IMPROVEMENTS TO THE LGRAPHICS PACKAGE ON THE TCA TOKAMAK PDP 11-60

J.B. Lister and M.-H. Poget

Centre de Recherches en Physique des Plasmas
Association Euratom - Confédération Suisse
Ecole Polytechnique Fédérale de Lausanne
1. Introduction

The LGRAPHICS package (INT 76/79) was written as a high-level set of routines which used either, or both of, the PLOT-10 and Versaplot software packages to control the visual Display Units or Versatek printer/plotter respectively. This total package was fairly large, as is the data retrieval package. The small addressable memory in the PDP 11-60 meant that we had to overlay programs which performed any detailed calculations on whole traces. We have counteracted this problem by rewriting separately the VDU control package and the Versaplot package, with minor restrictions. This led to an increased speed of execution in the case of the old-fashioned coding of PLOT-10, and a reduction in the task-image size. In the case of the Versatek, a considerable reduction in user task-image size was obtained and speed was maintained. The Versatek dump program, on the other hand, was increased in size. We now discuss the two modifications and the restrictions they impose.

2. VDU

The PLOT-10 package is an extremely universal and cumbersome software (20 routines) which spends a great deal of time doing very little. We wrote a very simple routine, HPDRAW, which uses the HP264RA escape sequence commands and the fairly efficient redundant byte rejection algorithm of the HP2648A. The latter avoids sending bytes which repeat themselves, using bits in the "LOW-X" byte to define the rejection. We have no subroutine calls within HPDRAW: which saves time. We have lost the possibility of declaring blank windows and other such subtleties which were never used on TCA. The full screen is
used and the aspect-ratio of a VDU drawing is not the same as that of
the Versatek drawing, which works in absolute units of cm. The trans-
mission to the terminal is by use of an asynchronous write via a QIO
request called in the FORTRAN program. HPDRAW is reproduced in
Appendix I.

3. VERSATEK

We have rejected all the Versaplot routines, with the exception
of the MTX routine which sends the QIO to the Versatek and is written
in MACRO, and the symbol-drawing routines. We have made the major
restriction of "one page equals one drawing" which should not in fact
be restrictive since this was the only way in which LGRAPHICS was used
on TCA. Instead of a MAP file, we therefore have an explicit interme-
diate file which simply contains the full-page Vector data, in point-
line units, and which is extremely simple. The LOW-X and LOW-Y bytes
are always sent with the 7 least significant bits of vector data. The
eight bit of each determines whether the most significant byte is sent
or implicitly repeated. The most significant bytes contain the pen-up,
pen-down information. In addition, we define a "page" coordinate (0,0)
and an "end" coordinate (1,0) to complete the definition of the inter-
mediate file. VFILE•BIN is written sequentially in records of
512 Bytes by a synchronous write. In this way we only have a simple
user routine, VCREER (see Appendix II), which enormously reduces the
task-image file, by in excess of 6 K-Bytes. In addition, the Graphics
Library, VLIB, became so small that we have simply made two object
code modules LGRAPH•OBJ and LGLOG•OBJ. The former contains all the
graphics package with the exception of LOGAXE and LCONV which are held
in the latter. Task-Build time is thereby also reduced.

The Vector-to-Raster phase is performed by the program RAST (not
appended). This program decodes the VFILE•BIN file, page by page, and
creates a scratch sliced-vector file, all file operations being asyn-
chronous. This second file contains the start-stop vector information
of each slice containing N_LINES (74) lines, separately. Each slice is
then a separate "drawing" containing Vector information, packed as in
VFILE, but with the "local" Y-coordinate. During this preparation
phase (in VPREP) the Versatek is not attached. When a page is completed an end-of-page code is written and the next page is prepared until the end-of-data code is met. The final Vector-to-Raster operation is then performed on each page. The date is printed by the print-software before the first slice is dumped. The black-bit calculation is extremely tedious in FORTRAN and could be optimised in MACRO, if ever... When a slice is completed it is sent asynchronously to the Versatek and the alternate buffer is filled with the next slice.

It is not very elegant but it has a subtle advantage in that whole pages are always printed and the users do not waste days writing Overlay Descriptions. In addition a logical unit (1) has been freed for general use.

We note finally that the slice-size can be tailored to suit. Having small but many slices reduces the RAST task-image size, but slows the performance. A final advantage of the new system is that two drawings can very simply be merged, should the need ever manifest itself.
SUBROUTINE HPDRAW(X,Z,IC)
DIRECT DRAWING ONTO HP SCREEN
IC=2 PENDOWN
IC=3 PEN-UP
IMPLICIT BYTE Y
REAL Y0
LOGICAL LFLAG
INCLUDE 'COMTEK.FTN'
INCLUDE 'COMVTK.FTN'
DATA LGRAP/.FALSE./,IUP/42/.NBUF/0/
DATA YFF,VGS,VYS,VESC/12,29,31,27/
LGSX=.TRUE.
C IF PENDOWN KEEP ADDING BYTES
C IF(LGRAP.AND.IC.EQ.2)GOTO 1
START NEW GRAPH SEQUENCE
YBUFF(NBUF+1)=VGS
NBUF=NBUF+1
LGSX=.FALSE.
1 CONTINUE
XLAST=X+X0
ZLAST=Z+Y0
IX=XLAST*719/26.5
IZ=ZLAST*359/19.5
IF(IX.GT.719) IX=719
IF(IX.LT.0) IX=0
IF(IZ.GT.359) IZ=359
IF(IZ.LT.0) IZ=0
YXL=MOD(IX,32)+64
YLY=MOD(IZ,32)+96
YXH=IX/32+32
YYH=IZ/32+32
PENDOWN?
IF(LGRAP.OR.IC.EQ.3)GOTO 12
SEND OLD DATA
YBUFF(NBUF+1)=YXYH
YBUFF(NBUF+2)=YOYL
YBUFF(NBUF+3)=YXH
YBUFF(NBUF+4)=YOXL
NBUF=NBUF+4
LGSX=.TRUE.
12 CONTINUE
CHECK CHANGED OR JUST AFTER GS
C IF(LGSX.AND.YOYL.EQ.YYH.AND.YOXH.EQ.YXH)GOTO 21
YBUFF(NBUF+1)=YXYH
YXYH=YYH
NBUF=NBUF+1
C CONTINUE
C IF(LGSX.AND.YOYL.EQ.YYL.AND.YOXH.EQ.YXH)GOTO 22
YBUFF(NBUF+1)=YYL
C CONTINUE
YXYL=YYL
NBUF=NBUF+1
C CONTINUE
IF(LGSX.AND.YOXH.EQ.YXH)GOTO 23
YBUFF(NBUF+1)=YXYH
YXYL=YXL
NBUF=NBUF+1
C CONTINUE
YBUFF(NBUF+1)=YXYL
NBUF=NBUF+1
LGRAP=.TRUE.
IF(NBUF.LT.74)RETURN
ENTRY HPSEND
SEND CURRENT BUFFER, ENDING WITH US
IF(NBUF.EQ.0)RETURN
YBUFF(NBUF+1)=YUS
NBUF=NBUF+1
GOTO 9
ENTRY HPCLER
INITIALISE TERMINAL
SEND ERASE
CALL GETADR(IPARAM(1),YSEND(1))
CALL SETEF(24)
TB=SECNDS(0.0)-3
YBUFF(1)=VESC
YBUFF(2)=VFF
YBUFF(3)=VUS
YXH=8
YOYL=8
YOYL=8
NBUF=3
SEND THE BUFFER
9 CONTINUE
CALL WAITFR(24)
CHANGE THE BUFFER
DO 51 I=1,NBUF
YSEND(I)=YBUFF(I)
IPARAM(2)=NBUF
NBUF=0
SEND
5 C DT=SECNDS(T0)
C IF(DT.LT.8.12)GOTO 51
CALL QIOQ(4105241,ISP,IPARAM)
LGRAP=.FALSE.
T0=SECNDS(0.)
RETURN
END
SUBROUTINE VCREER
C
INCLUDE 'VERSTK.COM'
C
ENSEMBLE DE Routines DESTINEES A CReER UN BUFFER
C CONTENANT LES COORDONNEES DES POINTS A TRACER EN VALEURS ENTIERES
C
DIMENSION IBUF(256),YBUF(1)
EQUIVALENCE(IBUF,YBUF)
C
ENTRY VGO
C
OPEN(UNIT=2,NAME='VP:VFILE.BIN',TYPE='UNKNOWN',
   * ACCESS='SEQUENTIAL',ERR=900,
   * FORM='UNFORMATTED')
   GO TO 901
C TRY AGAIN. MAYBE LOCKED
900 OPEN(UNIT=2,NAME='VP:VFILE.BIN',TYPE='NEW',
   * ACCESS='SEQUENTIAL',FORM='UNFORMATTED')
901 CONTINUE
IPT=1
IY30LD=0
C IPT EST LE POINTEUR DE IBUF
C
RETURN
C
ENTRY VFILE(X,Z,IC)
C
ECRITURE DANS LE BUFFER CREE PAR VGO
C
IX=X*PTSY
IF(IX.LT.1) IX=1
IF(IX.GT.2847) IX=2847
IY=Z*PTSY
IF(IY.LT.1) IY=1
IF(IY.GT.IYMAX) IY=IYMAX
C
YBUF(IPT)=IAND(IY,127)
YBUF(IPT-1) = IAND(IY,127)
C
IY3=IAND(IY,1928)/128
IY3=IY3 - IAND(IY,1928)/8
C
IF(IC.EQ.3) YBUF(IPT)=YBUF(IPT) - 128
C
IPT=IPT+2
IF(IY3.EQ.IY30LD) GO TO 71
C
IY30LD=IY3
IF(IY3.GT.127) IY3 = 127 - IY3
YBUF(IPT)=IY3
YBUF(IPT-1)=YBUF(IPT-1) - 128
IPT=IPT+1
C
CONTINUE
C
GO TO 71
C FIN DE L'OPERATION AVEC FERMETURE DU FICHIER
C
WRITE(2) IBUF
C
RETURN
C
ENTRY VSTOP
C
BUFFER PLEIN ENVOYEE SUR DM:
C
GO TO 2
C
ENTRY VSTOP
C
BUFFLE DERNIER BUFFER
C
CLOSE(UNIT=2)
END