NAME

memtester − stress test to find memory subsystem faults.

SYNOPSIS

memtester [-p PHYSADDR] <*MEMORY*> [*ITERATIONS*]

DESCRIPTION

memtester is an effective userspace tester for stress-testing the memory subsystem. It is very effective at finding intermittent and non-deterministic faults. Note that problems in other hardware areas (overheating CPU, out-of-specification power supply, etc.) can cause intermittent memory faults, so it is still up to you to determine where the fault lies through normal hardware diagnostic procedures; memtester just helps you determine whether a problem exists.

memtester will malloc(3) the amount of memory specified, if possible. If this fails, it will decrease the amount of memory requested until it succeeds. It will then attempt to mlock(3) this memory; if it cannot do so, testing will be slower and much less effective. Run memtester as root so that it can mlock the memory it tests.

Note that the maximum amount of memory that memtester can test will be less than the total amount of memory installed in the system; the operating system, libraries, and other system limits take some of the available memory. memtester is also limited to the amount of memory available to a single process; for example, on 32-bit machines with more than 4GB of memory, memtester is still limited to less than 4GB.

Note that it is up to you to know how much memory you can safely allocate for testing. If you attempt to allocate more memory than is available, memtester should figure that out, reduce the amount slightly, and try again. However, this can lead to memtester successfully allocating and mlocking essentially all free memory on the system -- if other programs are running, this can lead to excessive swapping and slowing the system down to the point that it is difficult to use. If the system allows allocation of more memory than is actually available (overcommit), it may lead to a deadlock, where the system halts. If the system has an out-of-memory process killer (like Linux), memtester or another process may be killed by the OOM killer.

So choose wisely.

OPTIONS

-p PHYSADDR

tells memtester to test a specific region of memory starting at physical address PHYSADDR (given in hex), by mmap(2)ing /dev/mem. This is mostly of use to hardware developers, for testing memory-mapped I/O devices and similar. Note that the memory region will be overwritten during testing, so it is not safe to specify memory which is allocated for the system or for other applications; doing so will cause them to crash. If you absolutely must test a particular region of actual physical memory, arrange to have that memory allocated by your test software, and hold it in this allocated state, then run memtester on it with this option.

MEMORY

the amount of memory to allocate and test, in megabytes by default. You can include a suffix of B, K, M, or G to indicate bytes, kilobytes, megabytes, or gigabytes respectively.

ITERATIONS

(optional) number of loops to iterate through. Default is infinite.

NOTE

memtester must be run with root privileges to mlock(3) its pages. Testing memory without locking the pages in place is mostly pointless and slow.

EXIT CODE

memtester's exit code is 0 when everything works properly. Otherwise, it is the logical OR of the following values:

x01 error allocating or locking memory, or invocation error

x02 error during stuck address test

x04 error during one of the other tests

AUTHOR

Written by Charles Cazabon.

REPORTING BUGS

Report bugs to <charlesc-memtester-bugs@pyropus.ca>.

COPYRIGHT