

MCR10123

Fix Issue with `meter_signal` When Invoked with `-nfaults`

2

Revision 1.0

Eric Swenson

September 3, 2022

1 Problem Description

Ticket <https://multics-trac.swenson.org/ticket/123> describes a problem with the `meter_signal` command, where it gets an `illegal_procedure` condition when invoked with the `-nfaults N` control argument, where N is greater than 1.

A trivial fix to the code causes the problem to disappear.

2 Analysis

If you invoke the `meter_signal` command with the `-nfaults 1` control argument, you get output like this:

```
meter_signal -nfaults 1
The following environment will be established:

1 stack frames will be laid down.
0 dummy interrupt handlers will be established in each frame.
1 zerodivide faults will be signalled.
```

Following are the times in microseconds for each fault:

```
419
```

```
Minimum value = 419 Mean = 419
```

However, if you invoke it with a `-nfaults N` control argument, where N is 2 or greater, you get output similar to this:

```
meter_signal -nfaults 2
```

The following environment will be established:

1 stack frames will be laid down.
0 dummy interrupt handlers will be established in each frame.
2 zerodivide faults will be signalled.

Following are the times in microseconds for each fault:

436

Error: illegal_procedure condition by meter_signal\$|1461
(>system_library_tools>bound_system_test_)
referencing stack_4|4010 (in process dir)

The illegal procedure condition is occurring in meter_signal at the DTB instruction shown below:

```
q = divide (1, 0, 17, 0);
```

001455	aa	100	004	227	404	dv3d	(ic),(ic),(pr)	
001456	aa	000322	01	0002	desc9ls	210,2,0		001777 = 053060000000
001457	aa	000323	01	0002	desc9ls	211,2,0		002000 = 053061000000
001460	aa	6	00156	01	0022	desc9ls	pr6 110,18,0	
001461	aa	000	100	305	500	dtb	(pr),(pr)	
001462	aa	6	00156	01	0022	desc9ls	pr6 110,18,0	
001463	aa	6	00123	00	0004	desc9a	pr6 83,4	q

The program intentionally divides by zero to generate the signal to be measured. The program's condition handler collects the timing data and if more faults are to be generated, returns from the condition handler. The execution is resumed at the DTB instruction. DV3D, by specification does not change the value of the result operand for the divide-by-zero case, so pr6|110 contains random stack values, and the DTB instruction faults on an illegal digit value.

This sequence seems predictable, so it is unclear why this issue was never reported.

In discussions with Olin Sibert, he raised the point that a DV3D/DTB sequence was almost certainly not what the program authors intended, and suggested re-coding the divide as:

```
q = divide (1b, 0b, 17, 0); /* divide by zero */
```

This, in turn, corrects the issue:

meter_signal -nfaults 2

The following environment will be established:

1 stack frames will be laid down.
0 dummy interrupt handlers will be established in each frame.
2 zerodivide faults will be signalled.

Following are the times in microseconds for each fault:

424 403

Minimum value = 403 Mean = 413

Examining the history of the source code, we see:

```
/* Originally coded by Paul Karger August 17, 1971 */
/* Updated by Alan Bier - March 1974. */
/* Fixed to eliminate ERROR 295, 05/01/81, W. Olin Sibert */
```

This implies that `meter_signal` was rebuilt in 1981, ten years after the original implementation. I hypothesize that a change in the compiler in that period caused the unwanted DV3D/DTB code generation, and the issue can be blamed on compiler regression.

3 Proposed Fix

The following trivial change fixes the issue:

`cpa meter_signal.pl1.orig meter_signal.pl1.new`

```
A166                    q = divide (1, 0, 17, 0);                    /* divide by zero */
Changed by B to:
B166                    q = divide (1b, 0b, 17, 0);                  /* divide by zero */
```

Comparison finished: 1 difference, 2 lines.

4 Testing

Invoking this little-used program multiple times with multiple control arguments shows that it no longer issues the `illegal_procedure` condition.

5 Documentation

6 Documentation

No changes to the documentation are required.

7 Version History

2022-09-03	1.0	Initial version of the MCR.
------------	-----	-----------------------------