Alma Mater Studiorum · Università di Bologna

SCUOLA DI SCIENZE
Corso di Laurea in Informatica

An Updated Emulated Architecture to Support the Study of Operating Systems

Relatore:
Prof. Renzo Davoli

Correlatore:
Prof. Michael Goldweber

Presentata da:
Mattia Biondi

Sessione Straordinaria
Anno Accademico 2018/2019

dedica

Abstract

One of the most effective ways to learn something new is by actively practising it, and there is—maybe—no better way to study an Operating Systems course than by building your OS. However, it is important to emphasize how the realization of an operating system capable of running on a real hardware machine could be an overly complex and unsuitable task for an undergraduate student. Nonetheless, it is possible to use a simplified computer system simulator to achieve the goal of teaching Computer Science foundations in the university environment, thus allowing students to experience a quite realistic representation of an operating system. μMPS has been created for this purpose, a pedagogically appropriate machine emulator, based around the MIPS R2/3000 microprocessor, which features an accessible architecture that includes a rich set of easily programmable devices. μMPS has an almost two decades old historical development and the outcome of this following thesis is the third version of the software, dubbed μMPS3. This second major revision aims to simplify even more the emulator's complexity to lighten the load of work required by the students during the OS design and implementation. Two of these simplifications are the removal of the virtual memory bit, which allowed address translation to be turned on and off, and the replacement of the tape device, used as storage devices, with a new flash drive device—certainly something more familiar to the new generation of students. Thanks to the employment of this software and the feed-backs received over the last decade it has been possible to realize not just this following thesis but also to develop some major improvements which concern everything from the project building tools to the front-end, making μMPS a modern and reliable educational software.

Contents

Chapter 1

Introduction

1.1 Background

The study and the consequent implementation of how an operating system works is by now a long-established and consolidated practice in every Computer Science's curriculum. It is, actually, one of the crucial components of a computer and it is responsible of ensuring it's basic operations, by coordinating and managing the system's resources like processor, memory, devices and processes, thus allowing hardware and software to interface each other. This is probably the first real example of "big project" which students must experience, thanks to complex intercommunication system that has to exist between the different machine's components, and which the study allows to comprehend the most common software engineering practices. The approach on practical contexts is essential to fully understand how a machine works behind the theoretical notions studied in the early stages of the course of study, and it is usually followed by the debate of which is the best teaching choice concerning processor architectures. Obviously, there is not only one way of how a central processor unit can be implemented, and while older realizations—although applicable for educational purposes—are now obsolete and incompatible with current platforms, modern ones are designed to achieve high speed and quality, which makes them overly complex and unsuitable for the pedagogic experience. The MIPS architecture has become one of the landmarks in this teaching choice over the years due to its clean and elegant instruction set, despite being excessively convoluted to student's perception because of the high level of details obscuring the basic underlying features of it. A potential solution to this problem is the adoption of a simplified computer system simulator like μMPS, to bring together an adequate level of understanding and a realistic representation of a real operating system.

1.2 History of μMPS

μMPS is based on the machine emulator MPS [1], designed and realized by Professor Renzo Davoli and one of his graduate students Mauro Morsiani in the late 1990s at the University of Bologna. The initial purpose was to bring back to life the layout and implementation experience of an operating system through an educational emulator which could be run on real hardware. This practice was already possible in the past years when the processor's architecture studied were the same available on real machines, and it has gone lost through the years because of the high-speed development of new more complex technologies used on physical implementations. MPS was able to emulate the MIPS R3000 processor along with five other different device categories: disks, tapes, network adapters, printers and terminals. The concerned CPU was genuinely emulated together with its complex virtual memory management system, which was the main subject of the feed-backs received during class testing of MPS as a pedagogical tool at the University of Bologna and Xavier University in an undergraduate operating systems courses taught respectively by Renzo Davoli and Michael Goldweber. The urge of simplification led to the creation of μMPS [2], virtually identical to MPS but with the addition of a virtual memory management subsystem which had resemble as much as possible to the conceptual one found in popular introductory OS texts. The only other difference was the new novice-friendly graphical user interface, significantly improved again in 2011 by Tomislav Jonjic during the development of the first major revision of the emulator, μMPS2 [3], which also implemented the support to up to sixteen MIPS R3000-style processors.

1.3 μMPS3

1.4 Document's Structure

Chapter 2

Memory Management

As for the two prior versions, the memory subsystem of μMPS3 is divided into physical and virtual, and both have undergone significant changes in this major revision. The main reasons for the modifications made are the further simplification of the work required by the user for the kernel implementation and the better clarification of the emulator's memory internal view which caused some confusion during the more than 10 years of use of μMPS as an educational tool in operating system courses.

2.1 Physical Memory

The physical address space is divided into two big areas: a kernel reserved space, from address `0x0000.0000` to `0x1FFF.FFFF`, and the installed RAM, from address `0x2000.0000` to `RAMTOP`. This last value is calculated upon the value retrieved from the configuration file, settable from the machine config dialog in the front-end emulator, which goes from a minimum of 8 to a maximum of 512 memory frames. Being the size of every frame 4 kilobyte, μMPS can have from 32KB up top 2MB of installed RAM. Hence, the value of `RAMTOP` range from `0x2000.8000` to `0x2020.0000`.

The first area was reserved for:

- Execution ROM code, which layed in a read-only segment from address

`0x0000.0000` to `ROMTOP`;

- prova.

This address space, corresponding to the first 0.5GB of physical memory, causes the raising of an Address Error when trying to access it while the processor was in user-mode.

2.2 TLB Floor Address

2.3 VM Bit Removal

Chapter 3

Exception Handling

3.1 Primary Design

3.2 BIOS Data area

Chapter 4

Flash Devices

4.1 Tape Readers

4.2 Flash Drives

Chapter 5

Project Modernization

5.1 CMake Migration

5.2 Qt5 Transition

5.3 Logo and Icon Theme

Chapter 6

Linux Packaging

6.1 Debian Package

6.2 Arch Linux Package

Chapter 7

Conclusions

Knowing how an operating system works should be common knowledge and not something restricted only to the ones who studied in the IT field. If you are reading this document there are high chances you are doing it on a device of your property, which is running an operating system, and you should know how all of this really works in other to really feel like you own this particular system.

Bibliography

[1] M. Morsiani, R. Davoli, *Learning Operating Systems Structure and Implementation through the MPS Computer System Simulator*, University of Bologna, April 1999.

[2] M. Goldweber, R. Davoli, and M. Morsiani, *The Kaya OS project and the μMPS hardware emulator*, SIGCSE Bull., vol. 37, pp. 49–53, June 2005.

[3] M. Goldweber, R. Davoli, and T. Jonjic, *Supporting operating systems projects using the μMPS2 hardware simulator*, in *Proceedings of the 17th ACM annual conference on Innovation and technology in computer science education*, ITiCSE 2012, (New York, NY, USA), pp. 63–68, ACM, 2012.