



# Sistemas de Operação / Fundamentos de Sistemas Operativos

## Course Overview

**Artur Pereira** <artur@ua.pt>

DETI / Universidade de Aveiro

## Outline

- ① Objectives and outcomes
- ② Prerequisites
- ③ Course contents
- ④ Bibliography
- ⑤ Practical classes schedule
- ⑥ Assessment

# Objectives and outcomes

- **Objectives**

- To present the most important concepts about the internal organization of present day operating systems
- To introduce concurrent programming and the core mechanisms for interprocess communication and synchronization
- To acquaint students with internal organization of Unix/Linux

- **Competencies to be acquired**

- To gain a good understanding of how multiprogramming works and of the general organization of present day operating systems
- To develop skills for the project and implementation of simple concurrent applications
- To be able to carry out productive work as a member of a team that develops system programming software

# Prerequisites

- **At the computer architecture level:**

- Basic notions on computer architecture
- Basic notions on communication protocols with input-output devices (pooled I/O, interrupt driven I/O and DMA based I/O)

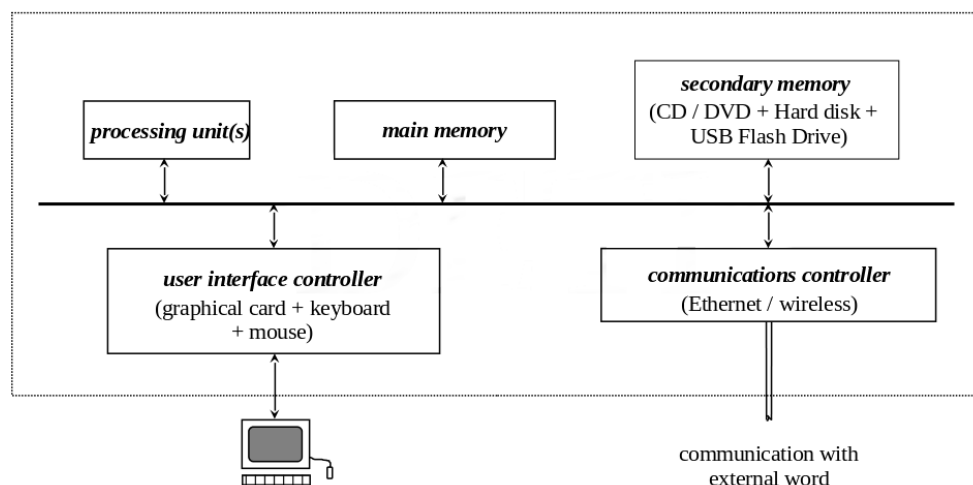
- **At the programming data structure level:**

- Programming skills in C/C++ language at a fair to good level
- Operational and conceptual knowledge of the most common static and dynamic data structures used to build different types of memory (RAMs, stacks, FIFOs and associative memories)

# Course contents

## Computational system

- Simple view of a computational system:



# Course contents

## Summary

- **Theoretical topics:**
  - Introductory concepts
  - Processor management in multiprogramming
  - Interprocess communication and synchronization
  - Memory management
  - Input / Output
  - File systems
  - Protection and Security (some introductory notions, if possible)
- **Practical and Lab topics:**
  - Concurrent programming, involving inter-process/thread communication and synchronization
  - Processor scheduling and memory management project

# Bibliography

- **Support bibliography:**

- Operating Systems: Internals and Design Principles, W. Stallings, Prentice-Hall International Editions, 7th Ed, 2012
- Operating Systems Concepts, A. Silberschatz, P. Galvin and G. Gagne, John Wiley & Sons, 9th Ed, 2013
- Modern Operating Systems, A. Tanenbaum and H. Bos, Pearson Education Limited, 4th Ed, 2015
- Sistemas Operativos, J. Marques, C. Ribeiro, L. Veiga, P. Ferreira and R. Rodrigues, FCA, 2012
- *Lecture Slides*

- 
- The lecture slides are not enough for a robust understanding of the course topics!

# Practical classes

## Schedule

- **General schedule:**

- C/C++ programming - 1 session
- Inter-process communication and synchronization (IPC) - 6 sessions
- Bash scripting - 1 session
- Support for the development of the practical group project - 5/6 sessions

- **IPC and concurrent programming:**

- Exercise on concurrent programming, based on processes and signals
- Exercise on concurrent programming, based on processes, shared memory and semaphores
- Exercise on concurrent programming, based on threads, mutexes and condition variables
- Training exercise for the practical exam

- **Group project:**

- Development of a processor scheduling and memory management simulation application

# Assessment

## General rules

- **2 components:**
  - Theoretical component: 45%, with a minimum of 7.0
  - Practical component: 55%, with a minimum of 8.0
- All intermediate grades are rounded to **one decimal place**
- **Theoretical component with 1 element:**
  - Written exam, at the exam periods
- **Practical component with 2 elements:**
  - Practical exam on concurrent programming: 25%
  - Practical group project (may include a defense): 30%
  - Marks above 17 may required some extra work
- **Repeating students:**
  - Can inherit, but ...

# Assessment

## Appeal and special exam periods

- In the appeal and special exam periods, the assessment elements are exactly the same
- The following **inheritance rules** apply:
  - The grade of the theoretical exam can be inherited from a previous exam period
    - but, if repeated, the previous grade expires
  - The grade of the practical exam can be inherited from a previous exam period
    - but, if repeated, the previous grade expires
  - The grade of the practical group project can be inherited from a previous exam period
    - repeating the practical group project involves a new project, not improvements in the former

# Assessment

## Inheritance rules for repeating students

- **By default:**
  - Grades obtained in previous years are not inherited directly
- **However, grades for assessment elements of this academic year can be obtained from previous grades based on the following rules:**
  - Theoretical exam: 100% of the grade obtained in the previous one
  - Practical exam: 100% of the grade obtained in the previous one
  - Practical group project (TP):  $\min(0.9 * TP, 14.0)$
- **Deadline:**
  - September 22<sup>th</sup>, 2023