

Operating Systems

INTRODUCTION

Giorgio Grisetti

`grisetti@diag.uniroma1.it`

Department of Computer Control and Management Engineering
Sapienza University of Rome

Contacts

Teacher:

Giorgio Grisetti

Tutors:

Lorenzo De Rebotti

Email:

- {[lastname](#)}@diag.uniroma1.it
- all emails you send us and concerning the course should have **[so]** as first string in the subject.

Teaching Material

- The primary source of information for the course is this web page

<https://sites.google.com/diag.uniroma1.it/sistemi-operativi-2023-24/>

- The material

- Slides
- Source code for practicals

Is also available at the following repository

https://gitlab.com/grisetti/sistemi_operativi_2023_24

Tools

- Linux (Ubuntu. Native install, please) (free)
- C/C++ (gcc)
- Arduino MEGA 2560 (or clones) ~11 eur
- 2 wires
(datasheet)

Exam

- Written Test(24 pts)
- An Individual Project (max 8 pts)
 - Rules for project evaluation
 - No rebase, we track the individual commits
 - Each one uses its own account
 - Code quality will be considered
 - Valgrind proof (when applicable)
 - $\text{project_mark} = \text{project_max_points} + f(\text{\#participants})$

Final mark is the sum

What will we Learn?

- Why are there operating systems?
- Tools: git, arduino
- Basics:
 - Object Oriented Programming in C
 - Hardware Overview (on Arduino)
 - Bare metal programming
 - Context Switch (bare metal and with ucontext)
- Dual Mode, System Calls
 - User Mode/Privileged Mode
 - Context switch in system call

What will we Learn?

- Processes:
 - State of a process
 - Basic Operations
 - Kernel Structures
 - Inter Process Communication (IPC)
 - Using Processes
- CPU Scheduling
 - Metrics
 - Batch: FIFO, SJF, Priorities
 - Time Sharing: Round Robin, Multiqueue

What will we Learn?

- Memory
 - Hierarchy, hardware support, metrics
 - Segmentation
 - Paging
 - Simple memory manager
- Virtual Memory
 - Hardware support
 - Metrics
 - Page replacement
 - Allocation
 - Copy-on-write

What will we Learn?

- File System Interface
 - Operations on Files and Directories
 - Permissions/Ownership
 - File System Abstraction
 - Dealing with POSIX file API
 - Memory mapped files
- File System Implementation
 - Disk
 - Basic Operations
 - Organizing disk space
 - Representing files and directories
 - Examples: FAT and UFS

What will we Learn?

- Signals
 - Why signals
 - Internals of a Signal Handler
 - Handling Signals
- Device Drivers and Linux Abstraction
- Kernel Modules