



Sistemas de Operação / Operating Systems

Course overview

António Rui Borges / Artur Pereira

Objectives and learning outcomes

- ♦ Objectives

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

- ♦ Outcomes

- ♦ 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 the implementation of simple concurrent applications
- ♦ to be able to carry out productive work as a member of a team that develops system programming software

Pre-requisites

- **Computer architecture:** basic notions on computer architecture and on communication protocols with input-output devices (pooled I/O, interrupt driven I/O and DMA based I/O)
- **Programming:** programming skills in C/C++ Language at a fair to good level
- **Data structures:** operational 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

- *Introductory concepts*
- *Processor management in multiprogramming*
- *Interprocess communication*
- *Memory management*
- *Input / Output*
- *File systems*
- *Protection and Security (only introductory notions)*

Support Bibliography

- *Modern Operating Systems*, A.S. Tanenbaum, Prentice-Hall International
- *Operating Systems*, W. Stallings, Prentice-Hall International
- *Operating Systems Concepts*, Silberschatz, Galvin e Gagne, John Wiley & Sons
- *Sistemas Operativos*, Alves Marques, ..., Rodrigo Rodrigues, FCA
- *Lecture Slides*

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

Practical classes

- ♦ Classes
 - ♦ Bash scripting - 1 session
 - ♦ File system project - 6 sessions
 - ♦ Inter process communications - 3 sessions
 - ♦ Concurrent programming project - 1 to 4 sessions
- ♦ File system project
 - ♦ Implementation of a file system, including its integration into the Linux operating system
- ♦ Concurrent programming project
 - ♦ Design and implementation of a concurrent solution for a given problem

Assessment

- ♦ 2 components
 - ♦ exam assessment: 50%, with a minimum of 7.5
 - ♦ lab and project assessment: 50%, with a minimum of 8.5
- ♦ intermediate marks are rounded to one decimal place
- ♦ Lab and project assessments
 - ♦ 3 components:
 - ♦ work assignment 1 (file system project): 40%
 - ♦ midterm quiz (on file system project): 30%
 - ♦ work assignment 2 (concurrent programming project): 30%
- ♦ Marks above 17 may required some extra work
- ♦ Repeating students

Groups

Work assignment 1 – file system

- 7/8 weeks work
- Groups with 5/6 elements each
- Presentation and defense session

Work assignment 2 – concurrent programming

- 3/4 weeks work
- Groups with 2/3 elements each
- Presentation and defense session