Abstract

Automatic memory management, also known as garbage collection, is a suitable means to increase productivity in the development of computer programs. As it helps to avoid common errors in memory management, such as memory leaks or dangling pointers, it also helps to make programs safer. Conventional garbage collection techniques are however not suited for use in hard real-time systems. As a failure in these systems can have catastrophic consequences—such as the loss of human life—it is of utmost importance to meet the timely requirements of the respective system. In the past few years, methods for garbage collection that are suitable for use in hard real-time systems have been developed. However, these techniques are suited only for uniprocessors. On multi-processors, garbage collection does not yet meet the requirements for hard real-time systems.

Garbage collection reclaims memory objects that are provably inaccessible to the application. Objects that are referenced by global or local variables can be accessed at any time. Starting at these objects, the object graph is traversed by the garbage collector to determine which objects are reachable. As there is the possibility that reachable objects are accessed by the application, they must not be reclaimed. Determining the object references in local variables is challenging in hard real-time systems, because the timely requirements usually do not permit stopping the application to do so. The development of appropriate techniques is one of the core areas of this dissertation.

Memory fragmentation cannot be tolerated in hard real-time systems, because otherwise, it would be impossible to determine reasonable bounds on the maximum memory consumption. In order to defragment memory, it is necessary to relocate objects. To achieve this without interrupting the application, a hardware unit has been developed, which enables transparent object relocation on chip multi-processors.

Multi-processors often use memory models that provide only few guarantees on the ordering of memory accesses. The effects of such memory models on garbage collection are investigated in this thesis. Also, an analysis to determine the maximum size of memory allocations of tasks is presented. Such an analysis is necessary to compute whether the allocation requests of an application can be met even in the worst case.

This thesis points out solutions which enable the reconciliation of garbage collection with the requirements of hard real-time systems on chip multi-processors. The theoretic results are supported by measurement results, which indicate that low release jitter can be achieved in multi-processor systems in the presence of garbage collection.