WIKIPEDIA
The Free Encyclopedia

Shadow memory

In computing, **shadow memory** is a technique used to track and store information on computer memory used by a program during its execution. Shadow memory consists of shadow bytes that map to individual bits or one or more bytes in main memory. These shadow bytes are typically invisible to the original program and are used to record information about the original piece of data.

Technique

The technique is utilized by memory-error checkers that can store information on which parts of memory have been allocated to the program being checked. This shadow memory is then used for detecting and reporting incorrect accesses of memory, even though the program may not be crashing due to a segmentation fault or similar. An error checker may also store additional information on memory such as which bits have been defined and which ones do not. Memcheck, part of the Valgrind suite, uses this to detect undefined behavior resulting from acting on or printing undefined memory values.

Use of shadow memory is however not limited to memory-error checkers, as what information is stored in these shadow bytes is not fixed. It is for instance used by ThreadSanitizer, a data race detector.

Shadow memory can be both implemented and used a lot of different ways, and have different performance characteristics. Memcheck for instance tracks values with bit precision, while AddressSanitizer, part of the clang compiler, is comparatively very fast. Memcheck, like all Valgrind tools, uses binary translation and instrumentation to run code manipulating the shadow memory corresponding to program memory use. AddressSanitizer on the other hand is created on compile-time and inserts error-checking code inline into a program during compilation. Its shadow-memory implementation uses a huge reservation of virtual memory for its shadow memory, giving very different performance characteristics.

References

General

- Nethercote, N.; Seward, J. (2007). "How to shadow every byte of memory used by a program". *How to shadow every byte of memory used by a program. In Proceedings of the 3rd international Conference on Virtual Execution Environments (San Diego, California, USA)*. pp. 65–74. CiteSeerX 10.1.1.643.7117 (https://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.643.7117). doi:10.1145/1254810.1254820 (https://doi.org/10.1145%2F1254810.1254820). ISBN 9781595936301. S2CID 10263496 (https://api.semanticscholar.org/CorpusID:10263496). {{cite book}}: |work= ignored (help)
- http://research.google.com/pubs/pub37752.html