SOFTECH MICROSYSTEMS

A SUBSIDIARY OF SOFTECH

UCSD PASCAL™

MAINTENANCE NEWS

March 1980 Vol. I No. 2

****

9494 Black Mountain Rd., San Diego, CA 92126 (714) 578-6105 TWX: 910-335-1594

UCSD Pascal™ Maintenance News

MARCH 1980 Vol. 1 No. 2

EDITORIAL

The first edition of the UCSD Pascal Maintenance Newsletter appears to have been well received. One reviewer said that the people in his company "were quite impressed. You have done a very complete job collecting the information, and it is well presented." We are of course happy to receive such kind words, but we are also interested in hearing about ways in which we could improve this publication. To date we have not received any criticism and we are sure that some useful suggestions could be passed our way.

More importantly, we really want to receive your "votes" and suggestions in the New Developments Poll. Since the response to the one published in the previous issue was considerably less than overwhelming, we are presenting the same questions again in this issue. Please let us hear from you this time.

Many interesting and exciting developments in the UCSD Pascal world are coming in the near future. For example, an ANSII-77 FORTRAN standard subset compiler, which is written in UCSD Pascal, and which compiles to P-code, will be available in just a few weeks. We can't say more about it now, but you'll be hearing more very soon.

EDUNET ACCESS SYSTEM IN UCSD Pascal

In the Spring 1980 (Number 15) edition of EDUNET News there is a news article and an editorial discussing "EASy", the EDUNET Access System. EASy was developed at the North Carolina Educational Computing Service (NCECS) in UCSD Pascal. It is prototype software which is meant to augment a stand-alone microcomputer system in such a way that it can also serve as a very intelligent terminal.

EASy is designed to provide services which simplify signing onto remote services; permit off-line editing and data entry; and simulation of a communications terminal. Its current capabilities include block transfers with EDUMAIL at the University of Wisconsin and the TSO system at NCECS/TUCC. It will be expanded in the future to handle other host block transfer protocols.

The current implementation of EASy is running on an Apple II, and has not yet been made available. It is planned that EASy will be distributed through EDUNET and the article promised that "further details on its availability" would appear in the next issue of EDUNET News.

IN THIS ISSUE

Editorial	Page	1	UCSD Pascal User's Group	Page	3
EDUNET Smart Terminal	"	1	Program Exchange Library	"	3
Medical Software Network	"	2	Distributor in UK	"	3
WAVE MATE Series 2000	"	2	Maintenance Report	"	4
New User's Manual	"	2	User's Poll	"	95

"MEDICAL SOFTWARE NETWORK" IS FORMED

Los Angeles, CA. - Datamed Research has recently announced SOFTDOC, a service to aid health professionals who are interested in utilizing computer systems in their practices.

Datamed said that SOFTDOC will support the emergence of high-quality, low cost, medical computing by publication of a quarterly machine-readable software journal which contains: health related software; indepth user reviews of both SOFTDOC and commercial software; and analyses of both vendor descriptions and user evaluations of software.

SOFTDOC is reported to differ from other physician's computer associations in that it exists primarily to distribute medical applications software on machine-readable media, and to assist its members in evaluating which programs work best.

Because the focus of software products for the health professional has been those for the business office, Datamed said that the initial concentration will be in that area.

Datamed plans to distribute SOFTDOC on 8" floppy disks, in both UCSD Pascal and CP/M formats, although they say that "Because of its greatly improved capabilities for serious programming, Pascal is the preferred language."

Contributions to the first issue of SOFTDOC are currently being solicited. Those received prior to 1 May 1980 may appear in the issue published in June of 1980. Subscriptions to SOFTDOC are priced at $55 per year, with single diskettes going for $18. Further information can be obtained from Datamed, 1433 Roscomare Road, Los Angeles, CA., 90024.

UCSD Pascal AND 'DAM' FLOPPY HIGHLIGHT WAVE MATE SERIES 2000

Carson, CA. - Wave Mate, Inc., manufacturer of the 6-year-old Jupiter family of microcomputers, has announced availability of its new Series 2000 microcomputer system, and to accompany it the UCSD Pascal System Software Package. The Series 2000 is a complete stand-alone system packaged in an attractive, compact, desk-top enclosure. It includes a double mini-floppy and controller, two microprocessors, 64KB of RAM, boot ROM, 12" CRT, keyboard with function keys and numeric pad, two serial ports and provision to attach a Winchester disk in the near future.

The compact packaging of the Series 2000 has been made possible by using the newly available double mini-floppy disk drive. It provides two 5 1/4" double density drives (368KB) in the space of a single 5 1/4" drive. The total weight of the Series 2000 is less than 50 pounds.

A complete Series 2000 microcomputer system including UCSD Pascal and 64KB RAM has an end user price of $3,450. A 32KB system without Pascal is also available for $2,995. OEM and dealer discounts are being offered. For more information contact WAVE MATE, Inc., 18005 Adria Maru Lane, Carson, CA., 90746. Tel. (213) 532-4532.

NEW MANUAL AVAILABLE

A new edition of the UCSD Pascal User's Manual has just come off the presses. It has been revised and extended and now includes over 400 pages. The assembler section has been completely rewritten and expanded to about 70 pages. Two sections on the Adaptable System have been added. The new manual, priced at $25, may be ordered from SofTech Microsystems.

UCSD Pascal USER'S GROUP MEETING PLANS MOVING FORWARD

The plans for this summer's meeting of the UCSD Pascal User's Group are moving along. Dr. Jim Gagne' has stepped forward to organize the program exchange library for the Group, and is moving very rapidly to get program exchanges going (see the article elsewhere in this issue).

We need some volunteers to organize sessions for the meeting. Sessions which have been suggested and are in need of volunteers are: Industrial Division; Word processing; Medical applications; Pascal Standardization; and the Portability Manual Project.

The Industrial Division will represent the many corporations which are licensed for distribution or internal use of UCSD Pascal. It will address such issues as corporate cooperation, jointly funded development of applications software, and perhaps media recording standardization efforts.

The Pascal Standardization Project will represent the UCSD Pascal User's Group in the Pascal Standardization efforts which are underway.

Much interest has been expressed in the preparation of a Portability Manual. Such a manual would be a guide to preparing the most portable programs possible. Since UCSD Pascal programs can be written by accident or design which are not fully portable, the goal of this project is to produce a guide which defines the programming practices which result in programs which are not fully portable.

Volunteers are needed for organizing sessions in all of the above areas. In addition, volunteers and nominations are needed for various officers of the User's Group, including the offices of President, Vice President, Secretary, and Treasurer.

USER'S GROUP PROGRAM EXCHANGE LIBRARY STARTS

Dr. Jim Gagne' of Datamed Research has temporarily taken on responsibilities for the UCSD Pascal User's Group Program Exchange Library. Jim is accepting programs submitted for the Library and will be doing all of the sorting, cataloguing, and reproduction (8" single density, IBM compatible soft-sectored floppies only now). The first three volumes will be ready for distribution, Jim said, by the middle of April.

The task which Jim has taken on is really immense, and he has proposed to do it at what amounts to his costs. He is requiring that all submissions include the documentation in "soft-copy" on the disks. He hopes that the Library will, in addition to programs, eventually contain useful procedures and functions, programming tips, algorithms and reviews of commercially available software.

If you want more information or want to volunteer to help in some way, contact: Dr. Jim Gagne', c/o Datamed Research, 1433 Roscomare Road, Los Angeles, CA., 90024.

DICOLL IS FIRST DISTRIBUTOR IN THE UK

DICOLL Datasystems, Ltd., has executed a license for distribution of UCSD Pascal on LSI/11 machines in the UK. DICOLL is said to be the largest DEC "OEMer" in England. The interest in the system is very high in the UK and DICOLL will be a very active distributor there.

The following two sections of the Newsletter contain the current Maintenance status. At this time there are about 180 Problem Reports on file. Most of them (more than 80%) have been discovered and reported by SofTech Microsystems Development staff during our software quality assurance activities. The organization of the status report has been designed to be as useful as possible within the constraint of getting the information to you as quickly as we can. The first section, "SYMPTOMS", is really an index to the second section which contains more detailed "PROBLEM REPORTS".

In the first section the problems are described in terms of the external behavior of the system, and a list of one or more Problem Report numbers is given with each symptom. You can approach the first section with the attitude of "if my system exhibits this symptom, then I should study the problem reports listed with it". We have also organized the symptom list into categories corresponding to the major system components.

SYMPTOM LIST FOR THE UCSD PASCAL SYSTEM

SYSTEM INITIALIZATION (either at boot time or after reinitialization)...

TERMINATES INEXPLAINABLY...

During bootstrap or reinitialization on PDP-11/LSI-11 systems. 2
During bootstrap on non LSI-11/PDP-11 systems. 3
When break key is pressed. 2
For no apparent reason. 42

On The 6502 Processor...
Due to lack of recovery from stack overflow. 115
During a procedure call between segments. 41

SYSTEM-WIDE FACILITIES...

BEHAVE INEXPLAINABLY...

When the assembler requires the use of a character not on the keyboard. 102
When the prompt line does not appear after typing a flush character. 103

When Improperly Handling Execution Errors...
After a stack overflow on the 6502. 115
When the code offset printed in the execution error message is unreasonable. 31

NOTE: Asterisk by report number in symptom list indicates new or revised report.

SCREEN ORIENTED EDITOR...

TERMINATES INEXPLAINABLY...

When running with memory space greater than 32k words. 52
After reading in a text file. 136
After a syntax error from the assembler. 92

BEHAVES INEXPLAINABLY...

When Z)ap command is used. 36
When F)ind command is used. 106
When D)elete command is used. 150
After a syntax error from the assembler. 92
After reading in a text file. 136
When the cursor is in the wrong position (two positions to the right) after an I)nsert. 137
When markers previously set are not in the correct position. 140
When doing a Q)uit and U)pdate to a different disk. 27
After D)elete appears to delete the wrong half of the last line of a text file. 101

Resulting In Unexpected Treatment Or Appearance Of Text...

After buffer overflow warning. 14
After insertion of very long lines. 14
When deleting off of the top or the bottom of the screen. 138
When deleting on a terminal that requires prefixed cursor moving commands. 150
After scrolling of the terminal screen. 54
Before binding in GOTOXY. 100
When the Z)ap command is used. 36
After use of RT11TOEDIT. 121
After copying text from buffer. 141, 142
After the use of the M)argin command 175

PASCAL COMPILER...

TERMINATES INEXPLAINABLY...

On a value range error when using large floating point constants. 89
After syntax error #253: procedure too long. 85
After processing a string constant longer than 80 characters. 87
After syntax error #10: error in type. 84
After syntax error : file of pointers not allowed. 83
When using large ranges on case statements. 111
When using real constants of the form '1E4'. 88

BEHAVES INEXPLAINABLY...

When attempting to escape from syntax error prompt. 25

When a program generating a compiled listing includes...
$P directive. 163
Include Files 169

On Detection Of Syntax Error...
When compiling the filehandler unit. 97
When no error message is displayed 176
#407: block 0 overflow. 8
#250: too many scopes of nested identifiers. 17
#7: error in parameter list. 33
#50: error in constant. 77
#104: undeclared identifier when using UNITSTATUS. 124
#400: illegal character. 139
#401: Unexpected end of input 151
#5: ':' expected 159

When Compiled Listing Is Directed To Disk...
And some or all files on the disk are lost after compilation. 86
And the listing file disappears after an unsuccessful compilation. 76
And the P-code offsets in the listing file decrease on successive lines. 7

FILE HANDLER...

BEHAVES INEXPLAINABLY...

Resulting in the workfile being lost when S)aved to a non-default volume. 117
Resulting in some or all files on the disk being lost after compiler produces a compiled listing to disk. 86
Resulting in unexplained appearance of the filer prompt line after a directory listing. 105
When using the eX)amine command to mark blocks. 50, 179
When using the Z)ero command. 109
When running with memory space greater than 32K words. 52
After an unexpected response to a 'yes/no' prompt. 144
When the promptline originates from the middle of a line 178

LINKER...

BEHAVES INEXPLAINABLY...

When running on a word machine. 35

When Linker Reports Error...
'constant <name> undefined' when linking an intrinsic unit. 118
'can't find x' where x is the name of an assembly routine. 47
'bad LI type' when using separate units. 51

ASSEMBLER...

TERMINATES INEXPLAINABLY...

When equating labels to external references. 2
When using .INTERP. 5
When two macros are declared with the same name. 5
When assembling more than eleven routines 16

BEHAVES INEXPLAINABLY...

When using .ASECT or .PSECT. 1
When trying to escape from the listing prompt. 3
When using backspace in the listing prompt. 3
When two macros are declared with the same name. 5
When passing a macro as a parameter. 6
When it fails to flag an invalid construct for hex constants in a decimal default radix assembler 15
When radix switch characters are used 17

When Generating A Listing File...
Resulting in unexpected output in listing file when using .NOMACROLIST. 6
Resulting in an incorrect backpatch message. 67, 15
Resulting in incorrect filename on a page header 16

Resulting In An Unexpected Error Generated...
When using .BLOCK to allocate large blocks of data. 6
When using .ASCII containing a semicolon. 6
With an incorrect message generated. 6
When radix switch characters are used 17
#15: backward ORG not allowed. 14
#36: Unexpected end of input, 15
#83: Unimplemented instruction on a PDP-11 15
#18: Invalid structure. 15

UTILITY PROGRAMS...

TERMINATE INEXPLAINABLY...

When the L2 editor is used on a text file produced with a machine of opposite byte sex. 4
When attempting to copy segments with the librarian. 146

BEHAVE INEXPLAINABLY...

When making a change using PATCH in the C)har mode of T)ype. 94
When making a change using PATCH in the H)ex mode of T)ype. 29
When making a D)ump using PATCH in 'M' (octal) mode 172
When using th COP$ function in the BASIC Compiler. 38
When using the L2 editor on a text file produced on a machine with different byte sex. 4
When RT11TOEDIT does not acknowledge device units other than 4 and 5. 15
Resulting in unexpected behavior of RT11TOEDIT. 121
When a value range error is generated when running SETUP. 10
When a promptline is hard to read in YALOE 162

BEHAVES INEXPLAINABLY WHEN IT INCLUDES...

An IF statement in which expression is FALSE constant. 7
A FOR loop which may execute an unexpected number of times. 81
A FOR loop with an upper bound of 32767 148
A procedure with very large variable declarations. 44
Incorrect evaluation of a segment function. 122
Calls to PRED and SUCC. 79
Signed character constants. 80
Calls to READCHAR and READLN. 20
An attempt to create a file on a disk. 40
Assignment of 32767 + 1 to a REAL variable on the 6502. 149
A String assigned a value from an assembled listing textfile 157
A negative index operation into a string 174
References to a user generated textfile that does not contain a NULL at the end of every page. 165

Set Expressions...
  Containing a one element set constant which is evaluated incorrectly. 9
  Containing a set declared with a negative lower bound. 74

Units...
  Containing a file declaration in the implementation section. 78
  When using noload (*$N+*) compiler directive. 82
  On byte flipped machines. 125

Intrinsic Units...
  With variable declarations, but no data segment number assigned. 147

Separate Units...
  Having more than one procedure in one separate unit. 133
  Having a call from a procedure within a separate unit to a procedure within a separate unit. 129

Long Integers...
  When performing a comparison on the 6502. 56
  When performing an addition on the 8080/Z80. 113
  When performing a multiplication on the PDP-11 or LSI-11. 171

Packed Arrays...
  When referenced on a word machine. 28
  Which may be incorrectly allocated by the compiler. 22

Intrinsic Routine(s)...
  PAGE. 128
  ARCTAN in which a bad value is returned. 98, 135
  CONCAT and COPY in string expressions. 99
  Real number output formatting. 37
  RELEASE. 21
  LN(0) 164

Output...
  Formatting of real numbers. 37
  Caused by using PAGE. 128
  Resulting in an I/O error when using unit 8 (REMOUT:). 90
  Where the file window buffer contains an incorrect value. 46, 123, 180
  Caused by functions not described in documentation. 120
  Resulting in an unimplemented instruction when using Unit 6 (PRINTER:) on a PDP-11 170

Input...
Where the file window buffer contains an incorrect value. 46, 123, 180
Caused by functions not described in documentation. 120
Where performing a READ on a real number will not allow backspacing 166
Where performing a READ on a long integer yields an incorrect value. 173

A USER ASSEMBLED ASSEMBLY LANGUAGE PROGRAM OR PROCEDURE...

BEHAVES INEXPLAINABLY...

On Any Processor When It Includes...

The .ASECT and .PSECT directives. 1
The .INTERP directive. 58
An assembly expression which is evaluated incorrectly. 49
An externally referenced constant. 134
The use of '%' to mean remainder division. 102
Passing a macro name as a parameter. 60
A predefined register name used in an expression. 69
A relative jump to an absolute address. 68
A nonunary operator used in a unary context. 69
The .BLOCK directive when used to allocate large blocks of data. 62
The .WORD directive when used with multiple arguments. 63
The .ORG directive. 64
Two or more assembly procedures within a single file linked to a host program. 133
The .PRIVATE directive when linked to an intrinsic unit. 132
An ASCII string passed as a macro parameter 156

On An 8080/Z80 Processor When It Includes...
The RESTART instruction. 34
Invalid register names on an 8080/Z80. 71,72

On A PDP-11/LSI-11 Processor When It Includes...
A jump instruction used with register address mode. 70

On A 6502 Processor When It Includes...
A JSR instruction with indirect addressing. 73
Non-standard syntax used for indirect addressing. 13

On A 6809 Processor When It Includes...
PC Relative addressing 160

On a Word Oriented Processor When It Includes...
A directive (.BLOCK, .BYTE, .ASCII) which generates an odd number of data bytes. 131

---

1. Problem Report Format

New problem entries are assigned a report number one larger than the number of existing entries, and added to the end of the problem report. Entries that are no longer valid are marked 'voided', and their report numbers are not reassigned. The assignment of report numbers is not meant to reflect any chronological ordering of the problem reports.

A problem report entry contains the following fields: report number, module, description, impact, versions affected, problem, temporary solution, and fix. Each field is required to appear in an entry. Fields that are irrelevant to the problem in question contain the symbol 'na'.

The Report Number Field contains the report number of a problem entry, which is used in the symptom list to reference entries in the problem report.

The Module Field identifies the part of the Pascal system from which the problem arises or the part of the system affected by the problem. The compiler, editor, and linker would be considered as separate modules. Problems that are not specific to one module will be identified by the name 'system' in this field.

The Description Field contains a one sentence overview of the problem.

The Impact Field describes the severity of an occurrence of the problem with respect to the integrity of the Pascal system. This field may contain one of the following: none, mild, moderate, severe, and lethal.

The Versions Affected Field lists all releases that may be affected by the problem. Releases covered in this report include II.0, II.1, and III.0. When appropriate, the specific versions are listed (e.g., II.1b3). Problem reports affecting the II.1 release may or may not exist in software released by Apple Computers, Inc. Problem reports affecting the III.0 release may or may not exist in software released by Western Digital, Inc.

The Problem Field provides a detailed description of the problem, including the enumeration of unexpected system actions that can result from specific user inputs.

The temporary solution field describes any detours that may be used to accomplish the desired task without encountering the described problem.

The Fix Field attempts to describe the change(s) required in the system program sources that will alleviate the problem. Since problem fixes may range from one line changes to a substantial redefinition and/or reconstruction of a major module, the completeness of the fix provided may vary to a great degree. The overriding constraint in the report is that the description be kept relatively brief. When known, simple fixes will have a description of the lines in the source programs that require modification, along with the changes needed. More complex fixes will provide a reference to the procedure in which the problem resides, along with a general idea of the changes required. Fixes beyond the scope of the problem list will be noted as such.

1. Problem Report Entries

Report number: 1
Module: assembler

Description:
Using .ASECT and .PSECT may cause unusual errors during assembly.

Impact: mild
Versions Affected: II.0, II.1, III.0

Problem:
These directives are undocumented and also not implemented correctly. Their lack of documentation invites misuse, which the assembler then flags with errors.

Temporary Solution:
which are used for assigning absolute addresses to labels. This can also be done using the .EQU directive.

Fix:
Generate documentation on use of absolute sections. Changes in procedure ASSEMBLE to allow blank lines and other directives in absolute sections. Changes in each directive handling procedure to control its use in absolute section.

Report number: 2
Module: system

Description:
Typing breaks makes LSI/11 system hang while booting.

Impact: moderate
Versions Affected: II.0, II.1, III.0

Problem:
An infinite loop in the interpreter occurs if the break key is typed while the system is booting or reinitializing.

Temporary solution:
Do not type break while booting or reinitializing.

Fix:
undecided

Report number: 3
Module: system

Description:
Non-11 system hangs during bootstrap.

Impact: moderate
Versions affected: II.0, II.1, III.0

Problem:
System goes into an infinite loop at boot time if BIOS returns a nonzero IORESULT on a UNITCLEAR. This occurred on an implementation where no unit 5 was configured into the system, so the BIOS returned an IORESULT of 9.

Temporary solution:
Implementations with a similar configuration must place all system files on the boot unit (unit 4) to boot successfully.

Fix: In operating system procedure INITUNITABLE, initialize UNITABLE before doing search for system files.

Report number: 4
Module: editor

Description:
L2 editor may crash on text files moved from another machine.

Impact: severe
Versions affected: II.0, II.1, III.0

Problem:
Text file headers have information which is byte sex dependent. Text files moved between machines of different byte sex cause this bug.

Temporary solution:
Use PATCH program to store chr(0) into the first 2 bytes of block 0 of the text file.

Fix:
Changes required in procedures INITIALIZE and COPYFILE.

Report number: 5

voided

Report Number: 6

voided

Report number: 7
Module: compiler

Description:
False boolean constants in if statements may cause bad code.

Impact: moderate
Versions affected: II.0, II.1, III.0

Problem:
An attempt at code optimization was made by throwing away all code generated by statements within the scope of an IF statement of the form:

```
CONST  short_circuit = FALSE;
BEGIN
  IF short_circuit THEN ...
```

Unfortunately, labels referenced by outside GOTO's and linker references to code within the IF are not taken into account, so bad code may be generated.

Temporary solution:
Change boolean constant to a variable initialized to false.

Fix:
Remove 'optimization' section in procedure IFSTATEMENT.

Report number: 8
Module: compiler

Description:
Syntax error #407 occurs, but is not described in the document.

Impact: moderate
Versions affected: II.1

Problem:
The message associated with #407 is 'Block 0 overflow'. This error is an implementation restriction caused by using too many units in a user program.

Temporary solution:
none

Fix:
undecided

Report number: 9
Module: compiler

Description:
Set constants of form [x] where ord(x)=15 are incorrectly evaluated.

Impact: mild
Versions affected: II.0, II.1, III.0

Problem:
Set constant of above form is generated as a word constant, but the word value in this case is 2's complement -32768. This value cannot be generated by our compiler due to the nature of the constant generation routine. The range of type integer is -MAXINT..MAXINT, where MAXINT = 32767.

Temporary solution:
Use "([x,y] - [y])", where x is not equal to y..

Fix:
Modification to procedure GENLDC.

Report number: 10
Module: setup

Description:
A value range error occurs while using SETUP.

Impact: severe
Versions affected: II.0, II.1, III.0

Problem:
When in C(hange mode, single command mode of SETUP, typing in an incorrect field name and following the reprompt with a carriage return will cause a value range error.

Temporary solution:
Avoid typing that sequence of inputs.

Fix:
Modification to procedure GETVAL.

Report number: 11

voided

Report number: 12

voided

Report number: 13
Module: assembler

Description:
6502 assembler accepts nonstandard syntax for indirect address mode.

Impact: mild
Versions affected: II.0, II.1, III.0.

Problem:
Instructions of the form:

```
        STA  (LABEL),X
        LDA  (LABEL,X)
        STA  (LABEL),Y
```

will assemble without being flagged as a syntax error. The code produced specifies indirect addressing, but the use of parentheses to indicate indirect addressing is not in the specification of our 6502 assembler's address mode syntax.

Temporary solution:
Use the standard syntax for indirect addressing.

Fix:
In procedure ZOP3, remove attempts to parse parentheses and replace with traps to flag their use as an error.

Report number: 14
Module: editor

Description:
Editor ruins a workfile with extra long lines and on buffer overflows.

Impact: severe
Versions affected: II.0, II.1, III.0

Problem:
In the regular (not L2) editor, random blocks of data in the workfile may be lost or duplicated after a work buffer overflow. Lines wider than 128 characters are always unreliable.

Temporary solution:
Avoid creating lines wider than the screen when possible. S(et E(nvironment at regular intervals to check 'bytes available' and avoid expanding files when less than 1000 bytes are available in the editor buffer.

Fix:
Change constant MAXCHAR for long line problem. Overflow problem is not found yet.

Report number: 15
Module: edittort11

Description:
EDITTORT11 should acknowledge device units other than 4 and 5.

Impact: mild
Versions affected: II.0, II.1, III.0

Problem:
The EDITTORT11 program is designed to run on Terak; it needs upgrade and reliability test.

Temporary solution:
unsupported

Fix:
unsupported

Report number: 16

voided

Report number: 17
Module: compiler

Description:
Erroneous error message #250 in compiler.

Impact: mild
Versions affected: II.0, II.1, III.0

Problem:
Error message 250 (too many scopes of nested identifiers) appears when one declares more than 6 segment procedures, none of which need be nested or contain any declarations. The error number should be 398.

Temporary solution:
Be aware of misleading error message.

Fix:
In procedure PROCDECLARATION (roughly line 1773), the source should be changed to generate error 398.

Report number: 18

voided

Report number: 19

voided

Report number: 20
Module: system

Description:
A program is able to continue reading data past the end of a file.

Impact: moderate
Versions affected: II.0, II.1, III.0

Problem:
The Pascal intrinsics READCH and READLN reset file attributes without regard to the EOF state. This allows I/O calls to return data subsequent to EOF condition coming true .

Temporary solution:
Check for the end of file condition (EOF) before READ operations.

Fix:
Modify the operating system procedures FREADLN and FREADCHAR to return no data if the EOF flag is true.

Report number: 21
Module: interpreter

Description:
Improper use of RELEASE may crash the system.

Impact: moderate
Versions affected: II.0, II.1, III.0

Problem:
There is no check in the standard procedure RELEASE that the address to which the heap pointer is to be set is valid (i.e., below the stack); however, enforcemment of this check would prevent users from taking advantage of this freedom to simulate dynamic allocation with RELEASE.

Temporary solution:
Avoid misusing RELEASE.

Fix:
Interpreter should trap illegal use of RELEASE.

Report number: 22
Module: compiler

Description:
A program using packed records has some data fields disturbed.

Impact: severe
Versions affected: II.0

Problem:
In certain cases, packed records may be allocated incorrectly by the compiler (the optimization of moving fields to byte boundaries is done improperly). The following is one example where this occurs:

```
packed record
  bool1, bool2, bool3, bool4, bool5, bool6 : boolean;
  twobits : (0..4);
  byte2, byte1 : char
end;
```

The fields 'twobit' and 'byte1' are both allocated to the high order byte of the packed word.

Temporary solution:
Use unpacked records or rearrange the declaration order of the record fields and try again.

Fix:
Modifications in procedure FIELDLIST.

Report number: 23

voided

Report number: 24
Module: assembler

Description:
Equating external references to labels causes unexpected results.

Impact: Moderate
Versions affected: II.0, II.1, III.0

Problem:
External references should not be used as operands to the .EQU directive. Attempts to do so will disturb the assembler's symbol table; this may cause the system to crash during assembly or enhance the probability of an ill-formed assembly code file.

Temporary solution:
If it is necessary to repeatedly use an expression containing an externally referenced label, substitute the expression as a whole everywhere it is needed.

Fix:
Modify procedure ZEQU to create the proper symbol table entry for labels equated to external references and remove the original label-type symbol table entry.

Report number: 25
Module: compiler

Description:
Escaping from the compiler after an error message may not work.

Impact: mild
Versions affected: II.0, II.1, III.0

Problem:
The escape character is fixed as chr(27) in the compiler. If a terminal configuration has another value defined as the escape character, it will not be recognized.

Temporary solution:
Type <control-[> to escape.

Fix:
Modify compiler to read escape character from USERINFO.

Report number: 26

voided

Report number: 27
Module: editor

Description:
Editor crashes when Q(uit U(pdating to a different disk.

Impact: severe
Versions affected: II.0, II.1, III.0

Problem:
The editor calls a segment procedure after the quit/update sequence. If the user pulls out the original disk and attempts to update the work file on another disk, the possibility arises that the editor's code file may well have been on the original disk; in this case, the segment procedure call will cause a segment fault and crash the system, destroying the updated work file.

Temporary solution:
The Q(uit W(rite sequence allows the original disk to be taken out and replaced with another disk for writing the updated work file to. Do not change the disk until after the prompt appears from W(rite.

Fix:
Difficult. Current P-machine architecture assumes that proper disk is online when a segment must be loaded into memory.

Report number: 28
Module: system

Description:
Gap in the P-machine specification affecting word addressed machines.

Impact: na
Versions affected: na

Problem:
P-machine specification does not state whether the byte portion of a word/byte address couple is signed or unsigned. This has some implications on the size of data items and negative addressing. Only word addressed machines are affected.

Temporary solution:
na

Fix:
na

Report number: 29
Module: patch

Description:
The PATCH utility does not work well on some terminals.

Impact: moderate
Versions affected: II.0, II.1, III.0

Problem:
When in the T)ype mode of PATCH, a change made in the H(ex mode isn't accepted, but a change made in the C)har mode is accepted. Also, the vector arrows on terminals with prefixes don't work.

Temporary solution:
unsupported

Fix:
unsupported

Report number: 30

voided

Report number: 31
Module: system

Description:
IPC value printed in execution error messages appears incorrect.

Impact: mild
Versions affected: II.0, II.1, III.0

Problem:
The name and proper setting of the 'BYTE-FLIPPED' field in SYSCOM is confusing. It is currently referenced for only one system function - printing an execution error's IPC value correctly. P-machine implementations that use a seg-relative IPC should set this field to true; all others (regardless of byte sex) should set it to false.

Temporary solution:
Use SETUP to change the value of 'BYTE-FLIPPED' field.

Fix:
Rename field to more accurately reflect its use by the system.

Report number: 32
Module: assembler

Description:
Escaping and backspacing the assembler listing prompt may not work.

Impact: mild
Versions affected: II.0, II.1, III.0

Problem:
Escape is fixed as chr(27) and backspace is fixed as chr(8) in the assembler's listing prompt. If a terminal configuration has other values defined for these characters, they will not be recognized.

Temporary solution:
Use break key to escape and type file names right the first time.

Fix:
Modify procedure INITIALIZE to do a read of a string to get the list file name, defining the escape sequence as '<esc>-<ret>'.
Use USERINFO.ALTMODE instead of chr(27) for the escape character.

Report number: 33
Module: compiler

Description:
Compiler flags untyped variable parameters in external procedures.

Impact: mild
Versions affected: II.0, II.1, III.0

Problem:
Compiler does not allow the last parameter of an EXTERNAL procedure or function to be an untyped variable parameter.

Temporary solution:
Reorder parameters.

Fix:
Modifications in procedure PARMLIST.

Report number: 34
Module: assembler

Description:
RST x (restart) in the Z80 assembler may assemble as RST 0.

Impact: Moderate
Versions affected: II.0, II.1, III.0

Problem:
Z80 assembler only accepts Zilog operand format (i.e., operand is one of first seven multiples of 8). Unfortunately, all other values are mapped to 0 without an error message, including Intel operand format (i.e., 0 to 7).

Temporary solution:
Use Zilog operand format carefully.

Fix:
In procedure ZOP17, flag any operands with an error if the operand value MOD 8 is nonzero.

Report number: 35
Module: linker

Description:
Linker does not work on a word addressed machine.

Impact: moderate
Versions affected: II.0, II.1, III.0

Problem:
Impossible to link separate units and assembly procedures on word addressed machines.

Temporary solution:
none

Fix:
In the procedure COPYINPROCS, the following code must be added. Existing code is in upper case, and addition is in lower case:

```
        PDP := GETCODEP (ORD(SEGBASE),SEGLENG - 2*NEWPROC - 2);
        if wordmachine then
        storeword(2*(ord(pdp) - ord(jtab)),pdp,0) else
        STOREWORD(ORD(PDP) - ORD(JTAB),PDP,0);
        WP := NEXT
      END;
```

Report number: 36
Module: editor

Description:
Strange behavior of editor after using Z(ap command.

Impact: severe
Versions affected: II.0, II.1, III.0

Problem:
Editor gets confused if a Z(ap is done without having done a F(ind, I(nsert, or R(eplace command immediately beforehand. Once confused, any editor commmands will behave incomprehensibly, and the problems will not vanish if attempts are made to save the strange work file and re-edit it.

Temporary solution:
Take care to use Z(ap only after the proper lead-in commands.

Fix:
Modifications in procedures ZAPIT and INSERT to invalidate bad zaps in a manner similar to COPYBUFFER.

Report number: 37
Module: system

Description:
A real number is displayed with full accuracy when none should show.

Impact: mild
Versions affected: II.0, II.1, III.0

Problem:

```
var x:real;
x:=1.00000;
writeln(x:2:0);
```

The ':0' specifies output should be 1., but output is 1.00000.
Positive argument values do work correctly .

Temporary solution:
Simulate by writing as follows: writeln(trunc(x),'.')
Note that this may cause an execution error for large real numbers.

Fix:
Change default communication between compiler and operating system to use -1 for default, rather than 0.

Report number: 38
Module: basic compiler

Description:
The COP$ function does not work.

Impact: unknown
Versions affected: II.0, II.1, III.0

Problem:
The BASIC compiler is currently unsupported.

Temporary solution:
unsupported

Fix:
unsupported

Report number: 39

voided

Report number: 40
Module: system

Description:
A file may be opened between two adjacent files in directory.

Impact: moderate
Versions affected: II.0, II.1, III.0

Problem:
The operating system occasionally opens new files between two existing files with no disk space between them. Subsequent attempts to write to this file cause the error 'no room on vol'.. This problem has occurred only after crunching a disk, when one contiguous space is available.

Temporary solution:
Create a dummy file to divide the available disk space into two chunks.

Fix:
undecided

Report number: 41
Module: interpreter

Description:
6502 based systems may crash without generating any error messages.

Impact: Moderate
Versions affected: II.0, II.1, III.0

Problem:
6502 interpreter doesn't attempt to detect a 'no proc in seg table' error during segment procedure calls.

Temporary solution:
none

Fix:
In procedure LODSEG, check that the value in the SEG field of a newly loaded segment matches the segment operand of the CXP instruction and cause an execution error if they are different.

Report number: 42
Module: assembler

Description:
Opcodes may get scrambled during relocation of assembly procedure.

Impact: Moderate
Versions affected: II.0, II.1, III.0

Problem:
Assembler retains the relocation type of operands used with the .EQU directive and neglects to initialize the relocation type when it generates code for one word instructions with no operands; thus, if a one word inherent instruction immediately follows an equate directive, the opcode itself may be put on the relocation list.

Temporary solution:
Arrange source lines of an assembly routine to avoid occurrences of this sequence of instructions.

Fix:
In procedure ASSEMBLE, set variable RELOCATE to NULLREL at the end of the repeat loop.

Report number: 43

voided

Report number: 44
Module: compiler

Description:
Programs with very large data segments may crash.

Impact: moderate
Versions affected: II.0, II.1, III.0

Problem:
No check is made that the size of a data segment exceeds the 32K word limit.

Temporary solution:
Avoid using huge data declarations.

Fix:
Modifications to procedure DECLARES.

Report number: 45

voided

NOTE: There is further information now available on this report. See report number 180.

Report number: 46
Module: compiler

Description:
GET may work improperly on files of char on byte-flipped systems.

Impact: moderate
Versions affected: II.0

Problem:
Code generated by the compiler is incorrect for byte-flipped machines when references are made to the window variable of a file of character. The operating system's GET routine treats the window variable as a packed array of char, and stores the input character in the 1st byte of the array; however, references to F^ are assumed by the compiler to deal with a full word variable of type char (which is byte sex dependent).

Temporary solution:
Write a short function that uses a record variant to exchange the bytes of a variable of type char. Use this function instead of making direct references to the window variable.

Fix:
Modifications to procedure SELECTOR. Add GENLDC(0) and change ACCESS to BYTE.

Report number: 47
Module: assembler

Description:
Assembly procedures may not be linkable to their host programs.

Impact: moderate
Versions affected: II.0, II.1, III.0

Problem:
If an assembly procedure name is also a reserved identifier in the assembler version, the assembler fails to emit linker information for the procedure. Reserved identifiers include machine instruction names and register names.

Temporary solution:
Choose names for assembly procedures that do not conflict with reserved identifiers.

Fix:
In procedure PROCEND, cause an error message if SYM^.ATTRIBUTE of the procedure name is not set to CURRENTATRIB. See procedure ZGLOBAL for ideas on how this is handled.

Report number: 48

voided

Report number: 49
Module: assembler

Description:
Assembler expressions may exhibit unusual behavior.

Impact: moderate
Versions affected: II.0, II.1, III.0

Problem:
The assembler's expression parser works strictly left-to-right on operands, with no operator precedence. This problem is caused by lack of proper documentation.

Temporary solution:
Order the operands to reflect the constraint mentioned above or use angle brackets ('<' and '>') to form separately evaluated subexpressions.

Fix:
Supply documentation.

Report number: 50
Module: filer

Description:
eX)amine command refuses to mark block 10 as a bad block.

Impact: unknown
Versions affected: II.0, II.1, III.0

Problem:
The eX)amine command refuses to mark block 10 when using a duplicate directory. This is apparently related to the fact that block 10 is part of a possible directory; thus, the problem may also exist for block 6.

Temporary solution:
none

Fix:
undecided

Report number: 51
Module: compiler

Description:
Separate units may not be linkable.

Impact: moderate
Versions affected: II.1b3 and II.1b2b only

Problem:
This bug results from an attempted optimization of linker information, and the linker info generated is unusable by the linker.

Temporary solution:
Use regular units.

Fix:
Remove the following lines from procedure GLOBALSEARCH:

```
IF LITYPE IN [SSEPPROC,SSEPFUNC] THEN
  LITYPE := SUCC(SUCC(LITYPE))
```

Report number: 52
Module: system

Description:
MEMAVAIL must be used carefully on machines with more than 32K words.

Impact: severe
Versions affected: II.0, II.1, III.0

Problem:
MEMAVAIL returns # of available words as an integer. If the address space is less than 32K words, this integer is positive; however, if greater than 32K words, it can become negative. System software having this problem include the editor and filer.

Temporary solution:
Add checks for negative result of MEMAVAIL, assuming that a negative value indicates a large amount of unused memory.

Fix:
undecided

Report number: 53

voided

Report number: 54
Module: editor

Descripton:
Long lines of text may cause problems when editor scrolls.

Impact: severe
Versions affected: II.0, II.1, III.0

Problem:
When a line which extends to the rightmost column of the terminal is scrolled onto the screen, a character is written in the rightmost column which causes a linefeed on most terminals. This leaves an empty line on the screen and causes any editing of the text above the line to be incorrect. The main problem is in the scrolling.

Temporary solution:
Type V(erify to redisplay screen correctly before continuing.

Fix:
Put SCROLLDOWN field in syscom to be set by SETUP.

Report number: 55

voided

Report number: 56
Module: long integers

Description:
6502 long integers may crash the system.

Impact: moderate
Versions affected: II.0, II.1, III.0

Problem:
The compare operator in the 6502 long integer package loses track of the stack when it is supposed to return false.

Temporary solution:
none

Fix:
In DECCMP routine, change branch instruction following the 'push false' code from BNE to BEQ.

Report number: 57

voided

Report number: 58
Module: assembler

Description:
Problems with .INTERP while assembling or running assembly code.

Impact: moderate
Versions affected: II.0, II.1, III.0

Problem:
The predefined symbol .INTERP works properly only when equated to a label and used by referencing the label. Using it in the operand field of instrucions or directives causes relocation to be mishandled; in addition, various words of memory are disturbed through NIL pointer references.

Temporary solution:
Always equate .INTERP (plus offset) to a label.

Fix:
Modifications in procedure PUTWORD to avoid references to RELOCATE.SYM^.ATTRIBUTE, as SYM = NIL when .INTERP is used directly.

Report number: 59
Module: assembler

Description:
Two macro definitions with the same name crashes the assembler.

Impact: moderate
Versions affected: II.0, II.1, III.0

Problem:
Assembler neglects to check if a name has been used when scanning a macro definition name; if name given identifies a previous macro definition, the first macro is expanded and the assembler acts unpredictably.

Temporary solution:
Define and use macro names carefully.

Fix:
In procedure PIDENT, cause an error if CURRENTATRIB = MACROS before switching SOURCE to MACROSOURCE.

Report number: 60
Module: assembler

Description:
Macros names passed as parameters to other macros don't work.

Impact: moderate
Versions affected: II.0, II.1, III.0

Problem:
This situation overextends the macro expanding facilities of the assembler, but is not checked for.

Temporary solution:
Avoid passing macro names as parameters.

Fix:
In procedure PIDENT, cause an error if SOURCE = MACROSOURCE before switching SOURCE to MACROSOURCE.

Report number: 61
Module: assembler

Description:
Macros may be expanded on assembled listings with .NOMACROLIST on.

Impact: none
Versions affected: II.0, II.1, III.0

Problem:
are suppressed, but subsequent calls are expanded.

Temporary solution:
none

Fix:
In procedure LEX, in procedure PKWORD:

Remove this code:

```
If listing and not display and (mclistlevel=mcstkindex) then
begin
  .
  .
  .
end;
lex;
```

Substitute this code:

```
If listing and not display and (mcstkindex = mclistlevel) then
  begin
    display := true;
    if console then writeln;
  end;
if mclistlevel > mcstkindex then
  mclistlevel := mcstkindex;
lex;
```

Report number: 62
Module: assembler

Description:
Unnecessary size restrictions on .BLOCK directive.

Impact: moderate
Versions affected: II.0, II.1, III.0

Problem:
data.

Temporary solution:
Use consecutive .BLOCK's.

Fix:
Replace the BUFLIMIT restriction with 32766.

Report number: 63
Module: assembler

Description:
The .WORD directive may act inconsistently with multiple arguments.

Impact: moderate
Versions affected: II.0, II.1, III.0

Problem:
Multiple arguments are delimited by commas. A null argument appearing between 2 commas is defined to be 0. This is true for .BYTE directive, but .WORD replaces null arguments with the value of the preceding argument.

Temporary solution:
Avoid using null arguments.

Fix:
In procedure ZWORD, move INITVALUE into the repeat loop.

Report number: 64
Module: assembler

Description:
The .ORG directive may cause strange behavior in assembly programs.

Impact: moderate
Versions affected: II.0, II.1, III.0

Problem:
(using .ABSOLUTE directive) to initialize the location counter.
Use in relocatable programs causes unexpected results.

Temporary solution:
Avoid using .ORG.

Fix:
Restrict the context in which it may be used and flag other uses with an error.

Report number: 65
Module: assembler

Description:
ASCII strings containing semicolons get flagged with misleading error.

Impact: mild
Versions affected: II.0, II.1, III.0

Problem:
Assembler will not accept semicolons in an ASCII string.

Temporary solution:
Split string around semicolons and use .BYTE 3BH to form semicolons.

Fix:
In procedure GETCHAR, check if NOTSTRING is true before scanning a comment.

Report number: 66
Module: assembler

Description:
Error messages for some errors are wrong or blank.

Impact: mild
Versions affected: II.0, II.1, III.0

Problem:
On system errors, the assembler uses IORESULT as an index into a file of error messages. The IORESULT specification has subsequently changed, so IORESULT values greater than 13 will map either into wrong or unused error messages.

Temporary solution:
Remove errors file from disk while assembling to force assembler to print the error number, which is defined as '46 + IORESULT'. Use BIOS document to determine system error from IORESULT value.

Fix:
Update the errors file, or define a system procedure that returns valid error messages.

Report number: 67
Module: assembler

Description:
Patch messages on listings seem to show bad code being assembled.

Impact: mild
Versions affected: II.0, II.1, III.0

Problem:
Patch messages on least-significant-byte-first machines seem to indicate that zero bytes are always patched into the generated code. The code is fine, but the patch messages are wrong.

Temporary solution:
Ignore patch messages.

Fix:
In procedure PATCHPRINT, 'IF LISTHIFIRST THEN' should be changed to 'IF LISTHIFIRST OR BYTESIZE THEN'.

Report number: 68
Module: assembler

Description:
Assembler may generate wrong code for relative addresses.

Impact: moderate
Versions affected: II.0, II.1, III.0

Problem:
For relocation reasons, the assembler cannot handle relative branches to absolute addresses; however, instead of flagging this as an error, it puts the value of the absolute address as the relative offset.

Temporary solution:
Use absolute branches to absolute addresses.

Fix:
In procedure PUTRELWORD, trap absolute addresses as invalid.

Report number: 69
Module: assembler

Description:
Expressions may exhibit strange behavior.

Impact: moderate
Versions affected: II.0, II.1, III.0

Problem:
The expression parser is rather shaky and allows the following atrocities: register names used as absolute values, no errors and incorrect code caused by unmatched angle brackets around subexpressions, use of nonunary operators in a unary context with unexpected results.

Temporary solution:
Use expressions sparingly and carefully.

Fix:
Rewrite function EXPRESS.

Report number: 70
Module: assembler

Description:
11 assembler allows JMP with register address mode.

Impact: moderate
Versions affected: II.0, II.1, III.0

Problem:
In the 11 assembler, JMP instruction is grouped and handled with instructions which allow registers as operands, but this mode is not defined for jump instructions. Attempts should be flagged with an error.

Temporary solution:
Stick to proper syntax.

Fix:
In procedure ZOP3, if MODE1 = 0 and OPBYTE.BWORD contains the opcode for jump, flag as an error.

Report number: 71
Module: assembler

Description:
8080 assembler interprets invalid register operands as constants.

Impact: moderate
Versions affected: II.0, II.1, III.0

Problem:
8080 assembler uses routines which do not differentiate between registers and constants. Almost no checking is done for operands which must be registers B, D, H, or SP; other register names will slip through without error messages and generate bad code.

Temporary solution:
Scrupulously obey Intel syntax rules.

Fix:
Better type checking of operands in all ZOP procedures.

Report number: 72
Module: assembler

Description:
Z80 assembler generates bad code instead of flagging invalid operands.

Impact: moderate
Versions affected: II.0, II.1, III.0

Problem:
Z80 assembler uses routines which do not differentiate between registers and constants. No checking is done for operands which must be subsets of the register pairs; other register names will slip through without error messages and generate bad code. The worst example is the instruction: LD HL,(DE) which treats DE as an absolute constant instead of causing an error message.

Temporary solution:
Scrupulously obey Zilog syntax rules.

Fix:
Better type checking of operands in all ZOP procedures.

Report number: 73
Module: assembler

Description:
6502 assembler may generate wrong opcode on JSR instructions.

Impact: moderate
Versions affected: II.0, II.1, III.0

Problem:
When the 6502 assembler encounters an attempt to use indirect addressing on a JSR instruction, it emits a JMP indirect opcode instead of flagging with an error message.

Temporary solution:
Scrupulously avoid JSR indirects.

Fix:
In procedure ZOP5, if an atsign is scanned in operand field and OPBYTE.GOODBYTE contains the JSR opcode, flag as an error.

Report number: 74
Module: compiler

Description:
Compiler allows invalid set declarations on sets of integers.

Impact: moderate
Versions affected: II.0, II.1, III.0

Problem:
A "set of -1..n" is not caught as an error, but the P-machine implementation of sets does not allow a negative lower bound; this can crash the system during execution.

Temporary solution:
Avoid negative lower bounds.

Fix:
Modifications in procedure TYPE.

Report number: 75

voided

Report number: 76
Module: compiler

Description:
The compiler removes compiled listings after some errors.

Impact: mild
Versions affected: II.0, II.1, III.0

Problem:
On a terminal error (error number > 400), the compiler does not lock the listing file before terminating.

Temporary solution:
none

Fix:
Modifications to procedure ERROR.

Report number: 77
Module: compiler

Description:
The compiler flags some long integer constants with errors.

Impact: mild
Versions affected: II.0, II.1, III.0

Problem:
Long integer constants (i.e., > 32767) cannot be declared unless they are negative.

Temporary solution:
Use a long integer variable.

Fix:
Modifications to procedure CONSTANT.

Report number: 78
Module: compiler

Description:
Units may not declare files in implementation section.

Impact: moderate
Versions affected: II.0, II.1, III.0

Problem:
A unit may not have a file declared in its implementation part. This restriction is in the specification, but is not enforced by the compiler. Programs that attempt to do so will compile, but won't execute properly.

Temporary solution:
Move file declaration to interface part.

Fix:
undecided

Report number: 79
Module: compiler

Description:
No value range checking is done for PRED and SUCC.

Impact: moderate
Versions affected: II.0, II.1, III.0

Problem:
Compiler doesn't emit range check code after PRED and SUCC.

Temporary solution:
Manual range checking in program.

Fix:
Modifications in procedure COROUTINE.

Report number: 80
Module: compiler

Description:
Unary plus signs on char variables are not flagged with a syntax error.

Impact: none
Versions affected: II.0, II.1, III.0

Problem:
Compiler does not detect the error in expressions like " +'.' ".

Temporary solution:
Avoid unary plus on character data.

Fix:
Modifications to procedure SIMPLEXPRESSION.

Report number: 81
Module: compiler

Description:
'for i := i+1 to i+n do' does not work as expected.

Impact: moderate
Versions affected: II.0, II.1, III.0

Problem:
For i:=i+1 to i+n do write('.')" compiles to code that writes out n+1 dots . i+1 is stored into i before i+n is evaluated, thus making the upper bound '(i+1)+n').

Temporary solution:
Use temporaries to hold bound expressions.

Fix:
Modifications to procedure FORSTATEMENT.

Report number: 82
Module: compiler

Description:
Intrinsic unit data segments are not unloaded when using {$N+}.

Impact: moderate
Versions affected: II.1b3

Problem:
When $N is used to make intrinsic units nonresident in memory, the compiler does not generate RELSEQs for the data segments of the intrinsic units.

Temporary solution:
Avoid using $N.

Fix:
Modifications to procedure BODY.

Report number: 83
Module: compiler

Description:
Declaring files of pointer variables crashes compiler.

Impact: moderate
Versions affected: II.0, II.1, III.0

Problem:
Declaring a "file of ^x" where x is a variable causes a syntax error to be generated, but crashes the compiler with a nil pointer reference after the error message.

Temporary solution:
Avoid usage.

Fix:
Modifications to procedure TYPE.

Report number: 84
Module: compiler

Description:
Invalid use of reserved word 'packed' crashes compiler.

Impact: moderate
Versions affected: II.0, II.1, III.0

Problem:
Whenever "packed" is used illegaly (e.g. packed real), the compiler gives a syntax error and then crashes.

Temporary solution:
Use "packed" legally.

Fix:
Modifications to procedure TYPE.

Report number: 85
Module: compiler

Description:
Compiler may crash after generating syntax error 'procedure too large'.

Impact: moderate
Versions affected: II.0, II.1, III.0

Problem:
Unresolved jumps after a procedure-too-long error cause the the compiler to attempt chaining down nonexistent links. This can manifest itself in case statements, if statements, gotos etc.

Temporary solution:
Always escape from procedure-too-long errors.

Fix:
Modifications to procedure PUTLABEL.

Report number: 86
Module: compiler

Description:
Compiled listings to a disk file will destroy the disk.

Impact: lethal
Versions affected: II.1

Problem:
When making a listing to a disk, the compiler invariably overwrites the disk directory.

Temporary solution:
Avoid sending list files to disk.

Fix:
Modifications to procedure COMPILER.

Report number: 87
Module: compiler

Description:
Constant character strings longer than 80 chars crashes compiler.

Impact: moderate
Versions affected: II.0, II.1, III.0

Problem:
Weak checking of an implementation restriction. This should be flagged with a syntax error.

Temporary solution:
Avoid long string constants.

Fix:
Modifications to procedure STRING.

Report number: 88
Module: compiler

Description:
Real numbers of the form '1E4' or '1e4' will crash compiler.

Impact: moderate
Versions affected: II.0, II.1, III.0

Problem:
Compiler expects a decimal point after the mantissa.

Temporary solution:
Use decimal point and fraction part on all real numbers.

Fix:
undecided

Report number: 89
Module: compiler

Description:
Large real constants may crash the compiler.

Impact: moderate
Versions affected: II.0, II.1, III.0

Problem:
Compiler assumes that real constants are within the range of the host machine's floating point package, due to the lack of a floating point standard. A real constant with exponent larger than the host implementation allows will get an execution error from interpreter's power-of-ten P-code.

Temporary solution:
Know thy machine's limitations.

Fix:
undecided

Report number: 90
Module: interpreter

Description:
Some Z80/8080 interpreters may not recognize unit 8 (REMOUT:).

Impact: moderate
Versions affected: II.0

Problem:
The Z80/8080 interpreter has only 7 I/O units; thus, REMIN: will be available, but REMOUT: will not.

Temporary solution:
Use unit 7 (REMIN:) as I/O device.

Fix:
Add unit 8 (REMOUT:) to Runtime Support Package (RSP).

Report number: 91

voided

Report number: 92
Module: editor

Description:
Editor crashes immediately after entry from an assembler syntax error.

Impact: moderate
Versions affected: II.0, II.1, III.0

Problem:
Editor uses a lame method for finding syntax error messages in the file system.syntax and crashes if an error message is missing. The message usually missing is 500 (assembly language error).

Temporary solution:
Remove system.syntax from disk before assembling. This causes the editor to report only the error number.

Fix:
Tighter range checking in procedure PUTSYNTAX.

Report number: 93

voided

Report number: 94
Module: patch

Description:
PATCH may not update a patched file.

Impact: moderate
Versions affected: II.0, II.1, III.0

Problem:
Changes made in C)har mode of T)ype are not done after Q)uit - S)ave - Q)uit sequence.

Temporary solution:
after Q)uit and S)ave, R)ead a different block and then Q(uit.

Fix:
unsupported

Report number: 95

voided

Report number: 96

voided

Report number: 97
Module: compiler

Description:
Compiler will not compile the file handler unit.

Impact: moderate
Versions affected: II.1

Problem:
The identifier 'filehand' is declared at system lex level.

Temporary solution:
Don't use this name.

Fix:
In procedure COMPINIT, make 'filehand' a 'module' and not
a procedure.

Report number: 98
Module: interpreter

Description:
Z80/8080 interpreter arctangent function works improperly.

Impact: moderate
Versions affected: II.0, II.1, III.0

Problem:
Arctan function for Z80/8080 returns correct value for arguments between -1 and high positive numbers. For arguments less than -1, it adds pi to the result.

Temporary solution:
Write envelope arctan function that calls regular arctan and adjusts result as follows:

```
if x < -1 then arctangent := arctan(x) - pi
else arctangent := arctan(x)
```

Fix:
In procedure FPFATAN, calculate arctangent with the absolute value of the argument (using subroutine FPFABS), and negate the result (using subroutine FPFNEG) if the argument is negative.

Report number: 99
Module: compiler

Description:
"copy(s1,1,2)=copy(s2,1,2)" generates bad code.

Impact: moderate
Versions affected: II.0, II.1, III.0

Problem:
Due to the misuse of a temporary variable, expressions of the form

Temporary solution:
Assign individual results into temps and compare temps.

Fix:
Modifications to procedure EXPRESSION.

Report number: 100
Module: system

Description:
Systems sent with improper GOTOXY procedure bound in operating system.

Impact: mild
Versions affected: II.0, II.1, III.0

Problem:
Systems were sent with a Datamedia GOTOXY bound to operating system.

Temporary solution:
Use YALOE to create new GOTOXY for terminal in use and bind it in.

Fix:
Send terminal independent GOTOXY that uses sequences of cursor move instructions to position cursor.

Report number: 101
Module: editor

Description:
D(elete seems to delete wrong part of the last line in a text file.

Impact: mild
Versions affected: II.0, II.1, III.0

Problem:
If last line of the text file is followed by EOF instead of EOLN, attempts to delete the last half of the line by typing <CR> will cause the first half to dissappear.

Temporary solution:
Delete one character at a time.

Fix:
Change <cr> command so that, on last line, the cursor is moved to the end of the line instead of the beginning.

Report number: 102
Module: assembler

Description:
Assembler has problems with its special character definitions.

Impact: moderate
Versions affected: II.0, II.1, III.0

Problem:
The assemblers use '!' for logical union, and ' ' for logical negation. Terminals that do not provide these keys cannot use the corresponding assembler operators. Ambiguity exists with the use of '%' to represent remainder division (mod) in expressions and its use in macros to indicate a macro parameter. The result is that the mod operation cannot be used in a macro definition.

Temporary solution:
none

Fix:
Replace special character definitions of operators with predefined symbols similar to .INTERP (e.g., .OR and .NOT).

Report number: 103
Module: system

Description:
User prompt does not appear after typing flush to stop output.

Impact: none
Versions affected: II.0, II.1, III.0

Problem:
The removal of the distinction between SYSTERM: and CONSOLE: with respect to disabling flush on UNITWRITES presents a potential problem in some UCSD supplied programs; i.e., the prompt following the flush state can never appear.

Temporary solution:
The next input cancels the flush state. Type a space after typing the flush key.

Fix:
na

Report number: 104

voided

Report number: 105
Module: interpreter

Description:
TI interpreter may disturb some of the filer's prompt line.

Impact: none
Versions affected: II.0, II.1, III.0

Problem:
This bug resides in TI 9900 RSP or BIOS. If a file directory listing is longer than one page, one may type <space> to continue the directory listing or <esc> to terminate it. After termination, the filer prompt line is only partially redisplayed.

Temporary solution:
Type space to redisplay complete prompt line.

Fix:
Modifications to RSP or BIOS.

Report number: 106
Module: editor

Description:
Finding empty strings does not work as expected.

Impact:mild
Versions affected: II.0, II.1, III.0

Problem:
F)ind // finds the delimiter '/'. This should find the null string. In addition, the pattern is set to a null string after a successful FIND.

Temporary solution:
none

Fix:
Modifications to procedure FIXD.

Report number: 107

voided

Report number: 108

voided

Report number: 109
Module: filer

Description:
All block numbers used with Zero are one less than they should be.

Impact: mild
Versions affected: II.0, II.1, III.0

Problem:
The filer assumes that block numbers are zero-based, and expects inputs to reflect that assumption. For instance, the user manual indicates to type 493 as the number of blocks when zeroing IBM 3740 disks, which actually have 494 blocks available.

Temporary solution:
Consult user manual before using Zero.

Fix:
Emphasize this convention in documentation.

Report number: 110

voided

Report number: 111
Module: compiler

Description:
Case statements with large selector range may crash the compiler.

Impact: moderate
Versions affected: II.0, II.1, III.0

Problem:
Big ranges on case statements (e.g. case i of -10000,10000: end) can cause trouble. This is an implementation restriction that is only partially protected. The current maximum range is 256.

Temporary solution:
Avoid ranges larger than 256.

Fix:
Modifications to procedure CASESTATEMENT.

Report number: 112

voided

Report number: 113
Module: long integers

Description:
8080/Z80 long integers may crash system on a simple addition.

Impact: moderate
Versions affected: II.0, II.1, III.0

Problem:
In the 8080/Z80 decops, an addition that creates a result one digit longer than operands with an even number of digits (i.e., 52 + 48 = 100) crashes the system.

Temporary solution:
none

Fix:
In decimal addition routine, fix code that handles rounding and generation of new lead digit '1' to prevent stack pointer from getting set to 100 hex.

Report number: 114

voided

Report number: 115
Module: interpreter

Description:
6502 interpreter does not recover from stack overflow.

Impact: moderate
Versions affected: II.0, II.1, III.0

Problem:
When loading code segments, the test for overflow is done after overwriting system data structures with new code.

Temporary solution:
Use swapping option when compiling large programs to avoid stack overflows.

Fix:
In LODSEG routine, check if the segment will overlay the heap before loading the segment into memory.

Report number: 116

voided

Report number: 117
Module: filer

Description:
Filer may lose track of the work file.

Impact: unknown
Versions affected: II.0, II.1, III.0

Problem:
File handler loses track of work file when it is saved to non-default volume. Subsequent W(hat commands responds with 'not named (not saved)'. Related problems may exist.

Temporary solution:
Save work file on default volume.

Fix:
undecided

Report number: 118
Module: linker

Description:
Assembly routines can't reference constants in an intrinsic unit.

Impact: mild
Versions affected: II.1

Problem:
Compiler bug in intrinsic units. No CONSTDEF linker info entry is emitted for constants in the interface section of an intrinsic unit.

Temporary solution:
Declare constant in assembly program.

Fix:
Emit CONSTDEF type linker info for intrinsic units.

Report number: 119

voided

Report number: 120
Module: system

Description:
Existing documentation on unit I/O doesn't describe control word.

Impact: none
Versions affected: II.0, II.1

Problem:
Unit I/O descriptions in system documents do not describe the additional functions available using the control word, which is described in the 10 April 79 BIOS specification. Control word functions are implemented on all II.0 and II.1 release systems.

Temporary solution:
Obtain BIOS document.

Fix:
Update system documentation.

Report number: 121
Module: rt11toedit

Description:
The RT11TOEDIT utility has numerous problems.

Impact: mild
Versions affected: II.0, II.1, III.0

Problem:
RT11TOEDIT has problems expanding tab characters, has a rather low maximum line length, and will only work with units 4 and 5. Other problems may exist.

Temporary solution:
unsupported

Fix:
unsupported

Report number: 122
Module: compiler

Description:
Segment functions do not work in III.0 system.

Impact: moderate
Versions affected: III.0

Problem:
Compiler emits getsegs and releasesegs around the call. When the release occurs, the function value is on the evaluation stack. Except on implementations with a separate evaluation stack, this causes two problems: relseg assumes an empty evaluation stack, and if the segment does get released, the function value is lost.

Temporary solution:
Avoid segment functions.

Fix:
Modifications to compiler such that the function value is stored in a temporary before the relseg and reloaded on the stack after the call. An alternative is to modify procedure RELSEG, but this change would not be absolutely secure.

Report number: 123
Module: system

Description:
Window variable may not be maintained correctly for interactive files.

Impact: moderate
Versions affected: II.0, II.1, III.0

Problem:
The window variable after reading from an interactive file should contain the last "gotten" variable; instead, it contains the next variable. This lookahead is consistent with Jensen & Wirth files, but not with interactive files.

Temporary solution:
Do not depend on window variable of an interactive file.

Fix:
undecided

Report number: 124
Module: compiler

Description:
UNITSTATUS is not supported on release compilers.

Impact: mild
Versions affected: II.0, III.0

Problem:
The standard procedure unitstatus (documented in the 10 April 79 BIOS document) is not supported in the II.0 or III.0 compiler, but only in II.1. Some of the utilities in the Adaptable system use unitstatus, but may not be compiled on same because they are supplied with II.0 compilers.

Temporary solution:
none

Fix:
undecided

Report number: 125
Module: compiler

Description:
The III.0 compiler may emit incorrect segment information.

Impact: moderate
Versions affected: III.0

Problem:
The III.0 compiler emits incorrect seginfo information (see II.1 intrinsic unit documentation) when run on a byte-flipped (most significant byte first) machine.

Temporary solution:
none

Fix:
Change compiler so that byte sex field of seginfo is not hard wired to LEAST_SIG_FIRST.

Report number: 126

voided

Report number: 127

voided

Report number: 128
Module: compiler

Description:
The PAGE intrinsic emits an ASCII form feed character.

Impact: mild
Versions affected: II.0, II.1, III.0

Problem:
The PAGE routine is parsed by the compiler and translated directly to a write of an ASCII FF (form feed). This character may not be recognized by some terminals and printers as a page command.

Temporary solution:
If your printer doesn't recognize form feeds, program the pagination.

Fix:
Parameterize the form feed character by adding a new field to SYSCOM or a new routine analagous to GOTOXY.

Report number: 129
Module: compiler

Description:
Compiler has problems with separate units.

Impact: moderate
Versions affected: II.0, II.1, III.0

Problem:
Separate units have the following problems:

1) When used by a program, the initialization section is ignored.

2) Procedures within a separate unit cannot call other procedures in separate units. Units that attempt to do so compile and link without error and then crash when executed.

Temporary solution:
Use regular units.

Fix:
The init section should either be flagged by the compiler or an EXTPROC linker entry should be generated in the host program for procedure #1 and the call should be made. The inter-unit procedure call should either be flagged by the compiler or should generate a SEPPREF linker entry.

Report number: 130

voided

Report number: 131
Module: assembler

Description:
Word oriented assemblers produce bad code if LC isn't maintained on word boundary.

Impact: moderate
Versions affected: II.0, II.1, III.0

Problem:
Programmer must ensure that LC is aligned to word boundary after generating byte data when instructions are to follow. Failure to do so will cause the assembler to generate a garbage instruction immediately following the data bytes.

Temporary solution:
Be sure to use even number of data bytes or use .ALIGN.

Fix:
In all ZOPs, remove 'IF ODD(LC) THEN PUTBYTE(NOP)'. Make provisions for a boolean constant WORD_ORIENTED to be placed in main part of assembler and add code to flag attempts to output data words or code on non aligned addresses.

Report number: 132
Module: linker

Description:
The .PRIVATE directive may not work in assembly routines linked to intrinsic units.

Impact: moderate
Versions affected: II.1

Problem:
If no data segment is declared in an intrinsic unit, the linker will perform a NIL pointer reference while unsuccessfully attempting to link the arguments of .PRIVATE. If the linker doesn't crash, the executed program will.

Temporary solution:
Declare a data segment.

Fix:
Linker should flag this bug.

Report number: 133
Module: linker

Description:
Linker bug when linking 2 or more assembly procs or separate units.

Impact: moderate
Versions affected: II.0, II.1, III.0

Problem:
The linker may not resolve all references when linking files that include 2 or more procedures within a separate unit or 2 or more assembly procedures residing in a single code (or lib) file. The resulting mislinked code will probably crash when executed. This error may be detected during the linking process by observing if the order of the procedure names listed with the message 'Copying proc <identifier>' differs from the order of their code within the lib file. If the order is changed by the linker, mislinking has occurred.

Temporary solution:
The guaranteed detour around this bug is to have each separate unit or assembly code file contain only one procedure. If this is inconvenient, it is highly probable that the bug won't occur if the order that the lib files are given to the linker is changed. The success of the new lib file ordering is determined by the criterion described above.

Fix:
Modifications to procedure PROCINSERT.

Report number: 134
Module: assembler

Description:
Externally defined and referenced constants don't work.

Impact: moderate
Versions affected: II.0, II.1, III.0

Problem:
The assembler and linker currently have the ability to resolve external addresses between assembly procedures. It is not possible to define constants to be externally referenced; however, the assembler does not flag attempts to do so. Therefore, references to the following code (which is not flagged by the assembler):

```
          .DEF   FOON
FOON      .EQU   2
```

but with the address of the second byte in the procedure that contains FOON.

Temporary solution:
Make only labels available for external referencing.

Fix:
In procedure ZEQU, flag attempts to equate a constant to an externally defined label.

Report number: 135
Module: transcendentals unit

Description:
Arctan function in Pascal transcendentals unit doesn't work properly.

Impact: moderate
Versions affected: II.0, II.1, III.0

Problem:
Arctangent function returns correct value for arguments between -1 and high positive numbers. For arguments less than -1, it adds pi to the result.

Temporary solution:
Write envelope arctan function that calls regular arctan and adjusts result as follows:

```
if x < -1 then arctangent := arctan(x) - pi
else arctangent := arctan(x)
```

Fix:
In procedure ATAN, calculate arctan using the absolute value of the argument. Negate the result if the argument is negative.

Report number: 136
Module: editor

Description:
Text files with nonstandard format may crash the editor.

Impact: moderate
Versions affected: II.0, II.1, III.0

Problem:
The editor makes some assumptions about text file format during editing. If the text file violates this format, the editor may crash or go into an infinite loop. One example of nonstandardness: a DLE character at the end of a source line is alleged to hang the editor.

Temporary solution:
Use the PATCH program and try to clean up any nonconforming characters or character sequences.

Fix:
Validate file format in procedures INITIALIZE and COPYFILE.

Report number: 137
Module: editor

Description:
Cursor in wrong position after I(nserting into an empty buffer.

Impact: mild
Versions affected: II.0, II.1, III.0

Problem:
If autoindent and filling are on, inserting into an empty buffer (only at initialize time) will leave the cursor 2 characters to the right. This only happens on one line insertions.

Temporary solution:
Move cursor to desired position.

Fix:
Initialize buffer with a DLE/32.

Report number: 138
Module: editor

Description:
Screen may be incorrect after deleting off top or bottom of screen.

Impact: mild
Versions affected: II.0, II.1, III.0

Problem:
D(elete cannot always handle screen properly when deleting.

Temporary solution:
none

Fix:
More robust screen handling in procedure DELETE.

Report number: 139
Module: editor

Description:
C(opy F(ile may put illegal characters in text.

Impact: moderate
Versions affected: II.0, II.1, III.0

Problem:
Copy file sometimes allows two blank compression sequences in a row. The compiler will flag this this sequence with an error.

Temporary solution:
A(djusting the affected lines will remove the DLE sequence.

Fix:
Check for illegal DLE sequence in first and last line of buffer in procedure COPYFILE.

Report number: 140
Module: editor

Description:
Editor markers get accidentally moved when inserting.

Impact: mild
Versions affected: II.0, II.1, III.0

Problem:
I(nserting characters with the last character being an
<eol> character will move the markers 2 characters.

Temporary solution:
Reset markers.

Fix:
In procedure INSERT, add adjust bias.

Report number: 141
Module: editor

Description:
C(opy B(uffer produces unexpected results.

Impact: mild
Versions affected: II.0, II.1, III.0

Problem:
Bug happens after an I(nsert. If the source and destination
overlap, the wrong characters are copied into the editor buffer.

Temporary solution:
Delete erroneous text, and then delete (followed with an <esc>)
the text to be copied. Copy buffer will then work.

Fix:
In procedure COPYBUFFER, replace 'moveleft' with 'moveright'.

Report number: 142
Module: editor

Description:
C(opy B(uffer produces unexpected results.

Impact: mild
Versions affected: II.0, II.1, III.0

Problem:
C(opy B(uffer on single line copies moves text to wrong line.

Temporary solution:
Delete erroneous text, and then delete (followed with an <esc>)
the text to be copied. Copy buffer will then work.

Fix:
Modifications to procedures DELETE, INSERT, and COPYBUFFER.

Report number: 143
Module: assembler

Description:
Using .ORG directive with argument >= 8000H may cause error message.

Impact: moderate
Versions affected: II.0, II.1, III.0

Problem:
absolute code file to initialize the location
counter; in this context, the error message does
not appear. Any use in a relocatable program
will probably mislead the user, for .ORG does
not produce linkable overlays; it emits zero bytes
until the LC is set to value of the argument and/or
causes errors. The problem is caused by insufficient
documentation.

Temporary solution:
Avoid using .ORG whenever possible.

Fix:
In procedure ZORG, flag all uses except at the
start of an absolute code file or within an
absolute section.

Report number: 144
Module: filer

Description:
Filer only checks for y(es on yes/no prompts.

Impact: mild
Versions affected: II.0, II.1, III.0

Problem:
Filer should check for (y,n) responses instead of just (y, not y); currently, it assumes any character other than "y" to be a negative response.

Temporary solution:
none

Fix:
undecided

Report number: 145

voided

Report number: 146
Module: librarian

Description:
Librarian crashes when copying large segments.

Impact: unknown
Versions affected: II.0, II.1, III.0

Problem:
Librarian has unnecessary restriction of only being able to copy over segments that will entirely fit in its buffer and crashes if a segment is too large to fit. This problem is particularly noticable in III.0 system.

Temporary solution:
none

Fix:
Librarian should be able to copy segments in pieces if the segment is too large to copy in one chunk.

Report number: 147
Module: compiler

Description:
Intrinsic units that declare data without a data segment will crash.

Impact: moderate
Versions affected: II.1

Problem:
The compiler allows data to be declared in the interface and implementation sections of an intrinsic unit that lacks a declared data segment number. The unit will then crash when executed.

Temporary solution:
Use data declarations only when a data segment is declared.

Fix:
Compiler should flag data declarations when a data segment is not declared.

Report number: 148
Module: compiler

Description:
Any FOR loops with an upper bound of 32767 will go into an infinite loop.

Impact: moderate
Versions affected: II.0, II.1, III.0

Problem:
FOR loop implementation causes termination on index value being one larger than upper bound. Unfortunatrly, one larger than 32767 is negative, so the termination test fails and the loop is infinite.

Temporary solution:
Avoid using 32767 for upper bound of FOR loop.

Fix: modifications to the code generated for loops.

Report number: 149
Module: interpreter

Description:
Assigning integer expression 32767 + 1 to a REAL variable on a 6502 will cause an infinite loop.

Impact: moderate
Versions affected: II.1

Problem:
Routine FPFLOAT cannot handle this value of integer.

Temporary solution:
none.

Fix:
Rewrite FPFLOAT.

Report number: 150
Module: editor

Description:
Typing moving commands (i.e. vector arrows) in DELETE mode does not cause any change to the textfile.

Impact: moderate
Versions affected: II.0. II.1, III.0

Problem:
The editor does not use function maptocommand in procedure deleting. As a result, if any moving commands are set to prefixed characters in SETUP, the editor will not recognize the commands to affect a delete. The text will remain unchanged.

Temporary solution:
Execute setup to redefine moving commands to be control characters rather than prefixed characters.

Fix:
Procedure deleting in the editor should use function maptocommand to obtain commands.

Report number: 151
Module: system

Description:
Attempting to execute a the assembler or compiler as a codefile (i.e. asm.8800.code) rather than a system file from the promptline results in the error message: Unexpected end of input.

Impact: mild
Versions affected: II.0. II.1, III.0

Problem:
The operating system opens the Input and Output files for system programs. If the programs are executed as though they were user programs, their I/O files are not open causing the error message.

Temporary solution:
Change the name of the file to a system file (i.e. system.assmbler) and invoke from the promptline.

Fix:
Either change documentation to reflect that these files must only be invoked as system files or rewrite the files so that they open their own input and output files rather than having the operating system do it.

Report Number: 152
Module: assembler

Description:
Assemblers with decimal default radix will accept single character hex constants (0A..0F) lacking a hexswitch character (trailing 'H') without flagging an error - though the value is correct.

Impact: mild
Versions affected: II.0, II.1, III.0

Problem:
Procedure PCONST allows this by not performing tight enough error checking.

Temporary Solution:
Always use the Hexswitch character 'H' on hexadecimal constants in decimal default radix assemblers.

Fix:
Tighter error checking in procedure PCONST.

Report number: 153
Module: assembler

Description:
The 11 assembler appears to generate bad code on unresolved conditional branches. Assembled listings prints high 3 octal digits (opcode) and 3* for offset. Patch message prints 3 low octal digits because the 3rd digit is split on a byte boundary. Bit 8 of the word (opcode field) is always printed as 0. The code itself is OK, just the listing is wrong.

Impact: mild
Versions Affected: II.0, II.1, III.0

Problem:

Temporary solution:
None.

Fix:
Unresolved. Octal word byte offset should print 4 octal digits and 2 **'s. Patch message 3 digits defined to overlay with op digit.

Report number: 154
Module: assembler

Description:
The 11 assembler doesn't accept the MARK instruction. It prints the error message 'unimplemented instruction'.

Impact: moderate
Versions affected: II.0, II.1, III.0

Problem:
The MARK instruction was never implemented.

Temporary solution:
Write a macro to perform a MARK instruction.

Fix:
Rewrite ZOP6 to accept MARK.

Report Number: 155
Module: assembler

Description:
A blank line in an absolute section (.ASECT) is flagged with an error message.

Imapact: mild
Versions Affected: II.0, II.1, III.0

Problem:
The assembler is too restrictive about what can appear in an absolute section.

Temporary Solution:
To put comments in absolute sections, use ORG in this manner:

```
BIMBO   .WORD
        .ORG BIMBO+2; <comment>
        .
        .
```

Fix:
In procedure ASSEMBLE, add ENDLINE to set where checking for 'valid structure' in Asect.

Report Number: 156
Module: assembler

Description:
In macros, passed a .ASCII 'ABC ' will display 'ABC'.

Impact: moderate
Versions Affected: II.0, II.1, III.0

Problem:
Leading and trailing blanks are stripped from ASCII strings passed as macro parameters.

Temporary Solution:
None.

Fix:
Rewrite procedure PSTRING to prevent backscanning of blanks in ASCII strings.

Report Number: 157
Module: assembler

Description:
Assembled listfiles have a null character [chr(0)] at the end of each line in the list file. This may affect user programs which attempt to read a list file as input.

Impact: mild
Versions Affected: II.0, II.1, III.0

Problem:
The assembler writes the string with an incorrect length byte and picks up an extra byte (value 0) at the end of the textline.

Temporary Solution:
None

Fix:
In procedure PRINTLINE set the string length byte one smaller.

Report Number: 158
Module: documentation

Description:
Two syntax diagrams are missing from the back of the manual:
Simple expression, variable.

Temporary Solution:
none.

Fix:
Generate these syntax diagrams for the next version of the manual.
Note: The 'Third printing with revisions' edition has the syntax diagram for simple expression.

Report Number: 159
Module: compiler

Description:
When an error occurs in a program involving the token '..'(i.e. in array bounds), the incorrect syntax error: " ':' expected " is generated.

Impact: mild
Versions Affected: II.0, II.1, III.0

Problem:
The compiler treats .. and : as the same token for the purposes of syntax errors.

Temporary Solution:
Use the '..' token correctly in user written programs.

Fix:
Add a new syntax error 'missing ..'.

Report Number: 160
Module: assembler

Description:
The 6809 assembler does not handle PC relative addressing properly.

Impact: moderate
Versions Affected: II.0, II.1, III.0

Problem:
The 6809 assembler puts the absolute address of the operand instead of the relative offset, for PC relative addressing.

Temporary Solution:
none.

Fix:
Set PCR boolean after the second call to EXPRESS in procedure Parse_indirect expression and rewrite code that determines whether target is addressable with a byte offset.

Report Number: 161
Module: assembler

Desription:
The assembler or system might crash when assembling more than eleven routines at once.

Impact: moderate
Versions affected: II.0, II.1, III.0

Problem:

The assembler assumes a maximum of eleven routines (.PROCS or .FUNCS) in one source file. It has an array[0..10] of procaddresses in the global variable declarations. No checking is done on the number of procedures which implies that there is no bound checking on the array. This results in memory being altered in a random way.

Temporary Solution:
Use only up to 11 routines in one assembly.

Fix:
In procedure PROCEND, if PROCNUM exceeds MAXPROC, flag with an error message.

Report Number: 162
Module: yaloe

Description:
After the '?' command is given, the final command prompt comes up after the last line.

Impact: mild
Versions affected: II.0, II.1, III.0

Problem:
No carriage return is performed after prompting a '?'.

Temporary solution:
none

Fix:
In procedure PROMPTS the last line should be a writeln instead of a write.

Report Number: 163
Module: Compiler

Description:
The compiler uses the page intrinsic to implement the $P directive.

Impact: mild
Versions affected: II.0, II.1, III.0

Problem: See report number 128 for the problem with intrinsic page.

Temporary Solution:
none

Fix:
none

Report Number: 164
Module: Transcendental Unit

Description:
On versions that use the transendental unit (9900, 6502, 6800) LN(0) gives a user programmed break rather than a floating point math error.

Impact: mild
Versions Affected: II.0, II.1

Problem:
The error handling for the transendentals unit is to cause a Pascal programme HALT.

Temporary Solution:
Recognize that if the system reports that a break has occurred, the real problem may be that a floating point math error has occurred.

Fix:
The transendental unit will report the error as a floating point math error.

Report Number: 165
Module: Operating System

Description:
User generated text files may be treated abnormally.

Impact: moderate
Versions Affected: II.0, II.1, II.0

Problem:
The specification of a textfile is that there must be at least one NULL at the end of each page. The operating system requires this assumption. No system program (i.e. editor) or procedure (i.e. writeln) will create a textfile that does not have at least one NULL at the end of each page. If a user program creates such a file, there will be unexpected handling of the file by the operating system.

Temporary Solution:
Reformat the textfile so that there is at least one NULL at the end of each page.

Fix:
none

Report Number: 166
Module: Pascalio

Description:
FREADREAL parses the real number directly as it is input in such a way that backspacing past the 'E' is impossible.

Impact: moderate
Versions Affected: II.0, II.1, II.0

Problem:
FREADREAL does its reads character by character so bit is not possible to backspace behind the E.

Temorary Solution:
none

Fix:
FREADREAL needs to buffer its input to allow backspacing.

Report Number: 167
voided

Report Number: 168
Module: assembler

Description:
Assembled listings sometimes display a strange file name on the page headings (instead of the name of the input file being assembled). Other times no file name is printed.

Impact: mild
Versions Affected: II.0, II.1, III.0

Problem:
The operating system gets confused about the current workfile and fails to update the workfile name. The assembler then uses the bad name provided by the operating system.

Temporary Solution:
G)et the file to be assembled (in the filer) and then assemble directly.

Fix:
undecided

Report Number: 169
Module: compiler

Description:
In a compiled listing of a program that uses include files, some line numbers may be inconsistant.

Imapact: mild
Versions Affected: II.0, II.1, III.0

Problem:
The line numbers and partial lines of text of the lines which have the include compiler directive are duplicated on the next line.
Temporary Solution:
none

Fix:
Procedures PRINTLINE and CHECKEND need to be interfaced properly.

Report Number: 170
module: interpreter

Description:
There is an inability to connect a DL(V) - 11 interface as a printer device on PDP-11 and LSI-11 processors.

Impact: moderate
Versions Affected II.0, III.0

Problem:
The PDP-11 and LSI-11 interpreters are not properly configured to allow a serial line interface (DL-11) to be connected as 'printer:'. It is assumed that the printer will interupt at vector 200 (octal). When the interface is a DL(V) - 11, and the vector jumpers are set to 200, the transmit interupt will occur at 204. The new PC in that vector is 0, and location 0 contains a trap instruction which causes an 'unimplemented instruction' execution error to be generated.

Temporary Solution:
Use utility Patch to correct block 0 of system.pdp-11. The four bytes starting at byte address 200 octal should be duplicated starting at 204.

Fix:
The printer interupt handler should be reachable through both vectors 200 and 204. The printer driver needs to be modified to achieve this.

Report Number: 171
Module: interpreter

Description:
On PDP-11 & LSI-11, multiplication of long integers may result in an incorrect answer if either operand contains a word whose value is 100000 octal.

Impact: moderate
Versions affected: II.0

Problem:
The soft multiply routine used by the PDP-11 long integer package is incorrect. It is called in the course of long integer multiplications to multiply corresponding words of the multilplier and multiplicand. Since this 16 bit multiply fails when either operand has octal value 100000, the long integer multiply will be incorrect if either operand contains this pattern. For instance, the product 32768*1 is done incorrectly.

Temporary solution:
none

Fix:
After the label SPECL1 in 11.DECOP.A.TEXT add the following code:

```
SPECL1:   CMP R4, #10000          ; if both numbers are 32768
          BNE $2
          MOV #40000,R4           ; then answer is known
          TST (SP)+               ; but sign flag must be discarded
          BR  OUTM                ; and then we're done
$2:       ASR (SP)+               ; this is old code
```

Also remove lines:

```
          TST R4                  ; fix for negative
          BPL OUTM                ; odd numbers
          INC R4
```

Report Number: 172
module: patch

Description:
In D)ump mode the program does not recognize 'M' for octal.

Impact: moderate
Versions affected: II.0, II.1, II.0

Problem:

Temporary solution:
Dump in Hex and convert to Octal manually

Fix:
undecided

Report Number: 173
module: pascalio

Description:
Using backspace when reading a long integer from console will not work as expected.

Impact: moderate
Versions affected: II.0, II.1

Problem:
READEC terminates reading digits from the keyboard after receiving the first <backspace>. Though the console is updated to display the fixed long integer, the number actually read only consists of the digits typed before the first <backspace>.

Temporary solution:
If one makes an error in entering a long integer, escape from the read and try again. Do not backspace!

Fix:
Rewrite READEC in PASCALIO to handle backspaces properly.

Report Number: 174
module: interpreter

Description:
Runtime errors dealing with negative string indexing on the 9900 processor may not be reported.

Impact: moderate
Versions affected: II.0, II.1

Problem:
The IXS operator does not give runtime errors if the index into the string is negative. For example WRITELN(S[-1]) does not generate an error at runtime.

Temporary solution:
None

Fix:
Tighten range checking on the IXS operator

Report Number: 175
module: editor

Description:
Cosmetic errors occur when the M)argin command is used on paragraphs containing the characters '?', '"', ')', ']', '.', or '-'.

Impact: mild
Versions affected: II.0, II.1, III.0

Problem:
M)argining in the editor will result in only one space after each of the abo characters (except '.') instead of two when they complete a sentence . This condition is also true for a period but only if it is the last characte on a line. Also, if a hyphen is M)argined so that it no longer is the last character on a line, a space is added after the hyphen.

Temporary solution:
F)ind the occurences of these characters in M)argined textfiles and manually change the text.

Fix:
undecided

Report Number: 176
module: system.syntax

Description:
Not all error messages generated by the compiler are contained in this file.

Impact: mild
Versions Affected: II.0, II.1, III.0

Problem:
The errors left out are: 50 and 178.

Temporary solution:
Look up these messages in Jensen and Wirth when they occur.

Fix:
Add these messages to system.syntax. The text of the messages is in Jensen and Wirth. The editor must be updated to know which new messages have been added.

Report Number: 177
module: assembler

Description:
The current release assemblers do not match the specifications documented in the 'Third printing with revisions' edition of the User Manual.

Impact: moderate
Versions Affected: II.0

Problem:
Various assemblers do not support all constant radices described in the new documentation. A table of the current radix switch characters follows. Note: the binary switch character is undocumented, and is available as a
Temporary solution:
This table is available to show the true functioning characteristics of the various assemblers:

Version	11	Z80	6500	6800	6809	8080	9900
Radix							
Binary	B	B	T	B	T	B	B
Decimal	.	.	.	none	.	.	none
Octal	none	none	O	none	Q	none	O
Hex	H	H	H	H	H	H	H

Note: For Octal O is the letter 'O' as opposed to zero '0'.

Fix:
The assemblers will be upgraded to match the specifications in the new documentation.

Report Number: 178
module: filer

Description:
On hardcopy terminals the next prompt comes out on the same line as the previous prompt.

Impact: mild
Versions Affected: II.0, II.1, III.0

Problem:
There are no writeln's performed after procedure PROMPT is called as the filer does a home to position the cursor for the next prompt.

Temporary solution:
Hard copy terminal users should execute Setup and define HOME to be carriage return.

Fix:
undecided

Report Number: 179
module: filer

Description:
Cosmetic errors occurr in the user interface of the X)amine command.

Impact: mild
Versions Affected: II.0, II.1, III.0

Problem:
1) When responding to 'Block Range?', any input '###:' will terminate the command and cause the ':' to become part of the next input. 2) If a good block number is input, a carriage return is not performed after the prompt 'fix them?'. 3) After the X)amine command terminates, the next character inputed is disregarded.

Temporary solution:
Be aware of the user interface the filer is performing to avoid confusion.

Fix:
undecided

Report Number: 180
Module: compiler

Description:
References to a reference ('var') parameter of type char are incorrectly handled on most-significant byte first machines when the corresponding actual parameter is the window variable of a file of char.

Impact: moderate
Versions Affected: II.0

Problem:
The fix reported in report number 46 (and installed in version II.1) is an incomplete solution to the problem. Window variables passed as 'var' parameters are not properly handled.

Temporary solution:
Move the window variable to an ordinary char variable before passing it as a parameter.

Fix:
Undecided

Software Discrepancy Report

Reported By _____

Corporation _____

Phone number (____) _____-_____ x _____

Document serial # _____ Today's Date ____/___/___

Version ____ ____ ____ ____(e.g. II.0A.3) Version date ___/___/___

__ Software error S#____P#____I#____

__ Documentation error Page # _____

__ Check here if reply not necessary.

Error description:

Notice: Your document serial number is required in order for us to get a reply to you concerning the reported error. Please be as explicit as possible in describing the error, and if applicable, please attach a listing. Thank you.

ROUTING (for SofTech Microsystems use):

1 to _____for_____Initial_____

2 to _____for_____Initial_____

3 to _____for_____Initial_____

NEW DEVELOPMENTS POLL

UCSD Pascal is the most complete integrated development system software available. We think it's also the best one, but we know that it's not perfect. We want to encourage the user community to express its views on what is needed and how important various enhancements will be to them.

In order to foster communication on these subjects we will frequently publish in this Newsletter those suggestions which appear to be the most popular, and we hope you will take the time to fill out the "ballot" and send us your views. We look on this ballot as an important part of our effort to provide you with the system features which you need.

SUGGESTION	Don't Do It 0	1	2	3	Important 4
1) Standardize real number representations so that data files and programs containing real number constants are machine independent.					
2) Extend the precision of real numbers by adopting the IEEE draft standard (64 bit).					
3) Extend the precision of real numbers by providing packed BCD floating decimal or other radix 10 arithmetic support.					
4) Decouple the system provided user interface so that a special (e.g. menu driven) interface could be built.					
5) Provide a "SINK" I/O UNIT which accepts output directed to it, discards the data and provides a good IORESULT.					
6) Provide a mechanism which allows a user to add foreign devices conveniently and completely (i.e. devices which support interactive files, direct access files, etc.).					
7) Provide a generalized "redirection" facility which permits input which would normally be obtained from the console to come from a designated file.					

SUGGESTION_____

_____

_____

_____

NAME_____

ADDRESS_____ PHONE NUMBER_____

SOFTECH MICROSYSTEMS
A SUBSIDIARY OF SOFTECH

9494 Black Mountain Road, San Diego, CA 92126 USA, TEL: 714/578-6105, TWX: 910-335-1594