

Laboratórios de Computadores:
Apresentação
Computer Labs: Introduction
2º MIEIC

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

September 17, 2018

Staff

- ▶ Nuno Paulino (nmcp@fe.up.pt)
- ▶ Pedro Silva (pmms@fe.up.pt)
- ▶ Pedro Ferreira do Souto (pfs@fe.up.pt)
- ▶ Ricardo Nobre (rjfn@fe.up.pt)

Context

- ▶ I/O devices are an integral part of a computer
 - ▶ Without them, a computer would be like a heater
- ▶ The relevance of I/O devices has increased significantly with the deployment of an increasing number of embedded systems
 - ▶ An "embedded system" is a computer-based system that is part of an "object" and interacts with the physical-world.
E.g. a car's ABS, a washing machine controller, ...
- ▶ However, programming of I/O devices requires specific knowledge and techniques

Objectives

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

- 1. Use the hardware 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
 - ▶ but also in assembly
 - ▶ ... for the IA-32 architecture

- ▶ **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 for extra programming effort

Syllabus

I/O devices

C programming with assembly

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 (excluding classes)

One integration project

- ▶ Must use at least 3 different I/O devices
- ▶ Must use interrupts
- ▶ Should use both C and assembly
- ▶ Should require 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)

(Not So) Recent Changes

What? There are no lab classes on the RTC or on the serial port (UART)

- ▶ However, we will still talk about them in the lectures
- ▶ Students wishing to get a grade of 16 or better in the final project, are expected to use these devices anyway
 - ▶ To use any of these devices you are required to use **all** the other, e.g. you **cannot** use the RTC but not the mouse

Why?

- ▶ Have more lab classes for the other I/O devices
 - ▶ Most of them now have 2 lab classes, instead of only 1
- ▶ Remove some pressure out of the graded lab assignments

Expected results (and fulfilled)

- ▶ Less cheating
- ▶ Higher passing rate

Brand new this year: The LCOM Framework (LCF)

- ▶ Thanks to Pedro Silva, who designed and implemented it
- ▶ It allows us to provide you
 - Helper functions, that will allow you to more easily debug some common and time consuming errors
 - Test cases so that you can be more confident that your code is working as requested
- ▶ Furthermore, the LCF will allow us to automatically grade your code
 - ▶ Actually, we will still need to do some grading "manually", but we expect it to be less than 10%

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	12,5	25
Labs.	3	12	36
Prep. L0	5	1	5
Prep L2-L5	6	8	48
Proj.	9	5	45
Slack			3
Total			162

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 (1/2)

Labs (Labs) Of the 4 graded labs, we will use 3 for computing this component

- ▶ For a final grade of 19 or 20, must use the grade of the mouse lab

Project (FP)

Class Participation (CP) covers lab preparation, class participation proper and contribution to the work submitted

Programming Test (PT) individual test similar to a lab, but much smaller, on a pseudo-device

- ▶ **Minimum score of 8 required for approval**
- ▶ Can be repeated during exam period
 - ▶ Maximum score in the **repeated PT** is 8
 - ▶ If you got the minimum score, you cannot repeat it

Formula $0.4 \text{ Labs} + 0.1 \text{ PT} + 0.4 \text{ FP} + 0.1 \text{ CP}$

Grading (2/2)

- ▶ All grades are individual
 - ▶ Even though labs and projects should be done in groups
- ▶ Lab/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 final result
 - ▶ 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) of each of the labs/projects submitted by the end of the day of the respective deadline
 - ▶ **Failure to comply, may penalize you**

Labs Grading

- ▶ You must submit your code via SVN
 - ▶ This year, we are making an experiment:
 - ▶ The submission deadline will be on the Saturday after the last lab class of every lab
 - ▶ But, if you do not work on the lab classes, we will revert the deadline to the day of your last lab class
- ▶ You must also fill a self-evaluation form for each lab.
- ▶ Grading will be mostly automatic
 - ▶ It will allow us to publish the lab grades much faster
- ▶ We will provide you a subset of the test cases used for grading, to prevent you from getting low marks, just because you did not understand the requirements

Programming Test

- ▶ It is an individual test similar to a lab, but smaller, and may be on a device that is not real
- ▶ You will have access to, at least, the handouts of both the lectures and the labs
 - ▶ We are analysing the feasibility of allowing you to access to your SVN repository, but we did not reach a conclusion yet
- ▶ Grading will be fully automatic
 - ▶ We will not consider criteria that we are unable to automate
- ▶ You must have a minimum score of 8 valores to pass in LCOM
- ▶ You can repeat the test during the exam period, but only if you failed because of the PT
 - ▶ Maximum score in the **repeated PT** is 8
 - ▶ This ensures that you cannot have a final grade higher than that of any student that got the same (aggregate) grade on all other components, and was not allowed to repeat the PT, because s/he had passed

Final Project Grading (1/2)

Execution: 40% + 10%

- ▶ 10% for demo in the last lab class

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

Tools: 5% (SVN) (We expect you to commit to the SVN repository at least once a week, and to log messages then)

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), sometime in March 2018
- ▶ in recent years, we have selected 3 or 4 projects per year

Final Project Milestones

Project proposal: Beginning of the 7th lab class (week starting 12th November)

- ▶ Half to one page description of the functionalities 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: Beginning of 8th lab class (week starting 19th November)

- ▶ Refinement of the proposal, identifying the modules to implement, their functionality and API.
- ▶ Should include planning of the project
- ▶ Must be rewritten in class, if the instructor does not accept it

First demo: At the last lab class of the semester.

Project submission: January 7th, 20:00

Project presentation: January: 8th to 10th

TEs Grading

Labs 3 of the 4 labs, like for other students, but with a weight of 45%

- ▶ Presentation/discussion in the week of that lab class
 - ▶ Student must get in touch with me (pfs@fe.up.pt) to arrange for an hour, at least 7 calendar days in advance
- ▶ Submission at the end of presentation/discussion

Project Similar to that of the other students, but with a weight of 45%.

- ▶ Presentation/discussion of proposal and specification in the same week as that of the other students
 - ▶ Student must get in touch with me (pfs@fe.up.pt) to arrange for a date and hour, at least 7 calendar days in advance
- ▶ Submission by the same deadline as other students
- ▶ Presentation/discussion in the same period as for other students (in January 8th through 10th).

Programming Test with a weight of 10%, like for other students

- ▶ On the same date as for the other students
- ▶ Minimum score of 8 valores for approval, with possibility of repetition under the same conditions as other students

“Época Especial”

► There is none.

Special Evaluation

IMPORTANT Students wishing to:

1. be assessed as TE's
2. use their labs/project (positive) grades from 2016/2017 or 2017/2018

must fill [this Google form](#) by the end of this week, i.e. 2018-09-22.

IMPORTANT Please note that by choosing this option, you may have to work alone in some or all grading components

- ▶ 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 zero in that lab **and** a penalty of “2 valores” in your final grade;
 - To failing the course (especially, in case of recidivism or of the project)
 - ▶ The delay in publishing the grades is no excuse.

Important Dates

Labs

Lab	First lab class	Topic	Deadline
Lab 0	21-09	Redmine, Devel. and SVN	Not graded
Lab 2	28-09	Timer	13-10 @ 20:00
Lab 3	15-10	Keyboard	27-10 @ 20:00
Lab 4	29-10	Mouse	10-11 @ 20:00
Lab 5	12-11	Video (graphics)	24-11 @ 20:00

- ▶ Friday sections will have a holiday on the 5th of October
 - ▶ To make up for that holiday, lab classes will start this Friday

Programming Test

- ▶ 05-12

Project

What	Week	Comments
Proposal	12-11	7th lab class
Specification	19-11	8th lab class
First demo	17-12	Last lab class
Submission	2019-01-07 @ 20:00	To be confirmed
Presentation	9, 10 and 11 January, 2019 (afternoon)	To be confirmed

Lab Sections

- ▶ I've instructed the secretariat not to accept more than 24 students per section
 - ▶ ensure you get help from staff, if you need it
 - ▶ each group should have only 2 students
- ▶ Any changes will have to be done by permutation among sections
 - ▶ We have created a forum in LCOM's Moodle explicitly with that purpose
- ▶ **In the lab classes, students will be allowed to join only groups in their sections**

Announcements

Lectures

- ▶ Start 10 minutes after the hour, e.g. 17:40 on Mondays
 - ▶ Actually, this is a FEUP's rule, and therefore applies also to lab classes

Labs

- ▶ Start this Friday, i.e. September 21st
 - ▶ All sections will have 12 lab classes
 - ▶ The sections with lab classes on Friday morning will not have any additional lecture before the first lab class. Before that lab class, please read;
 - ▶ next lecture's slides
 - ▶ Lab0's handout

Acknowledgments

- ▶ Prof. António Miguel Pimenta Monteiro (who designed the course)
- ▶ Prof. João Cardoso (not the same person as TC's lecturer) (who perfected it)
- ▶ Prof. Pedro Silva, who made possible all the changes from last year:
 - ▶ 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
 - ▶ SVN
 - ▶ doxygen
 - ▶ ar
- ▶ Redmine

Advice

For Lecture Preparation

- ▶ Read the material before the lecture:
 - ▶ In each lecture before a lab assignment, I'll present:
 - ▶ The concepts and the information required to complete that lab class
 - ▶ Provide hints to address the key issues of the lab
 - ▶ The lecture slides will be available at least the day before at: <http://web.fe.up.pt/~pfs/aulas/lcom2018/>
 - ▶ The handout of every graded lab will be also available by Thursday of the previous week via the same [URL](#)

so that you can:

- ▶ Understand better the lecture
 - ▶ Participate more actively in the lecture
 - ▶ Get your questions answered before the lab class
- ▶ If I'm late and you cannot wait, check last year's material available at <http://web.fe.up.pt/~pfs/aulas/lcom2017/>