JOHANNES KEPLER
UNIVERSITÄT LINZ

**o.Univ.-Prof. Dr.
Hanspeter Mössenböck**
Institute for System Software

T +43 732 2468 4340
F +43 732 2468 4345
hanspeter.moessenboeck@jku.at

Secretary:
Karin Gusenbauer
Ext 4342
karin.gusenbauer@jku.at

Master's Thesis

Supporting Multiple Garbage Collectors in a Single GraalVM Native Executable

Student:	Thomas Schrott (12007309)
Advisor:	Prof. Hanspeter Mössenböck
Co-Advisor:	Dr. Christian Häubl
Begin:	1.9.2024

GraalVM [1] Native Image provides ahead-of-time (AOT) compilation of Java code into optimized native executables. A native executable includes code and data for the application, the Java standard-library, a VM runtime, and statically linked native libraries.

When building a native executable, a garbage collector (GC) needs to be selected at image build time. The selected GC will manage the memory of the Java heap at run-time. Native Image supports various GCs, but each native executable can only contain a single GC. By default, a stop-the-world Serial GC is included, which is primarily optimized for low memory footprint and small Java heap sizes.

One goal of this project is to support that multiple GCs can be included in a single native executable and that the used GC can be selected at run-time. As most of the AOT compiled code needs to be shared between all GCs, it is necessary that they have a common understanding about the object layout and the used write barriers. This will require changes in the GCs, the write barriers, the card table, the object layout, and various other Native Image build and run-time data structures.

Another goal of this project is to optimize performance and memory footprint of the Serial GC. For example, thread-local allocation buffers (TLABs) always have a fixed size at the moment, which increases memory footprint unnecessarily. Instead, TLABs should be dynamically resized, based on the allocation rate of its thread. This will especially improve the memory footprint of multi-threaded applications that have a small Java heap (e.g., microservices).

The scope of this thesis is as follows:

- Support including multiple GCs in a single native executable.
- Optimize performance and memory footprint of the Serial GC.
- Test and benchmark the changes.
- Contribute the approach to the open-source repository [2] (requires signing OCA [3]).

The work's progress should be discussed with the supervisor at least every 2 weeks. Please note the guidelines of the Institute for System Software when preparing the written thesis. The deadline for the written thesis is 31.8.2025.

References:
[1] https://www.graalvm.org/
[2] https://github.com/oracle/graal
[3] https://oca.opensource.oracle.com/