

# Laboratórios de Computadores: Apresentação

## Computer Labs: Introduction

2º MIEIC

Pedro F. Souto (pfs@fe.up.pt)

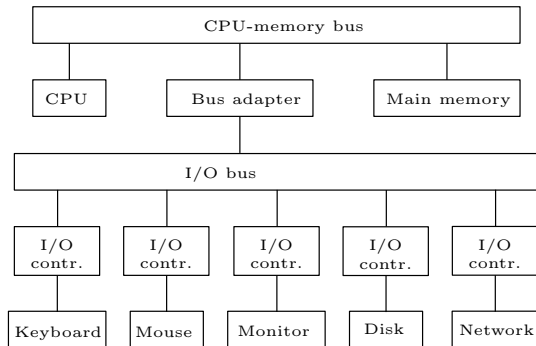
September 21, 2020

# Staff

- ▶ Sara Fernandes ([sfclf@fe.up.pt](mailto:sfclf@fe.up.pt))
- ▶ Mário Cordeiro ([mmfc@fe.up.pt](mailto:mmfc@fe.up.pt))
- ▶ Nuno Paulino ([nmcp@fe.up.pt](mailto:nmcp@fe.up.pt))
- ▶ Pedro Ferreira do Souto ([pfs@fe.up.pt](mailto:pfs@fe.up.pt))
- ▶ Rui Grandão ([ruirocha@fe.up.pt](mailto:ruirocha@fe.up.pt))

# Context

- ▶ I/O devices are an integral part of a computer

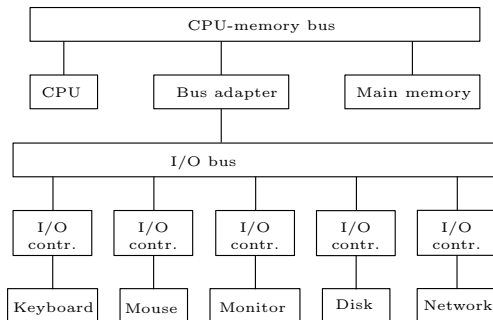


- ▶ The relevance of I/O devices has increased significantly with the deployment of an increasing number of embedded systems and the emergence of the Internet of Things
- ▶ However, programming of I/O devices requires specific knowledge and techniques

# Objectives

*This course aims to endow students with the knowledge and the skills required to:*

- 1. Use the programmatic interface of the most common computer I/O devices;***



- 2. Develop low-level/system-level and embedded programs*
- 3. Use software tools typical of large programming projects*

# Prerequisites

- ▶ **Programação**
  - ▶ You'll program a lot, mostly in C
- ▶ **Microprocessadores e Computadores Pessoais**
- ▶ Arquitectura de Computadores
- ▶ If you have **not** completed Programação, please consider to drop LCOM, if you do not have enough time to dedicate to programming

# Syllabus

I/O devices and their programming techniques

C programming

Programming tools

# Method

## Learn by doing

“I hear, I forget. I see, I remember. I do, I understand”

## Several short lab assignments

- ▶ Each focusing on one I/O device
- ▶ Some of them take only one lab class, others take two lab classes
- ▶ Requiring a preparation of about 6 hours per lab class (not including lectures or lab classes)

## One integration project

- ▶ Must use
  - ▶ at least 3 different I/O devices
  - ▶ interrupts
- ▶ Expected effort: about 9 hours per week (during 5 weeks)

**Note:** Both lab assignments and project should be done in groups of 2 students.

# Project Examples

- ▶ Games (video, timer, keyboard and mouse)
- ▶ Two user games (video, timer, keyboard and serial port)
- ▶ Electronic calendar (video, keyboard, mouse, RTC and timer)
- ▶ Music composer/player (video, keyboard, mouse and timer)
- ▶ Text editor (video, keyboard, mouse, timer and RTC)
- ▶ Typing tutor (video, keyboard, mouse, timer)
- ▶ File transfer between PCs (video, keyboard, serial port)
- ▶ Chat between PCs (video, keyboard, serial port)
- ▶ Video player (video, keyboard, mouse, timer and RTC)
- ▶ Drawing/painting program (video, keyboard, mouse, timer, RTC and serial port)



# COVID-19 Operating Mode

Lectures Remotely

Lab classes In person, every-other week for each student

- ▶ Assignment based on parity of the student's number:
  - ▶ Check [FEUP's Calendar for the School Year 2020/21](#)
- ▶ You are required to follow appropriate safety rules (masks, etc.)

## Consequences

Groups should (?must?) have students with both even and odd numbers – this way every group will have one student attending the lab class, allowing for easier interaction

- ▶ During the lab classes you are allowed to use headphones to collaborate with your team-mate
- ▶ We may be to help remote students, but only if available, i.e. if we are not helping students attending the class in person

# Work Load

- ▶ LCOM has 6 ECTS, i.e. about 160 hours
  - ▶ Assuming 1 ECTS equal to 27 hours
    - ▶ Check out the [European Credit Transfer and Accumulation System \(ECTS\)](#)
- ▶ If you share the load with your team-mate, this should not be a problem.

Unit	Hours/Week	No. Weeks	Total
Lectures	2	11	22
Labs classes	3	12	36
Prep. L0	5	1	5
Prep L2-L5	6	8	48
Proj.	9	5	45
Total			156

- ▶ Although, you have to attend lab classes every other week only, you are expected to work in every lab class (and prepare for it, as well).

# Bibliography and Other Resources

- ▶ PC HW is well documented on several books and online resources
- ▶ Book mentioned in SIFEUP  
***Mazidi, Muhammad**, The 80x86 IBM PC and Compatible Computers: Assembly Language, Design and Interfacing, 4th Ed., Prentice-Hal*

Note that it does not cover all the subjects, and that, on the other hand, it has a lot more material than needed for this class.

# Grading

**Two Programming Tests (PT)** each test is similar to a lab, but much smaller and likely on a pseudo-device, and is taken individually.

- ▶ Grading is fully automatic
- ▶ **Minimum average grade of 10 (out of 20) is required for passing**

**Project (FP)** minimum grade of 10 (out of 20) is required for passing

- ▶ **You cannot submit the project, if you do not have a minimum score of 8 (out of 20) in one of the tests**

**Formula**  $0.25 (PT1 + PT2) + 0.5 FP$

# Grading: PT Requirements

- ▶ **Minimum average grade of 10 (out of 20) is required for passing**
- ▶ You can take a make-up programming test during the exam period to meet this requirement

# Grading: PT Requirements

- ▶ **Minimum average grade of 10 (out of 20) is required for passing**
- ▶ You can take a make-up programming test during the exam period to meet this requirement
  - ▶ **As long as you have a minimum score of 8 in one of the tests**
  - ▶ The maximum score in the **make-up PT** is the grade required for you to get the average score of 10

# Grading: PT Requirements

- ▶ **Minimum average grade of 10 (out of 20) is required for passing**
- ▶ You can take a make-up programming test during the exam period to meet this requirement
  - ▶ **As long as you have a minimum score of 8 in one of the tests**
  - ▶ The maximum score in the **make-up PT** is the grade required for you to get the average score of 10
    - ▶ This means that students that take the make-up PT cannot have a final grade than that of a student that is not allowed to take the make-up PT (and has the same project grade)

# Grading: Project

- ▶ **All grades are individual**
  - ▶ Even though the project should be done in groups
- ▶ Project grades are normalized for groups of 2
- ▶ The grade of each group member is obtained by applying a factor to the group's grade
  - ▶ The factor depends on the member's **contribution** mostly **participation** also to the project's grade
    - ▶ In principle, if you choose to be the only group member, the factor's value will be 1.
  - ▶ We will use a piecewise linear function to determine the value of the factor
    - ▶ Likely "breakpoints" are: (0,0), (0.33,0.8), (0.5,1), (0.8, 1.1) and (1, 1.1)
- ▶ Each member of the group must fill a Google form with its own self-assessment (contribution and participation to the project) submitted by the respective deadline



# Final Project Grading (1/2)

## Execution: 45%

- ▶ 10% (i.e. almost 25% of the exec grade) for demo in the last lab class (if ...)

## Code: 20%

- ▶ Structure and Modularity
- ▶ Documentation (use Doxygen)
- ▶ Readability
  - ▶ Names and comments
  - ▶ Indentation
- ▶ Compilation warnings

## Final Report: 20%

- ▶ Summary of what is and what is not implemented;
- ▶ Usage instructions (with images)
- ▶ Description of the program's architecture
- ▶ **Relevant** implementation aspects (grades above 18)
- ▶ Function call diagram

## Video: 5% A short video with a demo of your project

## Tools: 5% (Git) (We expect you to commit to the Git repository at least once a week, with useful log messages)

## Project Specification: 5%

# Final Project Grading (2/2)

- ▶ To the grade obtained by applying the above criteria, we'll apply:

## Difficulty Factor

- ▶ number and type of I/O devices
- ▶ features used of the I/O devices
- ▶ I/O techniques used (interrupt vs. polling)
- ▶ use and extent of assembly programming

## Originality Factor

## Team Management Factor

- ▶ load share among group members

## Marketing Bonus

- ▶ of 1 valor for the participation in the Semana Profissão Engenheiro (SPE)
  - ▶ most likely it will not be awarded because of COVID-19
- ▶ in recent years, we have selected 3 or 4 projects per year

# Final Project Milestones (Provisional)

**Project proposal:** To be discussed in the lab class of the week starting on the 2nd of November

- ▶ Half to one page description of the functionality desired, of the devices used and their role in the program
- ▶ Must be rewritten in class, if the instructor does not accept it

**Project specification:** To be discussed in the following lab class

- ▶ Refinement of the proposal, identifying the modules to implement, their functionality and API.
- ▶ Should include planning of the project

**First demo:** In the last week of classes of the semester (starting on December 14)

**Project submission:** January 4 @ 20:00

**Project demonstration:** January 6 to 8

# TEs Grading

Similar to the grading of other students

- ▶ I.e. it is based on programming tests and project
- ▶ The student must demo her/his project in the same dates as other students (January 6 to 8, 2021)

# “Época Especial”

► There is none.

“Melhoria de nota”

► Next year.

# Special Evaluation

**IMPORTANT** Students wishing to:

1. be assessed as TE's
2. use their labs/project/PT (positive) grades from 2018/2019 or 2019/2020

must fill [this Google form](#) by the end of this week, i.e. 2020-09-25.

**IMPORTANT** Please note that by choosing this option, you may have to work alone in the project

- ▶ If you do not want to work alone, it will be up to you to create a group (together with other students) or find a group that accepts you as a new member.

# Academic Integrity

- ▶ The UP and we take this issue very seriously
  - ▶ Check the [Despacho do Reitor N° 08/09/2011](#)
  - ▶ We believe that the majority of you follow the rules
- ▶ You are allowed to discuss the labs
  - ▶ For each lab and for the project, there will be a discussion forum on Moodle
- ▶ However, all code submitted should be either:
  - ▶ Developed by the group members
  - ▶ Provided by me
- ▶ We will use tools to automatically detect common code
  - ▶ **All groups** with common code will be penalized
  - ▶ You **cannot show or share code**

The lab assignments are identical to those of last year, but this is no excuse

- ▶ The penalty may range:
  - From a penalty of “2 valores” in your final grade (if limited to a single device)
  - To **failing** the course



# Important Dates

## Labs

Lab	First lab class	Topic
Lab 0	22-09	Redmine, Devel. and SVN
Lab 2	29-09	Timer
Lab 3	13-10	Keyboard
Lab 4	27-10	Mouse
Lab 5	03-11	Video (graphics)

## Programming Tests

**First** Week starting on 2020-11-09 (FEUP's week)

**Second** Week starting on 2020-11-30, most likely on 2020-12-05,  
i.e. Saturday (i.e. about 4 weeks later)

## Project

What	Week (starting on)	Comments
Proposal	2-11	Last week of classes All To be confirmed
Specification	16-11	
First demo	14-12	
Submission	2021-01-04 @ 20:00	
Presentation	6, 7 and 8 January, 2021	

# Announcements

**Lectures** start 10 minutes after the hour, i.e. at 16:40

- ▶ This is a FEUP's rule, and applies also to lab classes

**Labs** start next Tuesday, i.e. September 29th

- ▶ Students with sections on Tuesdays have 5 lab classes
  - ▶ Assuming we have a make-up lab class on FEUP's week, otherwise even-numbered students will have only 4
- ▶ Students with sections on Fridays have 6 or 5 lab classes depending on parity of their numbers

**New Students** who are not registered yet

- ▶ You should be able to access the material (slides and lab handouts): it is available at the [LCOM's Web page](#)
- ▶ If you want to attend a given lab class while you are not registered, please check with the respective instructor
  - ▶ Given the epidemic, we will be very strict on the number of students per lab class
  - ▶ Of course, whatever decision we make regarding the parity of the student number will apply to all students

# Acknowledgments

- ▶ Prof. António Miguel Pimenta Monteiro (who designed the course)
  - ▶ Prof. João Cardoso (not the same person as TCOM's lecturer) (who perfected it)
  - ▶ Prof. Pedro Silva, who made possible major recent changes:
    - ▶ has proposed, designed and implemented the LCOM Framework;
    - ▶ implemented a set of utilities that make the development process in LCOM easier;
    - ▶ ported libx86emu to Minix, allowing us to use the most recent Minix version, and thus use:
      - ▶ more recent compiler with better error/warning messages;
      - ▶ VirtualBox shared folders
- all of which also simplify the development process in LCOM
- ▶ developed a set of scripts to automatically generate a new VirtualBox image with Minix 3.4.0rc6

# Thank You!

# Questions?

# Platform

## MINIX 3

Unix-like operating system that allows privileged user processes to:

- ▶ Access every memory address
- ▶ Access directly I/O devices
- ▶ Process interrupts

## Linux

- ▶ MINIX 3 is installed in a VirtualBox VM

# Software

- ▶ CLANG compiler and assembler
  - ▶ This is Minix's default compiler and provides much better messages than those of the GNU compiler available in Minix 3.1.8
- ▶ Other SW development tools
  - ▶ `make`
  - ▶ `git`
  - ▶ `doxygen`

# Advice on Course Participation

- ▶ It is important that you participate in the lectures
  - ▶ Especially this year, that each lecture is 2 hours
- ▶ It is also important that you prepare for the lab classes
  - ▶ Unless you prepare your labs, it is unlikely you will be able to complete them during the respective class
- ▶ To motivate you to do both:
  - ▶ In the 2nd hour of each lecture, we will present the main material required for the lab class of the following week.  
E.g., today, we will present the material for next week's lab: Lab 0
  - ▶ In the first hour of each lecture, we will answer to questions about that material as well as about the lab work planned for that week
    - ▶ We will give priority to questions made through a Moodle forum that we will create.
    - ▶ These questions, will be used to assign you a grade to be used in the computation of course participation.
- ▶ It is important that you provide us the information we request you in a timely fashion.
  - ▶ Failure to do so, will affect your course participation score.