

JZX: Java 'Sinclair ZX Spectrum' emulator

Evaluation and Optimization of Emulation Engine

Group 7

António Goulão
66950
antonio.m.goulao@ist.utl.pt

José Correia
67026
jose.p.correia@ist.utl.pt

Miguel Borges
67041
miguel.a.borges@ist.utl.pt

Abstract

In this project we will try to improve the implementation of JZX in a way that optimizes the emulation process and the memory management. We will apply techniques learned in the Virtual Environment Execution course. We will also apply benchmarking techniques to compare the performance of the actual and our solutions.

Keywords: JZX, emulator, ZX Spectrum, virtualization.

interpreter and run the existent programs in its time.

- **Motivation:** This project will allow us to acquire a deeper knowledge of how a System-VM works inside. This project was last updated in 2006 and has a basic emulation process that we pretend to improve and provide a tool to analyze which operations are more used.

1 Overview

JZX is an open source emulator for ZX Spectrum. JZX was written by Razvan Surdulescu, who worked previously in two other emulators, XZX (UNIX emulator) and WinXZX (Win32 port of XZX) with credits to Des Herriot and Erik Kunze. This emulator was written using the pieces of the source of the two emulators.

2 Virtualization Technology Studied

- **Name:** JZX
- **URL:** <http://www.sonic.net/~surdules/projects/jzx/>
- **VM Type:** System-VM
- **Common Usage:** A common ZX Spectrum emulator that can be used in any platform that runs Java. This Virtual Machine is capable of emulating the machine's

3 Internal Mechanisms Study

The main focus of our work in this System-VM is to study the mechanism of code emulation so we can improve it, and study the memory emulation mechanism.

4 Project Approaches

In the first place we will need to analyze the actual implementation of the emulator and understand its mechanisms of emulation. In particular we will study the memory mapping mechanisms used in the actual implementation. This first part has as objective discover ways in which we can optimize the actual implementation of the emulator.

After that we will apply techniques learned in the course in a way that improve the emulator. The main focus will be in optimizing the code decoding of the actual solution. To do that, we pretend to implement basic predecoding. The actual emulation procedure uses a switch statement, it is inefficient because the same instruction is analyzed every time it is executed. Our main focus is to analyze an instruction once and save it for later uses, simplifying interpretation and speeding it up. This procedure will raise memory consumption, but still acceptable according to the memory available in computers nowadays. While studying the code we will also study the documentation provided by the main devel-

oper of the emulator to helps us find places where we can improve it.

5 Evaluation

Along the way of the project we will apply benchmark techniques so that we can the best solutions. We will test the number of bytecodes executed and the number of frames per second before and after implement our solution. The tests will be performed using 128KB games.

6 Detail Section

This emulator is written in Java, so the project will be executed using this language. The changes proposed here, will use the algorithms and techniques presented in the lectures and in the course's bibliography.

7 Conclusion

At the checkpoint, we expect to have a good overview of how the emulator is implemented and have a strategy ready to implement. After concluding this project we hope to have a better understanding of the virtualization mechanisms used in the emulator. We hope to improve the performance and give an important contribute to this project.

References

- [1] R. Surdulescu. *JZX: Java 'Sinclair ZX Spectrum' emulator* <http://www.sonic.net/~surdules/projects/jzx/>.