



HERCULES:
**High-Performance Real-time Architectures for
 Low-Power Embedded Systems**



Project title:	High-Performance Real-time Architectures for Low-Power Embedded Systems
Acronym:	HERCULES
Project ID:	688860
Call identifier:	H2020 - ICT 04-2015 - Customised and low power computing
Project Coordinator:	Prof. Marko Bertogna, University of Modena and Reggio Emilia



D6.4: Report on Open-source results

Document title:	Open-source results
Version:	1.1
Deliverable No.:	D6.4
Lead task beneficiary	EVI
Partners Involved:	All
Author:	Claudio Scordino (EVI)
Status:	Final
Date:	December 30 th 2018
Nature ¹ :	R

Dissemination level		
PU	Public	X
PP	Restricted to other programme participants (including the Commission Services – CS & IAB)	
RE	Restricted to a group specified by the consortium (including the IAB)	
CO	Confidential to consortium (including CS & IAB)	

¹ For deliverables: **R** = Report; **P** = Prototype; **D** = Demonstrator; **S** = Software/Simulator; **O** = Other
 For milestones: **O** = Operational; **D** = Demonstrator; **S** = Software/Simulator; **ES** = Executive Summary; **P** = Prototype

Document history:

Version	Date	Author	Comments
0.1	2018-11-19	EVI	First draft
0.2	2018-11-30	CTU	Information for the tracker
0.3	2018-12-11	ETHZ	UIUC license for components by ETHZ
1.0	2018-12-12	ETHZ	Information about libpremnotify
1.1	2018-12-30	UNIMORE	Final revision

This document reflects only the author's view and the EU Commission is not responsible for any use that may be made of the information it contains.

Table of contents

Document history:	i
Table of contents	ii
1. EXECUTIVE SUMMARY.....	4
2. INTRODUCTION	5
2.1. Objectives	5
2.2. Licenses used for HERCULES	5
3. Open-source components.....	7
3.1. ERIKA RTOS	7
3.2. SCHED_DEADLINE	8
3.3. Jailhouse for NVIDIA TX1/TX2.....	9
3.4. Jailhouse cache colouring.....	9
3.5. Jailhouse GPREM (Guarded PREM).....	10
3.6. PREM Compiler	10
3.7. PREM library (libpremnofity).....	11
3.8. GPU memguarding (GPUGuard).....	11
3.9. Tracker	12
4. CONCLUSIONS	13
5. REFERENCES.....	15

1. EXECUTIVE SUMMARY

This deliverable aims at describing the final plan for open-source exploitation of the software developed within the HERCULES project. Besides listing the software components that are (or will be) released as open-source software, the document will also describe the specific license used as well as the dissemination activities planned for promotion.

2. INTRODUCTION

2.1. Objectives

Deliverable D6.3 “Open-source Strategy Plan”, released at M12, illustrated a preliminary plan for the exploitation strategy of open-source software. In particular, it listed the possible licenses, and made a first attempt in identifying which HERCULES components could be released as open-source software. At that stage of the project, however, the overall software infrastructure was not yet completely defined. The tight collaboration established by the partners in the subsequent two years has allowed the consortium to design and develop the full HERCULES software stack. The consortium has then been capable of analysing each single component developed in the project, considering the pros and cons of open-source licensing.

2.2. Licenses used for HERCULES

For convenience we here recall the characteristics of the subset of the licenses described in deliverable D6.3 that will be used for releasing the open-source software developed in HERCULES. Please refer to the original deliverable for further information.

GNU General Public License (GPL)

GPL [4] is a license published by the Free Software Foundation [2] allowing to inspect, modify, redistribute (even sell) the source code as long as the recipient maintains the same rights. This is one of the less industry-friendly licenses within the open-source domain. In fact, a proprietary application is forbidden to either directly include GPL source code or be linked against a GPL library. If it does, then the license starts affecting the proprietary code as well, requiring to disclose the source code to any recipient (which is what industrial companies usually do not want to). According to a recent survey [18], GPL is currently the second most used license.

GNU General Public License linking exception (GPL-LE)

GPL with linking exception [7], also known as ClassPath [6] (from the original project that used it), adds an exception to the standard GPL to allow linking of a proprietary object against the open-source library. At the same time, it does not impose the restrictions of the LGPL (i.e., possibility of re-linking). This is therefore the most permissive and industry-friendly license among GPL licenses.

MIT license

This is a family [17] of licenses by Massachusetts Institute of Technology (MIT) aiming at imposing minimal restrictions on the redistribution of the software. The license is very short and simple. With respect to the GPL licenses, this family of licenses also allows to include source code into a proprietary application by just acknowledging the original author of the code. According to a recent survey [18], this is currently the most used license.

BSD license

BSD [8] is a family of licenses made by the Berkeley University. The constraints are almost identical to the ones by the MIT license. The original “4-clause” BSD license contained a controversial “advertising clause” imposing to credit the original author in all advertising material. This license has been dropped in favour of the simpler “3-clause” or “2-clause” versions.

UIUC license

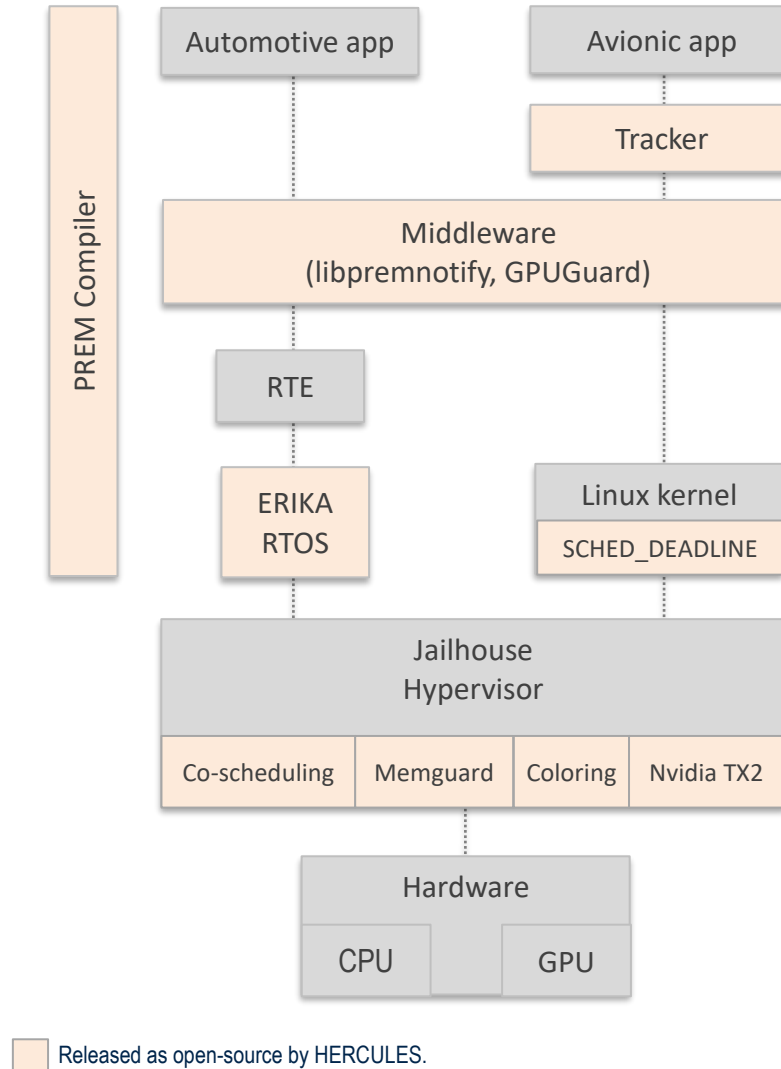
The University of Illinois/NCSA Open Source License (UIUC) [16, 20] is a permissive free software license, based on the MIT and the 3-clause BSD licenses. By combining parts of these two licenses, it attempts to be clearer and more concise than either. Redistributions of either source or binary code must reproduce the original copyright notice.

The following figure summarizes the constraints for the recipients of the software components developed in HERCULES, depending on the specific license:

	Include HERCULES code into proprietary software without releasing as open-source	Link HERCULES code with proprietary software without releasing as open-source
GPL	X	X
GPL Linking exception	X	√
BSD/MIT/UIUC	√	√
Commercial	√	√

3. Open-source components

The different components of the HERCULES software stack are summarized in the following figure:



We now list all the components that have been (or will be) as open-source software, also mentioning the specific license, the method used for delivering, and the activities planned for proper dissemination.

3.1. ERIKA RTOS

Description

ERIKA Enterprise [13] is an RTOS developed by partner **EVI** and specifically designed for the automotive market. It is already used in production by renowned companies (e.g. Magneti Marelli, Vodafone automotive).

Ownership

EVI

Licensing

ERIKA Enterprise v3 is already released as open-source software under dual licensing (i.e. plain GPL license, with linking exception with a fee).

The new features of the RTOS (i.e. porting on the NVIDIA TX1/Tx2 platforms and on the Xilinx ZCU102) have been released following the same dual licensing scheme of the RTOS. The code has therefore been published onto ERIKA's GitHub repository (<https://github.com/evidence/erika3>) while the package containing the RT-Druid Eclipse tool has been released on ERIKA's website [13].

Dissemination activities

- Talk at 2nd Linux-Lab, Florence, December 2018.

3.2. SCHED_DEADLINE

Description

SCHED_DEADLINE [1, 3] is a real-time CPU scheduler for Linux, originally born within the context of the ACTORS project [5], financially supported by the European Commission under the FP7 framework. Such scheduler was created in collaboration with the Real-Time Systems Laboratory of Scuola Sant'Anna (which already had a wide experience in development of real-time schedulers for Linux) and was merged into the official Linux kernel since release 3.14 (i.e. 2014).

As explained in deliverable D4.2, within the HERCULES project, partner **EVI** has collaborated with Scuola Sant'Anna and ARM Ltd. for improving the SCHED_DEADLINE scheduler by adding reclaiming (i.e. GRUB [9, 12]) and, especially, energy-efficiency (i.e. integration with the power management subsystem based on the GRUB-PA [10, 11] algorithm).

Ownership

EVI, Scuola Sant'Anna, ARM

Licensing

The Linux kernel is the biggest collaborative project developed by an open-source community. This piece of software is the basic brick of millions of devices running Linux or Android distributions, and it currently consists of about 20 millions of lines of code under the GPLv2 license. Since the kernel image and the user-space applications are not linked together, the GPL license does not affect the other components of the system.

The IPR ownership belongs to the Linux kernel community, consisting of thousands of developers who have collaborated to the project over the past 25 years. The HERCULES consortium did not have therefore the permission of releasing the mentioned functionalities under a different license; nor it had permission of adopting a dual licensing model. The consortium has therefore released such functionalities under the same open-source software of the rest of the kernel (i.e., GPLv2). Such functionalities have been released publicly (e.g., through the Linux Kernel Mailing List) and were merged into the official Linux kernel with release 4.14 (i.e. reclaiming) and 4.16 (i.e. power management).

Dissemination activities

Before being merged into the mainline kernel, the new features have been sponsored through the following dissemination activities:

- Talk at 1st OSPM Summit, Pisa, April 2017.
- Talk at Linux RT-Summit, Prague, October 2017.
- Demo at ARM's booth at Embedded Linux Conference Europe (ELCE) Prague, October 2017.

- Talk at 1st Linux-Lab, Florence, December 2017.
- C. Scordino, L. Abeni, J. Lelli, *Energy-Aware Real-Time Scheduling in the Linux Kernel*, 33rd ACM/SIGAPP Symposium On Applied Computing (SAC 2018), Pau, France, April 2018.

3.3. Jailhouse for NVIDIA TX1/TX2

Description

Jailhouse [14] is a hypervisor targeting the embedded and safety-critical domains. It is developed by Siemens and released as open-source software under the GPLv2 license through the GitHub platform. Partner EVI has added the support for NVIDIA TX1/TX2 platforms.

Ownership

EVI

Licensing

Being an open-source initiative without the need of any “contribution agreement”, the IPR ownership of the full hypervisor belongs partly to Siemens and partly to the development community. The HERCULES consortium did not have the IPR on the hypervisor. Therefore, any change or improvement (e.g., support for the PREM model [19]) must be necessarily released under the same GPLv2 license, without any possibility of adopting a dual licensing strategy. The consortium could only choose if releasing the source code publicly (e.g., through a website or a GitHub account) or only to the HERCULES end-users.

In particular, the support for the NVIDIA TX1/TX2 made by partner **EVI** has been publicly released and already merged into the official Jailhouse project [14]. Additionally, EVI maintains a specific version on its own GitHub account for supporting the vendor-specific Linux kernel by NVIDIA (<https://github.com/evidence/linux-jailhouse-jetson>).

Dissemination activities

- Talk at 2nd Linux-Lab, Florence, December 2018.

3.4. Jailhouse cache colouring

Description

Cache colouring extension to the Jailhouse hypervisor. Cache colouring is a technique for avoiding data eviction on the cache. It works by handling virtual memory virtual memory so that pages with different “colors” have different positions in cache.

Ownership

UNIMORE

Licensing

Due to the constraints given by the current license of Jailhouse, the new features developed by UNIMORE and CTU will be released under the same **GPL** license. The code will be published on the <https://git.hipert.unimore.it/jailhouse/jailhouse> repository at the end of the HERCULES project.

The partners will propose by the end of 2019 the upstreaming of the developed features by submitting a patch to the mailing list of the Jailhouse project.

Dissemination activities

- Scientific and industrial events such as OSPM and IWES (see deliverable D7.8)
- Direct exploitation with 3rd parties such as Xilinx Inc. or United Technology Research (see deliverable D6.2)

3.5. Jailhouse GPREM (Guarded PREM)

Description

Predictable memory hierarchy (PMH) extension to the Jailhouse hypervisor. It consists of the PREM co-scheduling algorithm plus the MemGuard mechanism, both implemented in the Jailhouse hypervisor.

Ownership

UNIMORE, CTU

Licensing

Due to the constraints given by the current license of Jailhouse, the new features developed by UNIMORE and CTU will be released under the same **GPL** license. The code will be maintained on UNIMORE's GitLab system (i.e. <https://git.hipert.unimore.it/hercules2020/jailhouse>). To have some time for exploiting this result, the partners will try to upstream the developed features by submitting a patch to the mailing list of the project before one year from the end of the HERCULES project (i.e. by December 31st 2019).

Dissemination activities

Scientific and industrial events such as OSPM and IWES -- see also D7.8 Final Dissemination Report.

3.6. PREM Compiler

Description

The LLVM compiler [15] is an open-source project started by the University of Illinois. Partner **ETHZ** has enhanced this compiler adding support for the PREM model [19].

Ownership

ETHZ

Licensing

The LLVM compiler is released under the UIUC license. Although the HERCULES consortium was free to relicense the new features under a proprietary license, partner ETHZ has decided to release them under the same **UIUC** license.

The initial open source release (containing PREM compiler, libpremnotify and GPUGuard) has been tagged with DOI

<http://doi.org/10.5905/ethz-1007-157>

while information on the continued development will be available at the following page:

https://iis-people.ee.ethz.ch/~bjoernf/h2020_hercules/index.html

Dissemination activities

- Forsberg et al, "Taming Data Caches for Predictable Execution on GPU-based SoCs", DATE'19.
- Matejka et al, "Combining PREM compilation and ILP scheduling for high-performance and predictable MPSoC execution", PMAM'18.
- B. Forsberg, L. Benini, A. Marongiu, "HePREM: Enabling Predictable GPU Execution on Heterogeneous SoC", DATE'18, 2018
- J. Matejka, B. Forsberg, M. Sojka, P. Sucha, L. Benini, A. Marongiu, Z. Hanzalek, "Combining PREM Compilation and Static Scheduling for High-Performance and Predictable MPSoC Execution", Journal of Parallel Computing, accepted

3.7. PREM library (libpremnofity)

Description

Libpremnofity is the low-level glue of the HERCULES infrastructure, it provides an API to which the Compiler inserts "hooks" to guard the entering of the PREM phases, thus preventing the execution to use memory when it is not permitted to by the system schedule. The implementations of Libpremnofity used within the HERCULES stack are forwarding memory requests to the memory scheduler, either via GPUguard (for GPU programs), or directly to the HERCULES Jailhouse Hypervisor (for CPU programs). The Libpremnofity API provides a portable interface for managing the memory guarding, enabling the reuse of the HerculesCompiler for PREMization on other systems.

Ownership

ETH, UNIMORE, CTU

Licensing

After the end of the project, the Libpremnofity library will be released under **UIUC** license. For information about how to download the code, see the Licensing section of the PREM Compiler.

Dissemination activities

As Libpremnofity is an integrated component in the Hercules stack, operating in the background, no specific dissemination has been done to external parties on this software in isolation. It can be noted however, that the portable API provided by Libpremnofity was fundamental for the joint work by project partners ETHZ and CTU, in which the HerculesCompiler was used for an early evaluation of PREM on the Tegra platform, before the final version hypervisor became available:

- J. Matejka, B. Forsberg, M. Sojka, Z. Hanzalek, L. Benini, A. Marongiu, "Combining PREM compilation and ILP scheduling for high-performance and predictable MPSoC execution", 9th International Workshop on Programming Models and Applications for Multicores and Manycores, 2018.
- J. Matejka, B. Forsberg, M. Sojka, P. Sucha, L. Benini, A. Marongiu, Z. Hanzalek, "Combining PREM Compilation and Static Scheduling for High-Performance and Predictable MPSoC Execution", Journal of Parallel Computing

3.8. GPU memguarding (GPUGuard)

Description

GPUGuard is a Linux kernel module that manages system-level PREM between CPU and NVIDIA GPUs.

Ownership

ETHZ

Licensing

The GPUGuard support will be released under the **GPL** license. For information about how to download the code, see the Licensing section of the PREM Compiler.

Dissemination activities

- Forsberg et al, "Gpuguard: Towards supporting a predictable execution model for heterogeneous soc", DATE'17.
- Forsberg et al, "On the Cost of Freedom From Interference in Heterogeneous SoCs", SCOPES'18.

3.9. Tracker

Description

The KCF tracker was originally developed by Tomas Vojir and released under the open-source BSD license. It should be conceptually similar to the AGI proprietary tracker but due to its open-source licensing it is better suited for certain dissemination and exploitation activities. Compared to the original version this tracker was 1) ported from x86 architecture to ARM, 2) parallelized for execution on the GPU, 3) extended for tracking of rotating objects, 4) improved by fixing a few bugs and 5) integrated with the HERCULES toolchain.

Ownership

CTU

Licensing

The KCF tracker, developed by partner **CTU**, is available under the **BSD** license on a GitHub repository at <https://github.com/CTU-IIG/kcf>.

Dissemination activities

- Sending a few pull requests (with bug-fixes and improvements) to the original version <https://github.com/vojirt/kcf/pulls?utf8=%E2%9C%93&q=is%3Apr+author%3Awentasah+>
- Bachelor thesis describing some of the changes and evaluation is publically available <https://dspace.cvut.cz/handle/10467/76311?locale-attribute=en>
- Planned demonstration of the PREMized tracker at the final IAB meeting, integrated in PITOM's use case.

4. CONCLUSIONS

The following table summarizes the licenses of the components developed in HERCULES:

Component	Partners	Source Undisclosed	Open-source only to recipients	Open-source to public	License
Avionic App	AGI, PIT	✓			
Automotive App	MM	✓			
PREM Compiler	ETHZ			✓	UIUC
Tracker	CTU			✓	BSD
Libpremnofity	ETHZ, UNIMORE, CTU			✓	UIUC
GPUGuard	ETHZ			✓	GPL
Run-Time Environment (RTE)	EVI	✓			
Jailhouse co-scheduling (PREM)	CTU, UNIMORE			✓	GPL
Jailhouse's colouring	UNIMORE			✓	GPL
Jailhouse's Memguarding	CTU, UNIMORE			✓	GPL
Jailhouse for TX1/TX2	EVI			✓	GPL (mainline)
ERIKA RTOS for Jailhouse on reference platforms	EVI			✓	Dual license. GPL (on GitHub)
Work by UNIMORE for NVIDIA	UNIMORE	✓			
Power management for SCHED_DEADLINE	EVI			✓	GPL (mainline)

It is important to highlight that most of the components have been released as open-source software. In some cases (e.g., Linux kernel, Jailhouse hypervisor) this choice has been forced by the existing licenses. In other cases (e.g. PREM compiler) the consortium has evaluated the potential benefits and consequences (i.e. better visibility of the new features, possibility of consulting and collaborations, etc.). Finally, the new features of the ERIKA Enterprise RTOS have been released following the original dual license scheme; such a scheme allows getting a fair trade-off between visibility (thanks to the open-source GPL license) and revenue (by selling the linking exception to all interested industrial partners).

5. REFERENCES

- [1] Juri Lelli, Claudio Scordino, Luca Abeni, Dario Faggioli, *Deadline scheduling in the Linux kernel*, Software: Practice and Experience, 46(6): 821-839, June 2016.
- [2] Free Software Foundation, <https://www.fsf.org/>
- [3] Wikipedia, SCHED_DEADLINE, https://en.wikipedia.org/wiki/SCHED_DEADLINE
- [4] GPL, <https://www.gnu.org/licenses/old-licenses/gpl-2.0.html>
- [5] Enrico Bini, Giorgio Buttazzo, Johan Eker, Stefan Schorr, Raphael Guerra, Gerhard Fohler, Karl-Erik Arzen, Vanessa Romero Segovia, Claudio Scordino, *Resource Management on Multicore Systems: The ACTORS Approach*, IEEE Micro, vol. 31, no. 3, pp. 72-81, May/June 2011.
- [6] GNU Classpath license, <https://www.gnu.org/software/classpath/license.html>
- [7] GPL Linking exception, https://en.wikipedia.org/wiki/GPL_linking_exception
- [8] BSD licenses, https://en.wikipedia.org/wiki/BSD_licenses
- [9] Giuseppe Lipari and Sanjoy K. Baruah. *Greedy reclamation of unused bandwidth in constant bandwidth servers*. In IEEE Proceedings of the 12th Euromicro Conference on Real-Time Systems, Stockholm, Sweden, June 2000.
- [10] Claudio Scordino, Giuseppe Lipari, *A Resource Reservation Algorithm for Power-Aware Scheduling of Periodic and Aperiodic Real-Time Tasks*, IEEE Transactions on Computers, December 2006.
- [11] Claudio Scordino, *Dynamic Voltage Scaling for Energy-Constrained Real-Time Systems*, PhD Thesis, University of Pisa, December 2007.
- [12] Luca Abeni, Juri Lelli, Claudio Scordino, Luigi Paolopoli, *Greedy CPU reclaiming for SCHED_DEADLINE*, Proceedings of 16th Real-Time Linux Workshop (RTLWS), Dusseldorf, Germany, October 2014.
- [13] ERIKA Enterprise, <http://www.erika-enterprise.com/>
- [14] Jailhouse, <https://github.com/siemens/jailhouse>
- [15] The LLVM Compiler Infrastructure, <http://llvm.org/>
- [16] University of Illinois/NCSA Open Source License, https://en.wikipedia.org/wiki/University_of_Illinois/NCSA_Open_Source_License
- [17] MIT License, https://en.wikipedia.org/wiki/MIT_License
- [18] Black Duck software, Top Open-Source licenses, <https://www.blackducksoftware.com/top-open-source-licenses>
- [19] Pellizzoni, R., Betti, E., Bak, S., Yao, G., Criswell, J., Caccamo, M., Kegley, R. *A predictable execution model for COTS-based embedded systems*. In 17th IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS), pp. 269-279, 2011.
- [20] Illinois Open Source License, <https://otm.illinois.edu/disclose-protect/illinois-open-source-license>