PUCRS/HP Agreement

LIS – Laboratory of Innovation in Software

**Non-Volatile Memory**

Technical report

TR 2014

Porto Alegre, June, 2014.

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Team  Version 1  Authors:   |  |  | | --- | --- | | Taciano Perez  taciano.perez@hp.com |  | | Micheli Sielecki  pm.lis@cpca.pucrs.br | Matheus Alves  tl.lis@cpca.pucrs.br | | Pedro Monteiro  pedro.monteiro@cpca.pucrs.br | Natan Facchin  natan.facchin@cpca.pucrs.br | | Cristovam Lage  cristovam.lage@cpca.pucrs.br |  | |  |  | |
| **PUCRS/HP Agreement**  **LIS – Laboratory of Innovation in Software**  Coordination - PUCRS: Prof. Marcelo Hideki Yamaguti, PhD  Projects Manager - HP: Paulo Sá  Technical Leader - Processes - HP: Dante Antunes  Technical Leader - Technologies - HP: Alexandre Soares  Technical Team:  Project Manager: Micheli Sielecki  Technical Leader: Matheus Alves  *Study developed by LIS - Laboratory of Innovation in Software of the PUCRS/HP Agreement, financed by Hewlett-Packard Computers Ltd. with resources of Law 8.248/91.* |



|  |
| --- |
|  |
|  |

**Summary**

1 Introduction 4

2 Stories and Tasks 4

2.1 Organize Repository 4

2.1.1 Organize Repository - Update configure file 4

2.1.2 Organize Repository - Remove project/configuration files 4

2.2 Define and Save Needed Structures 5

2.2.1 Define and Save Needed Structures - Save/Load heap 5

2.2.2 Define and Save Needed Structures - Save utf8/classes hash table 5

2.2.3 Define and Save Needed Structures - Find Needed Structures 5

2.2.4 Define and Save Needed Structures - Load utf8/classes hash table 5

3 Tests 6

4 Support Software 7

4.1 Github 7

5 Reviews 8

# Introduction

This document aims to provide a high level documentation of the NVM (Non-Volatile Memory) project. In its sections you will find information about the stories and tasks developed during the project sprints. Also, you can verify the testing methods and support software used by the team.

# Stories and Tasks

## **Organize Repository**

* Update configure file
* Remove project files
* Remove configuration files

## **Organize Repository - Update configure file**

We have encountered a problem when trying to run the JamVM without the -O2 optimization flag on the configure file. The solution we opted to use is to manually change the makefiles flags to -O0.

## **Organize Repository - Remove project/configuration files**

We have deleted all unnecessary files inside our repository and updated the ignore file so no more unnecessary files are committed.

## **Define and Save Needed Structures**

* Save heap
* Load heap
* Save classes hash table
* Load classes hash table (Nice to have)
* Save utf8 hash table
* Load utf8 hash table (Nice to have)
* Find Needed Structures (Recurrent task, Nice to have)

## **Define and Save Needed Structures - Save/Load heap**

We have included parameters to run the JamVM in persistent mode. In this mode, the heap allocated memory is linked to a file. This file is used on subsequent persistent mode runs and is also used for debugging.

## **Define and Save Needed Structures - Save utf8/classes hash table**

We have created new functions to initialize hash tables in persistent mode. On these functions, the hash table allocated memory is linked to a file to be used on subsequent persistent mode runs.

## **Define and Save Needed Structures - Find Needed Structures**

After some investigation, we found out that chunk's headers have information that should be saved. The first two bits of a header are flags to check if the chunk has hash code and if the hash code is taken, respectively. The last 3 bits are also flags but further investigation is needed on those.

## **Define and Save Needed Structures - Load utf8/classes hash table**

Unfortunately, we weren't able to complete this task with the available time for this sprint.

# Tests

We have created another initialization parameter called "testingmode".

We have created a test header that creates a log file. This header only runs if the testingmode flag is true.

# Support Software

## Github

We have been using github as our main repository for code. Our repository can be found in <https://github.com/icaro-henrique/JamVM-PH>.

# Reviews

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| June/20/2014 | 1.0 | Sprint: Welcome to the Jam | Pedro Monteiro |
|  |  |  |  |
|  |  |  |  |