

The Jelatine JVM

Summary

The Jelatine JVM is a new Java Virtual Machine targeted at small embedded systems like cell phones, network appliances, embedded controllers, and so on. It implements the **Java 2 Micro Edition Connected Limited Device Configuration** standard in a small, portable and customizable package.

The JVM is engineered to work on systems with as little as 32 KiB of volatile memory and 128 KiB of non-volatile memory, yet it offers reasonable performance levels. The machine offers a compact yet rich basic class library, but it can in addition be customized, by adding pure Java classes or native system-dependent functionalities. Notice that some functionalities may be disabled (for example floating-point support) in order to remove unneeded features and to further reduce the memory footprint.

The machine has been designed and implemented for portability onto different architectures. It is written in 100% pure ANSI C89, it uses only a minimal subset of the ANSI C standard library, and it works both on 16- and 32-bit processors.

The Jelatine JVM should enable a wide range of small embedded platforms to run Java code, providing a common, portable platform for software development. The objective is to simplify development and deployment of software on heterogeneous architectures, and to shorten development cycles thanks to convenience of the Java language. Also it enables developers to reuse popular tools (NetBeans, Eclipse) and existing libraries for developing new applications targeted at the embedded market.

The Java 2 Micro Edition standard

The Java 2 Micro Edition standard was designed to bring the ease of use and platform independence of the popular Java language to a wide range of embedded devices. For this reason it was split in a body of standards covering the various aspect of a complete Java environment. The main parts of the standard are distributed among a configuration which describes the virtual machine and the basic device-independent class library, a profile which specifies a number of additional classes specific to a certain device and the optional frameworks.

The Jelatine JVM implements the Connected Limited Device Configuration version 1.1 of the J2ME standard offering a complete virtual machine coupled with a full implementation of the default class library. Even though Jelatine doesn't include any ready-made application-oriented profile, it offers all the necessary handles to add one easily. Such profile classes can be efficiently implemented using the provided light-weight native-call mechanism, in order to offer more device- or system-dependent operations.

Features of the Jelatine JVM

Limited memory footprint

The Jelatine JVM was designed to work on very small, memory-constrained embedded systems. It requires a minimum of 32 KiB of volatile memory and 128 KiB of non-volatile memory to run and makes very efficient use of it. The machine was designed to fit in the constraints specified by the CLDC standard (32 KiB of volatile memory and 160 KiB of non-volatile memory).

The machine is robust by design in extreme low-memory conditions, thus enabling it

to safely run very close to the heap limit and to keep offering predictable behaviour even under potentially malicious pathological workloads.

Portability

In order to make the source-code easy to port, the code strictly complies with ANSI C89 standard. Optional compiler-specific features are used to improve performance, but can be disabled without hampering operations. The machine and the underlying standard class library make minimal use of system calls, basically using only very few functions from the ANSI C89 standard library. This reduces the effort for porting on a different platform. The machine natively support 16- and 32-bit processors, yet its design makes it work also on 64-bit machines, though they are not its main target.

Performance

Jelatine offers very good performance for the intended small embedded device target. For the desk-top machines, though it does not match the JIT based virtual machines, its speed is comparable with the best purely-interpreted, freely available desktop virtual machines. Compared to Sun CLDC reference implementation, Jelatine is 2 to 3 times faster and only marginally slower than the commercially available JIT-enhanced version of the machine.

Modularity

Many components of the virtual machine may be tuned for a specific device or simply disabled if they are not needed. The machine offers a command-line front-end which may be disabled, the machine then works as an external linkable library. In the library form the machine can be made fully re-entrant: several instances can be run together in the same context allowing for multiple Java applications to be launched and used. This provides an environment allowing multi-tasking of Java applications. Floating-point types support and JAR file loading support can also be disabled to reduce footprint even more. The interpreter may also be easily altered and modified as it offers a convenient set of macros for integrating external modifications.

Licensing

The Jelatine JVM is released under the Gnu Public License (GPL) and is thus freely available. Jelatine's source code is available for download from

<http://sourceforge.net/projects/jelatine>

Modifications and improvements to its source-code are encouraged and welcome. The authors may be contacted for related developments or for specific non-free licenses for commercial purposes.

Contacts

Jelatine JVM is one of the projects of :

Formal Languages and Compiler Group

Dipartimento Elettronica Informazione DEI, Politecnico di Milano

<http://compilergroup.elet.polimi.it/research.html>

Giovanni Agosta, giovanni.agosta@polimi.it

Stefano Crespi Reghizzi, stefano.crespireghizzi@polimi.it

Jelatine's designer: Gabriele Svelto, gabriele.svelto@gmail.com