ISEL
INSTITUTO SUPERIOR DE ENGENHARIA DE LISBOA

Computer Science and Engineering

System Virtualization Techniques

Autumn 2023

Professors

João Trindade	joao.trindade@isel.pt	F.0.21	LEIC51D, LEIC53D, LEIC51N
Jorge Martins	jorge.martins@isel.pt	F.0.16	LEIC52D

General Info

Original name [PT]	Técnicas de Virtualização de Sistemas [TVS]
Scientific Area	Computer Science and Engineering
ECTS Credit Points	6 credit points
Level, Year/Semester	Undergraduate, 3rd year / 5th semester
Available in	Autumn-Winter Semester (Sep. - Feb.)
In-class contact hours	15 weeks x 4.5 hours per week in class
Language	Portuguese (lectures, tests/exams) English (course materials, assignments, tests/exams)

Description

In order to optimize the use of existing hardware resources and to simplify the management and maintenance of computer systems, software deployment and execution frequently relies on virtualized computing infrastructures, where the virtualization occurs in different levels and forms. Operating systems use basic virtualization techniques to allow multiple programs to execute simultaneously in isolated environments called *processes*, where a program in execution appears to have a computing environment for itself. Some operating systems also offer, via *emulation*, the possibility of running programs originally produced for microprocessors with different instruction sets. In other cases, they allow running programs built for a different operating system. Additionally, a *hypervisor* is a specialized form of system software that is able to provide, on top of a single physical system, multiple instances of virtual computers, usually known as *virtual machines*. Also, in a different level and form, we have systems ready to run containerized software, where a particular arrangement of software parts is set to run as a group of processes in a dedicated computing environment, known as a *container*, on top of an operating system.

In this course unit, we will explore these virtualization techniques, mainly from the perspective of software deployment, but also, when reasonable, looking at how each type of virtualization is achieved. During the semester, students will have the opportunity to explore these subjects in practice, by working in groups to complete assigned coursework. One global or two partial written tests will assess students' knowledge individually.

Prerequisites

- Students will need to write software in C, including interacting with the operating system. We expect at least two (but preferably three) semesters of programming experience, using any language, but including some non-introductory programming experience in C. Equivalent experience in C++ is also acceptable, but it may require additional effort to adapt.
- Students are expected to know basic computer architecture, including assembly language. Ideally, they would understand how a native computer program is executed.
- We expect students to be familiar with multithreading and synchronization. Although the course is not particularly focused or heavy on these subjects, there will be no effort to circumvent them and they will be used whenever deemed necessary.
- We expect students to have basic knowledge about IP networks and to be able to write software using sockets, if needed.

Related Course Units

- [PSC] *Computer Systems Programming* – programming in C and x86 assembly with associated calling conventions; executable file organization; native programs execution
- [AC] *Computer Architecture* – computer organization, assembly programming, interrupts
- [PC] *Concurrent Programming* – multithreading and synchronization
- [RCp] *Computer Networks* – IP networks
- [LAE] *Managed Languages and Runtimes* – covers a particular form of virtualization, usually know as *managed runtime environments* or *virtual execution environments* (such as the Java Virtual Machine, Microsoft's Common Language Runtime or WebAssembly runtimes), which are therefore not covered in this course unit
- [CN] *Cloud Computing* – further explores the use of virtualization techniques in context

Topics

- Introduction to virtualization techniques
- Operating systems processes as isolated execution environments
- The UNIX/POSIX process model
- Hardware provided protections for isolation of operating systems processes
- Operating systems techniques for safe sharing of memory among processes
- Inter-process communication mechanisms
- Operating system services and their configuration and management
- Taxonomy of virtualization techniques
- Hypervisors for full computer virtualization
- Emulators for virtualization of a different processor
- Containers as isolated environments for groups of processes and their resources

Bibliography

- Andrew S. Tanenbaum, Herbert Bos, *Modern Operating Systems*, 5th edition, Pearson 2022
- Shashank Mohan Jain, *Linux Containers and Virtualization*, Apress 2020
- Sean P. Kane, Karl Matthias, *Docker: Up & Running*, 3rd edition, O'Reilly 2023

Coursework

Students will work in groups of 3 to complete **4 mandatory assignments**, with the following topics and expected start and end dates:

- CW1 – UNIX Programming — Sep. 22 – Oct. 8
- CW2 – HW Protections and OS Processes Isolation — Oct. 11 – Oct. 25
- CW3 – OS Services, Hypervisors, Emulators — Nov. 7 – Nov. 26
- CW4 – Containers — Nov. 30 – Dec. 16

Each group will have a dedicated GitHub repository, setup via GitHub Classroom. An access link will be provided to students before the first assignment start date.

All students are expected to contribute to assignments, and this should be visible in the multiple commits made to the group repository, which shall also reflect the work progress timeline. Do not perform bulk commits with vast changes and make sure that all commits are properly identified.

Comments to group work will be provided in **two feedback sessions** in the following dates:

- Feedback to CW1 and CW2 Oct. 26 – Oct. 30
- Feedback to CW3 and CW4 Dec. 18 – Dec. 20

For each feedback session, each group will select a date/time slot from a provided list. In exceptional cases, date/time slots may also be arranged directly with the professor.

Written Tests and Grading

Approval in this course unit requires a final weighted average of **at least 9.5 points** (in 20) in two assessment components:

- **[T]** Theoretical assessment via written tests, with a required minimum of 9.5 points, weighted at **50%**, in two alternatives:
  - one global test [TG], with a duration of 2h30m, and a minimum of 9.5 points
    or
  - two partial tests [TP1,TP2] of 1h15m, with a minimum of 8.0 points each, and a minimum average of 9.5 points
- **[P]** Practical assessment, with individual grading of the contribution to the 4 mandatory group assignments, with a required minimum of 9.5 points, weighted at **50%**.

There will be three dates for the written tests:

- [TI] around the 8th or 9th week of classes, where you can take TP1 only
- [EN] first examination period (Jan. 3 – Jan. 20), where you can take TP1, TP2, or TG
- [ER] second examination period (Jan. 22 - Feb. 3), where you can take TP1, TP2, or TG

In each of these three dates, you can take any of the available tests (TP1, TP2, or TG). If you take the same test multiple times, only the best grade will be considered.

If you get 8.0 points or more both in TP1 and TP2, their average will be the TP grade.

The highest between TP and TG will be your [T] component grade.

As soon as you have a [T] component grade of 9.5 or above, your professor will:

- propose a [P] component grade from the observation of your work during the semester
  or
- request a discussion with you, in order to determine your [P] component grade

In the first case, if you disagree with the proposed [P] component grade, you can request a discussion, in which case the previously proposed grade is discarded and the discussion grade prevails, even if it is lower.

Class Support Infrastructure

- Moodle for TVS
  - General class announcements
  - Links to general resources
  - Publication of grades
  - Link : https://2324moodle.isel.pt/course/view.php?id=7508

- Moodle for each TVS section
  - Announcements for each specific section
  - Lecture summaries
  - LEIC51D : https://2324moodle.isel.pt/course/view.php?id=7505
  - LEIC52D : https://2324moodle.isel.pt/course/view.php?id=7506
  - LEIC53D : https://2324moodle.isel.pt/course/view.php?id=7558
  - LEIC51N : https://2324moodle.isel.pt/course/view.php?id=7559

- GitHub repository for each TVS section
  - Source code of examples presented in lectures
  - LEIC51D : https://github.com/isel-leic-tvs/tvs-2324-1-leic51d/
  - LEIC52D : https://github.com/isel-leic-tvs/tvs-2324-1-leic52d/
  - LEIC53D : https://github.com/isel-leic-tvs/tvs-2324-1-leic53d/
  - LEIC51N : https://github.com/isel-leic-tvs/tvs-2324-1-leic51n/

- TVS Slack Workspace
  - Communication between TVS participants
  - Set your display name as: **Student_Number - Name Surname**
    - Example: 12345 - Fernando Pessoa
    - Erasmus students may use only **Name Surname**
  - Signup link (first access only): https://join.slack.com/t/isel-leic-tvs-2324-1/signup

- GitHub Classroom repositories
  - Git repositories for work groups
  - Set your git identity in all your work machines as in this example:
    - git config user.name "12345 - Fernando Pessoa"
    - git config user.email "A12345@alunos.isel.pt"
    - Erasmus students may use *Name Surname* and some working email
  - Link: *(to be provided later)*

Software to have installed

- VirtualBox : https://www.virtualbox.org/

- Lubuntu 22.04LTS : https://tinyurl.com/tvs-lubuntu-2204lts [4.1GB]
  - Uncompress the downloaded file
  - Move the uncompressed folder to an adequate location. In Windows, you may use the default location for VirtualBox images: *C:\Users\<username>\VirtualBox VMs*
  - Open VirtualBox and select *Machine > Add*
  - Navigate to the VM image folder and select file *Lubuntu22.04.vdi*
  - Select the newly installed VM from the VM list and press *Start*

Bom estudo, bom trabalho e bom semestre!