，反時計回りで示す. (単位: 度)
NC : -1 または -2 で，-1 なら (X,Y) までペンアップで，-2 なら (X,Y) までペンダウンで移動する.
ICODE : 描くべきセンタードシンボルに対応するコード番号を指定する.

(9) NEWP (0x08)

ペン種 (色，線の太さ，輝度レベルに対応) を変更する.

IPEN : 1 ～ 3 の整数を標準としてペン種を指定する.

(10) FCTR (0x09)

FCTR 命令以降に描くべき図形を拡大または縮小するためのスケーリングファクタを指定する.

FACT : 正の実数で, スケーリングファクタを示す.

(11) NEXT (0x0a)

データ作成時に CALL PLOT(x,y,ITYPE) の形で CALL した場合に作られる命令で, ITYPE の値によって, 次のような意味を持つ.

ITYPE : = 666 画面消去と原点リセット.
= 777 画面表示.
= 888 画面表示.

(12) NOTE (0x0b)

作成時に, 必要に応じて注釈文を挿入することができる. 実際の作図には影響を与えない.

NOTED : 任意の注釈文.

3.2 xpifの仕様

xpif は, 大型機で作成された図形中間ファイルを, EWS 上で処理するためのユーティリティ・プログラムである. piflib のインストールと同時に指定したディレクトリにインストールされる.

xpif は, 大型機で作成した図形の

(1) ウィンドウ表示,
(2) PostScript ファイルへの出力,
(3) EPS ファイルへの出力,

の 3 つの機能を持っている. これらは, 基本的には piflib と同等の機能である. 出力先の指定や, ファイル名の指定はプログラム起動時の引数で与える. コマンドの書式は以下の通りである.

```
% xpif [options] file1 [file2]
```

ここで, 各引数の意味は,

options : 出力先の指定する. 以下の 3 つのオプションのうち 1 つを選ぶ.
-x ウィンドウへの図形表示. (デフォルト)
-p PostScript ファイルへの出力
-e EPS ファイルへの出力
file1 : 図形中間ファイル
file2 : 出力ファイル名 (-p, または -e オプションを選択した時のみ必要)

である．EPS ファイルは，1 画面につき 1 ファイルを必要とするため，出力ファイル名に対してページ番号を自動的に付加する．また，出力ファイル名を入力しない場合はデフォルトのファイル名 (piflib.PS，または piflibnn.EPS ， nn はページ番号) となる．

3.3 使用方法

ここでは，大型計算機上での中間ファイルの作成方法，EWS への中間ファイルの転送方法，xpif の使用方法，大型計算機上での中間ファイルの図形出力方法について述べる．

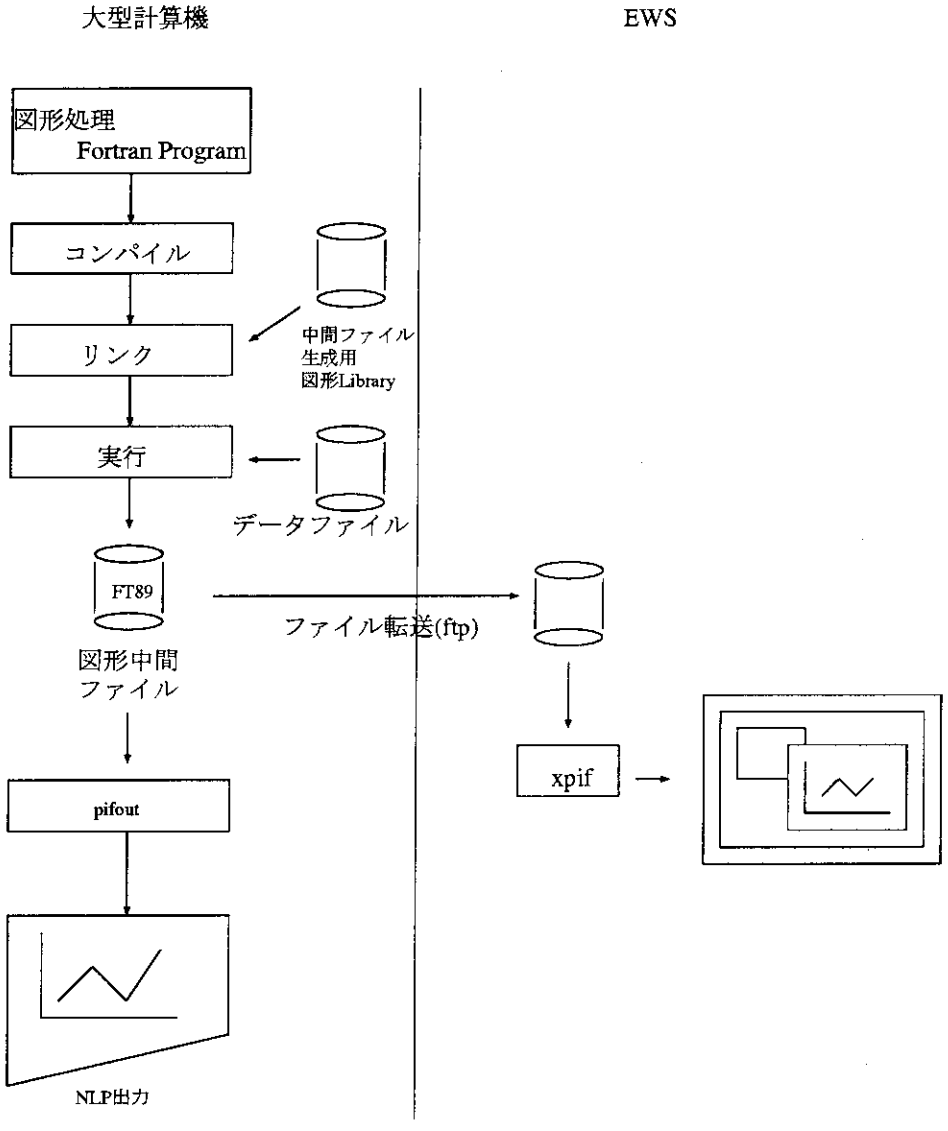

Fig. 3.2. Graphic Processing with Common Graphic Data File

中間ファイルを利用した場合の，大型計算機と EWS を併用した図形処理の基本的な手順は，以下のとおりである．

(1) 大型機上で図形処理プログラムを書く．

(2) 図形中間ファイルを作成する.
(3) EWS に図形中間ファイルを転送する.
(4) xpif を利用して EWS 上で図形表示を行う.
(5) 大型計算機上での中間ファイルの図形出力を行う.

ここで, (1), (2) は大型計算機上で, (3), (4) は EWS 上で行う作業である. また, (5) は大型計算機で図形を出力する場合の作業である.

3.3.1 図形中間ファイルの作成

大型計算機側で図形中間ファイルを作成するためには, 次の手順どおり行えば良い.

(1) まず, 大型計算機上でカルコンプ・ライブラリを利用した図形処理プログラムを作成する. 既存のプログラムも利用できる.
(2) プログラムを実行し, 結果を図形中間ファイルに出力する. 中間ファイルに出力するためのライブラリは, 'SYS9.PIFLIB.LOAD' に登録されている.
(3) 作成した図形中間ファイルは, 大型計算機上の図形出力装置へ図形を出力することもできる.

以下に, 中間ファイルを作成するための JCL 例を示す. 図形処理のための Fortran プログラムは, Jnnnn.WSGRDATA.FORT77(TEST11), 出力先の図形中間ファイルは Jnnnn.@TESTSYM.DATA とする.

```
  TWCEI(3 3 5 0 5) SRP
//FORT EXEC FORTEX,SO='Jnnnn.WSGRDATA',A='ELM(TEST11)'
//LINK EXEC LKEDEX
//SYSLIB DD DSN=SYS9.GGS.LOAD,DISP=SHR
// DD DSN=SYS9.PIFLIB.LOAD,DISP=SHR
//RUN EXEC GOEX
//FT89F001 DD DSN=Jnnnn.@TESTSYM.DATA,UNIT=TSSWK,SPACE=(TRK,(10,10)),
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=3120),DISP=(NEW,CATLG)
```

大型計算機上に作成された中間ファイルは大型計算機に接続されている図形出力装置へ出力することができる. 入力制御データは, SYSIN (FT05F001) から 1 行で与え, 図形中間ファイルは FT89F001 から与える. 入力制御データはフリーフォーマット形式で以下のように入力する.

```
FACTOR,IPAGES,IPAGEE,IUNIT,IREC/
```

FACTOR : 出力図形のスケールファクター (標準値 1.00)
IPAGES : 出力すべき最初のページ (標準値 1)
IPAGEE : 出力すべき最後のページ (標準値 999)
IUNIT : 中間ファイルの入力機番 (標準値 89)
IREC : 中間ファイルのレコード長 (標準値 80)

該当する項目についての入力が0の場合(又は無い場合)は標準値を使う．以下に，それぞれの図形出力装置へ図形を出力する場合のJCL例を示す．

(1) NLPへの出力JCL例

```
  TWCEI(1 1 4 0 5) SRP GRP
//GO EXEC LMGOEX,LM='SYS9.PIFOUT',PNM=NLP
//FT89F001 DD DSN=Jnnnn.@TESTSYM.DATA,DISP=SHR
//GRP EXPAND GRNLP
//SYSIN DD *
1.00,1,999,89,80/
/*
```

(2) OPR，Kシリーズ等プリンタへの出力JCL例

```
TWCEI(1 1 4 0 5) SRP GRP
//GO EXEC LMGOEX,LM='SYS9.PIFOUT',PNM=OPR
//FT89F001 DD DSN=J9136.@TESTSYM.DATA,DISP=SHR
//FT30F001 DD SYSOUT=* 2cm・・・図形出力用
//GRP EXPAND GRNLP 3cm・・・漢字パターンファイル使用時
//SYSIN DD *
1.00,1,999,89,80/
/*
```

(3) F6683等グラフィック端末へのTSSによる図形出力例

```
ALLOC F(FT89F001) DA(@TESTSYM.DATA) SHR
CALL 'SYS9.PIFOUT.LOAD(GSP)' 'GSP(NOBUF)'
nnnnnn?
1.00,1,999,89,80/
/*
```

3.4 大型計算機との協調処理

ここでは，大型計算機で作成された図形中間ファイルを，EWS上で表示させる方法について説明する．

(1) EWSに中間ファイルを転送する．
EWSからftpコマンドを利用して，大型計算機にログインする．中間ファイルはバイナリ形式であるため，バイナリモードにして転送をする．

```
get 'データセット名' ['ファイル名']
```

このとき，大型計算機上のデータセットがPOファイルの場合は，メンバ名まで指定する．すべてのメンバをgetしたい場合は，POファイルを疑似的なディレクトリ名とみなし，cdコマンドでPOファイル名を指定する．その後，mgetコマンドですべてのメンバがgetできる．

(2) xpifを利用して処理を行う．
EWS上で中間ファイルを読み込み，各出力先に応じた処理を行う．

以下に，ftpコマンドでファイル転送を行うときの例を示す．転送すべき大型計算機上の図形中間ファイルJnnnn.WSGRDATA.DATAはPOファイルとし，その中に3つのメンバが含まれているとする．また，'133.53.11.1'は東海研究所Fujitsu M780/10SのIPアドレスであり，下線部分がユーザの入力である．

```
furies[/home/tanabe/tmp/piflib] % ftp 133.53.11.1
Connected to 133.53.11.1.
220 Service ready for new user
Name (133.53.11.1:tanabe):  Jnnnn <-- User ID
331 User name okay, need password
Password:
230 User logged in, proceed
Remote system type is OSIV:FTP.
ftp> binary                       <-- バイナリ・モードの設定
200 Command okay
ftp> cd wsgrdata.data             <-- データセット名(POファイルの場合)
200 Command okay
ftp> mget *                       <-- 全メンバをget
mget SCIEN? y
200 Command okay
150 File status okay;about to open data connection
226 File transfer complete
73600 bytes received in 0.32 seconds (228.13 Kbytes/s)
mget TESTSYM? y
200 Command okay
150 File status okay;about to open data connection
226 File transfer complete
29600 bytes received in 0.10 seconds (293.16 Kbytes/s)
mget TEST11?  y
200 Command okay
150 File status okay;about to open data connection
226 File transfer complete
51440 bytes received in 0.13 seconds (388.39 Kbytes/s)
```

```
ftp> quit
221 Service closing CONTROL connection
furies[/home/tanabe/tmp/piflib] %
```

4. おわりに

本報告では，情報システムセンターで開発したX Window System上のカルコンプ・インターフェース・ライブラリpiflib Ver.1.0について述べた．本ライブラリを用いれば，従来大型計算機で行われていた図形処理をEWS上で行うことができる．しかし，カルコンプ仕様の基本ルーチンの整備だけでは，いくつか不十分な点がある．

- カルコンプの基本ルーチンには含まれないが，フォントに関するルーチンが十分に整備できていない．X Window System上のフォントは，EWSの各メーカーによって様々な種類をもっており，フォント名を統一的に扱うことが困難である．piflibでは，フォントの指定は，フォント名のすべてを入力しなければならず，使い勝手を低下させる原因となっている．

- プリンタへの出力においては，大型機に接続されているNLP (日本語ラインプリンタ) と比べて解像度が荒くなってしまう．これは，PostScriptファイル出力時の誤差による影響と，PostScriptにおける1ピクセルの大きさによるものと思われる．特に，小さい文字を描いている図形等の場合には，文字がつぶれて表示されるので大型機と同等の図形が得られない．

- PLOTルーチンによって描画を行う際，ストローク回数が多い場合は，PostScriptインタープリタが実行時にエラーを起こすことがある．これは，PostScriptの仕様によるもので，描画を行う際のストローク数がが多くなり過ぎた場合に，使用できるスタックの領域がなくなってしまうことにより発生する．この場合には，今のところプログラムを修正して対処する以外に回避する方法はない．

次のバージョンでは，これらの項目についての整備を行い，使いやすさを向上させていきたいと思う．

また，piflibのバグ修正や，改良等に関する情報は電子メール等を用いて行うことを考えている．このような方法はUNIXの世界ではごくあたり前に行われており，原研におけるUNIX文化の浸透につながれば幸いである．

```
ftp> quit
221 Service closing CONTROL connection
furies[/home/tanabe/tmp/piflib] %
```

4. おわりに

本報告では，情報システムセンターで開発したX Window System上のカルコンプ・インターフェース・ライブラリ piflib Ver.1.0について述べた．本ライブラリを用いれば，従来大型計算機で行われていた図形処理をEWS上で行うことができる．しかし，カルコンプ仕様の基本ルーチンの整備だけでは，いくつか不十分な点がある．

- カルコンプの基本ルーチンには含まれないが，フォントに関するルーチンが十分に整備できていない．X Window System上のフォントは，EWSの各メーカーによって様々な種類をもっており，フォント名を統一的に扱うことが困難である．piflibでは，フォントの指定は，フォント名のすべてを入力しなければならず，使い勝手を低下させる原因となっている．

- プリンタへの出力においては，大型機に接続されているNLP (日本語ラインプリンタ) と比べて解像度が荒くなってしまう．これは，PostScriptファイル出力時の誤差による影響と，PostScriptにおける1ピクセルの大きさによるものと思われる．特に，小さい文字を描いている図形等の場合には，文字がつぶれて表示されるので大型機と同等の図形が得られない．

- PLOTルーチンによって描画を行う際，ストローク回数が多い場合は，PostScriptインタープリタが実行時にエラーを起こすことがある．これは，PostScriptの仕様によるもので，描画を行う際のストローク数がが多くなり過ぎた場合に，使用できるスタックの領域がなくなってしまうことにより発生する．この場合には，今のところプログラムを修正して対処する以外に回避する方法はない．

次のバージョンでは，これらの項目についての整備を行い，使いやすさを向上させていきたいと思う．

また，piflibのバグ修正や，改良等に関する情報は電子メール等を用いて行うことを考えている．このような方法はUNIXの世界ではごくあたり前に行われており，原研におけるUNIX文化の浸透につながれば幸いである．

謝　辞

本報告書を執筆する機会を与えて下さいました情報システムセンター室長秋元正幸氏に深く感謝致します．また，UNIX及び，X Window Systemについて貴重なご意見，ご指導を賜わりました外来研究員(富士通)の皆様に深く感謝致します．

参考文献

1) Adrian Nye, Xlib Programming Manual, O'Reilly&Associates,Inc (1988)

2) 木下凌一, X-Window Ver.11 プログラミング, 日刊工業新聞社 (1989)

3) Adobe Systems, PostScript リファレンス・マニュアル, アスキー出版局 (1988)

4) Adobe Systems, PostScript チュートリアル & クックブック, アスキー出版局 (1989)

5) Adobe Systems, PostScript プログラム・デザイン, アスキー出版局 (1990)

6) 野寺隆志, 楽々 LaTeX, 共立出版 (1990).

7) 情報システムセンター, カラー・グラフィック・サブルーチンライブラリ (CGSL) 利用の手引, 情報システムセンター (1985)

謝　辞

本報告書を執筆する機会を与えて下さいました情報システムセンター室長秋元正幸氏に深く感謝致します．また，UNIX 及び，X Window System について貴重なご意見，ご指導を賜わりました外来研究員（富士通）の皆様に深く感謝致します．

参考文献

1) Adrian Nye, Xlib Programming Manual, O'Reilly&Associates,Inc (1988)

2) 木下凌一，X-Window Ver.11 プログラミング，日刊工業新聞社 (1989)

3) Adobe Systems, PostScript リファレンス・マニュアル，アスキー出版局 (1988)

4) Adobe Systems, PostScript チュートリアル & クックブック，アスキー出版局 (1989)

5) Adobe Systems, PostScript プログラム・デザイン，アスキー出版局 (1990)

6) 野寺隆志，楽々 LaTeX，共立出版 (1990).

7) 情報システムセンター，カラー・グラフィック・サブルーチンライブラリ (CGSL) 利用の手引，情報システムセンター (1985)

付録 A piflib の詳細仕様

A.1 ウィンドウ描画ルーチン群

(1) xplots

機能： サーバとの接続を確立し，ウィンドウの生成，表示，属性の初期化を行なう．また，座標原点を，ウィンドウの左下隅から上に 20.0mm，左に 20.0mm に設定する．

呼び出し形式：
```
void xplots(dummy, idummy);
char dummy;
int *idummy;
```

引数： どちらもダミー・パラメータ

(2) xplot

機能： ペンを現在位置から指定された座標まで，直線的に移動する．

呼び出し形式：
```
void xplot(x, y, ipen);
float *x;
float *y;
int   *ipen;
```

引数：
x, y : ペンを移動させる終点座標値（単位:mm）
ipen : ペンのアップ・ダウン及び，画面制御情報
- = 2 : ペンダウンで指定座標まで移動.
- = 3 : ペンアップで指定座標まで移動.
- = -2 : ペンダウンで指定座標まで移動後，その座標を原点とする.
- = -3 : ペンアップで指定座標まで移動後，その座標を原点とする.
- = 666 : 改ページ，原点のリセット及び，属性の初期化を行なう.
- = 777 : ウィンドウへの図形の表示を行なう.
- = 888 : ウィンドウへの図形の表示を行なう.
- = 999 : 処理の終了.

なお，ペンダウンで移動（すなわち，描画を行なう）の際は，図形はすべてピックスマップ上に描かれる．

(3) xsymbol

機能1： 指定された座標に，アルファベット，数字及び，特殊記号を描く．

呼び出し形式：
```
void xsymbol(x, y, h, string, th, n);
float *x;
float *y;
float *h;
float th;
char  *string;
int   *n;
```

引数：

x, y ： 最初に描かれる文字の左下隅の座標値を指定する．(単位:mm)

h ： 文字を描く高さを指定する．（単位:mm）

string ： 描きたい文字列．

th ： 描きたい文字列の，X軸とのなす角度．（単位: 度）

n ： 文字数．（ N > 0 ）

ここで，x及び，yを999.0と指定すると，前回コールされたxsymbolルーチンで描かれた文字列の後ろに連続して描かれる．

機能2： 指定した座標を中心として，センター・シンボルを描く．

呼び出し形式：
```
void xsymbol(x, y, h, item, th, n);
float *x;
float *y;
float *h;
float *th;
int   *item;
int   *n;
```

引数：

x, y ： センター・シンボルの中心座標．（単位:mm）

h ： 描きたいシンボルの高さ．（単位:mm）

item ： センター・シンボルのコード番号．（ 0 ～ 15 ）

th ： 描きたいシンボルのX軸とのなす角度（単位: 度）

n ： = -1 と指定する．

(4) xfactor

機能： 図形全体の拡大，縮小を指定する．

呼び出し形式：
```
void xfactor(fctr);
float *fctr;
```

引数： fctr ： 尺度因子を指定する.

注) 一度，このルーチンの呼び出しを行なうと，次の変更があるまで，今回の指定値が有効となる．このルーチンを使用しない場合は，fctr の値は 1.0 になる.

(5) xnewpen

機能： ペン種を変更する.

呼び出し形式：
```
void xnewpen(ipen);
int *ipen;
```

引数： ipen ： 1～3 の整数を標準としてペン種を指定する.

	1	2	3
ペン種	黒	赤	青

(6) xwhere

機能： 現在のペン位置の座標と，尺度因子を返す.

呼び出し形式：
```
void xwhere(x, y, fctr);
float *x;
float *y;
float *fctr;
```

引数： x, y ： ペンの現在位置の X, Y 座標をストアさせる.
fctr ： 現在の尺度因子をストアさせる.

いずれも，実数型変数名を指定する.

(7) xsetltp

機能： このルーチンの呼び出し以降，描かれる線の線種を変更する.

呼び出し形式：
```
void xsetltp(ltp);
int *ltp;
```

引数： ltp ： 線種インデックス番号. (0～11)

標準値は 0(実線).

(8) xsetlwd

機能： このルーチンの呼び出し以降，描かれる線の線幅を変更する.

呼び出し形式：
```
void xsetlwd(lwd);
int *lwd;
```

引数： lwd ： 線幅のインデックス番号. (1～5)

(9) xsetclr

機能： このルーチンの呼び出し以降，描かれる線の色を指定する．

呼び出し形式：
```
void xsetclr(iclr);
int *iclr;
```

引数： iclr ： カラーインデックス番号．（0～7）

Index	0	1	2	3
表示色	黒	白	赤	緑
Index	4	5	6	7
表示色	青	シアン	マゼンタ	黄

(10) xsetrgb

機能： このルーチンの呼び出し以降，描かれる線の色を RGB 値で指定する．

呼び出し形式：
```
xsetrgb(red, green, blue);
int *red;
int *green;
int *blue;
```

引数：
```
red   : 赤色の成分値．（0～255）
green : 緑色の成分値．（0～255）
blue  : 青色の成分値．（0～255）
```

(11) xsymfnt

機能： 指定した座標から，フォントを利用して文字列を描く．

呼び出し形式：
```
void xsynfnt(x, y, h, string, th, n);
float *x;
float *y;
float *h;
float *th;
char  *string;
int   *n;
```

引数：
```
  x, y : 最初に描かれる文字の，左下隅の座標値．（単位:mm）
     h : 現在ダミーパラメータ．
string : 描きたい文字列．
    th : 現在ダミーパラメータ．
     n : 文字数．
```

(12) xdot

機能： 指定した座標にドットを描く.

呼び出し形式：
```
void xdot(x, y);
float *x;
float *y;
```

引数： x, y ： ドットを描く座標値. （単位:mm）

(13) xcpfont

機能： このルーチンの呼び出し以降，使用するフォントを指定する.

呼び出し形式：
```
void xcpfont(font_name);
char *font_name;
```

引数： font_name ： 利用するフォント名. （X で利用可能なもの）

注）指定したフォントが存在しない場合，エラーとなる.

A.2 PostScript ファイル出力ルーチン群

(1) psplots

機能： 出力ファイル名の入力を要求し，出力ファイルをオープンする.

呼び出し形式：
```
void psplots(dummy, idummy);
char dummy;
int *idummy;
```

引数： どちらもダミー・パラメータ

(2) psplot

機能： 指定された座標値と，ペンの情報を PostScript の命令に直し，出力ファイルに書き出す.

呼び出し形式：
```
void psplot(x, y, ipen);
float *x;
float *y;
int   *ipen;
```

引数：
x, y : ペンを移動させる終点座標値（単位:mm）
ipen : ペンのアップ・ダウン及び，画面制御情報
- = 2 : ペンダウンで指定座標まで移動.
- = 3 : ペンアップで指定座標まで移動.
- = -2 : ペンダウンで指定座標まで移動後，その座標を原点とする.
- = -3 : ペンアップで指定座標まで移動後，その座標を原点とする.
- = 666 : 改ページ，原点のリセット及び，属性の初期化を行なう.
- = 777 : NOP
- = 888 : NOP
- = 999 : 処理の終了.

なお，この関数がページの最初に呼ばれた場合，PostScript ファイルのヘッダ部分を先に出力する．また，ipen の値が 666，または，999 である場合は，ページの終了を表す命令が出力される.

(3) pssymbol

機能1： 指定された座標に，アルファベット，数字及び，特殊記号を描く．

呼び出し形式：
```
void pssymbol(x, y, h, string, th, n);
float *x;
float *y;
float *h;
float th;
char  *string;
int   *n;
```

引数：

x, y ： 最初に描かれる文字の左下隅の座標値を指定する．（単位:mm）

h ： 文字を描く高さを指定する．（単位:mm）

string ： 描きたい文字列．

th ： 描きたい文字列の，X軸とのなす角度．（単位: 度）

n ： 文字数．（ N > 0 ）

ここで，x及び，yを999.0と指定すると，前回コールされたxsymbolルーチンで描かれた文字列の後ろに連続して描かれる．

機能2： 指定した座標を中心として，センター・シンボルを描く．

呼び出し形式：
```
void pssymbol(x, y, h, item, th, n);
float *x;
float *y;
float *h;
float *th;
int   *item;
int   *n;
```

引数：

x, y ： センター・シンボルの中心座標．（単位:mm）

h ： 描きたいシンボルの高さ．（単位:mm）

item ： センター・シンボルのコード番号．（ 0 ～ 15 ）

th ： 描きたいシンボルのX軸とのなす角度（単位: 度）

n ： = -1 と指定する．

(4) psfactor

機能： 図形全体の拡大，縮小を指定する．

呼び出し形式：
```
void psfactor(fctr);
float *fctr;
```

引数： fctr ： 尺度因子を指定する.

注) 一度，このルーチンの呼び出しを行なうと，次の変更があるまで，今回の指定値が有効となる．このルーチンを使用しない場合は，fctr の値は 1.0 になる.

(5) psnewpen

機能： ペン種を変更する.

呼び出し形式：
```
void psnewpen(ipen);
int *ipen;
```

引数： ipen ： 1～3の整数を標準としてペン種を指定する.

	1	2	3
ペン種	細線	中線	太線

(6) pssetltp

機能： このルーチンの呼び出し以降，描かれる線の線種を変更する.

呼び出し形式：
```
void pssetltp(ltp);
int *ltp;
```

引数： ltp ： 線種インデックス番号. (0～11)

標準値は 0(実線).

(7) pssetlwd

機能： このルーチンの呼び出し以降，描かれる線の線幅を変更する.

呼び出し形式：
```
void pssetlwd(lwd);
int *lwd;
```

引数： lwd ： 線幅のインデックス番号. (1～5)

(8) pssetclr

機能： このルーチンの呼び出し以降，描かれる線の色を指定する.

呼び出し形式：
```
void pssetclr(iclr);
int *iclr;
```

引数： iclr ： カラーインデックス番号. (0～7)

Index	0	1	2	3
表示色	黒	白	赤	緑
Index	4	5	6	7
表示色	青	シアン	マゼンタ	黄

(9) pssetrgb

機能： このルーチンの呼び出し以降，描かれる線の色をRGB値で指定する.

呼び出し形式：
```
void pssetrgb(red, green, blue);
int *red;
int *green;
int *blue;
```

引数：
- red : 赤色の成分値. (0～255)
- green : 緑色の成分値. (0～255)
- blue : 青色の成分値. (0～255)

A.3 EPS ファイル出力ルーチン群

(1) texplots

機能： 出力ファイル名の入力を要求し，出力ファイルをオープンする.

呼び出し形式：
```
void texplots(dummy, idummy);
char dummy;
int *idummy;
```

引数： どちらもダミー・パラメータ

(2) texplot

機能： 指定された座標値と，ペンの情報を PostScript の命令に直し，出力ファイルに書き出す.

呼び出し形式：
```
void texplot(x, y, ipen);
float *x;
float *y;
int   *ipen;
```

引数：
x, y : ペンを移動させる終点座標値（単位:mm）
ipen : ペンのアップ・ダウン及び，画面制御情報
= 2 : ペンダウンで指定座標まで移動.
= 3 : ペンアップで指定座標まで移動.
= -2 : ペンダウンで指定座標まで移動後，その座標を原点とする.
= -3 : ペンアップで指定座標まで移動後，その座標を原点とする.
= 666 : 改ページ，原点のリセット及び，属性の初期化を行なう.
= 777 : NOP
= 888 : NOP
= 999 : 処理の終了.

なお，この関数がページの最初に呼ばれた場合，PostScript ファイルのヘッダ部分を先に出力する．また，ipen の値が 666，または，999 である場合は，ページの終了を表す命令が出力される.

(3) texsymbol

機能： 指定された座標に，アルファベット，数字及び，特殊記号を描く．

呼び出し形式：
```
void texsymbol(x, y, h, string, th, n);
float *x;
float *y;
float *h;
float th;
char  *string;
int   *n;
```

引数：

x，y ： 最初に描かれる文字の左下隅の座標値を指定する．(単位:mm)

h ： 文字を描く高さを指定する．（単位:mm）

string ： 描きたい文字列．

th ： 描きたい文字列の，X軸とのなす角度．（単位：度）

n ： 文字数．（ N > 0 ）

ここで，x及び，yを999.0と指定すると，前回コールされたxsymbolルーチンで描かれた文字列の後ろに連続して描かれる．

機能： 指定した座標を中心として，センター・シンボルを描く．

呼び出し形式：
```
void texsymbol(x, y, h, item, th, n);
float *x;
float *y;
float *h;
float *th;
int   *item;
int   *n;
```

引数：

x，y ： センター・シンボルの中心座標．（単位:mm）

h ： 描きたいシンボルの高さ．（単位:mm）

item ： センター・シンボルのコード番号．（ 0 ～ 15 ）

th ： 描きたいシンボルのX軸とのなす角度（単位：度）

n ： = -1 と指定する．

付録 B xpifの詳細仕様

(1) main

機能： xpifのメインルーチン．コマンドライン引数の解釈，及び入出力ファイルのオープン・クローズを行なう．

呼びだし形式：
```
void main(argc, argv)
int argc;
char *argv[];
```

引数： argc：コマンドライン引数の数（自身も含む）．
argv：引数の文字列．

(2) psout

機能： 図形中間ファイルから命令，及びデータを読み込んで，PostScript形式でファイルに出力する．

呼びだし形式：
```
void psout();
```

引数： なし．

(3) epsout

機能： 図形中間ファイルから命令，及びデータを読み込んで，EPSF形式でファイルに出力する．

呼びだし形式：
```
void epsout();
```

引数： なし．

(4) xwinout

機能： 図形中間ファイルから命令，及びデータを読み込んで，ウィンドウに図形を描画する．

呼びだし形式：
```
void xwinput();
```

引数： なし．