

Release Notice
CONVEX ALL V2.0.1
Document No. 720-004930-003

December 1992

CONVEX Computer Corporation
Richardson, Texas USA

Release Notice, CONVEX ALL V2.0.1

© 1992 CONVEX Computer Corporation
All rights reserved.

This document is copyrighted. This document, however, may be copied, duplicated, reproduced, translated, stored electronically, or reduced to machine-readable form without prior written consent from CONVEX Computer Corporation.

Although the material contained herein has been carefully reviewed, CONVEX Computer Corporation (CONVEX) does not warrant it to be free of errors or omissions. CONVEX reserves the right to make corrections, updates, revisions or changes to the information contained herein. CONVEX does not warrant the material described herein to be free of patent infringement.

UNLESS PROVIDED OTHERWISE IN WRITING WITH CONVEX COMPUTER CORPORATION (CONVEX), THE PROGRAM DESCRIBED HEREIN IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. SOME STATES DO NOT ALLOW THE EXCLUSION OF IMPLIED WARRANTIES. THE ABOVE EXCLUSION MAY NOT BE APPLICABLE TO ALL PURCHASERS BECAUSE WARRANTY RIGHTS CAN VARY FROM STATE TO STATE. IN NO EVENT WILL CONVEX BE LIABLE TO ANYONE FOR SPECIAL, COLLATERAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES, INCLUDING ANY LOST PROFITS OR LOST SAVINGS, ARISING OUT OF THE USE OR INABILITY TO USE THIS PROGRAM. CONVEX WILL NOT BE LIABLE EVEN IF IT HAS BEEN NOTIFIED OF THE POSSIBILITY OF SUCH DAMAGE BY THE PURCHASER OR ANY THIRD PARTY.

C200 Series architecture is a registered trademark of CONVEX Computer Corporation.
CONVEX and the CONVEX logo ("C") are registered trademarks of CONVEX Computer Corporation.
COVUE is a trademark of CONVEX Computer Corporation.
Apollo is a trademark of Apollo Computer, Inc.
Cray is a trademark of Cray Research, Inc.
IBM is a trademark of International Business Machines Corporation.
IEEE is a trademark of Institute of Electrical and Electronics Engineers, Inc.
Sun is a trademark of Sun Microsystems, Inc.
UNIX is a registered trademark of UNIX System Laboratories, Inc.
VAX and VAX FORTRAN are trademarks of Digital Equipment Corporation.

Printed in the United States of America

1**Introduction**

This document describes CONVEX ALL V2.0.1. It obsoletes all previous Release and Update Notices. This document is intended to enhance and clarify the existing permanent documentation for this product with information that is up-to-the-minute or was developed too late for inclusion in the permanent documentation. Always refer to this release notice before reporting questions or problems with CONVEX ALL utilities; your questions may be answered here.

The remaining sections in this document describe the contents of this release:

- Section 2 describes the contents of this distribution.
- Section 3 contains notes and warnings about the use of the software.
- Section 4 describes how to use this software.
- Section 5 describes software changes and enhancements.
- Section 6 lists fixes for previously reported problems.
- Section 7 lists known problems.

CONVEX ALL is a set of language utilities and the FORTRAN and C runtime libraries. ALL consists of the CONVEX assembler (`as`), CONVEX loader (`ld`) and the CONVEX FORTRAN and CONVEX C run-time libraries (`libI77.a`, `libU77.a`, ...).

2

Contents of this distribution

The distribution package for this release of CONVEX ALL consists of software and documentation. The contents of the software and documentation are described in the following tables:

ALL software media

Item	Qty	Part Number	Description	Format
1	1	720-003515-003	CONVEX ALL V2.0.1	GIP

CONVEX ALL documentation

Item	Qty	Type	Part Number	Description
1	1	Doc.	720-005030-003	CONVEX ALL Installation Procedure, V2.0.1
2	1	Doc.	720-004930-003	CONVEX ALL Release Notice, V2.0.1

3**Notes and warnings**

This section contains general information and words of caution about the product.

- This release requires the installation of CONVEX Utilities V9.1 or higher, and the ConvexOS Operating System V9.1 or higher.
- The FORTRAN libraries must be installed before the FORTRAN compiler. The FORTRAN libraries are only compatible with CONVEX FORTRAN V8.0 and higher. If you are not going to install CONVEX FORTRAN V8.0 or higher, (or if it not already installed), do not install the FORTRAN libraries from this tape, because these libraries are not compatible with older versions of the compiler and the older compilers may not be compatible with these libraries.
- The C runtime libraries must be installed before the C compiler. The C runtime libraries are only compatible with CONVEX C V5.0 and higher. If you are not going to install CONVEX C V5.0 or higher compiler (or if it not already installed), do not install the C runtime libraries from this tape, because these libraries are not compatible with older versions of the compiler and the older compilers may not be compatible with these libraries.
- If you are doing an initial install of the operating system, you must install ALL before you do a sysgen or before you install any of the compilers.
- You will need approximately 26 megabytes of disk space in both /tmp and /usr/lib to install CONVEX ALL V2.0.1.

4**Using ALL V2.0.1**

The assembler (as) and the loader (ld) are installed in /usr/bin with symbolic links to them from /bin. All of the CONVEX FORTRAN libraries and the CONVEX C bit and math libraries are installed in /usr/lib. All of the lint libraries are installed in /usr/lib/lint.

5

Software changes and enhancements

Assembler

There has been a minor syntax change in the assembler. The change involves using load immediate instructions. You used to be able to use integer constants with floating point load immediate. Example:

```
ld.du      #0x8000000000000000,s0
ld.dl      #0x8000000000000000,s0
```

Now the assmbler rejects this code with an invalid operand error message. The intent of the assembler was always to reject this code, but due to a bug, it was previously accepted.

Libraries

The pattern matching routines in the FORTRAN and C math libraries have been fixed to generate correct results. These changes are required by both CONVEX FORTRAN V8.0 and CONVEX C V5.0.

The various routines in the CXpa instrumented libraries have been fixed to not abort when you execute these routines in CXpa.

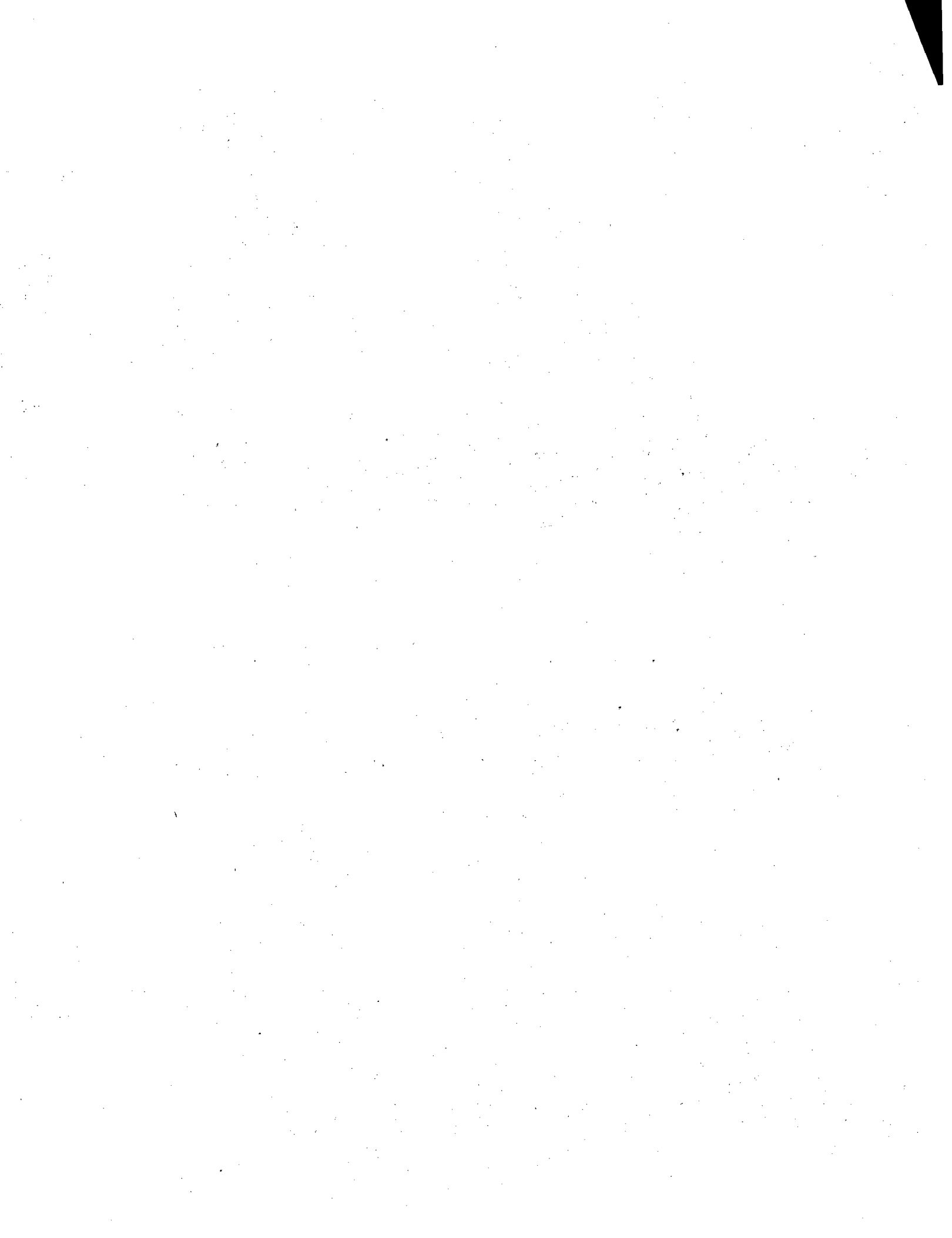
FORTRAN I/O Changes

The FORTRAN list directed I/O has been changed to print out a leading blank, as required by the ANSI FORTRAN 77 standard.

The FORTRAN I/O library (`libI77.a`) has been fixed to not hang when a `ERR=` or `END=` branch has been taken, while performing I/O in parallel code.

FORTRAN 90 Library

The `cshift` and `eoshift` intrinsics have been added to the Fortran 90 support library. Refer to the FORTRAN documentation for more details about these intrinsics.



6

Corrected problems

Assembler

X-25317
PR-26803

Assembly of an invalid `ld.l` instruction produces no warning or error. Assembling this instruction:

```
ld.l    #0x100000000,s0
```

produces exactly the same code as

```
ld.l    #0,s0
```

There is no warning or error to indicate that the immediate operand was truncated.

Resolution:

The assembler (`as`) will now generate a warning message when the constant operand of `ld.l` is greater than or equal to `0x100000000`.

X-26520
PR-28419

The assembler aborts when there is only a `.cxdbsmp` directive in an assembly source file.

Resolution:

The assembler has been fixed to not abort when it only sees the `.cxdbsmp` directive.

X-26531
PR-28426

The `as` utility does not complain when it is passed a command line option that it does not support. Therefore, it gives the user the impression that it accepted it and that it was a valid option when it was not.

Resolution:

The assembler has been fixed to ignore certain invalid options and it will always print a warning message on options that it ignores.

Loader

X-25486
PR-27104

The loader will sometimes treat `.tcbs` (uninitialized thread common) data as `.bss` instead of `.tbss`.

Resolution:

The loader has been fixed to treat all `.tcbss` data as `.tbss` data when creating thread common blocks.

X-25540
PR-27136

The warning message associated with the `-W` option of `ld` should be changed from

ld: warning - Multiply defined symbols founded.

to

ld: warning - Multiply defined symbols found.

Resolution:

The grammar error has been fixed.

FORTRAN Libraries

X-22647
PR-23472

If any of the Cray routines `GETPOS`, `SETPOS`, `UNIT`, or `LENGTH` are called before an actual data transfer has taken place via a `BUFFERIN/OUT` statement, the program fails with a `SIGIOT` error and dumps core. In Cray FORTRAN, this type of coding is common practice and thus does not produce a fatal error.

Resolution:

`GETPOS`, `SETPOS`, `LENGTH` and `UNIT` will no longer cause a program abort error when used before `BUFFERIN/OUT` statements. They will now return a `-1` of the appropriate type when used before any `BUFFERIN/OUT` statements.

X-24048
PR-24935

If FORTRAN I/O is made to a named pipe all works well if the amount of data is greater than the buffer size (4K bytes). If less data are transferred to the named pipe, the FORTRAN program finishes although no data are read from the pipe.

Resolution:

The problem was caused by the FORTRAN I/O system buffering the I/O. The FORTRAN OPEN routines have been fixed to not buffer I/O when the device is a named pipe.

X-24882
PR-26302

The following 3f routines will not work with largefiles: `fgetc`, `fputc`, `fseek`, `getc`, and `putc`. They need to call `fseek64` instead of `fseek`.

Resolution:

All of these routines have been changed to use `fseek64(3)`. Also FORTRAN versions of `fseek64` and `ftell64` have been added to the `libU77` libraries.

X-25573
PR-27200

According to the ANSI standard (1978), paragraph 13.6.2, page 13-16, list directed I/O should output each record such that it begins with a blank character. The attached example shows that this does not happen:

```
write(6,*) 'hello world'  
end
```

Resolution:

The FORTRAN list directed I/O has been fixed to output a blank at the beginning of each record.

X-25820
PR-27450

The `for$flush` function does not work when compiled with any of the following `fc` options: `-cfc`, `-p8` and `-pd8`.

Resolution:

The `for$flush` function has been fixed to work correctly with the above compiler options.

X-25559
PR-27162

Some programs when using the FORTRAN power intrinsic and compiled at `-O2` can die with a floating point overflow. At `-O1` it runs fine. Also if the double precision intrinsic is used instead of the single precision intrinsic, then at `-O2` it runs fine.

```
*** Floating Point Exception: Floating point overflow: at 80033fb6  
signal(8,3,c000ff6c,80001b92,80033fb6) from ffffd0a6 [ap = c000ff58]  
_end+3ff35740() from 80033fb6 [ap = ffffc9e8]  
_MAIN__() from 80001afa [ap = ffffc9e8]  
_main(1,ffffca48,ffffca50) from 80001108 [ap = ffffc9e8]  
IOT trap (core dumped)
```

Resolution:

The power intrinsic that was causing the floating point exception has been fixed to work correctly.

X-26265
PR-28064

The REAL*16 divide intrinsics `meth$q_div`, `meth$q_divx` and `meth$q_divq` cause a user's program to abort, producing a core file from a segmentation violation when running under CXpa.

Resolution:

The CXpa instrumented versions of the REAL*16 divide routines have been fixed to not abort and are now instrumented correctly for CXpa.

X-26266
PR-28065

The REAL*16 multiply intrinsics `meth$q_mul`, `meth$q_mulx` and `meth$q_mulq` cause a user's program to abort, producing a core file from a segmentation violation when running under CXpa.

Resolution:

The CXpa instrumented version of the REAL*16 multiply routines have been fixed to not abort and are now instrumented correctly for CXpa.

X-26665
PR-28603

FORTRAN programs will fail with error number 24, too many files open, when an OPEN statement uses the STATUS keyword OLD.

Resolution:

When status was "OLD", the FORTRAN open routine would open the file twice. The FORTRAN open routine has been fixed to only open the file once, when status is "OLD."

X-26918
PR-28934

FORTRAN list-directed output inserts a newline prior to printing character strings which are greater than or equal to eighty characters.

```
character*81 world
data world/'hello world'/
write(6,*) world
end
```

Resolution:

The FORTRAN list-directed I/O routines have been fixed to not print out the extra newline before the string.

X-26960
PR-28988

If a FORTRAN program does an OPEN on a tty and then tries to read from that tty, nothing is read until a Control-D is performed.

Resolution:

The problem was that the I/O to the tty was being buffered. The FORTRAN open routines have been fixed to not buffer I/O when the device is a tty.

X-26994
PR-28959

A FORTRAN program can hang while doing I/O in parallel code. The problem is caused by taking the END= or ERR= branch in a READ or WRITE statement and then doing any type of FORTRAN I/O operation.

```
c$dir force_parallel
  do i = 1,20
    read(9,err=200,end=300) a
    ...
  end do
  ...
200 write(6,*) 'error in read'
  ...
300 write(6,*) 'unexpected eof in read'
  ...
```

Resolution:

The FORTRAN I/O routines have been fixed to clear the I/O semaphore in parallel code when they have a error or end-of-file condition. Not clearing the I/O semaphore was the cause of the above problem.

X-27139
PR-29312

If you do a ENDFILE on a unit, reopen that unit to the same file or a different file and then write to the file, the write will fail.

```

      program x
      character*40 name
      logical opened
c
      open ( unit=27, status='scratch', form='unformatted' )
      write (unit=27 ) 1.0
      endfile (unit=27)
      close (unit=27, status= 'delete' )
c
c      many statements later.
c
      open ( unit=27, status='scratch', form='unformatted' )
      write (unit=27 ) 1.0
      close (unit=27, status= 'delete' )
c
      end

```

! this will fail

Resolution:

The FORTRAN open/close routines have been fixed to reset an internal flag that indicates the last command was a ENDFILE statement. This fixed the above problem.

X-27194
PR-29314

At the FORTRAN compiler optimization -O1 or higher, the following program will hang trying to dump core.

```

      real a(1)
      do 1 i=1,1000
1      a(i)=1.0
      print 'Vector length : '
      end

```

Resolution:

The above problem has been fixed.

Lint Libraries

X-26078
PR-27766

The man page suggests this for utimes function:

```

utimes(file, tvp)
char *file;
struct timeval tvp[2];

```

but /usr/lib/lint/llib-1c has this:

```

int utimes(const char *f, struct timeval *t[2]) { return 0 ;}

```

In particular, note the second argument.

Resolution:

The lint libraries have been changed to reflect the man page for `utimes(2)`.

MAN Pages

X-25325

PR-26817

The `-p` flag needs to be documented on the `as` man page.

Resolution:

The `as` man page has been updated to document this option.

X-25433

PR-27040

The word `dissect` is misspelled as 'disect' in the third paragraph of the description in the `stime(3f)` man page.

Resolution:

The misspelling of `dissect` has been corrected.

7

Known problems

Assembler

X-25403
PR-26963

An assembly language source causes the assembler to generate bad relocation records, such that `sod(1)` will core dump on the file and `ld(1)` will die with an error message about illegal relocation length.

X-26707
PR-28388

When defining a value as `VAL=label2-label1`, and using `VAL` as an immediate value in an instruction, it can get the wrong answer. If the instructions that fall between `label2` and `label1` can have different opcode sizes depending on whether a long or short immediate is needed, `VAL` can get confused.

```

        .globl _get_table
        .text
_get_table:
        xor    s0,s0
        add.w #SIZE,s0
        rtn
fake1:
        lck   @u64
        rtn
fake2:
SIZE=fake2-fake1

        .data
        .globl _Table_size
_Table_size:
        ds.w  SIZE
u64: ds.w  0x8000

```

X-27092
PR-29252

as will not assemble source files in a directory that is write protected.

FORTTRAN libraries

X-14000
PR-13387

The ANSI FORTRAN standard says that for sequential access: "On output, a new record is created and becomes the last record of the file." [page 12-14, paragraph 12.9.4.1]. However, when writing sequential access tape records, a backspace or rewind that has been preceded with a write, does not write an EOF to

the tape. This can leave the tape without an EOF if the tape is subsequently dismounted as part of a multi-volume job.

X-26929
PR-28833

FORTTRAN (while catching floating point exceptions) goes into an endless loop when it executes a `sqrt(-1)` inside of a subroutine. This has to do with the example in the *CONVEX FORTRAN User's Guide* page 109 figure 16.

Compiler language utility guide

X-23740
PR-24926

The assembler will treat the digits 8 and 9 in octal constants as the values 010 and 011, respectively. This is not documented in the *CONVEX Compiler Utilities User's Guide*.

X-23438
PR-24585

Header files shown in the figures do not appear in the index.