ワークステーション用簡易グラフィックライブラリの作成

南斉 清巳

Simple Graphic Library for Workstation

Kiyomi NANSAI

1 はじめに

本校の情報センターにはunixワークステーションが設置されており，学生実験，卒業研究，教官研究等に広く利用されている。しかし，GKS（Graphic Kernel System）と呼ばれるグラフィックライブラリがインストールされているにも関わらず，計算結果をグラフ化するというような利用方法がほとんどされていないのが現状である。この要因として，

（1）GKSの利用法法が難解である
（2）マニュアルが英語である
（3）GKSの解説書が少ない
（4）サンプルプログラムがない

といった点があげられる。
GKSやXライブラリの知識を必要とせず，BASIC言語のグラフィックコマンドを使用するような感覚で，グラフィックプログラミングが作成できれば，学生実験や卒業研究等での利用に大変有益であると考える。

本稿では，unixワークステーション上で容易にグラフィックプログラミングを可能とする簡易グラフィックライブラリを作成したので報告する。

2 グラフィックライブラリの構成

GKSを利用してプログラムを作成する場合，ユーザープログラムからGKSのサブルーチンを直接呼び出すのであるが，これらを利用するためにはかなりの知識と経験が必要となる。このため，学生実験などで限られた時間内にグラフィックプログラムを作成するのは困難である。今回作成した簡易グラフィックライブラリは図1に示すようにGKSとユーザプログラムとの間に位置するもので，ユーザはGKSの知識を必要とせず，非常に簡単なサブルーチン呼び出しでグラフィックプログラミングを可能にするというものである。このグラフィックライブラリはGKSを基にしている。ここで簡単にGKSの特徴をあげておく。GKSは1985年にISOの国際規格に定められた2次元のグラフィックシステムで次のような特徴を持つ。

（1）仮想装置による装置独立性の保証
（2）メタファイルによる共通の図形保存
（3）モデリング変換と視野変換の分離
（4）ワークステーションの概念によってグラフィックス装置を統一的に管理する
（5）コンピュータ言語での表現も統一して，プログラムの可搬性を保証
（6）機能を水準（レベル）化して，プログラムの可搬性を高めている

しかし，GKSのライブラリ関数を使用してグラフィックプログラムを組むためには，かなりの知識と経験を要する。これらのグラフィック機能を，学生実験や卒業研究等で簡単に利用できるようにするため，つぎのような点に考慮して，グラフィックライブラリを作成した。

（1）使用対象者をはっきりさせる
（2）関数名を単純にする
（3）関数の機能を理解しやすくする
（4）使用方法を単純化する

（1）については，使用対象者をはっきりしておくことで，関数の機能を絞り込むことができ，全体的に統一されるという利点がある。今回は本校の電子制御工学科の2学年時に行われるC言語の授業（年間60時間）と同程度の知識を持つ者の利用を想定した。配列

と簡単なポインタの知識があれば十分利用可能である。（2）については，関数の名前を単純にし，ユーザがすぐ思いつくようなものにした。具体的にはBASIC言語にあるようなコマンド名を参考にし，それらに該当しないものについては名前から機能を連想できるようなものにした。（3）については，GKSでは低レベルの関数があり，複数の関数を組み合わせて初めて機能するようなものがある。今回作成した関数は，基本的には1つの関数で1つの機能が実行できるような高機能なものにし，ユーザが使いやすいように考慮した。（4）については，簡便さを第一に考え，汎用性は多少犠牲になるが機能を限定させた。こうすることによって，引数を少なくすることができ，ユーザ側からは利用しやすくなる。

●使用機器

ワークステーション	：東芝AS4050FGX
CPU	：SPARC
メモリー	：32MB
OS	：AS／OS ver4.1.3
X端末	：高岳製作所XMiNT
CPU	：68030
メモリー	：8MB

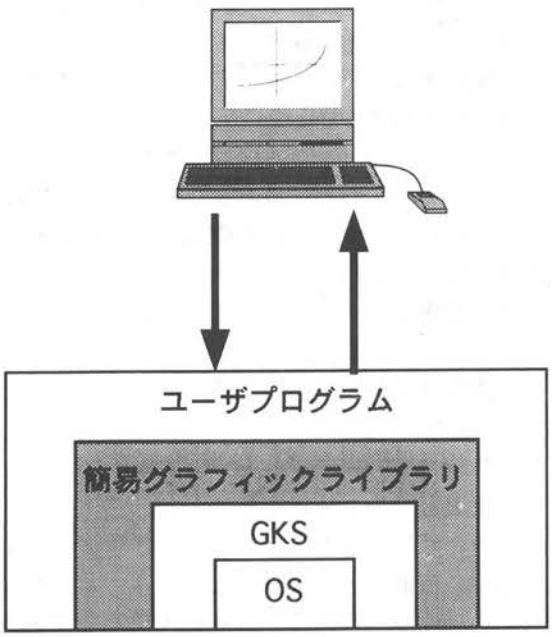


図1　簡易グラフィックライブラリの位置付け

●グラフィック関数

今回作成したグラフィックライブラリは制御，描画，入力，セグメント，その他の5つの機能に分かれており，制御関数6，描画関数12，入力関数4，セグメント関数9，その他3の合計33の関数から構成されている。表1に今回作成した関数群の一覧表を示す。

●ライブラリファイル "libglib.a" について

作成したグラフィック関数をすべてコンパイルし，オブジェクトファイルにした後，unixコマンドであるarコマンドおよびranlibコマンドを用いてアーカイブファイル "libglib.a" を作成し，これをライブラリファイルとする。

●ヘッダファイル "glib.h" について

今回作成したグラフィックライブラリを使用する場合，プログラム先頭においてヘッダファイル "glib.h" をインクルードしなければならない。このヘッダファイルでは，グラフィックライブラリ "libglib.a" 内で使用している関数のプロトタイプ宣言を行っている。またグラフィックライブラリ関数で使用している定数の定義用のヘッダファイル "def.h" および，GKSを使用するためのヘッダファイル "ansicgks" のインクルードも行っている。

●定数定義ファイル "def.h" について

ヘッダファイル "def.h" は，グラフィックライブラリ関数で使用する定数の定義を行っている。このヘッダファイルは，先に述べたヘッダファイル "glib.h" から参照されるようになっているので，ユーザは特に意識する必要はない。

●グラフィックライブラリの使用方法

グラフィックライブラリを使用する場合，プログラムの先頭で必ずヘッダファイル "glib.h" をインクルードする必要がある。その後，グラフィック関数の使用開始のための準備をする "gstart()" 関数を呼び出し，必要なグラフィック処理を行い，最後にグラフィック関数の使用終了手続きを行う "gstop()" 関数を呼び出す。

```
...
#include          <stdio.h>
...
main()
{
```

```
...
    gstart(xmin,ymin,xmax,ymax,rate);
...
    グラフィック関数呼び出し
...
    gstop();
...
}
```

●コンパイルの方法

グラフィックライブラリを使用したソースプログラムをコンパイルするときは"gc"コマンドを使用する。

```
...
gc　ソースファイル名
...
```

と入力するとソースプログラムのコンパイル，リンクが自動的に行われ，実行ファイルが作成される。"gc"コマンドはつぎのようなシェルスクリプトになっている。

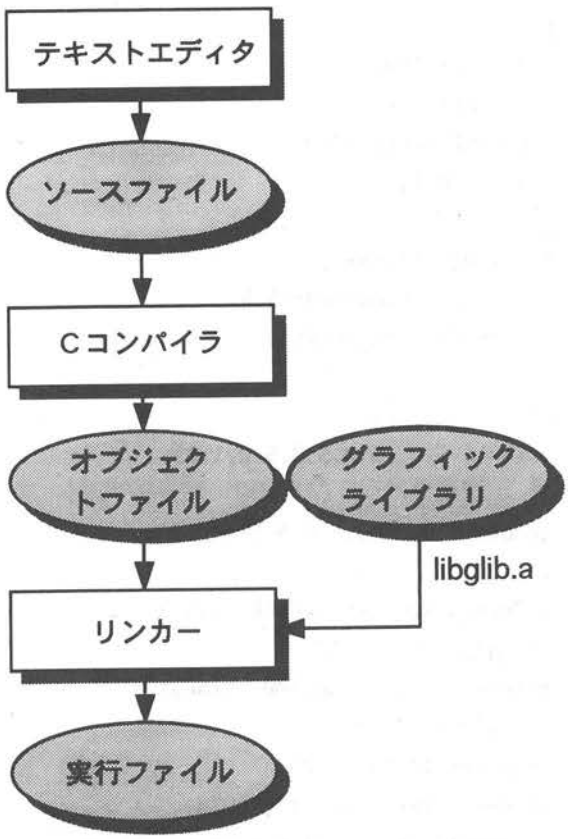


図2　プログラム作成の流れ

表1　グラフィック関数一覧表

制御関数	
gstart	グラフィック関数の使用開始宣言
gstop	グラフィック関数の終了宣言
gconsole	画面のモード変更
gcomment	コメントの表示
gcoord	座標軸の描画
描画関数	
gcls	画面の消去
gpset	点の描画
gline	線の描画1
glines	線の描画2
gbox	長方形の描画
gcircle	円，楕円の描画
garc	扇形の描画
glineset	線の設定
gmarkerset	点の属性設定
gtextput	文字列の描画
gtextset	文字の設定1
gtextloc	文字の設定2
入力関数	
gvinput	数値の入力
gcinput	文字列の入力
glinput	座標値の取得
gchoice	選択ボックスの使用
セグメント関数	
gsegopen	セグメントのオープン
gsegclose	セグメントのクローズ
gsegdel	セグメントの削除
gsegrename	セグメント番号の変更
gsegredraw	セグメントの再表示
gsegscale	セグメントの拡大・縮小
gtranroll	セグメントの回転
gtranshift	セグメントの直行座標移動
gsegset	セグメントの可視／不可視設定
その他	
gsave	グラフィックのセーブ
gload	グラフィックのロード
gprintout	グラフィックのプリンタ出力

```
cc  -I$GKSDIR/../../include
 -I$OPENWINHOME/include
 -I/usr/local/graphic/include
  ソースファイル名.c -o実行ファイル名
  -L$OPENWINHOME/lib
  -L$GKSDIR/.. -L/usr/local/graphic/lib
  -lglib -lgks -lxview -lolgx -lX11 -lm -lmle
```

3　プログラム例

ここで実際に，今回作成したグラフィックライブラリを使用したプログラム例をいくつか紹介する。

（例１）GKSウインドウを開いて座標軸のみを出力する。

```
/* ex1.c */
#include <stdio.h>
#include <stdlib.h>
#include <glib.h>

main()
{
  gstart(-100.0,100.0,100.0,100.0,_AUTO);
  gcoord(_ON,10.0,10.0,5,5);
  gcomment( "Already open GKS window." );
  gstop();
}
```

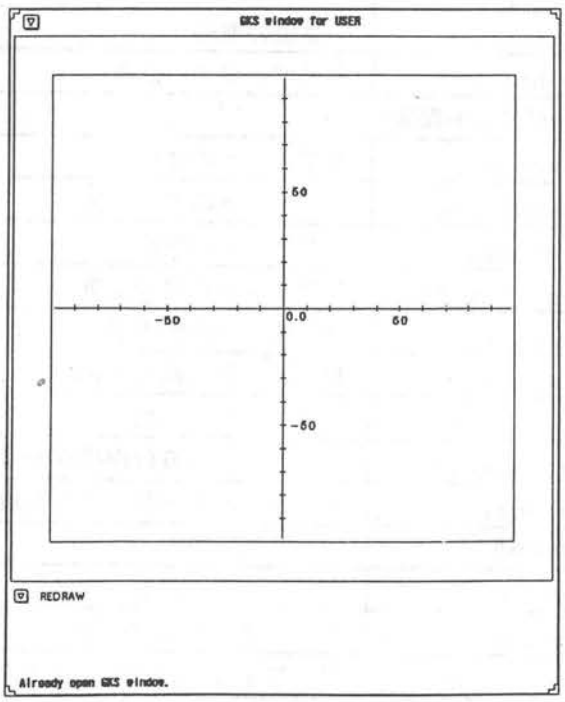


図３　例１の実行画面

（例２）$y = 0.5x$, $y = x^{\frac{1}{2}}$, $y = e^x$ のグラフを描く。

```
/* ex2.c */
#include <stdio.h>
#include <stdlib.h>
#include <glib.h>

main()
{
  int i;
  double x_sqrt[50],y_sqrt[50];
  double x_exp[100],y_exp[100];
  double x_pro[11], y_pro[11];
  double x1=0.0,x2=-5.0,a=0.5;
  /* 配列に計算値を代入 */
  for(i=0;i<50;i++){
    x_sqrt[i]=x1;
    y_sqrt[i]=sqrt(x1);
    x1 += 0.1;
  }
  for(i=0;i<100;i++){
    x_exp[i]=x2;
    y_exp[i]=exp(x2);
    x2 += 0.1;
  }
  for(i=0;i<11;i++){
    x_pro[i]=(double)i-5.0;
    y_pro[i]=a*x_pro[i];
  }
  /* グラフの作成  */
  gstart(-5.0,-5.0,5.0,5.0,1.0);
  /* 座標軸 */
  gcoord( _ON,1.0,1.0,5,5);
  /* べき乗のグラフ */
  glines(x_sqrt,y_sqrt,50,_LINE);
  /* 指数のグラフ */
  glines(x_exp,y_exp,100,_LINE);
  /* 比例グラフ */
  gmarkerset(_CIR,1.0);
  glines(x_pro,y_pro,11,_BOTH);
  gcomment( "Completion draw primitive." );
  gstop();
}
```

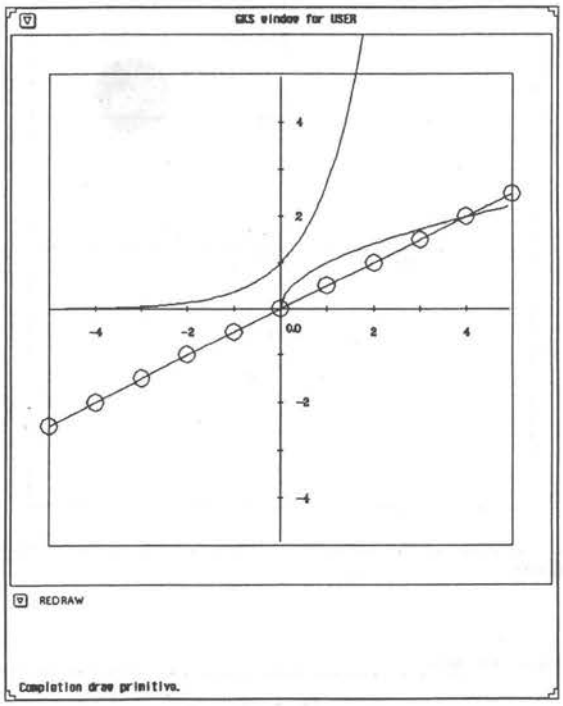


図4　例2の実行画面

（例3）ユーザの選択により，線，文字，箱，円，扇形を描いたり，消去を行う。

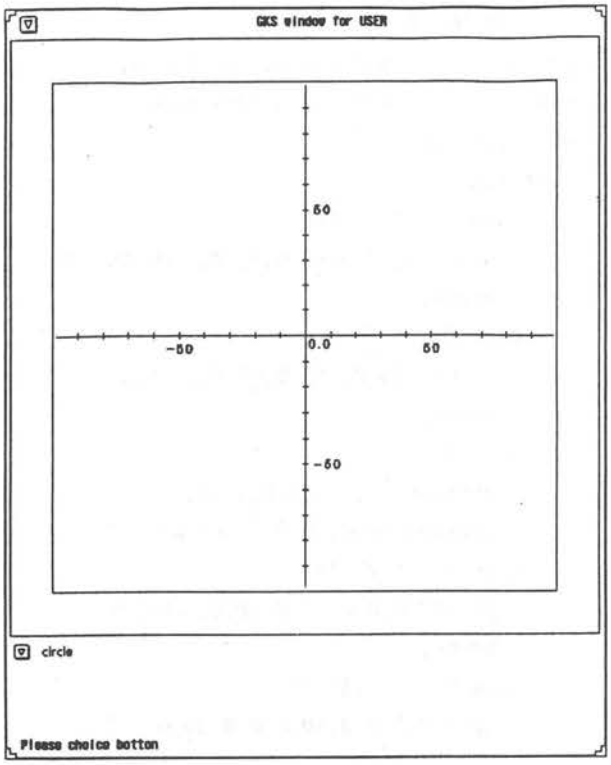
```
/* ex3.c */
#include <stdio.h>
#include <stdlib.h>
#include <glib.h>

main()
{
  int choice;
  int dev=1;
  int alts=7;
  static char *strings[]={ "circle" ,
                           "line" ,
                           "text" ,
                           "box" ,
                           "arc" ,
                           "cls" ,
                           "bye" };
  gstart(-100.0,-100.0,100.0,100.0,
         _AUTO);
  gcoord(_ON,10.0,10.0,5,5);
  gcomment( "Please choice botton" );
  choice=gchoice(dev,alts,strings);
  while(choice!=7){
    switch(choice){
      case 1: /* 円 */
        gcircle(60.0,60.0,15.0,1.0,_SOLD);
        break;
      case 2: /* 線 */
        gline(-45.0,-45.0,65.0,20.0);
        break;
      case 3: /* テキスト */
        gtextset(3,6.0,1.0,1.0);
        gtextput(0.0,-5.0," message" );
      case 4: /* 箱 */
        gbox(0.0,0.0,50.0,50.0,_HOLLOW);
        break;
      case 5: /* 扇形 */
        garc(0.0,0.0,50.0,0.0,90.0,1.0,1,
             _SOLD);
        break;
      case 6: /* 画面消去 */
        gcls();
        bcoord(_ON,10.0,10.0,5,5);
        break;
      case 7: /* 終了 */
        break;
    }
  choice=gchoice(dev,alts,strings);
  }
  gstop();
}
```

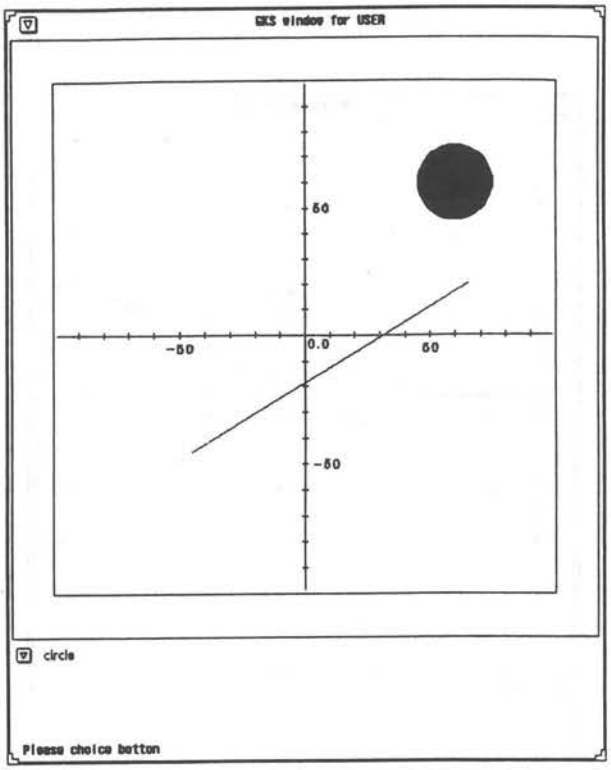
4　むすび

今回，ワークステーション用の簡易グラフィックライブラリを作成したことにより，以前に比べ，初心者にとってグラフィックプログラムが書きやすくなったと考える。基本的なC言語の知識だけで実験データのグラフ化等が可能である。今後，ワークステーションの有効活用の手段の1つとなることを期待する。

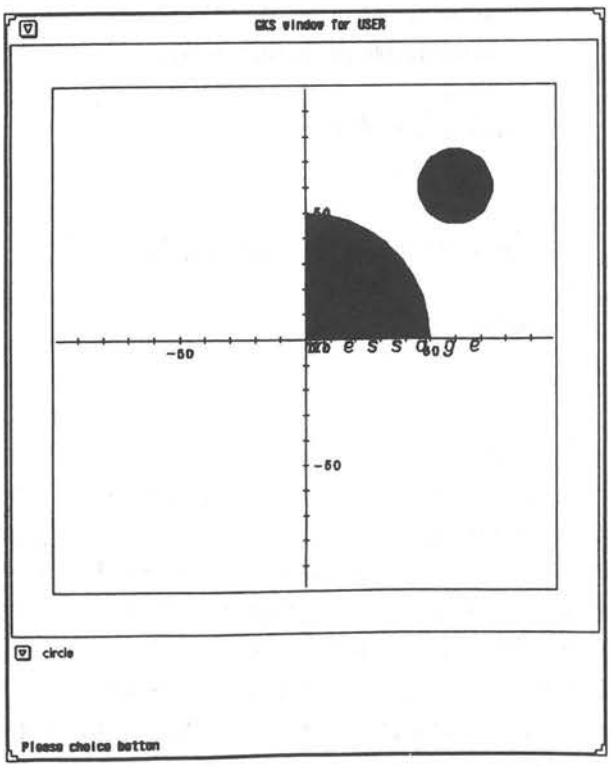
最後に，本研究にあたり多大な御協力を頂いた本校電子制御工学科平成8年卒業の諸裕之君ならびに吉田孝平君に感謝いたします。



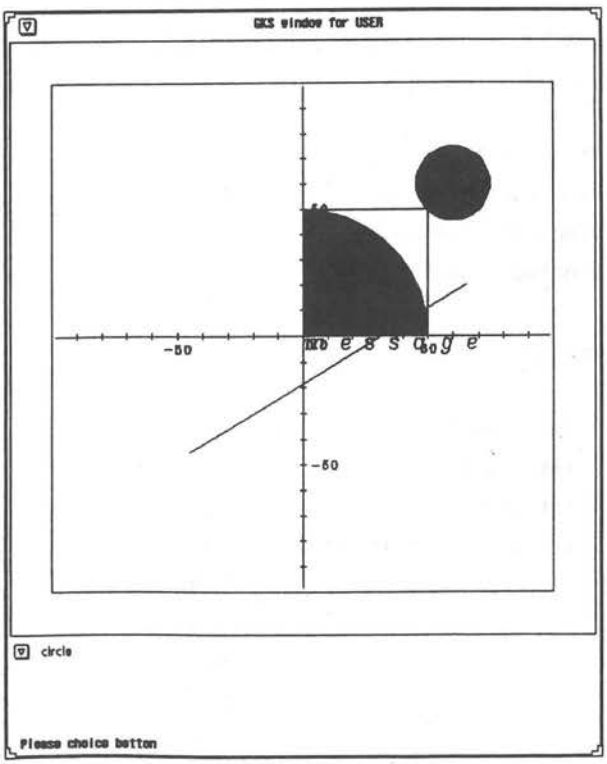
（ａ）　実行直後



（ｂ）　円と線を選択



（ｃ）　円，扇形，テキスト選択



（ｄ）　すべてを選択

図５　例３の実行画面

参考文献

（1）ルシア・マッケイ著，松本眞理訳，永塚匡編：ＧＫＳ入門　－ＧＫＳ　ＰＲＩＭＥＲ－，ノバ・グラフィックス・ジャパン（株）
（2）鈴木陽一：初めて学ぶＧＫＳ－国際標準コンピュータグラフィックスことはじめ－，共立出版
（3）岡田清，辻満夫：ＧＰＬＯＴの使い方，長岡高専情報処理センター
（3）金野茂男：グラフィックサブルーチン解説書，小山高専情報科学研究センター
（4）坂本文：ＵＮＩＸツール・ガイドブック，共立出版
（5）平林雅英：ＡＮＳＩ　Ｃ言語辞典，技術評論社

（受理年月日 1996年９月26日）