

GKS入門(4)

橋倉, 聡

九州大学大型計算機センターシステム管理掛

浦川, 伸治

九州大学大型計算機センターシステム運用掛

藤井, 龍磨

九州大学大型計算機センターシステム管理掛

河津, 秀利

九州大学大型計算機センターシステム管理掛

<https://doi.org/10.15017/1470148>

出版情報 : 九州大学大型計算機センター広報. 23 (4), pp.295-322, 1990-07-25. 九州大学大型計算機センター

バージョン :

権利関係 :

G K S 入門 (4)

橋倉 聡*, 浦川 伸治**, 藤井 龍磨*, 河津 秀利*

1. はじめに

前回までは、G K S の基本的な使用法について説明してきた。通常の使用には前回までで十分であると思われるが、G K S の水準が 0 b から 2 b へアップし、さらに多くの機能が実現できるようになった。今回は K S T (ストローク) フォントの使用法、ワークステーション (G K S で用いる図形装置) の複数動作、セグメント機能について説明する。

2. K S T (ストローク) フォントの使用法

G K S 8 5 の文字のフォントには C G 文字とストロークがあった。この 2 つのフォントは、ワークステーションの能力に依存するため、これまでは C G 文字のみが使用可能であった。今回のレベルアップでサポートされた K S T は、ワークステーションに依存しないストローク文字である。K S T を使用するには、以前説明した文字のフォント、精度を指定するサブルーチン G S T X F P を使用する。

```
CALL GSTXFP(-100,2)
```

上の例のように、最初の引数 (フォント) に -100 (K S T) を指定する。2 番目の引数 (文字列表示精度) は 0 (文字列精度), 1 (文字精度), 2 (図形精度) のいずれを指定しても図形精度として扱われる。K S T は日本語の場合にも有効である。

2. 1 K S T を用いたプログラム例

K S T を用いたプログラム例を以下に示す。グラフの目盛り数値には K S T を、タイトルには C G 文字を使用しており、文字の高さ、文字幅比はどちらも同じにしてある。このプログラムの出力結果を図 2. 1 に示す。C G 文字 (タイトル) は、これ以上小さくならないが、K S T ならば、さらに小さくすることも可能である。

(プログラム例)

```
0001      PROGRAM STROKE
0002 C
0003      INTEGER STAT, CHNR
0004      REAL X(91), Y(91), FX(4), FY(4), UP(2), LO(2), LE(2), RI(2), CX(2), CY(2),
0005 +      YMIN, YMAX, XLEN, YLEN, XSTEP, YSTEP, RAD, T, CHH, CHXP
0006      CHARACTER*7 XCHAR, YCHAR
0007 C
0008      PARAMETER (RAD=3.1415/180.0, T=240.0)
0009 C
0010      DO 1 I=1, 91
```

平成 2 年 5 月 2 5 日受理

* 九州大学大型計算機センター システム管理掛
 ** " システム運用掛

解 説

```

0011      X(I) = -360.0+FLOAT(I-1)*12.0
0012      Y(I) = X(I)*COS(X(I)*RAD)/T*SIN(T*RAD)
0013      IF (I.EQ.1) THEN
0014          YMIN = Y(I)
0015          YMAX = Y(I)
0016      ELSE
0017          IF (Y(I).LT.YMIN) YMIN = Y(I)
0018          IF (Y(I).GT.YMAX) YMAX = Y(I)
0019      ENDIF
0020      1 CONTINUE
0021 C
0022      FX(1) = X(1)
0023      FX(2) = X(91)
0024      FX(3) = FX(2)
0025      FX(4) = FX(1)
0026      FY(1) = YMIN-(YMAX-YMIN)/10.0
0027      FY(2) = FY(1)
0028      FY(3) = YMAX+(YMAX-YMIN)/10.0
0029      FY(4) = FY(3)
0030 C
0031      XLEN = FX(2)-FX(1)
0032      YLEN = FY(3)-FY(1)
0033      XSTEP = XLEN/10.0
0034      YSTEP = YLEN/10.0
0035      UP(1) = FY(3)
0036      UP(2) = UP(1)-YLEN/100.0
0037      LO(1) = FY(1)
0038      LO(2) = LO(1)+YLEN/100.0
0039      LE(1) = FX(1)
0040      LE(2) = LE(1)+XLEN/100.0
0041      RI(1) = FX(2)
0042      RI(2) = RI(1)-XLEN/100.0
0043 C
0044      CALL GOPKS (0,99)
0045      CALL GOPWK (1,1,1)
0046      CALL GACWK (1)
0047 C
0048      CALL GSWN (1,FX(1)-XSTEP,FX(2)+XSTEP,FY(1)-YSTEP,FY(3)+YSTEP)
0049      CALL GSVP (1,0.0,1.0,0.0,1.0)
0050      CALL GSELNT (1)
0051 C
0052      CALL GFA (4,FX,FY)
0053 C

```

```

0054 CALL GSTXFP (-100,2) ..... 文字のフォントに -100 (K S T) を指定する
0055 CHH = YLEN/75.0
0056 CALL GSCHH (CHH) ..... 文字の高さの設定
0057 CHXP = (XLEN/75.0)/CHH
0058 CALL GSCHXP (CHXP) ..... 文字幅比の設定
0059 C
0060 DO 2 I=0,10
0061   CX(1) = FX(1)+FLOAT(I)*XSTEP
0062   CX(2) = CX(1)
0063   CY(1) = FY(1)+FLOAT(I)*YSTEP
0064   CY(2) = CY(1)
0065   CALL GPL (2,CX,UP) ..... 上の X 軸の目盛りを作画
0066   CALL GPL (2,CX,LO) ..... 下の X 軸の目盛りを作画
0067   CALL GPL (2,LE,CY) ..... 左の Y 軸の目盛りを作画
0068   CALL GPL (2,RI,CY) ..... 右の Y 軸の目盛りを作画
0069   WRITE (XCHAR,'(F7.2)') CX(1)
0070   WRITE (YCHAR,'(F7.2)') CY(1)
0071   CALL GSTXAL (2,1) ..... 文字列配置 (X 軸の目盛り数値用) の設定
0072   CALL GTX (CX(1),LO(1)-YLEN/100.0,XCHAR) ..... X 軸の目盛り数値の作画
0073   CALL GSTXAL (3,3) ..... 文字列配置 (Y 軸の目盛り数値用) の設定
0074   CALL GTX (LE(1)-XLEN/100.0,CY(1),YCHAR) ..... Y 軸の目盛り数値の作画
0075 2 CONTINUE
0076 C
0077 CALL GSTXFP (-3,2) ..... 文字のフォントに -3 (C G 文字) を指定する
0078 CALL GSTXAL (2,5) ..... 文字列配置 (タイトル用) の設定
0079 CALL GSCHSP (XLEN/YLEN) ..... 文字と文字の間隔を設定
0080 CALL GTX (FX(1)+XLEN/2.0,FY(3)+YSTEP/3.0,'SAMPLE GRAPH') ..... タイトルを
0081 C                                     作画
0082 CALL GPL (91,X,Y)
0083 C
0084 CALL GRQCH (1,1,STAT,CHNR)
0085 CALL GDAWK (1)
0086 CALL GCLWK (1)
0087 CALL GCLKS
0088 C
0089 STOP
0090 END

```

(プログラムの説明)

0003 ~ 0008 必要な変数, 配列の宣言と定数の設定。
0010 ~ 0020 座標値の計算を行い, Y 座標値の最小値および最大値を求める。
0022 ~ 0029 曲線を囲む枠 (座標軸) の座標値を設定する。曲線が上下の X 軸と重ならないように余裕をとっている。

解 説

- 0031 ~ 0042 目盛りおよび目盛り数値のための計算.
- 0044 ~ 0046 G K Sを開始し, ワークステーションを出力可能状態にする.
- 0048 ~ 0050 座標変換. 図形より大きめにウィンドウを設定している.
- 0052 座標軸の作画.
- 0054 ~ 0058 文字のフォントにはK S Tを指定し, 文字の高さはY軸の1 / 7 5に設定, 文字幅比はX軸とY軸との比と同じにする.
- 0065 ~ 0068 X軸およびY軸の目盛りを作画.
- 0071 ~ 0072 文字列配置を 水平位置: 中間, 垂直位置: 上限 に設定して, X軸の目盛り数値を出力させている.
- 0073 ~ 0074 文字列配置を 水平位置: 右端, 垂直位置: 中間 に設定して, Y軸の目盛り数値を出力させている.
- 0077 ~ 0080 フォントをK S TからC G文字に変更し, 文字列配置を 水平位置: 中間, 垂直位置: 下限 に設定し, 字間をX軸の長さ/Y軸の長さに設定してタイトルを出力させている.
- 0082 曲線の作画.
- 0084 ~ 0087 ワークステーションを使用不可能状態にし, G K Sを終了する.

(プログラムの出力結果)

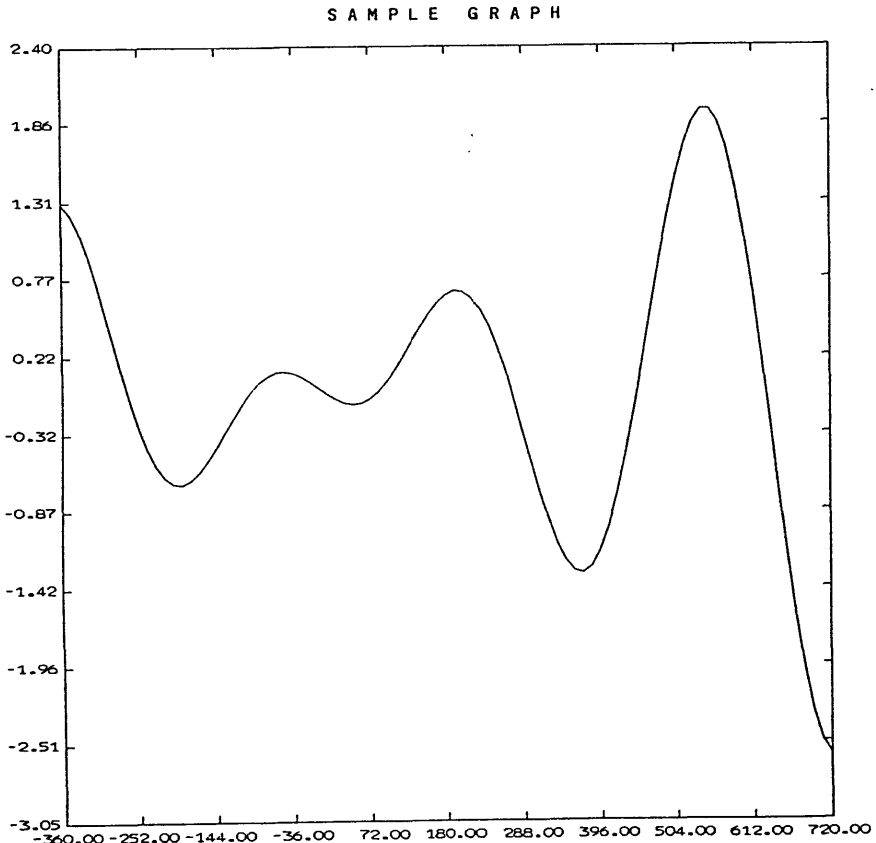


図 2. 1

3. 複数のワークステーションの動作

GKS 85 は、いくつものワークステーションを一度に動作させることができる。2つ以上のワークステーションを使用するときは、サブルーチン GOPWK, GACWK, GDAWK, GCLWK を、使用する数だけ用いれば良い。

3. 1 ワークステーションの複数動作を用いたプログラム例

複数のワークステーションに図形を出力させるプログラムを以下に示す。このプログラムの出力を図3. 1に示す。

(プログラム例)

```

0001      PROGRAM SAMPLE
0002 C
0003      INTEGER STAT, CHNR
0004      REAL   KDX(6), KDY(6), KTFDX(5), KTFDY(5), KX(6), KY(6),
0005      +      KTDX(6), KTDY(6), PXA(9), PYA(9)
0006      DATA  KDX/0.5, 0.5246, 0.5246, 0.5, 0.4754, 0.4754/,
0007      +      KDY/0.47, 0.49, 0.51, 0.53, 0.51, 0.49/,
0008      +      KTDX/0.0246, 0.0492, 0.0246, -0.0246, -0.0492, -0.0246/,
0009      +      KTDY/-0.0400, 0.0, 0.0400, 0.0400, 0.0, -0.0400/,
0010      +      KTFDX/0.0492, 0.0738, 0.0, -0.0738, -0.0492/,
0011      +      KTFDY/-0.08, 0.04, 0.08, 0.04, -0.08/,
0012      +      PXA/0.3, 0.35, 0.4, 0.45, 0.5, 0.55, 0.6, 0.65, 0.7/,
0013      +      PYA/0.7, 0.75, 0.7, 0.75, 0.7, 0.75, 0.7, 0.75, 0.7/
0014 C
0015      CALL GOPKS  (0, 1)
0016      CALL GOPWK  (1, 11, 111)
0017      CALL GACWK  (1)
0018      CALL GOPWK  (2, 22, 222)
0019      CALL GACWK  (2)
0020 C
0021      DO 10 I=1, 5
0022          DO 20 J=1, 6
0023              KX(J)=KDX(J)+KTFDX(I)
0024              KY(J)=KDY(J)+KTFDY(I)
0025      20  CONTINUE
0026          CALL GFA  (6, KX, KY)
0027      10  CONTINUE
0028 C
0029      CALL GSFAIS (0)
0030      CALL GSFACI (2)
0031 C
0032      DO 30 I=1, 6
0033          KX(I)=0.5+(KDX(I)-0.5)*3.0

```

ワークステーション
1の動作区間

ワークステーション
2の動作区間

解 説

```

0034      KY(1)=0.5+(KDY(1)-0.5)*3.0
0035  30 CONTINUE
0036      CALL GFA      (6, KX, KY)
0037 C
0038      CALL GSFAIS(0)
0039      CALL GSFACI(4)
0040 C
0041      DO 40 I=1, 6
0042          DO 50 J=1, 6
0043              KX(J)=KDX(J)+KTDX(1)
0044              KY(J)=KDY(J)+KTDY(1)
0045  50 CONTINUE
0046      CALL GFA      (6, KX, KY)
0047  40 CONTINUE
0048 C
0049      CALL GRQCH  (2, 1, STAT, CHNR)
0050      CALL GDAWK  (2)
0051      CALL GCLWK  (2)
0052 C
0053      CALL GOPWK  (3, 33, 333)
0054      CALL GACWK  (3)
0055 C
0056      CALL GSMK   (-3)
0057      CALL GSMKSC (3, 0)
0058      CALL GSPMCI (1)
0059      CALL GPM    (9, PXA, PYA)
0060 C
0061      CALL GRQCH  (3, 1, STAT, CHNR)
0062      CALL GDAWK  (3)
0063      CALL GCLWK  (3)
0064 C
0065      CALL GSCHH  (0, 05)
0066      CALL GSTXCI (5)
0067      CALL GSTXAL (2, 3)
0068      CALL GTX   (0.5, 0.90, 'SAMPLE GKS85')
0069 C
0070      CALL GRQCH  (1, 1, STAT, CHNR)
0071      CALL GDAWK  (1)
0072      CALL GCLWK  (1)
0073      CALL GCLKS
0074 C
0075      STOP
0076      END

```

ワークステーション
3の動作区間

(プログラムの説明)

- 0003 ~ 0013 変数の宣言および初期値設定。
 0015 ~ 0017 G K S を開始し、ワークステーション 1 を出力可能状態にする。
 0018 ~ 0019 ワークステーション 2 を出力可能状態にする。
 0021 ~ 0027 カメの手足の出力。
 0029 ~ 0030 カメの甲羅のスタイルと色の設定。
 0032 ~ 0036 カメの甲羅の出力。
 0038 ~ 0039 カメの甲羅の模様スタイルと色を設定し、出力する。
 0041 ~ 0047 カメの甲羅の模様の出力。
 0049 ~ 0051 ワークステーション 2 を使用不可能状態にする。
 0053 ~ 0054 ワークステーション 3 を出力可能状態にする。
 0056 ~ 0059 マーカの出力。
 0061 ~ 0063 ワークステーション 3 を使用不可能状態にする。
 0065 ~ 0068 文字列の出力。
 0070 ~ 0073 ワークステーション 1 を使用不可能状態にし、G K S を終了する。

(プログラムの出力結果)

ワークステーション 1

ワークステーション 2

ワークステーション 3

SAMPLE G K S 8 5

```

  *   *   *   *
 * * * * *

```

```

  *   *   *   *
 * * * * *

```

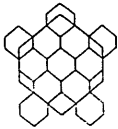


図 3. 1

4. セグメント

4. 1 セグメントの概要

G K S では、マーカ列や折れ線などの出力基本要素をひとつの集まりとして格納することができる。この集まりをセグメントと呼び、図形を操作する最小の単位となる。

作成されたセグメントは、オープンされた(出力可能状態)ワークステーションのセグメント記憶域(W D S S : Workstation Dependent Segment Storage)に格納される。この際、セグメントに名前を付けて管理している。セグメントの機能を表 4. 1 に示す。

表4.1 セグメントの機能

機能	サブルーチン名	備考
セグメントの作成を開始する	G C R S G	
セグメントの作成を終了する	G C L S G	
セグメントの名前を変更する	G R E N S G	
セグメントを削除する	G D S G	
	G D S G W K	
セグメントの属性を変更する	G S S G T	セグメント変換
	G S V I S	セグメント可視
	G S H L I T	セグメント顯示
	G S S G P	セグメント優先度
	G S D T E C	セグメント検知
セグメント操作	G C S G W K	セグメントの複写
	G A S G W K	セグメントの送出
	G I N S G	セグメントの挿入

以下に各セグメントの機能を説明する。

4.2 セグメントの作成開始

セグメントの作成を開始するには、サブルーチン G C R S G を使用する。このサブルーチンを呼び出すことにより、以後作成する出力基本要素はセグメント内の出力基本要素となる。セグメントには、名前を付けて識別する。この名前をセグメント名と呼び、セグメントの作成開始時に指定する。以下に呼び出し形式を示す。

CALL GCRSG(SGNA)

引数 SGNA (整数型) はセグメントを識別するセグメント名を指定する。指定有効な値は 0 以上である。

4. 3 セグメントの作成終了

セグメントの作成を終了するには、サブルーチン `GCLSG` を使用する。セグメントの作成開始から、このサブルーチンが呼び出されるまでに作成した出力基本要素は、出力可能状態ワークステーションの `WDS S` に格納される。以下に呼び出し形式を示す。

```
CALL GCLSG
```

このサブルーチンが呼び出された後、再びサブルーチン `GCRSG` が呼び出されるまでに作成した出力基本要素は、セグメント外の出力基本要素として、`WDS S` には格納されない。また、作成の完了したセグメントに対しては、新たに出力基本要素を追加したり、削除したりすることはできない。

4. 4 セグメント名の変更

作成済みのセグメントまたは、現在作成中のセグメントの名前を変更することができる。セグメントの作成開始時に指定したセグメント名を変更するには、サブルーチン `GRENSG` を使用する。以下に呼び出し形式を示す。

```
CALL GRENSG(OLD, NEW)
```

引数 `OLD` (整数型) に、変更したいセグメントの旧セグメント名を指定し、`NEW` (整数型) に新しいセグメント名を指定する。新しい名前には、存在していない名前を指定する。すでに存在している名前を指定するとエラーになる。

4. 5 セグメントの削除

セグメントを削除するには以下の二つの方法がある。

- ・現在使用可能状態のすべてワークステーションからセグメントを削除する。
- ・指定したワークステーションからセグメントを削除する。

現在使用可能状態のすべてのワークステーションからセグメントを削除するには、サブルーチン `GDSG` を使用する。以下に呼び出し形式を示す。

```
CALL GDSG(SGNA)
```

引数 `SGNA` (整数型) に、削除するセグメント名を指定する。

指定したワークステーションからセグメントを削除するには、サブルーチン `GDSGWK` を使用する。以下に呼び出し形式を示す。

```
CALL GDSGWK(WKID, SGNA)
```

引数 `WKID` (整数型) に、削除するセグメントを格納しているワークステーション名を指定し、`SGNA` (整数型) に削除するセグメント名を指定する。

どちらの方法でも削除できるセグメントは作成済みのものに限られ、現在作成中のセグメントは削除できない。

また、サブルーチン `GCLWK`、または `GCLRWK` を呼び出した場合にも、指定ワークステーションに格納しているすべてのセグメントが消去される。

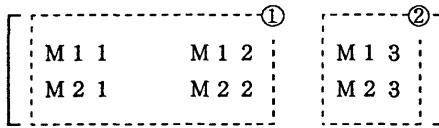
4. 6 セグメント属性

4. 6. 1 セグメント変換

セグメント変換とは、セグメント単位での図形の拡大、縮小、回転移動、および平行移動を行うことである。セグメント変換を行うサブルーチンは `GSSGT` である。以下に呼び出し形式を示す。

CALL GSSGT(SGNA, M)

引数 SGNA (整数型) は変換の対象とするセグメント名であり, M (実数型) はセグメント変換を行うための 2×3 の変換行列である. セグメント変換行列 M の値は図4. 1 のように使用される.



行列① 拡大/縮小の倍率および回転角度 (ラジアン)
 行列② 平行移動量 (正規化装置座標)

図4. 1 セグメント変換行列

さて, 2 倍に拡大するセグメント変換を行い, 続けて3 倍に拡大するセグメント変換を行ったとする. 表示される図形は, 元の図形の6 倍ではなく3 倍である. なぜなら, サブルーチン GSSGT は, 指定したセグメントのセグメント変換を行うための変換行列を変更するだけであるから, 後に行ったセグメント変換が有効になる.

図4. 1 の説明だけでセグメント変換行列を作成するのはかなり困難である. GKS には, この変換行列を簡単に作成するためのユーティリティ (表4. 2) が用意されている.

表4. 2 セグメント変換行列作成のためのユーティリティ

ユーティリティ	サブルーチン名
セグメント変換行列の作成	GEVTM
セグメント変換行列の合成	GACTM

GEVTM の呼出し形式を以下に示す.

CALL GEVTM(X0, Y0, DX, DY, PHI, FX, FY, SW, MOUT)

引数 X0, Y0 (実数型) にはそれぞれ変換の中心となる X, Y 座標, DX, DY (実数型) にはそれぞれ X, Y 方向の平行移動量, PHI (実数型) には回転角度 (ラジアン), FX, FY (実数型) にはそれぞれ X, Y 方向の拡大/縮小の倍率を指定する. SW (整数型) には X0, Y0 および DX, DY で指定した値が世界座標 (0) か正規化装置座標 (1) かを指定する. MOUT (実数型) は作成されたセグメント変換行列を格納する領域 (2×3 の2次元配列) である.

GACTM の呼出し形式を以下に示す.

CALL GACTM(MINP, X0, Y0, DX, DY, PHI, FX, FY, SW, MOUT)

GEVTM との違いは, 合成の元となる変換行列を指定する引数 MINP (2×3 の実数型2次元配列) だけである.

4. 6. 2 セグメント可視

図形をセグメント単位で可視（見える）にするか、不可視（見えない）にするかを指定する属性である。使用するサブルーチンはG S V I Sである。呼出し形式を以下に示す。

```
CALL GSVIS(SGNA, VIS)
```

引数 SGNA（整数型）にはセグメント名を、VIS（整数型）には不可視（0）か可視（1）かを指定する。

出力遅延モードがA S A P（As Soon As Possible - 出力基本要素をできる限り早く表示する）のワークステーション（ディスプレイなど）では、このサブルーチンを呼出すことによって即座に図形を表示したり消去したりすることができる。

4. 6. 3 セグメント顯示

図形をセグメント単位で顯示（強調）するか、標準（非顯示）にするかを指定する属性である。使用するサブルーチンはG S H L I Tである。呼出し形式を以下に示す。

```
CALL GSHLIT(SGNA, HIL)
```

引数 SGNA（整数型）にはセグメント名を、HIL（整数型）には標準（0）か顯示（1）かを指定する。

顯示の方法はワークステーションに依存している。一般に、カラーディスプレイでセグメントの顯示を指定すると、そのセグメント内の図形は赤で表示される。その他のワークステーションでは線幅が太くなるか、または、何も変わらないかのどちらかである。

4. 6. 4 セグメント優先度

セグメントを表示する順番（優先度）を指定する属性である。優先度を指定しない場合、セグメントは作成した順に表示されるため、互いのセグメント内の図形に重なりがあると、後から作成されたセグメント内の図形が上に表示される。しかし、優先度を指定すると、より優先度の高いものが上に表示されるように順番が変更される。すなわち、優先度の高いセグメントほど後から表示される。セグメント優先度を指定するサブルーチンはG S S G Pである。以下に呼び出し形式を示す。

```
CALL GSSGP(SGNA, PRIOR)
```

引数 SGNA（整数型）にはセグメント名を指定する。PRIOR（実数型）には優先度（0.0～1.0）を指定する。優先度は 0.0 が最も低く、優先度を指定しないセグメントにはこの値が設定されている。

4. 6. 5 セグメント検知

セグメントをピック入力装置よりピック可能か否かを指定する属性である。使用するサブルーチンはG S D T E Cである。呼出し形式を以下に示す。

```
CALL GSDTEC(SGNA, DET)
```

引数 SGNA（整数型）にはセグメント名を指定し、DET（整数型）には非検知（0）か検知（1）かを指定する。検知属性を指定しないセグメントには非検知（0）が設定されている。

ピック入力は“検知”でかつ“可視”のセグメントだけを入力の対象（ピック可能）としているため、この属性を“検知”に設定しても、可視属性が“不可視”ならピック入力することはできない。ピック入力は REQUEST PICK 機能（サブルーチンG R Q P K）で行う。

4. 6. 6 セグメントのプログラム例

このプログラムは 'LIB. SAMPLE. FORT77(GKS85E)' に登録している。

(プログラム例)

```

0001      INTEGER   WKID/1/, ERRIND
0002      CHARACTER DATREC*80
0003      REAL      X(2), Y(2), M(2, 3)
0004      RAD(IX)=3. 141592*REAL(IX)/180. 0
0005 C
0006      CALL GOPKS(0, 1)
0007      CALL GOPWK(WKID, 1, 1)
0008      CALL GACWK(WKID)
0009 C
0010      CALL GSFAIS(1)
0011      CALL GSFACI(7)
0012      CALL GPREC(0, 0, 1, 0. 003, 0, 0, 0, 1, ERRIND, LDR, DATREC)
0013      X(1)=0. 5
0014      DO 10 I=1, 7
0015          Y(1)=0. 5*REAL(I)/7. 0
0016          CALL GCRSG(I)          ..... セグメントの作成開始
0017          CALL GGDP(1, X(1), Y(1), -3, 1, DATREC)          ↓
0018          CALL GCLSG          ..... 作成終了
0019 10 CONTINUE
0020      DO 20 I=8, 15
0021          CALL GSPLCI(16-I)
0022          CALL GSFACI(I-7)
0023          CALL GCRSG(I)          ..... セグメントの作成開始
0024          DO 21 J=0, 359, 15          |
0025              X(1)=SIN(RAD(J))*REAL(I-7)/17. 0+0. 5          |
0026              Y(1)=COS(RAD(J))*REAL(I-7)/17. 0+0. 5          |
0027              X(2)=SIN(RAD(J))*REAL(I-7)/16. 0+0. 5          |
0028              Y(2)=COS(RAD(J))*REAL(I-7)/16. 0+0. 5          |
0029              CALL GPL(2, X, Y)          |
0030              CALL GGDP(1, X(2), Y(2), -3, 1, DATREC)          |
0031 21 CONTINUE          ↓
0032          CALL GCLSG          ..... 作成終了
0033 20 CONTINUE
0034 C
0035      DO 30 I=1, 15
0036          CALL GSVIS(I, 0)          ..... 不可視
0037 30 CONTINUE
0038 C
0039      DO 41 I=1, 7

```

```

0040      CALL GSVIS(1,1)          ..... 可視
0041      CALL GSHLIT(1,1)        ..... 顯示
0042      CALL GSVIS(1,0)          ..... 不可視
0043  41 CONTINUE
0044      DO 42 I=8,15
0045      CALL GSVIS(1,1)          ..... 可視
0046  42 CONTINUE
0047      DO 43 I=8,15
0048      CALL GSHLIT(1,1)        ..... 顯示
0049      CALL GSVIS(1,0)          ..... 不可視
0050  43 CONTINUE
0051 C
0052      CALL GEVTM(0.5,0.5,0.35,0.35,0.0,0.2,0.2,1,M) ..... 変換行列 M の作成
0053      DO 51 I=8,15
0054      CALL GSSGT(1,M)          ..... 変換行列 M によるセグメント変換
0055      CALL GSHLIT(1,0)        ..... 非顯示
0056      CALL GSVIS(1,1)          ..... 可視
0057  51 CONTINUE
0058      CALL GEVTM(0.5,0.5,-0.25,0.1,0.0,0.5,0.5,1,M) ..... 変換行列 M の作成
0059      DO 52 I=8,15
0060      CALL GSSGT(1,M)          ..... 変換行列 M によるセグメント変換
0061  52 CONTINUE
0062 C
0063      CALL GDAWK(WKID)
0064      CALL GCLWK(WKID)
0065      CALL GCLKS
0066 C
0067      STOP
0068      END

```

(プログラムの説明)

```

0014 ~ 0019   セグメント 1 から 7 を作成する。
0020 ~ 0033   セグメント 8 から 15 を作成する。
0035 ~ 0037   セグメント 1 から 15 を不可視にする。
0039 ~ 0043   セグメント 1 から 7 を可視→顯示→不可視にする。
0044 ~ 0046   セグメント 8 から 15 を可視にする。
0047 ~ 0050   "          顯示→不可視にする。
0052 ~ 0057   変換行列を定義し、セグメント 8 から 15 をセグメント変換→非顯示→可視にする。
0058 ~ 0061   変換行列を新たに定義し、セグメント 8 から 15 をセグメント変換する。

```

4. 7 セグメント操作

セグメント操作では W I S S (Workstation Independent Segment Storage) という特別なワー

クステーションに作成したセグメントを用いる機能である。WISSはWDSSと違って通常のワークステーションに存在しないセグメント記憶域であり、GKSとして1つだけ持つことができる。また、WISSに作成したセグメントは、ワークステーションのWDSSに格納されるセグメントと違い、他のワークステーションへの複写、送出、挿入が可能である。

4. 7. 1 セグメントの複写

WISSに作成済みのセグメントを、そのセグメントが持っているセグメント変換行列を用いてセグメント変換を行い、その後他のワークステーションに出力する機能である。この場合、出力（複写）の対象となるWISSに作成済みの図形（セグメント）は複写先のワークステーションのWDSSには格納されない。すなわち、セグメントではない（セグメント外の）出力基本要素の集まりとして出力される。また、セグメント変換を行うため、WISSに作成した図形を拡大、縮小、回転、平行移動しワークステーションに出力することが可能である。セグメントの複写を行うサブルーチンはGCSGWKである。以下に呼び出し形式を示す。

CALL GCSGWK(WKID, SGNA)

引数 WKID（整数型）は複写先のワークステーション名を指定し、SGNA（整数型）には複写しようとするWISSに作成済みのセグメント名を指定する。

4. 7. 2 セグメントの送出

WISSに作成済みのセグメントを他のワークステーションのWDSSに出力する。すなわち出力（送出）先のワークステーションに、WISSと同じセグメントを作成したことと同じ状態になる。なお、セグメント変換は行わない。セグメントの送出を行うサブルーチンはGASGWKである。以下に呼び出し形式を示す。

CALL GASGWK(WKID, SGNA)

引数 WKID（整数型）は送出先のワークステーション名を指定し、SGNA（整数型）には送出しようとするWISSに作成済みのセグメント名を指定する。

4. 7. 3 セグメントの挿入

セグメントの挿入では、GKSの動作状態により機能が異なる。

ワークステーションが出力可能状態（サブルーチンGACWKをCALLした後の状態）では、WISSに作成済みのセグメントをセグメント変換、挿入変換した後、他のワークステーションへ出力する。この場合は、出力（挿入）の対象となるWISSに作成済みの図形（セグメント）は挿入先のワークステーションのWDSSには格納されない。すなわち、セグメントではない（セグメント外の）出力基本要素の集まりとして出力される。また、セグメント変換を行うため、WISSに作成した図形を拡大、縮小、回転、平行移動しワークステーションに出力することが可能である。

セグメントの作成中（サブルーチンGCRSGをCALLし、サブルーチンGCLSGをCALLする前の状態）では、WISSに作成済みのセグメントを、セグメント変換、挿入変換した後、出力可能状態にあるすべてのワークステーションの、現在作成中のセグメント内に出力する。この場合は、出力（挿入）の対象となるWISSに作成済みの図形（セグメント）は挿入先のワークステーションのWDSSにセグメントの構成要素の一つとして格納される。また、出力先ワークステーションとしてWISS（自分自身）を指定できる。セグメント変換を行うため、WISSに作成した図形を拡大、縮小、回転、平行移動しワークステーションに出力することも可能である。

セグメントの挿入では、前述したようにセグメント変換の後、挿入変換が行われる。挿入変換には、セグメント変換とは別の変換行列を指定するが、その行列（ 2×3 の 2次元配列）は形式、機能ともにセグメント変換行列と同じである。また、セグメント変換、挿入変換の効果を累積することに注意する。すなわち、セグメント変換時に2倍に拡大させ、挿入変換時に2倍に拡大させるようにすると、元の図形（W I S S に作成済みの図形）より縦横とも4倍になる。

セグメントの挿入を行うサブルーチンはG I N S Gである。以下に呼び出し形式を示す。

```
CALL GINS G(SGNA, M)
```

引数 SGNA（整数型）は送出しようとするW I S S に作成済みのセグメント名を指定し、M（実数型）は挿入変換を行うための変換行列（ 2×3 の 2次元配列）を指定する。

4. 7. 4 セグメントとクリッピング枠

セグメント内の出力基本要素のクリッピング枠はセグメント記憶域（W D S S, またはW I S S）に格納されている。しかし、W I S S に格納されたセグメントの場合には、セグメントの複写、送出、挿入によってクリッピング枠の扱いが異なる。

セグメントの複写では、W I S S にセグメントを作成した時点のものが有効になる。クリッピングは複写先のワークステーションに表示する際に行われる。この場合、クリッピング枠は、複写の対象となるW I S S 内の図形と同様に、複写先のワークステーションのW D S S には格納されない。

セグメントの送出では、W I S S にセグメントを作成した時点のものが有効になる。クリッピングは送出先のワークステーションに表示する際に行われる。この場合、クリッピング枠は、送出の対象となるW I S S 内の図形と同様に、複写先のワークステーションのW D S S に格納される。

セグメントの挿入では、W I S S にセグメントを作成した時点のものは無効となり、クリッピングを行うか行わないか（この設定は、サブルーチンG S C L I Pで行う）によってクリッピング枠が付けられる。クリッピングを行う場合には、現在、正規化変換に用いられているビューポートがクリッピング枠として付けられ、クリッピングを行わない場合には、省略値のビューポート枠、すなわち、左下（0.0,0.0）右上（1.0,1.0）である四角形の枠がクリッピング枠として付けられる。また、G K S の動作状態がセグメント作成中であれば、挿入先のワークステーションのW D S S, またはW I S S に格納され、出力可能状態であれば、セグメント記憶域には格納されない。

4. 7. 5 W I S S を用いたプログラム例

W I S S に曲線を囲む枠を作成し、セグメント変換を行うプログラム例を示す。このプログラムの出力を図4. 2に示す。

（プログラム例）

```
0001      PROGRAM SEG
0002 C
0003      INTEGER STAT, CHNR
0004      REAL   XA(91), Y1(91), Y2(91), Y3(91), FX(4), FY(4), MSML(2, 3),
0005      +      RAD, T, YMIN, YMAX, WYMIN, WYMAX
0006 C
0007      PARAMETER (RAD = 3.1415/180.0, T=12.0)
```


解 説

```

0008 C
0009     FUNC(X, F) = X*COS(X*RAD)/T*F*SIN(T*RAD)
0010 C
0011     DO 1 I=1, 91
0012         XA(I) = -360.0+FLOAT(I-1)*12.0
0013         Y1(I) = FUNC(XA(I), 1.0)
0014         Y2(I) = FUNC(XA(I), 5.0)
0015         Y3(I) = FUNC(XA(I), 10.0)
0016         IF (I.EQ.1) THEN
0017             YMIN = MIN(Y1(I), Y2(I), Y3(I))
0018             YMAX = MAX(Y1(I), Y2(I), Y3(I))
0019         ELSE
0020             WYMIN = MIN(Y1(I), Y2(I), Y3(I))
0021             WYMAX = MAX(Y1(I), Y2(I), Y3(I))
0022             IF (WYMIN.LT.YMIN) THEN
0023                 YMIN = WYMIN
0024                 FY(1) = YMIN
0025                 FY(2) = FY(1)
0026             ENDIF
0027             IF (WYMAX.GT.YMAX) THEN
0028                 YMAX = WYMAX
0029                 FY(3) = YMAX
0030                 FY(4) = FY(3)
0031             ENDIF
0032         ENDIF
0033     1 CONTINUE
0034 C
0035     FX(1) = XA(1)
0036     FX(2) = XA(91)
0037     FX(3) = FX(2)
0038     FX(4) = FX(1)
0039 C
0040     CALL GOPKS (0, 99)
0041     CALL GOPWK (10, 10, 10)
0042     CALL GACWK (10)
0043 C
0044     CALL GSWN (1, FX(1), FX(2), FY(1), FY(3))
0045     CALL GSVL (1, 0.0, 1.0, 0.0, 1.0)
0046     CALL GSELNT (1)
0047 C
0048     CALL GCRSG (100) ..... W I S S (ワークステーション 10) に 100 というセ
0049     CALL GFA (4, FX, FY) グメントを作成する
0050     CALL GCLSG

```

```

0051 C
0052     CALL GOPWK (1, 1, 1)
0053     CALL GACWK (1)
0054 C
0055     CALL GEVTM (0.5, 0.5, 0.0, 0.0, 0.3, 0.0, 1.0, 0.2, 1, MSML) ..... 変換行列の作成
0056     CALL GSSGT (100, MSML) ..... セグメント変換
0057     CALL GCSGWK (1, 100) ..... W I S S に作成したセグメント 100 を複写
0058     CALL GSWN (11, FX(1), FX(2), FY(1), FY(3))
0059     CALL GSVF (11, 0.0, 1.0, 0.7, 0.9)
0060     CALL GSELNT (11)
0061     CALL GPL (91, XA, Y1)
0062 C
0063     CALL GEVTM (0.5, 0.5, 0.0, 0.0, 0.0, 0.0, 1.0, 1.0, 1, MSML) ..... 変換行列の作成
0064     CALL GSSGT (100, MSML) ..... セグメント変換
0065     CALL GEVTM (0.5, 0.5, 0.0, 0.0, 0.0, 0.0, 1.0, 0.2, 1, MSML) ..... 変換行列の作成
0066     CALL GINSF (100, MSML) ..... W I S S に作成したセグメント 100 を挿入
0067     CALL GSWN (12, FX(1), FX(2), FY(1), FY(3))
0068     CALL GSVF (12, 0.0, 1.0, 0.4, 0.6)
0069     CALL GSELNT (12)
0070     CALL GPL (91, XA, Y2)
0071 C
0072     CALL GEVTM (0.5, 0.5, 0.0, -0.3, 0.0, 1.0, 0.2, 1, MSML) ..... 変換行列の作成
0073     CALL GSSGT (100, MSML) ..... セグメント変換
0074     CALL GEVTM (0.5, 0.5, 0.0, 0.0, 0.0, 0.0, 1.0, 1.0, 1, MSML) ..... 変換行列の作成
0075     CALL GCRSG (200)
0076     CALL GINSF (100, MSML) ..... W I S S に作成したセグメント 100 を挿入
0077     CALL GSWN (13, FX(1), FX(2), FY(1), FY(3))
0078     CALL GSVF (13, 0.0, 1.0, 0.1, 0.3)
0079     CALL GSELNT (13)
0080     CALL GPL (91, XA, Y3)
0081     CALL GCLSG
0082 C
0083     CALL GRQCH (1, 1, STAT, CHNR)
0084     CALL GDAWK (10)
0085     CALL GCLWK (10)
0086     CALL GDAWK (1)
0087     CALL GCLWK (1)
0088     CALL GCLKS
0089 C
0090     STOP
0091     END

```

(プログラムの説明)

- 0003 ~ 0005 必要な変数・配列の宣言。
- 0011 ~ 0033 座標値の計算を行い、Y座標値の最小値および最大値を求め、曲線を囲む枠のY座標値を設定する。
- 0035 ~ 0038 曲線を囲む枠のX座標値を設定する。
- 0040 ~ 0042 G K Sの開始およびワークステーション 10 (このプログラムではW I S S) を出力可能状態にする。
- 0044 ~ 0046 座標変換。
- 0048 ~ 0050 セグメント 100 に枠を作画する。
- 0052 ~ 0053 ワークステーション 1 を出力可能状態にする。
- 0055 セグメント変換に使用する変換行列を作成する (変換の中心点は (0.5, 0.5), X方向の平行移動量は 0.0, Y方向の平行移動量は +0.3, 回転角度は 0.0, X方向の倍率は 1.0, Y方向の倍率は 0.2 で、いずれも正規化装置座標値で設定)。
- 0056 0055 行目で作成した変換行列を用いてセグメント変換を行う。
- 0057 W I S S に作成済みのセグメント 100 を複写する。
- 0058 ~ 0061 座標変換を行い、曲線を描く。
- 0063 セグメント変換に使用する変換行列を作成する (変換の中心点は (0.5, 0.5), X方向, Y方向の平行移動量は共に 0.0, 回転角度は 0.0, X方向, Y方向の倍率は共に 1.0 で、いずれも正規化装置座標値で設定)。
- 0064 0063 行目で作成した変換行列を用いてセグメント変換を行う。
- 0065 挿入変換に使用する変換行列を作成する (変換の中心点は (0.5, 0.5), X方向, Y方向の平行移動量は共に 0.0, 回転角度は 0.0, X方向の倍率は 1.0, Y方向の倍率は 0.2 で、いずれも正規化装置座標値で設定)。
- 0066 W I S S に作成済みのセグメント 100 を挿入する。
- 0067 ~ 0070 座標変換を行い、曲線を描く。
- 0072 セグメント変換に使用する変換行列を作成する (変換の中心点は (0.5, 0.5), X方向の平行移動量は 0.0, Y方向の平行移動量は -0.3, 回転角度は 0.0, X方向の倍率は 1.0, Y方向の倍率は 0.2 で、いずれも正規化装置座標値で設定)。
- 0073 0072 行目で作成した変換行列を用いてセグメント変換を行う。
- 0074 挿入変換に使用する変換行列を作成する (変換の中心点は (0.5, 0.5), X方向, Y方向の平行移動量は共に 0.0, 回転角度は 0.0, X方向, Y方向の倍率は共に 1.0 で、いずれも正規化装置座標値で設定)。
- 0075 セグメント 200 の作成開始。
- 0076 W I S S に作成済みのセグメント 100 を挿入する。
- 0077 ~ 0080 座標変換を行い、曲線を描く。
- 0081 セグメント 200 の作成終了。
- 0083 ~ 0088 ワークステーション 10 (このプログラムではW I S S) とワークステーション 1 を使用不可能状態にまで移行し、G K Sを終了する。

(プログラムの出力結果)

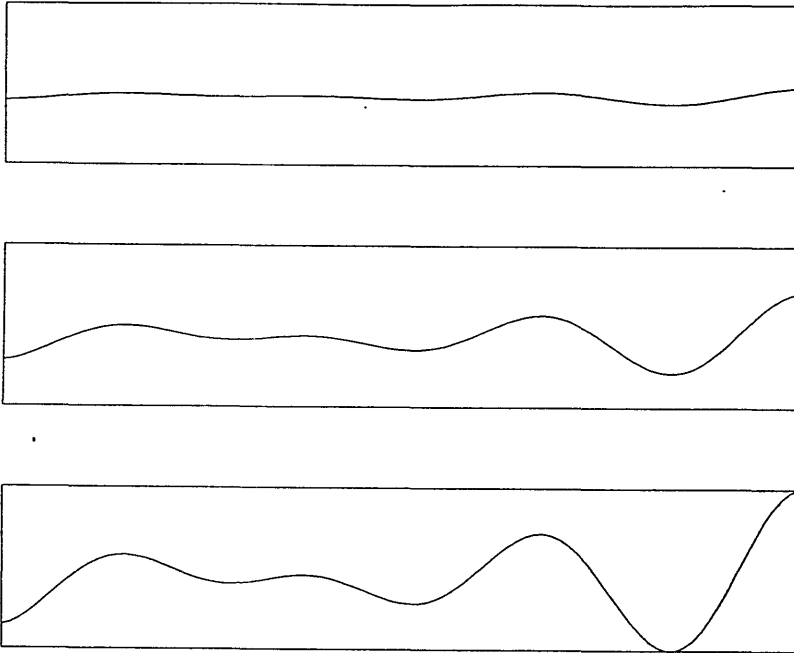


図4. 2

5. GKS 85 コマンドプロシジャ・カタログドプロシジャの変更について

5. 1 GKS 85 コマンドプロシジャ

GKS 85 コマンドプロシジャが、テクトロ端末, CLP, および複数のワークステーションの同時動作機能のサポートに伴い変更になった。以下に入力形式を示す。

(入力形式)

コマンド名	オペランド
GKS 85	{データセット名 @} [WS ([数値1/数値2:] ワークステーション種別 [,])]] [NLP (O A K S U)]] [OPR (OPR機番)]] [PRVLIB (私用ライブラリデータセット名)]] [ERR (0)]] [SIDE (FRONT BOTH)]] [BIND (LEFT RIGHT UPPER LOWER)]]

(オペランドの説明)

データセット名 | @ : ソースプログラムが入っているデータセット名を指定する。@ を指定すると前回に作成したロードモジュールを実行する。

WS ([数値1 / 数値2 :] ワークステーション種別 [,])
 : 数値1 はサブルーチンGOPWKの2番目の引数の値を指定する。数値2 は同サブルーチンの3番目の引数の値を指定する。ワークステーション種別には以下のものがある。
 WSパラメタを省略した場合には一覧を表示し、入力を促す。

- ・富士通カラーディスプレイ
 F 6 2 4 2 A F 6 6 5 3 A F 6 6 5 3 C F 6 6 5 8 K F 6 6 8 3 A C
- ・富士通モノクロディスプレイ
 F 6 6 5 8 A F 6 6 5 8 B F 6 6 8 3 A M F 6 6 8 3 B
- ・TTY端末
 T 4 0 0 6 T 4 0 1 0 T 4 0 1 2 T 4 0 1 4 T 4 0 1 4 E G M
 F 9 4 3 2 A
- ・日本語ラインプリンタ
 N L P
- ・カットシートラインプリンタ
 C L P A 4 L . . . A 4 サイズ ランドスケープ
 C L P A 4 P . . . A 4 サイズ ポートレート
 C L P B 4 L . . . B 4 サイズ ランドスケープ
 C L P B 4 P . . . B 4 サイズ ポートレート
- ・オフィスプリンタ
 O P R A 4 L . . . A 4 サイズ ランドスケープ
 O P R A 4 P . . . A 4 サイズ ポートレート
 O P R B 4 L . . . B 4 サイズ ランドスケープ
 O P R B 4 P . . . B 4 サイズ ポートレート
 O P R B 5 L . . . B 5 サイズ ランドスケープ
 O P R B 5 P . . . B 5 サイズ ポートレート
- ・メタファイル (入力)
 M I [(データセット名)]
- ・メタファイル (出力)
 M O A 4 L [(データセット名)] . . . A 4 サイズ ランドスケープ
 M O A 4 P [(データセット名)] . . . A 4 サイズ ポートレート
 M O B 4 L [(データセット名)] . . . B 4 サイズ ランドスケープ
 M O B 4 P [(データセット名)] . . . B 4 サイズ ポートレート
 M O B 5 L [(データセット名)] . . . B 5 サイズ ランドスケープ
 M O B 5 P [(データセット名)] . . . B 5 サイズ ポートレート
- ・セグメント記憶域
 W I S S

NLP (O | A | K | S | U)

: ワークステーション種別にNLPを指定した場合の出力クラスを指定する。省略時は O。

OPR (OPR機番) : ワークステーション種別にOPR系を指定した場合に、出力するOPRの機番を指定する。省略するとプロンプトが出力され入力を促す。

PRVLIB (私用ライブラリデータセット名)

: 組み込みたい私用ライブラリのデータセット名を指定する。

ERR (0) : サブルーチンGOPKSの最初の引数を指定する。省略時は 0。

SIDE (FRONT | BOTH)

: CLP出力時に片面印刷にする (FRONT) か、両面印刷にする (BOTH) かを指定する。省略時は FRONT。

BIND (LEFT | RIGHT | UPPER | LOWER)

: CLP出力時のとじしろを指定する。省略時は LEFT。

(注意)

- ・ WS パラメータは、ソースプログラムのサブルーチンGOPWKが例えば、

CONID = 12

WTYPE = 34

CALL GOPWK (WKID, CONID, WTYPE)

である場合、CONID = 数値1, WTYPE = 数値2であるから、

WS(12/34:ワークステーション種別)

のように指定するが、CONID = 1 であつ WTYPE = 1 であれば、

WS(ワークステーション種別)

のように、数値1 / 数値2 : を省略できる。

- ・ ソースプログラムのサブルーチンGOPWKが例えば、

CONID = 1234

WTYPE = 5678

CALL GOPWK (WKID, CONID, WTYPE)

であるとすると、図形データは GKSC1234 という論理機番から出力 (あるいは入力) される。WTYPE はGKS85が使用するワークステーション型ファイルに記述される数値である。

- ・ ワークステーションの同時動作を行なう場合、ソースプログラムのサブルーチンGOPWKが例えば

CALL GOPWK (WKID1, CONID1, WTYPE1)

CALL GOPWK (WKID2, CONID2, WTYPE2)

とすると、

WKID1 = WKID2 であってはならない。

CONID1 = CONID2 であってはならない。

WTYPE1 = WTYPE2 であってはならない。

- ・ WS パラメータに複数のワークステーション種別を指定する場合には、WSパラメータ内を ' で囲み、ワークステーション種別ごとにカンマで区切る。
- ・ 同じワークステーション種別を同時に指定してはならない。

- ・ E R R パラメタは、ソースプログラムのサブルーチン G O P W K が例えば、
`CALL GOPKS (12,34)`
 であれば、
`ERR(12)`
 と指定する。この場合、論理機番 GKSE0012 に G K S 8 5 のエラーメッセージが出力される。
- ・ G K S 8 5 コマンドプロシジャは、作業用データセットとして以下のデータセットを使用する。

オブジェクトモジュール	:	# G K S # . O B J
ロードモジュール	:	# G K S # . L O A D
ワークステーション型ファイル	:	# G K S # . W T Y P E
割当て用データセット	:	# G K S # . D E V

1) 画面出力

プログラムを実行し、端末に図形を表示させる。使用するワークステーションは F 6 6 8 3 A 型カラー端末とする。ソースプログラムに用いているサブルーチン G O P W K は、
`CALL GOPWK(1,1,1)`
 としている。このサブルーチンの 2 番目、3 番目の引数は、共に 1 であるから W S パラメタは以下のようにワークステーション種別のみを指定することができる。
 READY
GKS85 'LIB.SAMPLE.FORT77(GKS85A)' WS(F6683AC)

2) N L P への出力

プログラムを実行し、N L P に図形を出力させる。出力クラスは U とする。ソースプログラムに用いているサブルーチン G O P W K は、
`CALL GOPWK(1,1,2345)`
 としている。このサブルーチンの 2 番目の引数は 1、3 番目の引数は 2345 であるから以下のように W S パラメタを指定する。
 READY
GKS85 'LIB.SAMPLE.FORT77(GKS85B)' WS(1/2345:NLP) NLP(U)

3) O P R への出力

メニューを経由して、2) で作成したロードモジュールを実行し、O P R に出力する。

READY
GKS85 @ WS パラメタを省略するとメニューが現われる。
<<< GKS85 V10L20 START 15:38:40 05/29/90 >>>

 === FOLLOWING DEVICES ARE SUPPORTED NOW. ===
 1. FUJITSU COLOR DISPLAY
 2. FUJITSU MONOCHROME DISPLAY
 3. TTY TERMINAL
 4. OFFICE PRINTER
 5. LINE PRINTER (NLP AND CLP)

6. METAFILE
7. WISS
8. < MENU END >

PLEASE ENTER WORKSTATION NO ==> 4 O P R を選択する。この後、O P R シートのメニューが現われる。

--- OFFICE PRINTER ---

1. A4 SIZE LANDSCAPE
2. A4 SIZE PORTRAIT
3. B4 SIZE LANDSCAPE
4. B4 SIZE PORTRAIT
5. B5 SIZE LANDSCAPE
6. B5 SIZE PORTRAIT

PLEASE ENTER CONID/WTYPE:OPR NO --> 1/2345:1

..... サブルーチン G O P W K の、2 番目の引数 / 3 番目の引数：ワークステーション種別の番号（ここでは 1，すなわち A 4 サイズランドスケープを選択している）を入力する。サブルーチン G O P W K の 2 番目および 3 番目の引数が共に 1 ならばワークステーションの種別番号のみでよい。この後、最初のメニューに戻る。

=== FOLLOWING DEVICES ARE SUPPORTED NOW. ===

1. FUJITSU COLOR DISPLAY
2. FUJITSU MONOCHROME DISPLAY
3. TTY TERMINAL
4. OFFICE PRINTER
5. LINE PRINTER (NLP AND CLP)
6. METAFILE
7. WISS
8. < MENU END >

PLEASE ENTER WORKSTATION NO ==> 8 ワークステーションの指定がすべて終了したならば、8 を入力する。

PLEASE ENTER OPR UNIT NO ==> XXXX O P R 機番を入力する。

4) 複数の図形装置への出力

3. 1 で説明したプログラムを実行する。ソースプログラムに用いているサブルーチン G O P W K は、

```
CALL GOPWK(1,11,111)
CALL GOPWK(2,22,222)
CALL GOPWK(3,33,333)
```

の 3 つがある。すなわち、3 種類のワークステーションを使用するため、WS パラメタは以下のように指定する。ここでは、N L P, C L P (A 4 サイズランドスケープ), O P R (A 4 サイズランドスケープ) に出力させる。O P R の装置機番は XXXX である。


```
READY
GKS85 'LIB. SAMPLE. FORT77(GKS85E)' +
      WS('11/111:NLP, 22/222:CLPA4L, 33/333:OPRA4L') OPR(XXXX)
```

5) メタファイルを用いる場合

META. DATA というメタファイルに図形を格納する。メタファイルのデータセット名を省略した場合にはプロンプトが出力され、入力を促す。

```
READY
GKS85 'LIB. SAMPLE. FORT77(GKS85A)' WS(MOA4L(META. DATA))
```

6) W I S Sを用いる場合

4. 7. 5で説明したW I S Sを用いたプログラムを実行し、結果をF 6 6 8 3 A型カラー端末に表示させる。ソースプログラムに用いているサブルーチンG O P W Kは、

```
CALL GOPWK(10, 10, 10) . . . W I S Sのオープン用
CALL GOPWK(1, 1, 1) . . . 出力装置のオープン用
```

の2つがあるため、以下のように指定する。

```
READY
GKS85 'LIB. SAMPLE. FORT77(GKS85D)' WS('F6683AC, 10/10:WISS')
```

5. 2 G K S 8 5カタログドプロシジャ

G K S 8 5カタログドプロシジャが、C L P、複数のワークステーションの同時動作機能のサポートに伴い変更になった。以下に指定形式を示す。

(指定形式)

カタプロ名	パラメタ
G K S 8 5	[, STEP = { C C L <u>C G O</u> C G C L G G O }] [, SYSOUT = { O <u>A</u> K S U }] [, OPT = { 1 <u>2</u> 3 }] [, PRVLIB = ' 専用ライブラリデータセット名'] [, WS = ' [数値1 / 数値2 :] ワークステーション種別 [,] '] [, PROG = プログラム名] [, LOADDS = ' データセット名']

(パラメタの説明)

STEP = { C | C L | C G O | C G | C L G | G O }

: 処理過程を選択する。省略時は C G O.

C : 翻訳のみ行う。

C L : 翻訳, リンケージエディタによる結合編集を行い、ロードモジュールを作成する。

CGO : 翻訳, 結合編集, 実行を1プロシジャステップで行う。

CG : 翻訳, ロードによる結合編集および実行を行う。

CLG : 翻訳, リンケージエディタによる結合編集, 実行を行う。

GO : ロードモジュールを実行する。

SYSO UT = { O | A | K | S | U }

: 出力クラスを指定する。省略値は A。

OPT = { 0 | 1 | 2 | 3 }

: FORTRANの最適化のレベルを指定する。省略値は 2。

PRVLIB = 'データセット名'

: 組み込みたい私用ライブラリのデータセット名を指定する。

WS = ' [数値1 / 数値2 :] ワークステーション種別 [,] '

: 数値1はサブルーチンGOPWKの2番目の引数の値を指定する。数値2は同サブルーチンの3番目の引数の値を指定する。ワークステーション種別には以下のものがある。

・日本語ラインプリンタ

NLP

・カットシートラインプリンタ

CLPA4L . . . A4サイズ ランドスケープ
CLPA4P . . . A4サイズ ポートレート
CLPB4L . . . B4サイズ ランドスケープ
CLPB4P . . . B4サイズ ポートレート

・オフィスプリンタ

OPRA4L . . . A4サイズ ランドスケープ
OPRA4P . . . A4サイズ ポートレート
OPRB4L . . . B4サイズ ランドスケープ
OPRB4P . . . B4サイズ ポートレート
OPRB5L . . . B5サイズ ランドスケープ
OPRB5P . . . B5サイズ ポートレート

・メタファイル (入力)

MI

・メタファイル (出力)

MOA4L . . . A4サイズ ランドスケープ
MOA4P . . . A4サイズ ポートレート
MOB4L . . . B4サイズ ランドスケープ
MOB4P . . . B4サイズ ポートレート
MOB5L . . . B5サイズ ランドスケープ
MOB5P . . . B5サイズ ポートレート

・セグメント記憶域

WISS

PROG = プログラム名

: 実行するロードモジュールのメンバ名を指定する。このパラメータは、STEP = GO を指定した時のみ有効となる。省略値は MAIN。

LOADDS = '区分データセット名'

: 実行するロードモジュールが格納されている区分データセット名を指定する。このパラメータは、STEP = GO を指定した時のみ有効となる。

(注意)

・プロシジャステップについて

- ① STEP = C : FORT
- ② " CL : FORT, LKED
- ③ " CGO : FORTCGO
- ④ " CG : FORT, LOADGO
- ⑤ " CLG : FORT, LKED, GO
- ⑥ " GO : GKSGO

・関連する DDD 名について

- FORT. SYSIN : ソースプログラム用 (①, ②, ④, ⑤)
- FORTCGO. SYSIN : " (③)
- LKED. SYSLMOD : ロードモジュール用 (②)
- GO. SYSIN : 実行データ用 (⑤)
- LOADGO. SYSIN : " (④)
- FORTCGO. SYSGO : " (③)
- GKSGO. SYSGO : " (⑥)

・WS パラメータは、ソースプログラムのサブルーチン GOPWK が例えば、

```
CONID = 12
WTYPE = 34
CALL GOPWK (WKID, CONID, WTYPE)
```

である場合、CONID = 数値 1, WTYPE = 数値 2 であるから、

WS = '12/34:ワークステーション種別'

のように指定するが、CONID = 1 でかつ WTYPE = 1 であれば、

WS = 'ワークステーション種別'

のように、数値 1 / 数値 2 : を省略できる。

また、CONID = 1, WTYPE = 1, ワークステーション種別が NLP で、他のワークステーションを同時に使用しないのであれば WS パラメータは省略できる。

・ソースプログラムのサブルーチン GOPWK が例えば、

```
CONID = 1234
WTYPE = 5678
CALL GOPWK (WKID, CONID, WTYPE)
```

であるとすると、図形データは GKSC1234 という論理機番から出力 (あるいは入力) される。WTYPE は GK S 8 5 が使用するワークステーション型ファイルに記述される数値である。

- ・ワークステーションの同時動作を行なう場合、ソースプログラムのサブルーチン G O P W K が例えば

```
CALL GOPWK (WKID1, CONID1, WTYPE1)
CALL GOPWK (WKID2, CONID2, WTYPE2)
```

とすると、

```
WKID1 = WKID2 であってはならない。
CONID1 = CONID2 であってはならない。
WTYPE1 = WTYPE2 であってはならない。
```

- ・同じワークステーション種別を同時に指定してはならない。
- ・C L P への出力で両面印刷を行う場合には、プロシジャステップが C L G で、図形データが論理機番 GKSC0001 から出力され、とじしろを UPPER に変更する場合

```
//GO.GKSC0001 DD SY=U, PRTFORM=(SIDE=BOTH, BIND=UPPER)
```

とする。印刷モードの省略値は F R O N T (片面印刷)，とじしろの省略値は L E F T である。

1) N L P への出力

プログラムを実行し、N L P に出力させる。ソースプログラムに用いているサブルーチン G O P W K は、

```
CALL GOPWK(1,1,1)
```

としている。このサブルーチンの 2 番目、3 番目の引数は、共に 1 であり、N L P への出力であるから W S パラメタは省略できる。

```
// EXEC GKS85, STEP=CLG
//FORT.SYSIN DD DSN=LIB.SAMPLE.FORT77(GKS85A), DISP=SHR
```

2) O P R への出力

プログラムを実行し、O P R に図形を出力させる。ソースプログラムに用いているサブルーチン G O P W K は、

```
CALL GOPWK(1,1,2345)
```

としている。このサブルーチンの 2 番目の引数は 1、3 番目の引数は 2345 であるから以下のように W S パラメタを指定する。

```
// EXEC GKS85, STEP=CLG, WS='1/2345:OPRA4L'
//FORT.SYSIN DD DSN=LIB.SAMPLE.FORT77(GKS85B), DISP=SHR
//GO.GKSC0001 DD DSN=A79999A.OPR.DATA, DISP=(NEW, CATLG),
// SPACE=(TRK, (10, 10), RLSE), UNIT=PUB
```

上記ジョブを実行した後、O P R 機番 XXXX へ出力するために、T S S で次のコマンドを実行する。

```
READY
DSPRINT OPR.DATA XXXX DOC
```

3) 複数の図形装置への出力

3. 1 で説明したプログラムを実行する。ソースプログラムに用いているサブルーチン G O P W K は、

```
CALL GOPWK(1, 11, 111)
CALL GOPWK(2, 22, 222)
CALL GOPWK(3, 33, 333)
```

の3つがある。すなわち、3種類のワークステーションを使用するため、WSパラメタは以下のように指定する。ここでは、NLP、CLP（A4サイズランドスケープ、およびA4サイズポートレート）に出力させる。

```
// EXEC GKS85, STEP=CLG, WS='11/111:NLP, 22/222:CLPA4L, 33/333:CLPA4P'
//FORT.SYSIN DD DSN=LIB.SAMPLE.FORT77(GKS85C), DISP=SHR
//GO.GKSC0022 DD SYSOUT=U
//GO.GKSC0033 DD SYSOUT=U
```

4) メタファイルを用いる場合

応用プログラムを実行し、データセット名 META.DATA というメタファイルに図形を格納する。

```
// EXEC GKS85, STEP=CLG, WS='MOA4L'
//FORT.SYSIN DD DSN=LIB.SAMPLE.FORT77(GKS85A), DISP=SHR
//GO.GKSC0001 DD DSN=A79999A.META.DATA, DISP=(NEW, CATLG),
//          SPACE=(TRK, (10, 10), RLSE), UNIT=PUB
```

5) WISSを用いる場合

4.7.5で説明したプログラムを実行し、セグメント操作を行った結果をNLPに出力させる。ソースプログラムに用いているサブルーチンGOPWKは、

```
CALL GOPWK(10, 10, 10) . . . WISSのオープン用
CALL GOPWK(1, 1, 1) . . . 出力装置のオープン用
```

の2つがあるため、以下のように指定する。

```
// EXEC GKS85, STEP=CLG, WS='NLP, 10/10:WISS'
//FORT.SYSIN DD DSN=LIB.SAMPLE.FORT77(GKS85D), DISP=SHR
```

6. おわりに

今回は、KST（ストローク）フォントの使用法、ワークステーションの複数動作、セグメントを説明した。次回は最終回、メタファイルについて説明したい。

参考文献

- [1] GKS入門(1), 九大大型計算機センター広報, Vol.22, No.5, 1989 P446 ~ P468
- [2] GKS入門(2), 九大大型計算機センター広報, Vol.22, No.6, 1989 P729 ~ P753
- [3] GKS入門(3), 九大大型計算機センター広報, Vol.23, No.1, 1990 P001 ~ P013
- [4] 平野, グラフィックツールを使った図形出力の方法, 九大大型計算機センター広報, Vol.21, No.5, 1988 P432 ~ P452
- [5] ルシア マッケイ著, 松本 眞理訳, GKS入門, ノバグラフィックス ジャパン(株).
- [6] 計算機マニュアル, 富士通 GKS85文法書(FORTRAN編)(99SP-8240-1), 富士通(株).
- [7] 計算機マニュアル, 富士通 GKS85使用手引書V10(99SP-4080-2), 富士通(株).