ALPHA

MICRO

USERS

SOCIETY

Newsletter

VOL. 1 no. 8 & 9 Sept. & Oct., 1978

AMUS NEWSLETTER
VOLS. 8 & 9

SOFTWARE BUGS, FIXES, ANNOUNCEMENTS, COMPLAINTS & COMMENTS VERSION 3.4

We will list some of the "foibles of AMOS & ALPHA" in as decent an order as we can. If it is out of sequence or you don't understand what we're trying to do, just wait: it WILL get better.

AlphaBASIC may generate & display a 'SYNTAX ERROR' statement when the program command [ON (statement expression) GOTO] is used.

Also, there are a lot of flaky things that happen with the error trapping. Use at your own risk.

Curt Rostenbach has sent us a patch for LPTSPL that will print from DEVICE. other than DSK: if such is named in the PRINT statement. Also, the extensions:

/K	kills the file in QUEUE and also kills any printing in progress.
/L	same as no option=LIST
/D	deletes file from LIST after printing

NOTE: Curt is sending this patch to AMUS and we will include it in the next newsletter and on the utility and game disk.

Phil Hunt in Columbus, Ohio tells us that a null string used in a file specification in the OPEN statement crashes the system. For those of you who don't believe, just try:

```
        10 TEST$ ="     "
        20 OPEN #1, TEST$, INPUT
   or:
        10 OPEN #1, " ", INPUT
```

Also, he says that the CLOSE and TAB work in different ways than most BASICs......

CLOSE on an unopened file now causes an "I/O TO UNOPENED FILE" error message. This will mess up looped I/O. OPEN really should do an implicit CLOSE, but instead, you get a "FILE ALREADY OPEN" message.

TAB does not ignore TAB if it is already past the argument position. It will do a CR (carriage return) NOT A LINE FEED and goes to the position specified in the arguement. This is a handy way to do underlining, but you should be aware of it for other cases where you don't want stuff written on top of a previous line.

If you load a program form disk into BASIC and type a new line containing an array variable with subscripts, and the array was defined in a previous MAP statement, you will get a "WRONG NUMBER OF SUBSCRIPTS" error message. Recompiling the program will allow you to insert the new array variable.

If an "OUT OF MEMORY" message occurs during a BASIC COMPILE, it will cause an ASCII memory dump of the program at odd times. A 'control C' will return you to READY, but any other command returns to the dump. Version 4.0 cures this problem.

Nobody seems to be able to get PASCAL to use alternate work libraries or read the SYSTEM WORK.LIB to edit or change a system PASCAL file. Bob Dougherty at CRPLi in Boulder says he got around the problem by copying the files to another disk, then emptying their contents. Of course, he was still limited to 50 blocks. There will be a discussion of PASCAL at the Gala Inn Towne Hotel during the seminars at the WCCF.

We find that DSKANA might run faster if there were an option to delete the printout of block numbers.

Control C doesn't seem to work when you compile a BASIC program with /O because AlphaBASIC can't resume at specific line numbers. Version 4.0 corrects this problem.

Phil Hunt says that the macro assembler still hangs up in phase 4 (LINK). Alpha Micro has said that version 3.4 has been corrected but we can't all be making this many misteaks, can we? Also, is there a way to recover from ERROR 20's? (ERROR 10's can be fixed with DSKDDT, but not ERROR 20's) Phil also says that he has a program that recovers files lost because of ERROR 20's in the directory blocks. Eds note: ERROR 20's are caused because the disk cannot locate the block you are looking for. Reformatting the diskette will usually solve the problem if the diskette is not physically bad. If you must recover the data on the diskette, you might try adjusting one of the potentiometers (on Persci drives only) or adjusting the heads themselves. How to adjust the "pots" is described in an earlier AMUS newsletter.

BASORT only works if the key is a string. If a binary key is used, the items being sorted must fit in main memory. Polyphase merge sort does not work in version 3.4, but 4.0 cures the problem.

PRINT USING ignores all statements after a RESTORE command. Any statement after PRINT USING is also ignored unless that statement is enclosed in parentheses. Speaking of RESTORE, you cannot use RESTORE in an IF THEN statement.

Richard McClendon says that when structuring ISAM files, a CODE 2 by itself will skip one record after a record delete. You must use a CODE 1 and then a CODE 2 to maintain file record continuity.

If you haven't noticed by now, AlphaBASIC will accept all of the following variables in a program as separate variables:

```
THING   THING$  THING(N)   THING(N,J) .
```

You can also have a subroutine named THING: if you wish in the same program.

4.0 AlphaBASIC WILL NOT RUN PROGRAMS COMPILED IN VERSION 3.4.

Version 4.0 will not CHAIN to programs in another PPN.

Any system command file that is exactly 510 bytes long will not work. The link pointer gets written over and the file cannot get to the next block.

AMOS BUGS (Version 3.4)

(1) ALPHABASIC --- bad syntax within IF statements is not detected

try entering   IF(0) THEN PRINT "A" ELSE PRINT "[illegible]
it gets no syntax error.

(2) ALPHABASIC --- UCS,LCS functions go ape on null strings

Try typing   PRINT UCS("")
it crashes the system.

(3) ALPHABASIC --- FILEBASE command isn't documented

The user may set the filebase record number by including
    FILEBASE n
in an ALPHABASIC program, where n is an integer. Thereafter, whenever a READ or WRITE is made to a random file, the record number actually accessed will be equal to the value of the record number variable for this file decreased by n. Effectively, this says that the first record number in the random file is #n.

(4) ALPHABASIC --- ALLOCATEing too large a file crashes the system

If there is not enough contiguous room on the available disk (floppies; haven't tried a hard disk), then ALLOCATE will return a coherent error message, but the system is very sick and will do some of the most unpredictable things I've ever seen. Buss error, dead, crash an MFD, etc.

(5) ALPHABASIC --- BASORT doesn't work

Just for the record, here is the fix from Alpha Micro :

```
.DDT BASORT.PRG
PROGRAM BASE IS 132442
PROGRAM SIZE IS 1556
100/    MOV #1,36    MOV #1,36(R4)
ctrl-C
```

where the underlined parts are given by AMOS.

AMOS BUGS (Version 3.4)

(6) ALPHABASIC --- too bizarre to summarize on one line ....

Enter the following lines in BASIC :

```
    100 OPEN #1, "TRM:PRINTR", OUTPUT
    200 FOR I=1 TO 10
    300   PRINT #1
    400   PRINT #1, USING "#", VAL("1")
    500 NEXT I
    600 CLOSE #1
    700 END
    RUN
```

where PRINTR is the TRMDEF name for the printer being used (if it makes any difference, ours runs on the TELTYP.TDV driver). When the program runs, the output from line 300 correctly comes out on the printer, but the output from line 400 comes out on the user's crt !! The FOR-NEXT loop is not part of the bug, but it makes the blank lines printed stand out. The bug only occurs with PRINT USING, with a numeric USING field, with a VAL function. Take any of these out and it's ok.

(7) KILL --- small letters on a job name causes crash

Also small letters for any filename in Basic

Try typing ;

```
    KILL Job2
```

where JOB2 is a jobname. Or try one of your own jobnames.

(8) SORT.PRG --- the merge phase doesn't work

If a file fits into memory, SORT works fine. If the file is larger than available core, then a lot of time goes by, and at the end, an AMOS "." is given, but the original file is now empty (SIZE is 1 byte).

(9) SPOOLER --- can't PRINT from any DSK except 0 or 1

Matson & Moss in SanLuis Obispo report that two dual Persci drives can now execute the same program from different drives by different users, but a new bug has crept in which wasn't there before. If a file is spooled from DSK2 or DSK3, the system will promptly go dead. This happens whether the user is logged onto DSK0,1,2 or 3.

AMOS BUGS (Version 3.4)

(10) ADDS.TDV --- watch out for TAB(-1,n) agreement with other TDVs

The ALPHABASIC documentation describes the TAB(-1,n) features for n from 0 to 16, The features of n up to 19 are

17 --- delete character
18 --- insert character
19 --- read cursor address

These agree with the HAZEL.MAC and ACTIV.MAC listings. Technically, these should have been added to the ALPHABASIC manual before Version 3.3. In addition, the features 11,12 should be re-worded in the manual. As of now they read,

11 --- Protected field (reduced intensity)
12 --- Unprotected field (normal intensity)

This conforms to SOROC crts. The Hazeltine,however reverses the relation of protected fields to intensity level (using reduced intensity for unprotected fields). Accordingly to the HAZEL.MAC listing, the manual should now read

11 --- low intensity
12 --- high intensity

And the protection status would follow the particular crt's convention. Now, to get to the ADDS.MAC (remember ?). It does not have any features 11 or 12 implemented; this is probably because this would interfere with any other of the "attributes" that may be set on the ADDS crt beforehand. Think over that yourselves; it is an arbitrary choice. However, the features 17 to 19 do not agree with the other TDVs (above), and the features from 20 on should be promptly added to the BASIC manual. If it helps, I can summarize all the "attributes" for the ADDS in terms of bits (the manual omits this). Thus, to set an ADDS attribute, send out the ASCII sequence :

CHR(27)+CHR(48)+CHR(n)

where the bit configuration of n sets each attribute as follows

```
n =   64
    + 32 (to activate underlining)
    + 16 (to activate reverse video)
    +  4 (to activate zero intensity, for blank echo)
    +  2 (to activate blinking)
    +  1 (to activate half intensity)
```

By doing some logical bit setting, these attributes could probably be set independently by the ADDS.MAC driver, without disturbing the other attributes.

[illegible] INCORPORATED

500 [illegible] AVENUE, PASADENA, CALIFORNIA 91101 TELEPHONE (213) 684-33[illegible]

AMOS BUGS (Version 3.4)

[illegible] --- cannot open disk file (or printer) for output

See listing on following page. In essence, if a batch job (with a PSEUDO terminal driver) tries to open a disk file from BASIC, it will get hung up. The same thing happens if the batch job tries to open the printer for output, using the

```
    OPEN #1, "TRM:PRINTR", OUTPUT
```

statement to open the printer. This happens even if the memory given to the batch job is the same as the user's crt, where the program executes correctly.
Without these two capabilities in a batch job, the batch job loses most of its practical value (except as the spooler).

SYSTEM.INI [1,4]

```
:T
JOBS JOB1,BATCH
TRMDEF CRT,AM300=1:16,ADM3,100,100,200
TRMDEF NOCRT,PSEUDO,NULL,25,25,2
DEVTBL DSK1,TRM/LPT
BITMAP DSK,32,0
BITMAP DSK,32,1
SYSTEM BASIC
SYSTEM RUN
SYSTEM
CLKFRQ 60
ATTACH NOCRT,BATCH
FORCE BATCH MEMORY 4K
WAIT BATCH
MEMORY 4K
```

BOB.CMD [2,2]

```
LOG DSK0:1,4
BASIC
OPEN #1, "BOB.DAT", OUTPUT
CLOSE #1
BYE
```

① BOOTSTRAP AMOS USING SYSTEM.INI LISTED HERE

② EXECUTE BOB.CMD FROM JOB1 BY TYPING "BOB", & THE NET END RESULT IS THAT "BOB.DAT" IS NOW IN THE DIRECTORY [THIS IS OK]
[ERASE BOB.DAT BEFORE PROCEEDING]

③ EXECUTE BOB.CMD FROM BATCH JOB BY TYPING "FORCE BATCH BOB" & THE NET RESULT IS NO "BOB.DAT" IN DIRECTORY, & A HUNG BATCH JOB,

COMPUTER SYSTEMS INCORPORATED

500 SOUTH LAKE AVENUE, PASADENA, CALIFORNIA 91101 TELEPHONE (213) 684-3311

AMOS SUGGESTIONS (Version 3.4)

(1) TXTFMT --- simultaneous right/left justify

Some other processors do it. We'd like it. Several customers have said they'd really like it. I've just started using the TXTFMT program, and I'd sure love it. Just find out the number of blanks in the line, divide by the number of word gaps, and if the answer isn't an integer, then stuff the blanks in via some simple pattern that alternates between the lesser number of blanks and the greater.
Also, I might have missed it, but I think there is no right justify command.

(2) BASIC source code limit --- about 90 blocks ??

We have compiled off disk a BASIC source program with 90 blocks. It required COMPIL/O (no line numbers).

(3) BASIC MAP statement --- allow initialization of array values

The ALPHA Business Software has at least one line of source code that uses the following type of syntax :

```
    MAP1 A$(5),S,1,"A","B","C","D","E"
```

this does not currently compile, meaning that it will either be re-coded when it is de-bugged, or else this syntax will be implemented in ALPHABASIC. I'd prefer the later. There are other ways to initialize arrays, but this is the nicest.

(4) COMPIL --- makes line numbers totally obsolete !!!

If an ALPHABASIC source program is built up with the EDIT program, with no line numbers on the left, and (of course) no line number references within the text (such as GO TO 200), then COMPIL FILNAM will create the same file (FILNAM.RUN) as would have been created if the source had line numbers and was compiled with the /O switch.

(5) SYSACT --- short format possible

To do a SYSACT on disk 0, type

```
    SYSACT:
```

Unfortunately, this was accidentally discovered by someone who was trying to SYSACT disk 1, and did him no good until he made a phone call.

```
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;                                               ;
;       ALPHABASIC COMMON SUBROUTINE            ;
;                                               ;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

; BY R. FOWLER (KHALSA COMPUTERS) --- LAST UPDATE 1978 SEP 12

        COPY    SYS

COMMON: CMP     @R3,#2          ; CHECK FOR > 1 OPS SENT
        BGE     TYPCHK
        TYPE    <COMMON NEEDS 2+ ARGUMENTS>             ; ERROR MESSAGE
        RTN
TYPCHK: ADD     #4,R3           ; SET UP ADDRESS OF FIRST XCALL PARAMETER
        MOVB    @(R3),IOTYP     ; MOVE S/R BYTE
        CMPB    IOTYP,#83.      ; CHECK FOR S(END) CALL
        BEQ     IO
        CMPB    IOTYP,#82.      ; CHECK FOR R(ECEIVE) CALL
        BEQ     IO
        TYPE    <COMMON ONLY S(ENDS) OR R(ECEIVES)>     ; ERROR MESSAGE
        RTN
IO:     LEA     R1,DATBEG       ; SET UP COMMON ADDRESS COUNTER
        MOV     #DATEND-DATBEG,ROOM     ; SET UP COMMON "ROOM" COUNTER
LOOP:   ADD     #6,R3           ; COMPUTE NEXT "IMPURE" VARIABLE ADDRESS
        CMP     R3,R4           ; CHECK FOR LAST VARIABLE
        BGE     END
        MOV     2(R3),R0        ; LOAD VARIABLE'S LENGTH
        MOV     (R3),R2         ; LOAD VARIABLE'S ADDRESS
        SUB     R0,ROOM         ; DECREASE "ROOM" COUNTER
        BPL     SEND            ; CHECK FOR OVERFLOW OF COMMON AREA
        TYPE    <DATA EXCEEDS COMMON AREA>              ; ERROR MESSAGE
        RTN
SEND:   CMPB    IOTYP,#83.      ; CHECK FOR "S(END)"
        BNE     RECV
        MBBU    R2,R1           ; SEND NEXT VARIABLE TO COMMON AREA
        ADD     R0,R1           ; CALCULATE END OF VARIABLE
        BR      LOOP
RECV:   MBBU    R1,R2           ; RECEIVE NEXT VARIABLE FROM COMMON AREA
        ADD     R0,R1
        BR      LOOP
END:    RTN
IOTYP:  BLKB    2               ; S(END) OR R(ECEIVE)
ROOM:   BLKW    1               ; CONTAINS ROOM (BYTES) LEFT IN COMMON AREA
DATBEG: BLKB    1000.           ; (MAXIMUM) COMMON DATA AREA
DATEND: BLKB    1

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;                                                               ;
;       CALLING SEQUENCE IN ALPHABASIC :                        ;
;       XCALL COMMON, <MODE>, <VAR1>, <VAR2>, <VAR3>, ...       ;
;                                                               ;
;       WHERE <MODE> IS A STRING CONSTANT OR VARIABLE WHOSE     ;
;       FIRST CHARACTER IS "S" OR "R" TO SEND OR RECEIVE        ;
;       DATA TO THE COMMON AREA.                                ;
;       THE REMAINING PARAMETERS ARE VARIABLES OR CONSTANTS     ;
;       TO BE SENT/RECEIVED TO THE COMMON AREA (IN FORTRAN      ;
;       COMMON FASHION).                                        ;
;                                                               ;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
```

Command file DSK0:BACK.CMD[2,2]

```
:<

>
LOAD DSK0:LOG[1,4]
LOAD DSK0:ERASE[1,4]
LOAD DSK0:DIRSEQ[1,4]
LOAD DSK0:MOUNT[1,4]
LOAD DSK0:COPY[1,4]
LOAD DSK0:DING[1,4]
:R
LOG
:S
:<

THIS PROGRAM WILL COPY ALL THE FILES FROM THE LOGIN NUMBER SHOWN ABOVE
ONTO YOUR BACKUP DISK.  (IF THE ABOVE NUMBER DOES NOT INCLUDE 'DSK1:',
HIT CONTROL 'C' TO ABORT THIS PROGRAM.)

REMOVE THE 'SYSTEM' DISK FROM THE LEFT-HAND SLOT AND REPLACE IT WITH
YOUR BACKUP DISK.

NOW PUSH 'RETURN':>
:K
ERASE *.BAK
DIRSEQ
MOUNT DSK0:
LOG DSK0:
:<
ERASING THE FOLLOWING FILES FROM THE BACKUP DISK:
>
:R
ERASE *.*
DING 1
:<
COPYING THE FOLLOWING FILES ONTO THE BACKUP DISK:
>
COPY/X=DSK1:*.*
:S
LOG DSK1:
:R
DING 3
:S
:<
NOW, REMOVE YOUR BACKUP DISK FROM THE LEFT-HAND SLOT AND REPLACE IT
WITH THE 'SYSTEM' DISK YOU REMOVED EARLIER.

PUSH 'RETURN' ONCE MORE:>
:K
MOUNT DSK0:
DEL *.*
:<
FINISHED.
>
```

To Alpha Micro & AMUS

Conversion Error

From time to time we all need to pack a little more data onto the diskette. One way to do this is to store numeric data as binary instead of floating point. A contender for this is money data. Binary can only handle integers thus we must multiply our figure by 100, store it and divide it by 100 when we pull it back off of the disk.

```
MAP1 CUST'REC
        MAP2   NAME$,S,30
        MAP2   LIMIT,B,3
        . . .
```

With a 3 byte binary field we can store a dollar amount =$167,712.xx Thus:

```
        INPUT A
        CALL MONEY'CK
        LIMIT = A * 100
        . . .
```

If you try this you will find that what is stored in limit if cents are entered is 1 cent less that what was entered, i.e.

```
        A = 112
        LIMIT GETS 11200
        A = 112.23
        LIMIT GETS 112.22
```

The reason for this is that there is a rounding error in convertin g floating point to binary in some cases. In most higher level machines the internal representation for the floating point values 112 and 112.23 are handled differently.

To get around this problem, until Alpha Micro has a chance to get it fixed.is the following

```
        A$ = STR$(A*100)
        LIMIT = VAL(A$)
```

Thus you are changing the internal representation to one that will convert correctly. Also the program will still work OK when the conversion routine is fixed.

RUN.PRG ERROR

If a program is run that dimensions an array that eats up almost all of the user's core then it cannot be re-run without reloading the program again. The reason is that the run package does not reclaim the array space when it is called into action again. The is a real hassle when you are debugging and testing a program.

We also had this problem occur on one of our systems out in the field. It was in a high school where the teacher was giving a test on a 4 user system. The first student would take the test ~~and~~ his answers would be written to the disk and the program would end. The next student would come up to the same terminal and type RUN TEST1A and get an * OUT OF MEMORY* error message. The grades from the kids who used the CRTs first were much higher that the 2nd and 3rd students whom figured they had broken the system.

Two ways around the problem are to never let the program end and let the 2nd user just enter some key to start the program over again or to use a menu situation and keep rechaining back to the program but none the less its a real hassle when debugging programs.

MULTI-USER

Another problem we're having, running under 3.4, is when we take a 4 or 5 terminal system to a show. When we run the same demo programs chained thru a menu program and 2 users go after the same program at once, either both crash with program not found or one gets it and the other crashes.

THE HAWK

We desperately need some response times from anyone out there using the 10 meg system with several users running concurrently. One of our best customers is interested in running 10 terminals in a CAI lab environment. If you have had any experience in this area please give us a call.

THE BEST

Speaking with experience from the big systems on down, we sell both Honeywell and Alpha Micro, the cost/performance ratio of the Alpha Micro is unsurpassed. It's the best there is in the 10,000 to 50,000 range.

Franklin P. Parker
Capt. Tim Leslie
Computers Unlimited
7724 East 89th Street
Indianapolis, IN 46256
(317) 849-6505

HARDWARE BUGS, FIXES, ANNOUNCEMENTS, COMPLAINTS, & COMMENTS

Make sure you have a good cooling & ventilation system for your CDC 9740 HAWK disk drives. They have a tendency to overheat during a DEVCPY.

For those of you with an LA-36 DECwriter running at 300 baud, LARK ELECTRONICS & DATA, P.O. Box 22, Skokie, Ill. 60076 has a modification kit that will speed up your DECwriter to 600 baud. They claim no loss of reliability at the higher speed. It's called the EXCELERITER and retails for about $100. You must supply your own null bytes in the software driver during the CR & LF.

DIGITAL MICRO SYSTEMS has a 16K static memory board with memory management compatible with A/M and it sells for $595.0C, assembled and tested. It is guaranteed for one year. They may be reached at:

DIGITAL MICROSYSTEMS
BOX 1212
OREM, UTAH 84057
(801) 224-2102

The new terminal from Lear-Siegler is a replacement for the ADM-3, called the ADM-31. In addition to all of the features and options of the older terminal, it has an expanded keyboard with a separate keyboard, 2 pages of screen memory, and a field protect feature for screen formatting.

For those of you with TEI, EQUIBOXES, IMSAI, and other mainframes with rear panel cut-outs used for connector hole blanks, it is recommended that you tape them up or put a cover of some kind over them so that proper airflow is distributed to your S-100 boards. Remember to tape shut the long slot across the top of the rear panel where the ribbon cables come out of the cabinet. Also, if the ICOM model 3712W dual drives are giving anyone fits, it is because they also have a cooling problem with the controller/formatter P.C. boards. You can cure the problem by taping up 85% of the area of the hole located at the right rear of the cabinet as viewed from the front, so that most of the air is forced out through the front of the box. If that doesn't cure the problem completely, try rplacing the 74164 chip on board #1 (top) near the front with a 74LS164. We also replaced the fan on our 3712W with a muffin type fan that was quieter and had greater air flow. Always use an air filter, and change it every 6 months.

ALPHA MICRO SYSTEMS EXPANDS

Alpha Micro Systems has expanded its operations to include a staff of fifty people and over 24,000 square feet including a new 12,000 square foot manufacturing area.

REQUESTS

We have a request for an interface to the Frieden Flexowriter 2300 Series. Does anyone have any ideas?

AM-100 Users are also looking for a card reader.

Two other rumors going around are that there is an interface between the S-100 Bus and the HP 1B - RS448, also, is someone writing a "C" compiler for the AM-100?

KHALSA BUGLIST

Robert Fowler has supplied us (and Alpha Micro Systems) with another well documented buglist for 3.4 Incedently, those wonderful documents that outlined the history of AMOS published in previous AMUS newsletters were done by Robert.

AMOS BUGS (Version 3.4)

(1) AlphaBASIC - bad syntax errors within IF statements are not detected:
IF(Ø) then print "A" else print "B"
gets no syntax error

(2) Alph·BASIC - forcing a string to print a null with a "" tends to crash the system.

(3) AlphaBASIC - ALLOCATEing too large a file crashes the system.

(4) BASORT doesn't work

(5) KILL - small letters on a job name causes a crash. Also, BASIC will not accept small letters for filenames.

(6) SORT.PRG doesn't work if it has to go into poly-phase merge.

(7) SPOOLer can't print from any DSK except 0 or 1. If a file is spooled from DSK2 or DSK3, the system promptly goes dead. This happens whether the user is logged into DSK0, 1, 2, or 3.

(8) AMOS - cannot open disk file (or printer) for output. In essence, if a batch job with a PSEUDO terminal tries to open a disk file from BASIC, it will get hung up. The same thing happens when the batch job tries to open the printer for output using the:

```
    OPEN #1, "TRM:PRINTR", OUTPUT
```

statement to open the printer. This happens even if the memory given the batch jobs is the same as the user's CRT where the program executes correctly.

AM-100 Users Group

The following letter is from Lefford F. Lowden, who publishes a newsletter from Rochester, New York. We have agreed to trade newsletters, and will print any items from that newsletter not duplicated by AMOS in this newsletter.

Thanks for your newsletters. Note, in the June issue there is an error in the proposed Printer Spooler Request program changes. The line labeled QKIL2: MOVI 5,R3 should be: QKIL2: MOVI 6,R3 Fortunately, this error will not be detected until a file is removed from the queue with the /K option. I agree with the suggestion made in the July Newsletter that PRINT FRED.DAT/D be implimented to delete a file after printing. I would also suggest something like PRINT FRED.DAT/C35D which would print 35 copies and then delete the file. It would also be nice to be able to simulate form feeds when one is running the spooler on a terminal rather than on a printer -- in my case, I run it on a model 43 TTY. Which brings me to this -- does anyone mind if I try to implement some, or all, of my suggestions with the program and them send the resulting program to you for distribution, publication, or whatever with your group? On my end, I would dump it into the library.

Sincerely,
Lefford F. Lowden
616 Long Pond Road
Rochester, New York 14612

Some *.CMD's of our own

Tom Fox's Command file suggestions brought a couple to mind that we use a lot. One is used to load the three programs needed to perform BASORT. Called SORTLD.CMD, it looks like this:

Another quick and dirty one is called MYSTAT and is a poor man's substite for DYSTAT

```
TIME
SYSTAT
MYSTAT
```

It does take up a portion of the AM-100's effort to run the thing, but once in a while, you want to look at what is going on over a period of time.

AM-100 Medical Applications

Sydney Z. Spiesel of Sisyphus Iterations, 77 Everit St., New Haven, Conn. 06511 writes that he has a driver available for the Perkin-Elmer Fox CRT that anyone is welcome to. He has been using the AM-100 to develop a commercial turnkey advisory system for hospital emergency rooms, etc. and has also developed a considerable amount of statistical and utility software.

AMUS MEMBER'S NEEDS, WANTS, & SUGGESTIONS

BITMAP in memory should be expanded by one 16-bit word for every drive using that bitmap. This word should include a binary number representing the total free blocks available on that disk drive and that word should be updated at the same time that the BITMAP is updated. Reading 1's and 0's every time is time consuming, especially on 10M and larger disks.

FLTCNV.PRG and TODCNV.PRG source codes should be made available. They are sorely needed for all kinds of timing, interval, and math routines.

The MEMORY command needs work. If a user types MEMORY 6....Well, just try it and see (tee hee). Any MEMORY allocation that cannot handle MEMORY will result in a locked up terminal.

How about job swapping if the user has a large enough disk to support same?

MACRO should have a 'NOTE' pseudo OP. As used in other macroassemblers we have seen, it would generate no code, but would appear in the listing. It would generate an 'L' error message when error conditions exist that would bomb the assembly process when a bad term or argument is passed to the macro. Normally this would blow the assembly with no error messages whatsoever.

Also needed is a 'TITLE' pseudo OP, as typing in titles on every sub assembly is a pain ...

THE SOFTWARE EXCHANGE is a new publication devoted to the exchange of business and personal computing software. We talked to Alan Bartholomew, the Publisher/Editor, at the Personal Computing Faire in Anaheim at NCC '77. At that time, he was just getting the publication off the ground, and even then, it looked pretty good. You can write for your subscription (6 issues per year) by mailing $5.00 to:

THE SOFTWARE EXCHANGE
P.O. BOX 55056
VALENCIA, CALIF. 91355

NEWSLETTER FOR AM-100 USER'S GROUP   Produced by Lefford F. Lowden
616 Long Pond Road
1 OCTOBER 1978   Rochester, New York 14612

NEWS FROM ALPHA MICRO

Version 4.0 is scheduled for release about the end of October. It will contain a completely revamped version of COMPIL which will compile very large programs and will produce intelligible error messages. Other BASIC bugs have been fixed as well as some operating system glitches. Bob says that many of the bugs mentioned in the September Newsletter have been fixed. Version 4.0 will also contain a new screen oriented editor (VUE) which was on display at the Alpha Micro booth at PC-78; this one requires a CRT with cursor addressing.
A new language, MPL, is scheduled to be released. MPL is similar to TRAC and SAM-76 if you know anything about either of those.

Bob says that the M command in EDIT is not documented because it does not work well, and it is not supported by Alpha Micro. I found about it and some of the others by experimentation. I have had access to a DECsystem 10 and operated TECO, the editor on which the AM-100's EDIT is based. In addition, when I worked on the Sigma 9, I wrote an implementation of TECO for the Sigma 9. Thus, what I did was to try a wide selection of the commands to TECO in EDIT to see what happened. Most of the exotic commands failed, but M worked. Thus after some further testing, I reported it as being available.

WARNING ON FOR - NEXT LOOPS

FOR - NEXT loops in BASIC and DO loops in FORTRAN present problems in that implementations are not always consistent. In running a check on the operation of ALPHABASIC (TM) with the statement:
FOR I=1 TO 8: NEXT I: PRINT I it is found that the value output is 9!!!
Note: this is not an error as both FORTRAN and BASIC do not require that the index on a DO or a FOR-NEXT loop be defined outside the loop.

PROGRAMS USERS ARE LOOKING FOR

Bob Chaput has written a few days ago and said: "With regard to the types of programs I would like to see [are] additional business application external subroutines, similar to the 'unpublished, undistributed' Alpha Business Subroutines (copy enclosed). I have found these to be of great value, especially for data entry." I am including below an abbreviated listing of the contents of the program descriptions that he sent so that others who don't know of the package will be at least advised of what is in it. Since Bob says that it is undistributed, he may become its distributor. However, I think that you might check with your dealer first. I haven't had time to check with mine as I am writing this, but I will check with him early next week.

ALPHA BUSINESS SUBROUTINES

XCALL ACCEPT,CHAR

accepts any character (rubout, control-c, etc.) and returns floating point value in CHAR.

XCALL ANYCN,CNGCTL,WHATNO

for editing data input via a CRT.

XCALL BASORT,.......

this is discussed in most manuals, so nothing further.

XCALL COMMON,.......

this is discussed in most manuals, so nothing further.

XCALL DSPLY,XMAX,ROW,COL,DECMAL,OPTION

XMAX max num of digits in decimal field, controls output of trailing blanks unless option 6 is specified.

ROW ROW*2 = actual row where output appears

COL screen column where output appears

DECMAL numeric field to be output

OPTION type of editing.

1 => straight numeric editing.
2 => date editing (MM/DD/YY)
3 => dollar editing (###,###,###.##)

```
                  4 => three digits to right of decimal point.
                  5 => one digit to right of decimal point.
                  6 => straight number, only 1 blank after field is output.

XCALL INPUT,ROW,COL,XMAX,XMIN,TYPE,ENTRY,INXCTL
              ROW    ROW+2 = actual row on which to accept data
              COL    screen column from which to accept data
              XMAX   max num of char to be accepted
              XMIN   min num of char to be accepted
              TYPE   a two character string defining type of data to be entered
                  "A " => alphanumeric
                  "D " => date as MMDDYY
                  "# " => numeric field - positive only
                  "#-" => numeric field - positive or negative
                  "#N" => numeric field - same as "# " but displays one blank after dots
                  "N " => numeric field - same as "# " but does not default to zero
                  "NN" => numeric field - same as "N " but one blank after dots
                  "C " => verification - will accept only carriage return
                  "$ " => dollar field, pos only, enter $1.00 as 100
                  "$-" => dollar field, positive or negative
                  "H " => hours field - processed the same as "$ "
                  "?E" => where ? is any other type, the field will accept "END"
                          as a response
                  "?T" => allows <TAB> as the terminating char, sets INXCTL to 3
                  "YN" => YES/NO answer - accepts only "Y" or "N"
                  "YE" => YES/NO/END - same as "YN" but also allows "END" as response
                  "YY" => YES/NO - same as "YN" but defaults to "Y"
              ENTRY  string variable which contains response on return
              INXCTL code returned by INPUT
                  0 => input OK, no message
                  1 => "Y" answer to YES/NO field
                  2 => "N" answer to YES/NO field, or "END" for a "?E" field
                  3 => <TAB> key used to default on "?T" field

XCALL MESAG,MSG,MSXCTL
                  subroutine to display standard error messages
              MSG    string to be displayed
              MSXCTL specifies message option
                  1 => display message on line 24 with "CR TO RECOVER"
```

2 => display message on line 24 with "CR TO CONTINUE"
3 => display "INVALID ENTRY SEE INSTRUCTIONS CR TO RECOVER" on
on line 24. MSG can be null for this one
4 => display message only on line 24
5 => display "FILE NOW FULL - SEE INSTRUCTIONS" and stop. If
this option is used, all files should be closed prior to
calling this routine
6 => display "INVALID SELECTION CR TO RECOVER". MSG may be null

XCALL MMENU,MENUID,RECNAM,SELECT

subroutine to display standard master menu. It clears the screen displays application in upper left and in the middle is following

```
PLEASE SELECT APPLICATION
    1) ADD NEW recnam
    2) CHANGE/INQUIRE recnam
    3) DELETE recnam
    4) PRINT-OUT recnam
```

MENUID string specifying the title of the application. MENUID is displayed in upper left-hand corner of screen

RECNAM string specifying the name of the key field of the master record to be maintained (CUSTOMER, VENDOR, etc.)

SELECT value of the application selected by the user, where:
0 => "END" has been typed
1 => ADD mode
2 => CHANGE/INQUIRE mode
3 => DELETE mode
4 => PRINT-OUT
5 => PRINT-OUT with sort before printing

XCALL MOUNT,NAME,DEVICE

NAME   string specifying name of item to be mounted (like 3 part bill heads, etc.)

DEVICE string specifying device on which item is to be mounted

XCALL PGMEND

clears screen and displays "END OF PROGRAM" in middle of screen

XCALL PRINT,LINCNT,PGCNT,PLINE,TITLE,HDR1,HDR2,HDR3,LEGND1,LEGND2,LEGND3,
LINCTL,PAGSIZ,PRCTL2,LGNCTL,PCHANL

This routine is complex enough that I won't try to describe the arguments except to say that it provides a flexible, sophisticated method of formatting printer output. It handles the job of pagination, titling, and page numbering as well as providing several options for headers and legends. Output goes to a file on the disk or any other device capable of handling sequential ASCII output.

XCALL RDATE,DATE

DATE floating point variable into which the decoded date is to be placed. Format of variable is ((month*10000)+(day*100)+year) or MMDDYY.

XCALL RENAME,OLDNAM,NEWNAM,STATUS

OLDNAM string specifying current file name
NEWNAM string specifying new name for the file.
STATUS floating point variable 0 => success, nonzero => failure

XCALL SERCH,...........

SERCH is an AlphaBasic subroutine which handles binary and sequential searching of disk files. It allows the user to access a file via symbolic keys without paying the overhead costs of ISAM.

XCALL SRCH2,......

SRCH2 is a simplified version of SERCH

XCALL STENO,STRTNO.ENDNO,STNFLD,XMAX,STXCTL

subroutine to input and edit starting and ending numbers

XCALL STRIP,STRING

subroutine to remove trailing blanks. It runs about 100 times faster than can be done in BASIC.

XCALL TMENU,MENUID.SELECT

subroutine to display standard transaction menu. screen is cleared, title is put on line 1 and following menu is centered on the screen:

```
    PLEASE SELECT APPLICATION:

            1. ENTER (ADD) NEW TRX

            2. CHANGE EXISTING TRX

            3. DELETE EXISTING TRX

            4. PRINT TRX EDIT LIST

            5. POST TRX TO MAIN FILE

            6. CREATE BACKUP WORK FILE

          MENUID string specifying user title
          SELECT returned response

XCALL TMEN2,MENUID,SELECT
                        a modified version of TMENU

XCALL WAIT,OPTION
                        subroutine to display various wait messages
          OPTION specifies the message to be output:
               1 => clear screen, display "PROCESSING OCCURRING ... PLEASE WAIT"
                    on line 12
               2 => clear screen from row 4, column 29 on, display
                    "PRINTING OCCURING ... PLEASE WAIT" on line 12
               3 => display "PLEASE WAIT ..." on line 24
               4 => clear screen from row 4, column 1 on, display
                    "PROCESSING OCCURING ... PLEASE WAIT" on line 12
```

This has been compressed from 25 sheets; I hope I have preserved the flavor of the routines.

USER DISTRIBUTED PROGRAMS ON SUPPLEMENTAL SHEETS

```
        OCVT    4,^H32              ; 4 DIGITS, TRAILING SPACE
        MOV     (R3)+,R1            ; ADDRESS OF VARIABLE
        MOV     R1,R0               ; SAVE ADDRESS FOR LATER
        OCVT    4,^H32
        MOV     (R3)+,R1            ; SIZE OF VARIABLE
        DCVT    4,^H33
        MOV     R1,R5               ; SAVE SIZE OF VARIABLE
PLOOP2: MOVB    (R0)+,R1            ; GET A BYTE OF ARGUMENT
        OCVT    2.^H2
        SOB     R5,PLOOP2
        CRLF                        ; CARRIAGE RETURN
        BR      PLOOP1

PRTN:   ADDI    4,SP                ; POP THE TWO ITEMS
        MOV     R4,@R2              ; RESTORE HEX OR OCTAL AS IT WAS
        RTN                         ; AND RETURN TO BASIC
        END
```

For a call of the type XCALL PARG,1.5,3.4,"HI THERE",A$ where A$ is a null string, the output appears as:

```
 #  TYPE  ADDR   LEN VAL
 1  0004  8FF6     6 C04000000000
 2  0004  8FF0     6 594199999999
 3  0002  8FE8     8 4849205448455245
 4  0002  8F60    10 00000000000000000000
```

In this version, the variable number and length are output in decimal; the type, address, and value are output in hexadecimal. Note, the value is output as consecutive bytes, not as consecutive words. Thus, be careful in interpreting the value. In particular, character strings are addressed as consecutive bytes; however, a word is assembled from two bytes addressed by the even byte with the even byte on the right and the following odd byte on the left. Yes, I know it is strange, but that is the way it is.

FLOATING POINT REPRESENTATION

After reading the brown manual "WD16 MICROCOMPUTER" pages 34 and 35 in trying to determine what the bit pattern was for the floating point numbers, I decided it would be easier to have the machine tell me. The following ALPHABASIC (TM) program was written and run.

```
10 MAP1 A
11   MAP2 FLT,F,6
12 MAP1 BIN
13   MAP2 DIG(3),B,2
14 STRSIZ 1
15 DIM DIGIT$(16)
16 FOR I=1 TO 16: READ DIGIT$(I): NEXT I
17 DATA "0","1","2","3","4","5","6","7","8","9","A","B","C","E","F"
19 STRSIZ 18
20 INPUT "FLT = ?"; FLT
22 BIN=A
24 C$=""
30 FOR I=1 TO 3
31 D1=INT(DIG(I)/4096): DIG(I)=DIG(I)-D1*4096: D2=INT(DIG(I)/256)
32 DIG(I)=DIG(I)-D2*256: D3=INT(DIG(I)/16): D4=DIG(I)-D3*16
33 C$=C$+DIGIT$(D1+1)+DIGIT$(D2+1)+" "+DIGIT$(D3+1)+DIGIT$(D4+1)+" "
34 NEXT I
40 PRINT TAB(20); C$
80 GOTO 20
90 END
```

The following is a table of the results of the program. Using this information and the data from the manual, the floating point representation is now understandable. Note, the hex representation consists of the component words.

DECIMAL NUMBER	HEX REPRESENTATION
1.	40 80 00 00 00 00
2.	41 00 00 00 00 00
4.	41 80 00 00 00 00
-1.	C0 80 00 00 00 00
-2.	C1 00 00 00 00 00
-4.	C1 80 00 00 00 00
3.14159265	41 49 0F DA 9E 46
-3.14159265	C1 49 0F DA 9E 46

HIGHLY PLACED SOURCES

Highly Placed Sources is intended to be a resource of rumors, gossip, future p·oduct announcements, and other various items of interest that I'd like to pass on to AMUS newsletter readers and have a little fun in the process.

This AMUS NEWSLETTER is late because Steve Patterson played Manana with it. So I will admit to this publicly (Elliott & Taylor's excuses will appear in the next exciting issue of AMUS confessions.) We will see you at the next showing of the SHORT LIMBED FOLLIES or: HOW I WAS SAWED IN HALF AT WARREN'S WORST WORRIES: see you at the faire!

Here's a goody that I think you'll like: A/M reportedly plans to release the (AM-600?), a complete 1/2" nine-track tape drive system that will include: an S-100 interfce card, controller/formatter that will handle at least four drives, and possibly more. It will have 45 IPS R/W speed, dual density 800/1600 BPI NRZI format. The drive itself will probably use tension-arm tape control, be rack mountable, and be a miniture version of the biggies with a 2400 ft. reel capacity. No dates, prices, deliveries, or further details available (of course). My educated? guesstimate is that the software, S-100 interface, controller/formatter, and one drive will retail for under $6,000.00.......and I could be surprised in April '79?

FORTRAN IV is in development by an outside contractor. No commitments and no delivery dates from contractor or A/M. I don't blame them. My idea of a future release announcement is that you announce when you see at least SOME light at the end of the tunnel. However, if they don't hurry along, Ray Rosich will beat them with his interactive version of FORTRAN 77.

When I was in Dallas at the IME, I saw and used release 4.0 of AMOS, which was pretty darn good. I also used version 1.0 of the business software, which was damned good. This software should be available from your dealer now....but Richard McClendon's business software surpasses anything available on the market today, in my opinion. It is reported to be uncrashable, except for input service routines. I could not crash the system--- I will detail the system in a future issue.

Rumor has it that a new users, operators, and programming procedures manual is coming... but not from A/M. The Boulder Free School has put together a rather nice manual that will sell for $7.50. Prices go up at the first of the year to $10.00. It will be available at your local dealer in about four weeks.

Look for the micro computing industry to form an association at the West Coast Computer Faire. It will be their intention to sponsor only three shows nationally in the coming year. A West coast show, an

East coast show, and NCC. Most manufacturers (including A/M) find that $25K is not hard to spend for a first-class show, let alone some of the smaller regionals. There are simply too damn many shows with sparse attendance. The new all-industry organization is headed up by none other than your friend and mine....JOHN FRENCH.

It will cost $100.00 per year to join, but will be extremely well worth it from any point of view. If you are an A/M dealer, and even if you're not, attendance is an absolute must! There are simply going to be too many things going on in Washington, D.C. and other lofty government places that will require representation, and this is the best way I know to get it. You can get more information on joining this group by calling John at (714) 768-3411, or writing to him at 3095 Red Hill Dr., Costa Mesa, Calif. 92626.

Attendance at the Faire is going to be a must as there will be many things going on that will require your attenance, especially the two-day AMUS seminars. AMUS and A/M will be sponsoring them as a joint venture. Everyone is urged to attend them on Wednesday and Thursday, November 1st & 2nd.

I will be holding an Alpha Micro Dealer's meeting on the evening of November 2nd at 7:30 PM in the same meeting room that will be used for Alpha Micro seminars during the Faire. Some very interesting ideas will be discussed. Attendance by all A/M dealers is well advised.

STEPHEN B. PATTERSON

Computer Applications Research

7 August 1978

James S. Taylor, Esq.,
Data Processing Consulting Inc.,
P.O. Box 1723,
Boulder,
COLORADO 80306

Dear James,

Tony has passed on your letter of the 25 July for which I thank you. It is nice to have someone to share our problems with! We hope that we can also be of use to another member of Amus and we will therefore keep you constantly informed of all our development in both hardware and software.

We enclose herewith specifications of our first software package which is a highly specialized, comprehensive, and sophisticated fashion retailing management system. Source code is not made available. The system will be configured by us according to information given to us on a questionnaire in respect of size ranges, number of departments, branches, sales assistants, etc. The package will be fully maintained by us for an annual licence fee.

The licence fee varies according to the size of the installation, being mainly dependent upon the number of branches and departments. The single branch system with 10 departments would be £3,000 per annum. Otherwise we can quote prices after completion of a specific questionnaire. We would then also be able to indicate the disc and core size required for such a system. A copy of this specification and questionnaire is available upon request.

We have another package nearing completion which I hope to write to you about in a couple of weeks time.

Yours sincerely,

Leo Scheiner

Encl.

FOX ASSOCIATES
COMMUNICATION SYSTEMS ENGINEERING

September 8, 1978

Mr. Steve Elliott
Community Free School
P.O. Box 1724
Boulder, Colorado 80306

Dear Steve:

Those of us who have lived with the Alpha Micro for any length of time have discovered what a handy thing the COMMAND files can be. My user number (2,2) seems to keep filling up with all sorts of useful "keystroke savers". One of the first ones to show up was CLEAN.CMD, which merely reads:

```
ERASE *.BAK
DIRSEQ
DIR
```

It's short but quite useful after a heavy session with the EDIT program.

A more elaborate example is BACK.CMD, which our non-programmer types use to make a backup copy of their data disks (a habit we <u>all</u> develop after our first 'disaster' with the one-and-only disk with that critical data). The command file assumes that an AMOS system disk is in floppy drive #0, which includes the normal programs to be found in PPN (1,4) and the BACK.CMD file in (2,2). It further assumes that the user is logged on to his or her own PPN on disk drive #1. The user merely types "BACK" and follows instructions in order to make that all-important backup disk.

There must be several such "goodies" like this for every Alpha Micro System installation. How about an exchange via the AMUS newsletter?

Sincerely,

Tom Fox

Tom Fox

New Address: 17925 Sky Park Circle, Suite #G, Irvine, CA 92714

616 Long Pond Road
Rochester, New York 14612

2 September 1978

James S. Taylor
Data Processing Consulting, Inc.
P. O. Box 1723
Boulder, CO 80306

Dear Jim:

Enclosed is the September Newsletter. Feel free to use whatever you think useful to you.

Thanks for your Newsletters. Note, in the June issue there is an error in the proposed Printer Spooler Request program changes. The line labeled
QKIL2: MOVI 5,R3 should be QKIL2: MOVI 6,R3
Fortunately, this error will be not detected until a file is removed from the queue with the /K option. I agree with the suggestion made in the July Newsletter that PRINT FRED.DAT/D be implemented to delete a file after printing. I would also suggest something like PRINT FRED.DAT/C35D which would print 35 copies and then delete the file. It would also be nice to be able to simulate form feeds when one is running the spooler on a terminal rather than on a printer -- in my case, I run it on a model 43 TTY. Which brings me to this -- does anyone mind if I try to implement some, or all, of my suggestions with the program and then send the resulting program to you for distribution, publication, or whatever with your group? On my end, I would dump it into the library.

Sincerely,

Lefford F Lowden

Lefford F. Lowden

INFORMATION SYSTEMS RESEARCH LABORATORY
PO Box 4158
Medford, Or 97501
(503) 773-6558

Jim Taylor
AMUS
PO Box 1723
Boulder, Colo 80306

June 9,1978

Dear Jim:

ISRL is currently developing a "data management system" for the AMS called ISRL-DIS (ISRL's Data Information System). It consists of the two programs:

DISIO - an I/O management package which features

- data and program independence
- password security on an item (i.e. field) level
- file sharing by multiusers
- sequential I/O
- indexed I/O (partially inverted)
- direct I/O by record number

DIS - an application program built around DISIO with

- editing
- sorting
- updating
- searching (full conditional)
- reporting

etc. capabilities on DISIO data files.

The DISIO program is designed so that it may reside in monitor memory and has an AlphaBasic interface package of .SBR subroutines. DIS is an integrated package which includes DISIO and is designed for user memory residence. Each is available in object form only for the following costs:

DISIO -	2,500	Available in July
DIS -	5,000 (includes DISIO)*	Available in August.

Please include this information in the AMUS Software Directory.

Respectfully,

Jack N. Hatfield

Jack N. H atfield

* assumes DISIO is separate in either user or monitor memory.

Payne, Jackson and Associates
447 East Fifth Avenue, Anchorage, Alaska 99501
Accounting ● Data Processing ● (907) 272-7261

July 26, 1978

PJA ACCOUNTING SYSTEM

We have a generalized interactive bookkeeping and accounting system which was written in Alpha Basic by our staff of Certified Public Accountants for our accounting practice. It was created primarily during the last six months of 1977 and placed into service on January 2, 1978. To date we have invested in excess of six thousand hours in its development, including improvements and updates with each new AMOS release. The system has been in constant use during 1978 with numerous clients covering a wide range of business activities. The system is a complete accounting system and includes the following subsystems: accounts payable, accounts receivable, cash disbursements, cash receipts, financial statements, fixed assets, general journal, general ledger, inventory, payroll, and sales.

The entire system is menu driven and chained together allowing the user to execute all functions (with the exception of AMOS functions like disk copies) without leaving the PJA system's control. Advance CRT menu screens are used throughout, permitting the use of personnel less familiar with computers and accounting. Data entry defaults and edit controls are used whenever possible to increase productivity and accuracy. The system is designed to provide detailed audit trails thereby reducing the job of tracing and identifying past transactions.

GENERAL LEDGER

The PJA G/L system has been designed to provide a basic accounting system for a wide range of businesses and non-profit organizations. An 8 digit account number is used including a 2 digit department and a 2 digit section number. This allows the system to maintain separate accounts for each department, fund, job or company and still produce a consolidated general ledger and financial statements. Year to date historical transactions may be maintained within the general ledger and are limited only by the file size. In addition, monthly transactions may be either summarized or posted in detail. Postings may be reversed and removed from the G/L for error correction and then reposted.

Standard G/L reports consist of the general ledger detail, general ledger summary, chart of accounts plus the various batch reports and journals. The Cash Disbursements, Cash Receipts, and Sales Journals can be produced in account number, document or vendor/customer name order. Automatic depreciation and amortization computations are also included in the system for both interim and annual entries. All standard depreciation methods are supported.

FINANCIAL STATEMENTS

Since one of the primary purposes of accounting is to provide financial statements to management, creditors, etc., we have placed particular emphasis on the financial statement subsystem. It allows the user to design their own financial statements as needed. The statements can be either on a cash or accrual basis. This has proven to be a particularily useful feature since the most cost effective method of maintaining an accounting system is on the cash basis. The PJA financial statement subsystem will take a general ledger maintained on a cash basis and produce accrual basis financial statements by using special techniques which we developed. In addition, the financial statements may be either departmentalized and/or sectionalized, with or without footnotes. Supporting schedules and statements if used, may be referenced either by the primary statements or the footnotes.

PAYROLL

The payroll system supports the following employee types: hourly, salaried, salary plus hourly, commission with or without a minimum, salary plus commission, hourly plus commission and bonuses. Computation periods may be monthly, semimonthly, weekly, biweekly or other periods desired. In addition, different employees may have different computation periods within the same payroll. Standard earning categories include regular, overtime, and sick pay, with six other earning categories possible. Each of the six other earning categories can be taxable or not by each of the seven tax categories. Nine other deductions are also handled. All other earnings, tax and deduction categories are user defined. The operator may override all computations in unusual situations.

ACCOUNTS RECEIVABLE

The accounts receivable system will support in excess of 2000 individual customer accounts and produce the following reports: month to date activity report, A/R aging, month end statements, plus various batch reports. This system interfaces directly with the G/L cash receipts and sales subsystems to eliminate the need to enter data more than once.

The PJA accounting system also includes other subsystems which when combined, provides a complete accounting system for most types of businesses and non-profit organizations.

The system will be sold initially for $2000.00 for all systems except accounts payable and inventory. The complete package is available for $2500.00. For more information contact:

PAYNE, JACKSON AND ASSOCIATES
447 EAST 5TH AVENUE
ANCHORAGE, ALASKA 99501
(907) 272-7261

the fox office

17925 SKY PARK CIRCLE, SUITE #G
IRVINE, CALIFORNIA 92714
(714) 957-9331

SECRETARIAL SERVICE USING AM 100
CAN PROVIDE DATA INPUT SERVICES
FOR YOUR SYSTEM.

COMPETENT - FAST - ACCURATE - REASONABLE - QUICK TURNAROUND

ON-TRAK
1009 E. Mt. Curve St.
Altadena, California
91001
ON-TRAK

SUMMER 1978

SOFTWARE Special Applications Programming AM-100

- Programming to specification
- Analysis
- Documentation

We write programs in BASIC for AM-100 users. We're capable of either scientific or business systems analysis and specification writing, or if you wish, we will work to your spec. We currently have one AM-100 system dedicated to new program development and would be quite happy to give you a demo on our system or yours.

For references or further information please write us or call

(213) 794-1439

ON-TRAK

ON-TRAK
1009 E. Mt. Curve St.
Altadena, California
91001
ON-TRAK

AMUS
Attn: Jim Taylor
P.O. Box 1723
Boulder, Colorado 80306

9/08/78

Dear Mr. Taylor,

I have just reviewed Mattel's copy of the July AMUS, and was truly and favorably impressed. Please accept our payment for subscription to the newsletter (enclosed).

In addition to our own AM-100 here at ON-TRAK, I am also the operator of an AM-100 system at the Sierra Engineering Company R&D facility in Arcadia, where we will soon have a 10MB CDC Hawk. Regarding your request in the newsletter, either myself or my partner, Henry S. Cordova, would be happy to talk at length with potential AM-100 users about our experiences.

ON-TRAK is a custom programming house, but we are also offering these utilities for sale.

1. INPUT USING

   This is a BASIC subroutine that makes use of an XCALLed assembly routine to enable all sorts of fancy input. Examples: dates (--/--/--), telephone numbers ((---) --- - ----), Social Security numbers, or ANY thing you can think of.

2. RETAB

   This utility program will read your BASIC program and produce a copy with the x and y tab function coordinates reversed. The user has the option of adding an integer constant to each coordinate.

3. CPY01 & CPY10

   These command files help avoid accidents with DSKCPY.

Other utilities and systems are forthcoming. For more information, or whatever, call us at (213) 794-1439.

Thankyou,

Duane W Cargill

ON-TRAK

(213) 385-9388

KITZMILLER SYSTEMS

JAMES W. KITZMILLER, MS
PRESIDENT

252 SOUTH OXFORD
LOS ANGELES, CALIFORNIA 90004

INFORMATION LETTER

24 July 1978

The Business Management System software for the Alpha Microsystems computer is now complete. It works on both floppy disk systems and hard disk systems.

Those who ordered pre-completion copies will be receiving the complete system shortly.

The software contains quite a few capabilities beyond what was originally promised.

We are preparing demo disks to demonstrate the complete system. We will be sending these domo disks to those who have our earlier demo as well as to those who bought the source code for the Business Management System.

Be sure to let me know of any suggestions or comments that you have.

The entire system is easy to use because of the self-explanatory menus. Even so, we are preparing detailed documentation.

Hope you make good use of the system.

Jim Kitzmiller

WORKSHOP SCHEDULE

	WED.	THUR.	FRI.	SAT.
9:00 AM	SYSTEM ARICTECTURE	WRITING PROGRAMS		EDUCATIONAL SPECIAL INTEREST GROUP
10:30	AMOS *.PRG	CONFIGURING SYSTEMS	BOARD OF DIRECTORS MEETING	MEDICAL SPECIAL INTEREST GROUP
1:30 PM	BASIC	EDIT TXTFMT & VUE	COMMUNICATIONS	CROSS LICENSING AGREEMENTS
3:00	PASCAL		AMUS COMMUNICATIONS NETWORK	WORD PROCESSING
6:00		DEALERS (STEVE PATTERSON)		AMUS GENERAL MEETING

Time will be scheduled at the Faire for demonstration of systems, program swapping, or any other meeting on Sunday. A system will be available. Bring programs to swap.

All meetings will be held at the Gala Inn Towne Motor Hotel where Alpha Micro has reserved a meeting room for the entire week, except for the general AMUS meeting which will be held in a room at the Faire. Location of the general meeting will be listed in the Faire schedule.

AMUS
c/o Community Free School
P O Box 1724
Boulder, Colorado 80306