

Google Summer Of Code 2009 Projects Proposals

Please add your proposals to the list

Subject ID	<i>harmony-tools-1</i>
Title	Implement a Java developer's command line tool
ASF Project	Apache Harmony http://harmony.apache.org
Keywords	Java, tools
Description	Harmony is missing several of the tools that ship with the JDK, including jar, jconsole and javaws. For this task you would implement one of these tools, either in Java or C/C++ if preferred.
Possible Mentors	Oliver Deakin (committer), Mark Hindess (committer)
Status	Unassigned

Subject ID	<i>harmony-tools-2</i>
Title	Implement the jdb command line debugger tool
ASF Project	Apache Harmony http://harmony.apache.org
Keywords	Java, tools, jdb
Description	jdb is the command line debugger tool provided as part of the basic JDK tools (http://java.sun.com/javase/6/docs/technotes/tools/index.html). For this task you would implement the jdb debugger, in C/C++ and/or Java
Possible Mentors	Oliver Deakin (committer), Mark Hindess (committer)
Status	Unassigned

S u b j e c t I D	<i>harmony-tools-3</i>
Ti t l e	Extend VMTT bytecode assembler
A S F P r o j e c t	Apache Harmony http://harmony.apache.org
K e y w o r d s	Java, tools, vmtt
D e s c r i p t i o n	VMTT is a bytecode assembler used to test the Apache Harmony VM and class library. For this task, you would extend it in several ways: adding real labels (so one could write jsr loop instead of jsr -5), adding flexibility to emit more bad classes (specifying padding and values explicitly for tableswitch/lookupswitch, allowing double/long constant pool entries without a subsequent constant pool entry), allowing for mixed asm and bin in methods, allowing for insertion of binary data in other places (such as the constant pool), warning users of possible errors (such as bad code_length for a given code array (configurable)), and fixing existing defects. If legal approval is available, you might extend VMTT to read the JASM format as well.

Possible Mentors	Andrew Cornwall
Status	Unassigned

Subject ID	<i>harmony-i18n-tool</i>
Title	Automatic localization tool for open source project
ASF Project	Apache Harmony http://harmony.apache.org
Keywords	open source project, localization, automatic translation
Description	The Harmony Project has no localized data for worldwide customers. This project plans to create an automatic tool that helps projects to translate the necessary messages. The tool will automatically extract the messages, conduct analysis if necessary to translate, translate and adjust, and at last inject into the source code with ResourceBundle or property loading mechanism and apply the localization. There are many open source projects in existence, most of which do not have localized messages for customer convenience, including the Harmony project, so the tool may also be helpful on a wider scale. (http://wiki.apache.org/harmony/Harmony_i18n_tool)
Possible Mentors	Jing Lv (Jimmy) (committer)
Status	Unassigned

Subject ID	<i>harmony-daemon</i>
Title	Running multi-programs on one Harmony VM
ASF Project	Apache Harmony http://harmony.apache.org
Keywords	multi-programs on single VM
Description	In Java, we usually run one program on one VM. However there are many advantages to executing multiple programs on one jvm, e.g. memory usage, startup time, etc. However there are many potential conflicts that could be caused e.g. the vm will meet an exit() or fatal error in one program, which causes the other programs to fail together. This project is concerned with finding a clever solution for running multiple Java programs on a single VM.
Possible Mentors	Xu ruizhi(Regis), Jing Lv (Jimmy) (committer)
Status	Unassigned

Subject ID	<i>harmony-classes-selector</i>
Title	Smallest classes set for customer application*

ASF Project	Apache Harmony http://harmony.apache.org
Keywords	smallest class set for application
Description	Many Java desktop applications offer a JRE in the installation package, which may be big. Customising the JRE to be the smallest size to run an application takes a lot of effort. We are looking for an automatic tool to find out all classes used by an application and build up a smallest JRE for the customer. (http://wiki.apache.org/harmony/Harmony_classes_selector)
Possible Mentors	Xu ruizhi(Regis), Jing Lv (Jimmy) (committer), Li JinQing (Charles)
Status	Unassigned

Subject ID	<i>harmony-osgi</i>
Title	• Enable OSGi features for Harmony JDK*
ASF Project	Apache Harmony http://harmony.apache.org
Keywords	OSGi, JDK
Description	OSGi is hot and Apache Harmony has already make it's modules into OSGi bundles. If the Harmony runtime itself can include an OSGi framework, it will help other projects executing on a Harmony JRE to be OSGi without implementing an OSGi framework themselves. If we can go a little further, it may make the Harmony Project the first OSGi-ed JDK on the world and could enable an update mechanism for Harmony easily. (http://wiki.apache.org/harmony/OSGi_Harmony)
Possible Mentors	Zhou Kai(Kevin), Jing Lv (Jimmy) (committer)
Status	Unassigned

Subject ID	<i>harmony-portlib</i>
Title	Portability in Apache Harmony: portlib, the Apache Portable Runtime or a new alternative
ASF Project	Apache Harmony http://harmony.apache.org
Keywords	Harmony, APR
Description	To enable the Harmony class library to be easily ported to new platforms a portability library exists. By replacing components in the portability library a VM may integrate with the Harmony libraries. Alternatively a VM may use and build on the portability library with Harmony. No current VM builds on top of just the portability library. The IBM VMs use bespoke OS integration, DRLVM uses this and the Apache Portable Runtime (APR), Jikes RVM uses the Harmony portability library and limited bespoke OS integration. The Apache Portable Runtime differs from the Harmony portability library in that virtual memory functions are exposed via mmap/munmap/mprotect rather than reserve/free/commit/decommit. The Harmony portability library used to but no longer provides a portable signal handling mechanism. This project will create a portability strategy for VMs using the Harmony class library either by using the Harmony portability library, the Apache Portable Runtime or potentially with no wrapper layer. One option is to recode the Harmony portability library entirely using the APR, adding support for signal handling whilst also adding support to APR for a different means of managing virtual memory. Another option is to put the portability library functionality into APR. Alternatively no wrappers and adopting UNIX conventions may best support portability. Ideally multiple open source Harmony VMs can be tested with this portability layer such as DRLVM and Jikes RVM. This project will require consultation with Harmony developers to form agreement and consensus on a portability strategy. Documenting this strategy will also form part of the project.

Possible Mentors	Ian Rogers, Alexei Fedotov (committer)
Status	Unassigned

Subject ID	harmony-zlib
Title	Improve Apache Harmony zlib/zip performance*
ASF Project	Apache Harmony http://harmony.apache.org
Keywords	Java, zlib, Zip
Description	There are a number of changes that should improve zlib performance for Harmony such as using the inflateBack api (compare gzip/gun in the zlib distribution), exploiting parallelism (compare gzip/pigz) and/or using mmap rather than file operations. The project would be to implement these techniques to see what works in practice.
Possible Mentors	Mark Hindess (committer)
Status	Unassigned

Subject ID	harmony-JIT-1
Title	Modularize Harmony JIT by separating JET as a standalone JIT compiler*
ASF Project	Apache Harmony http://harmony.apache.org
Keywords	Harmony, JVM, JIT, modularity
Description	So far the JIT component (called Jitrino) of Harmony has virtually two JIT implementations: JET and OPT. Jitrino.JET is a fast but non-optimizing JIT, and Jitrino.OPT is an optimizing JIT. The code base of JET and OPT shares lots of code hence they are mixed in one module. This is undesirable for situations where people need only JET, for fast compilation, for small footprint. This project proposes to create a standalone JET-based JIT module for Harmony. It does not require to remove JET from Jitrino, but to create a new JIT module with JET. This project is also a very good exercise to examine the JIT modularity design, the interface between JIT and other components, the interaction between multiple co-existing JIT modules.
Possible Mentors	Xiao-Feng Li (committer)

St at us	Unassigned
S u b j e c t ID	harmony-GC-1
Ti tle	Implement WeakReference support in Harmony concurrent GC*
A S F P r o j e c t	Apache Harmony http://harmony.apache.org
K e y w o r d s	Harmony, concurrent GC, weak reference
D e s c r i p t i o n	Harmony already has a concurrent GC (called Tick, with three concurrent GC algorithms). It runs well with standard benchmarks. The only remaining unfinished feature is WeakReference support . Weakly referenced object (i.e., referent) is accessed through get() interface. That means, get() operation can make a weakly reachable referent strongly reachable. During concurrent collection, the system must monitor the get() operation to catch this change of reachability, otherwise the referent could be reclaimed. This project also includes to integrate the WeakReference processing with Finalization process. Other optimizations in Tick are also desirable, such as to reduce the amount of floating garbage.
P o s s i b l e M e n t o r s	Xiao-Feng Li (committer)
St at us	Unassigned