Machine Check Exception

OPERATING SYSTEM – SEMINAR

April 25, 2015

- NISHANSHI SHUKLA

(121030)

Index

Machine Check Exception:

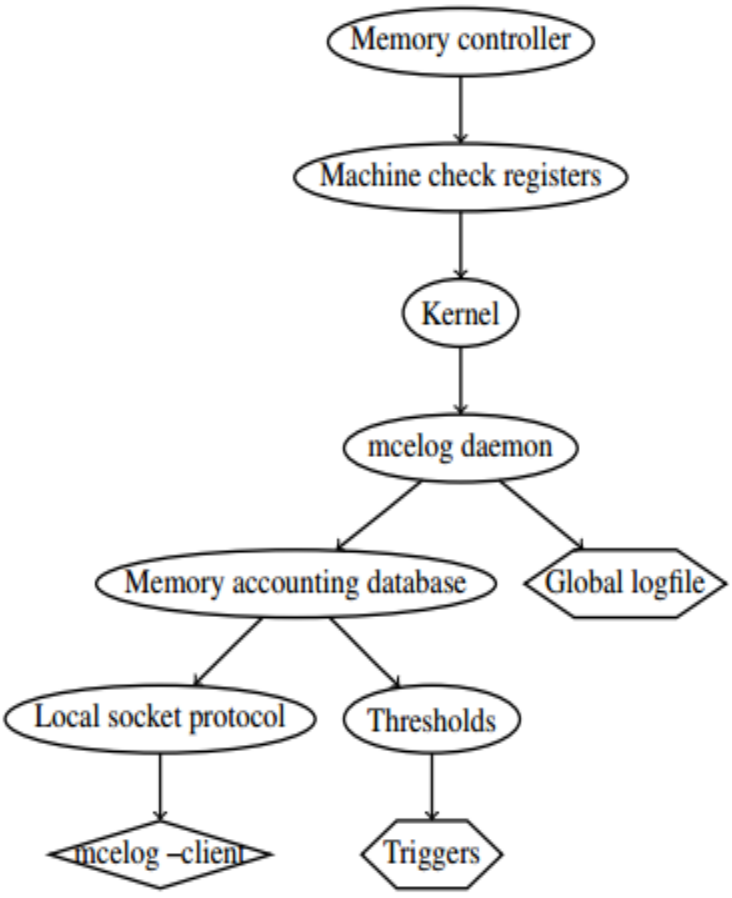
Machine Check Exception or MCE is an exception thrown by the hardware and is dependent on the model of the processor. Its implementation is different for different processors. Machine check exception indicates that the working processor has detected an error in the hardware or in the bus. Most common reasons for machine check exceptions are system bus errors, memory cache errors, inadequate memory errors or inappropriate heating or cooling of the system. Machine check exception is generated when error information is loaded onto machine check registers. There are two types of machine check exceptions, correctable and uncorrectable. In correctable machine check exceptions, they can be recovered from without the hardware being reset. Here action is taken by the processor by reporting the information related to correction machine check error and sending a programmable interrupt to allow the software to respond. While in uncorrectable machine check exceptions there arises a need to reset the system in order to recover. Here the processor concludes that it is no more safe and reliable to operate on the system and thus reset is forced as the cost of recovering from the error might be prohibitive. Machine check exceptions are difficult to debug mostly because of the large number of potential causes like violation to board design guidelines, operating the processor out of specification, incorrect hardware installation or BIOS setup issue.

Machine Check Architecture:

Some processors implement machine check architecture for detecting and reporting hardware i.e. machine errors. These errors include system bus errors, error correction code errors, parity errors, cache errors and TLB errors. A set of model specific registers are used to do machine checking and recording the detected errors. The detection of an uncorrected machine check error is signaled by the processor by generating a machine check exception. Machine check exception may or may not permit the processor to restart reliably after generating an exception. This is an abort class exception. Machine check exception handler can collect information about the machine check error from the machine check model specific registers. Machine check architecture communicates critical hardware errors to software and to possibly recovery from catastrophic system failures. It provides error handling features which contribute to processor reliability, reliable error containment and recording, serviceability, and error correction without program interruption. The Pentium processor reports machine check errors like data parity error during read cycle and unsuccessful completion of a bus cycle. These are reported using machine specific registers in the processor. As the processors are more integrated with additions of memory and fast input output devices, the types of errors that machine control architecture can cover becomes diverse with broader features. These machine check exceptions can be debugged by confirming that platform is operating under given specifications, gathering the error code displayed by the operating system, identifying error frequency, confirm the reproducibility of the machine check exception, decoding and understanding the machine check exception message and ensuring that the updates are installed. A group of error situations visible to the CPU at the point of failure is captured by the machine check architecture.

MCE log tool:

Mcelog is the user space backened for logging machine check errors reported by the hardware to the kernel. Mcelog decodes the errors and manages various other responses like offlining memory, trigerring events or CPUs. It also handles the corrected errors by logging and accounting them. But primarily it handles the machine checks and thermal events which are reported for errors by the processor. Mcelog runs on several modes like cronjob, trigger and daemon. In cronjob, mcelog tool runs every five minutes from cron and checks for errors. This method does not allow mcelog to keep extended state. In trigger, kernel runs the mcelog tool when some error occurs. This method has a relatively high overhead for each error because a program has to be initialized from scratch. In daemon mode mcelog tool runs continuously as a daemon in the background and waits for errors. This is the fastest of all modes and is recommended because several new functions require a continuously running daemon. Basic error flow can be described as follows:



References:

1. Intel® 64 and IA-32 Architectures Software Developer's Manual Volume 3A: System Programming Guide, Part 1, order number: 253668-039US, May 2011.
2. Intel® Embedded Debugging Machine Check Exceptions on Embedded IA Platforms, White Paper: Ashley Montgomery (Platform Application Engineer Intel Corporation), Tian Tian (Platform Application Engineer Intel Corporation), July 2010.
3. http://www.advancedclustering.com/act-kb/what-are-machine-check-exceptions-or-mce/
4. mcelog: memory error handling in user space, Andi Kleen, Intel®Corporation.
5. http://www.mcelog.org/README.html