FLEX™ NEWSLETTER NO. 6
September 1982

Copyright (c) 1982 by Technical Systems Consultants, Inc.
111 Providence Road, Chapel Hill, NC 27514

There has been one big change at Technical Systems Consultants, Inc. since the last issue of the FLEX Newsletter was mailed. We moved! Actually, you all should already know this since a change of address notice was mailed to all subscribers. For any who did not hear the news, our new address, phone, and Telex are:

Technical Systems Consultants, Inc.
111 Providence Road
Chapel Hill, NC 27514   USA

Telephone: (919) 493-1451
Telex II: 510-920-0540

We have greatly increased our facilities and are in the process of increasing our staff. The result will be more software and better service for our customers.

1) FLEX™ News

The FLEX Disk Operating System is now running on Radio Shack's TRS-80 Color Computer. Two firms have licensed FLEX and are selling versions that are ready to run on the Color Computer. These are:

Frank Hogg Laboratory
130 Midtown Plaza
Syracuse, NY 13210
(315) 474-7856

Spectral Associates
139 Harvard Avenue
Tacoma, WA 98466
(206) 565-8483

Another firm offers a set of patches which can be used with our General version of FLEX to produce a version of FLEX for the Color Computer. The firm is Data-Comp, P.O. Box 794, Chattanooga, TN 37443.

With any of these versions, you must start with a Color Computer modified to contain 64K of RAM. What you end up with is a powerful little system at a very low cost. The system uses standard FLEX disks and is therefore compatible with software available from any of the many FLEX-based software vendors. The only limitation a user sees is the number of lines and characters per line displayed on the Color Computer CRT monitor.

For some reason, a number of people have expressed concern that Technical Systems Consultants might drop support of FLEX and put all efforts into UniFLEX. This is totally unfounded. It is true that a considerable portion of our development work is now done on UniFLEX and on the 68000, but this in no way indicates an abandonment of FLEX. On the contrary, we plan to continue our support of FLEX indefinitely. Our support of 6800 FLEX will be only in the form of sales and support of existing products - we do not plan to produce new software for the 6800. For the 6809, however, we have several new products available, under development, or in the planning stages. You can read about two new products in the next section. We also still have our 6809 C Compiler under development. This project has seen several delays and is still a ways from completion, but there will be a FLEX version. Anyway, the point is this: do not be concerned about support of FLEX. It will be around for a long time.

2) New Products

We have two new products to announce in this newsletter, a 6809 FORTRAN compiler and a Relocating Assembler and Linking Loader package. A brief description of each follows. If you have further questions, feel free to contact Technical Systems Consultants, Inc. or your dealer directly.

<u>6809 FLEX FORTRAN 77</u>

This is a true, 6809 native-code FORTRAN 77 Subset compiler. It produces assembler language output compatible with Technical Systems Consultants' relocating assembler and linking loader. This implementation conforms to the ANSI FORTRAN 77 (ANSI X3.9-1978) subset of the FORTRAN language, with the following exceptions:

The INTRINSIC and SAVE statements are ignored.
The EQUIVALENCE statement is not implemented.
The BACKSPACE statement is not allowed.
The ENDFILE statement performs no useful function.
Statement functions are not supported.
Variable names may be of any length with 7 characters significant.
All keywords are reserved names.
Direct access files are not available under FLEX.

In addition, Technical Systems Consultants' FORTRAN contains some features of the full FORTRAN language, most notably list-directed I/O and expanded form of the OPEN statement. Also included in the extensions are the ability to open any file name and access to command line arguments. The FORTRAN library includes modules for 16.8 digit floating point arithmetic, all standard scientific functions, complete file manipulations, runtime trace back features, and post-mortem dump capability. 6809 FLEX FORTRAN requires a full 56K FLEX system and requires the FLEX Relocating Assembler and Linking Loader.

6809 FLEX Relocating Assembler and Linking Loader

Technical Systems Consultants now has available a full 6809 relocating assembler and linking loader package. Three separate programs are included in the package, the relocating assembler, the linking loader, and a library generator program. The assembler accepts standard 6809 instruction mnemonics (it also accepts 6800 mnemonics) but does not conform to any established directives for control of the relocatable output. Output of the assembler can be relocatable modules or absolute modules. The assembler supports macros and conditional assembly as in our standard 6809 assembler. The linking loader accepts multiple modules and libraries and performs the desired relocation, linking, and satisfying of external symbols. Output is either a relocatable module or an executable absolute module. The library generator program allows a user to create his own specialized libraries of relocatable modules.

Both the FORTRAN and the Relocating Assembler packages are available immediately on 5 or 8 inch floppy diskettes. The FORTRAN 77 Compiler alone is part number SP09-16 and costs $275.00 (Note that the Relocating Assembler is required in order to use SP09-16). The 6809 Relocating Assembler and Linking Loader alone is part number SP09-17 and costs $150.00. Both of these packages may be purchased together at the same time for a discounted price of $375.00. This combined package of the FORTRAN 77 and Relocating Assembler is part number SP09-18.

3) Current Versions

Once again we are listing the current versions of our FLEX based software products. Our update policy for FLEX software is as follows: If you have owned a package for under two months an update is free. Beyond two months there is a $10.00 updating fee. To obtain an update, you must return the original disk, or supply proof of purchase and an additional $10.00 for us to supply a new disk. The following version numbers are current as of September 22, 1982.

Program Name	6809 Version	6800 Version
Extended BASIC	24	22
BASIC	17	15
6809 Pascal	12	-
FORTRAN 77	3	
Relocating Asinb & Linking Loader	1	-
Extended BASIC Precompiler	4	2
BASIC Precompiler	3	2
Text Editing System	2	n/a
Assembler	2	n/a
Text Processing System	4	n/a
Sort/Merge	3	3
Debug	19	n/a
6809 Cross Assembler	-	2
68000 Cross Assembler	7	-

4) FLEX Based Products from Other Firms

There is a great wealth of FLEX-based software available from various vendors. One such package which became available recently might be of interest to many of you. It is called DYNACALC™ (trademark of Computer Systems Center) and is a powerful spread-sheet calculator much like the popular VisiCalc™ (trademark of VisiCorp). The program is available from:

Computer Systems Center
13461 Olive Blvd.
Chesterfield, MO 63017
Phone: (314) 576-5020

We have not actually run the FLEX version, but we did receive a copy of the UniFLEX version and it seems to run quite well. VisiCalc-like programs have been in high-demand by users of most all personal computers and we are happy to now see this capability under FLEX.

5) FLEX™ Tips

A number of people have expressed confusion over the "PRINT USING" statement in our Extended BASIC package. This is a very powerful feature and while we feel the manual's description of PRINT USING is sufficient, a few examples may be beneficial to the novice user. Examples of various PRINT USING techniques follow:

A) STRINGS (Back Slash):

```
BASIC  => PRINT USING '\23456\', 'THE RAIN IN SPAIN FALLS'
Output => THE RAI
Comment=> Total of 7 characters printed (count both backshlashes).
```

B) NEGATIVE NUMBERS (using the Pound Sign):

```
BASIC => PRINT USING  "####", -235
Output => -235
Comment=> A minus sign can be printed before number using pound sign.
```

C) FLOATING DOLLAR SIGN ($):

```
BASIC   => PRINT USING "$$###,###.##", 23.05
Output  =>      $23.05

BASIC   => PRINT  USING  "$$###,###.##", 38293.4
Output  =>   $385,293.40

BASIC   => PRINT USING "$$###.##-", -3.5
Output  =>    $3.50-
Comment=>  Note leading spaces, right justification, & floating "$".
```

D) $ and * RESERVE ONE NUMERICAL SPACE:

```
   BASIC  => PRINT USING '$$#.## W/ PROTECTED FIELD IS $**#.##', 12.5, 12.5
   Output => $12.50 W/ PROTECTED FIELD IS $*12.50
   Comment=> Both $ and * reserve a space for a numerical field.
   In this case it is the tens position.
```

E) EXAMPLE WITH SEVERAL TYPES OF PRINT USING:

```
   BASIC  => PRINT USING "THE BALANCE OF \23456789\ IS $##,###.##- AND
             $**#,###.##", 'AJAX, INC.', -2345.7, 3568.91
   Output => THE BALANCE OF AJAX, INC. IS $ 2,345.70- AND $**3,568.91
   Comment=> Here, the BASIC statement was broken across two lines.
             In a real BASIC program, it must be one physical line.
```

6) Free FLEX™ Utility!

This issue of the FLEX Newsletter includes a free utility for renaming a FLEX disk. When a FLEX disk is created, the user may specify a volume name and a volume number. This information is stored in the "System Information Record" of the disk (track 0 sector 3), along with the date on which the disk was created. The utility we are giving you permits the user to alter any or all of these three items. The utility is called "NAMEDISK" and the calling line should look like this:

```
      +++NAMEDISK <Drive No.>
```

where <Drive No.> is a valid drive number between 0 and 3 (inclusive). The program will then prompt for the new volume name, volume number, and creation date. If the user wishes to leave any of these items as is, he may simply hit a carriage return in response to the prompt for the item.

Technical Systems Consultants, Inc. makes no guarantees on the operation of this program and may not be held responsible in any way for any consequences of it's use. Furthermore, no technical support of any form will be provided.

The complete, assembled source listing follows. You will note that the program is written in 6809 assembler language. The user may certainly convert the program to 6800 if desired.

```
* NAMEDISK UTILITY
*

* COPYRIGHT (C) 1982 BY
* TECHNICAL SYSTEMS CONSULTANTS, INC.
* CHAPEL HILL, NC 27514
*

* THIS PROGRAM PERMITS THE USER TO GIVE A NEW NAME, VOLUME
* NUMBER, AND DATE TO THE SPECIFIED DISK. CALLING LINE IS:
*                +++NAMEDISK <drive no.>
* AND THE PROGRAM WILL PROMPT FOR ALL NECESSARY INPUT.  IF
* A PARTICULAR ITEM IS TO BE LEFT AS IS, SIMPLY HIT RETURN.

                  * EQUATES

            D403  FMSCLS  EQU    $D403
            D406  FMS     EQU    $D406
            CD03  WARMS   EQU    $CD03
            CD2D  GETFIL  EQU    $CD2D
            CD42  GETHEX  EQU    $CD42
            CD48  INDEC   EQU    $CD48
            CD1E  PSTRNG  EQU    $CD1E
            CD1B  INBUF   EQU    $CD1B
            CD3F  RPTERR  EQU    $CD3F
            C840  FCB     EQU    $C840
            CC14  BUFPNT  EQU    $CC14
            C880  SIR     EQU    FCB+64

C100                      ORG    $C100

C100  20   1A     NAMDSK  BRA    NMD1
C102  01                  FCB    1        SHOW VERSION NUMBER 1

                                          TEMPORARY  VARIABLES

C103 00           DRIVE   FCB    0
C104 00           NAMSET  FCB    0
C105 00           VOLSET  FCB    0
C106 00           DATSET  FCB    0
C107 0000         VOLUME  FDB    0
C109 00 00 00     DATE    FCB    0,0,0
C10C 0000 0000    FAKE    FDB    0,0,0,0
C114 0000 0000            FDB    0,0,0,0

                  * MAIN ROUTINE STARTS HERE

C11C BD   CD42    NMD1    JSR    GETHEX    GET DRIVE NUMBER
C11F 1025 00DB            LBCS   ERROR
C123 5D                   TSTB             ANY DRIVE SPECIFIED?
C124 1027 00D6            LBEQ   ERROR     ERROR IF NOT SPECIFIED
C128 1F   10              TFR    X,D
C12A 1083 0003            CMPD   #3        ENSURE 0 TO 3
C12E 1022 00CC            LBHI   ERROR
C132 F7   C103            STB    DRIVE     SAVE DRIVE NUMBER
```

```
C135 8E   C20F            LDX    #NPRMPT   PROMPT FOR NAME
C138 BD   CD1E            JSR    PSTRNG
C13B BD   CD1B            JSR    INBUF     GET REPONSE
C13E A6   9F CC14         LDA    [BUFPNT]  CHECK FOR NULL RESPONSE
C142 80   0D              SUBA   #$0D      CARRIAGE RETURN?
C144 27   08              BEQ    GOTNAM    SKIP IF SO (NO NAME)
C146 8E   C10C            LDX    #FAKE     POINT TO FAKE FCB
C149 BD   CD2D            JSR    GETFIL    GET NAME FROM INPUT
C14C 86   01              LDA    #1        NAME SPECIFIED FLAG
C14E B7   C104    GOTNAM  STA    NAMSET    SET FLAG

C151 8E   C222    GETVOL  LDX    #VPRMPT   PROMPT FOR VOLUME NO.
C154 BD   CD1E            JSR    PSTRNG
C157 BD   CD1B            JSR    INBUF     GET RESPONSE
C15A BD   CD48            JSR    INDEC     CONVERT TO DECIMAL
C15D 25   F2              BCS    GETVOL    LOOP IF ERROR
C15F F7   C105            STB    VOLSET    VOLUME SPECIFIED FLAG
C162 BF   C107            STX    VOLUME    SAVE VOLUME NUMBER

C165 8E   C232    GETDAT  LDX    #DPRMPT   PROMPT FOR DATE
C168 BD   CD1E            JSR    PSTRNG
C16B BD   CD1B            JSR    INBUF     GET RESPONSE
C16E A6   9F CC14         LDA    [BUFPNT]  CHECK FOR NO DATE
C172 80   0D              SUBA   #$0D      CARRIAGE RETURN?
C174 27   17              BEQ    GOTDAT    SKIP IF SO (NO DATE)
C176 8D   1A              BSR    GETITM    GET MONTH
C178 25   EB              BCS    GETDAT
C17A F7   C109            STB    DATE
C17D 8D   13              BSR    GETITM    GET DAY
C17F 25   E4              BCS    GETDAT
C181 F7   C10A            STB    DATE+1
C184 8D   0C              BSR    GETITM    GET YEAR
C186 25   DD              BCS    GETDAT
C188 F7   C10B            STB    DATE+2
C18B 86   01              LDA    #1        GET DATE SET FLAG
C18D B7   C106    GOTDAT  STA    DATSET    SET FLAG
C190 20   10              BRA    PUTINF    GO WRITE INFO

                  * GET DATE ITEM

C192 BD   CD48    GETITM  JSR    INDEC     GET NUMBER
C195 25   0A              BCS    GETIT2
C197 5D                   TSTB             ANY NUMBER SPECIFIED?
C198 27   05              BEQ    GETIT1
C19A 1F   10              TFR    X,D
C19C 1C   FE              CLC              CLEAR ERROR
C19E 39                   RTS
C19F 1A   01      GETIT1  SEC              SET ERROR
C1A1 39           GETIT2  RTS

                  * THIS ROUTINE STORES NEW VALUES INTO SIR

C1A2 8E   C840    PUTINF  LDX    #FCB      FIRST READ CURRENT SIR
C1A5 F6   C103            LDB    DRIVE     SETUP DRIVE NUMBER
C1A8 E7   03              STB    3,X
```

```
C1AA CC   0003            LDD    #$0003    TRACK 0 SECTOR 3
C1AD ED   88 1E           STD    30,X      PUT SIR ADDRESS INTO FCB
C1B0 86   09              LDA    #9        READ SINGLE SECTOR CODE
C1B2 A7   84              STA    0,X
C1B4 BD   D406            JSR    FMS       CALL FMS
C1B7 26   4D              BNE    DSKERR
                  * PUT IN NEW VALUES
C1B9 7D   C104            TST    NAMSET    NAME SPECIFIED?
C1BC 27   10              BEQ    DOVOL     SKIP IF NOT
C1BE 8E   C110            LDX    #FAKE+4   POINT TO NAME
C1C1 108E C890            LDY    #SIR+16   POINT TO CORRECT SPOT
CIC5 C6   0B              LDB    #11       NUMBER OF BYTES TO COPY
C1C7 A6   80      COPY    LDA    0,X+
C1C9 A7   A0              STA    0,Y+
C1CB 5A                   DECB
C1CC 26   F9              BNE    COPY
C1CE 7D   C105    DOVOL   TST    VOLSET    VOLUME SPECIFIED?
C1D1 27   06              BEQ    DODAT     SKIP IF NOT
C1D3 FC   C107            LDD    VOLUME
C1D6 FD   C89B            STD    SIR+27    PUT INTO CORRECT SPOT
C1D9 7D   C106    DODAT   TST    DATSET    DATE SPECIFIED?
C1DC 27   0C              BEQ    WRTINF    SKIP IF NOT
C1DE FC   C109            LDD    DATE
C1E1 FD   C8A3            STD    SIR+35    PUT INTO CORRECT SPOT
C1E4 B6   C10B            LDA    DATE+2
C1E7 B7   C8A5            STA    SIR+37
                  * NOW WRITE SIR BACK TO DISK
C1EA 8E   C840    WRTINF  LDX    #FCB
C1ED CC   0003            LDD    #$0003    TRACK 0 SECTOR 3
C1F0 ED   88 1E           STD    30,X      PUT SIR ADDRESS INTO FCB
C1F3 86   0A              LDA    #10       WRITE SINGLE SECTOR CODE
C1F5 A7   84              STA    0,X
C1F7 BD   D406            JSR    FMS       CALL FMS
C1FA 26   0A              BNE    DSKERR
C1FC 20   0B              BRA    EXIT

                  * ERROR ROUTINES

C1FE 8E   C24D    ERROR   LDX    #ERRS     REPORT INVALID DRIVE
C201 BD   CD1E            JSR    PSTRNG
C204 20   03              BRA    EXIT
C206 BD   CD3F    DSKERR  JSR    RPTERR
C209 BD   D403    EXIT    JSR    FMSCLS
C20C 7E   CD03            JMP    WARMS

                  * STRINGS

C20F 44 49 53 4B  NPRMPT  FCC    'DISK VOLUME NAME? ',4
C222 56 4F 4C 55  VPRMPT  FCC    'VOLUME NUMBER? ' ,4
C232 43 52 45 41  DPRMPT  FCC    'CREATION DATE (MM,DD,YY)? ',4
C24D 4D 55 53 54  ERRS    FCC    'MUST SPECIFY VALID DRIVE NUMBER#,4

                          END    NAMDSK
```