```
#include <stdlib.h>
#include <string.h>
#include <ctype.h>

#define MAXPAROLA 30
#define MAXRIGA 80

int main(int argc, char *argv[])
{
  int freq[MAXPAROLA] ; /* vettore di contatori
  delle frequenze delle lunghezze delle parole */
  char riga[MAXRIGA] ;
  int i, inizio, lunghezza ;
  FILE * f ;

  for(i=0; i<MAXPAROLA; i++)
    freq[i]=0 ;

  if(argc != 2)
  {
    fprintf(stderr, "ERRORE, serve un parametro con il nome del file\n");
    exit(1);
  }
  f = fopen(argv[1], "rt") ;
  if(f==NULL)
  {
    fprintf(stderr, "ERRORE, impossibile aprire il file %s\n", argv[1]);
    exit(1);
  }

  while( fgets( riga, MAXRIGA, f ) != NULL )
```

System and Device Programming

Course Introduction

Stefano Quer
Department of Control and Computer Engineering
Politecnico di Torino

General Information

- System and Device Programming
  - $01NYH_{OV}$, ING-INF/05
- Master of science degree
  - Computer Engineering
    - 1° year, 10 credits, **100** hours
    - All students (from A to Z)
- The course includes 2 separate parts
  - Operating system design and internal features
  - System and device programming

SDP 3nd Edition
Drastical changes (program and exam) with respect to past editions

First part

Second part

Instructors

First part

- Operating System Design
  - Gianpiero Cabodi
    - Dept. of Control and Computer Engineering
    - Phone: 011 090 - 7082
    - E-mail: gianpiero.cabodi@polito.it

  - Danilo Vendraminetto
    - Dept. of Control and Computer Engineering
    - Phone: 011 090 - 7048
    - E-mail: danilo.vendarminetto@polito.it

Instructors

Second part

- System and Device Programming
  - Stefano Quer
    - Dept. of Control and Computer Engineering
    - Phone: 011 090 - 7076
    - E-mail: stefano.quer@polito.it

  - Antonio Vetrò
    - Dept. of Control and Computer Engineering
    - Phone: 011 090 – 7052
    - E-mail: antonio.vetro@polito.it

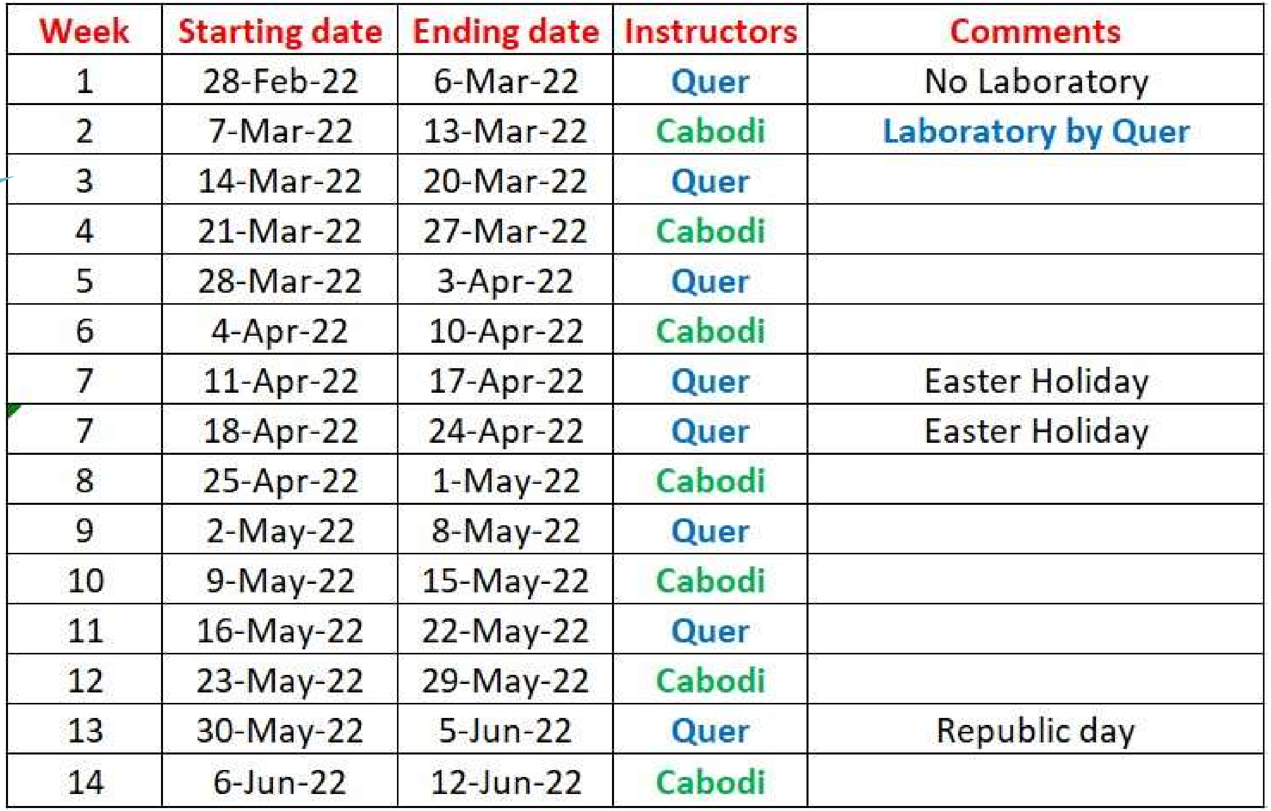
Tentative Plan

- Due to the uncertainty of the moment, please consult the “Portal Web Page” for
  - A more detailed version of the schedule
  - Changes and updates

First part (50%)

Second part (50%)

More comments will follow



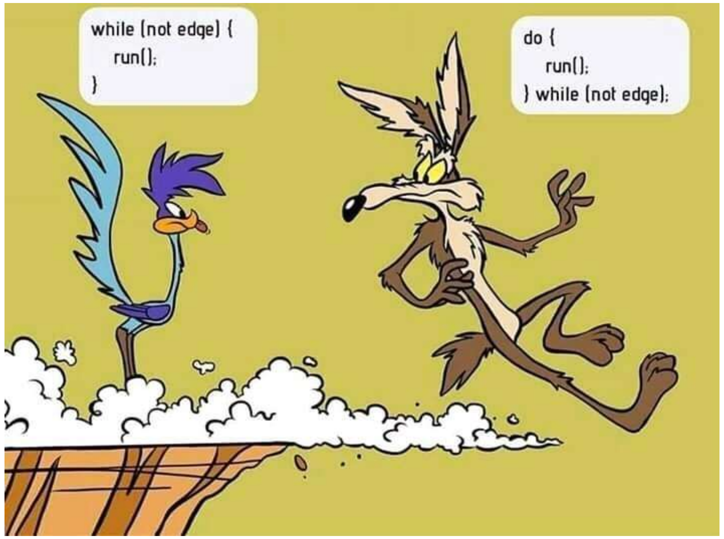
Week	Starting date	Ending date	Instructors	Comments
1	28-Feb-22	6-Mar-22	Quer	No Laboratory
2	7-Mar-22	13-Mar-22	Cabodi	Laboratory by Quer
3	14-Mar-22	20-Mar-22	Quer	
4	21-Mar-22	27-Mar-22	Cabodi	
5	28-Mar-22	3-Apr-22	Quer	
6	4-Apr-22	10-Apr-22	Cabodi	
7	11-Apr-22	17-Apr-22	Quer	Easter Holiday
7	18-Apr-22	24-Apr-22	Quer	Easter Holiday
8	25-Apr-22	1-May-22	Cabodi	
9	2-May-22	8-May-22	Quer	
10	9-May-22	15-May-22	Cabodi	
11	16-May-22	22-May-22	Quer	
12	23-May-22	29-May-22	Cabodi	
13	30-May-22	5-Jun-22	Quer	Republic day
14	6-Jun-22	12-Jun-22	Cabodi	

Background Requirements

- The course is incremental with respect to
  - The class of Operating System at the BS level
  - All programming-oriented classes at the BS level
- **Strict** prerequisites in terms of
  - Basic **operating system** features
    - UNIX/Linux Operating System
    - Processes
    - Threads
    - Synchronization paradigm and related primitives

Background Requirements

- **Programming skills** and problem solving
  - C language and Java
  - Problem solving including dynamic memory allocation, recursion and basic parallel programming with UNIX processes and POSIX threads
  - APIs under
    - The Windows OS (e.g., CodeBlocks, CLion, etc.)
    - The Unix OS (e.g., gcc, gdb, etc.)



```
while (not edge) {
   run();
}
```

```
do {
   run();
} while (not edge);
```

Learning Outcomes

- Acquire **adequate** knowledge of
  - Operating system internal features
  - System, device and parallel programming

First part (50%)

Second part (50%)

- Develop applications
  - To mimic the main features of real operating systems
  - Under different
    - Environments (UNIX-like and Windows systems)
    - Standards (C, POSIX, Windows API, C++)

Delivery Mode

- The course includes
  - Lectures
  - Practice lessons
  - Laboratories
- There is no distinction between theory lectures and practice lessons
  - Lectures include practice lessons
    - 3 (+1 extra) blocks of 1.5 hours every week
  - Laboratories
    - 2 blocks of 1.5 hours (1 single team) for (about) 12/13 weeks

Usually 3, sometimes 4 blocks

Contents

- Lectures and practice lessons
  - First part
    - Illustrate theoretical aspects of operating system design
  - Second part
    - UNIX-like (POSIX) programming
      - Revision
      - Extensions
    - Cross-platform programming in C++

More details on the first part will be given by Prof. Cabodi

Overheads are organized into units (**u**) and sections (**s**)

All sections include **parallel programming, system programming** and **problem solving**

On-line an registered (on the portal WEB page) for "off-line" students ...

Contents

More details on the first part will be given by Prof. Cabodi

- Laboratories
  - First part
    - Analysis and implementation of operating system internal features
  - Second part
    - System and device programming in
      - UNIX/Linux (revision and advanced)
      - Microsoft Visual C++ Developer's Studio
      - C++ (cross-platform)
    - Problem solving
      - From specs to code through editing, compilation, debugging, and execution of programs
      - Fix defects, fill small missing code sections, improve programs, solve classical problems

Contents

- Laboratories
  - First part
    - Analysis and implementation of operating system internal features
  - Second part
    - System and device programming in
      - UNIX/Linux (revision and advanced)
      - Microsoft Visual C++ Developer's Studio
      - C++ (cross-platform)

More details on the first part will be given by Prof. Cabodi

Each of you will need
A Linux machine with all main apps
A Windows machine with Visual Studio

Contents

- Labs are mandatory (as lectures)
  - Every week
    - You will find a new laboratory test
      - One or more questions/exercises
    - You will get the solutions of the previous lab
  - They are part of the course and there may be specific questions during the exams targeting laboratory topics

But the first one and, possibly, another selected one

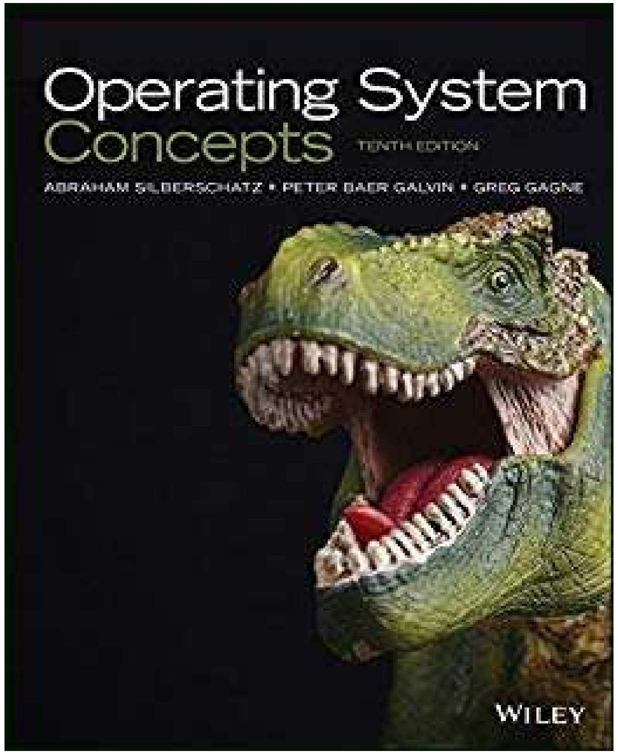
Texts, Readings, Handouts

- Material
  - We will create a **Slack channel** (with several threads) as soon as possible to reply to common questions/problems
  - Personal student's page (Politecnico portal) through **Dropbox**
    - Calendar, rules and deadlines
    - Exams bookings and exam results
    - Material used during all lectures and practice
      - Overheads
      - Laboratory exercises and solutions
      - Examination texts

Texts, Readings, Handouts

- Printed material
  - A. Silberschatz, P. B. Galvin, G. Gagne, "Operating System Concepts", 10th edition, Wiley, 2019, ISBN: 978-1119456339

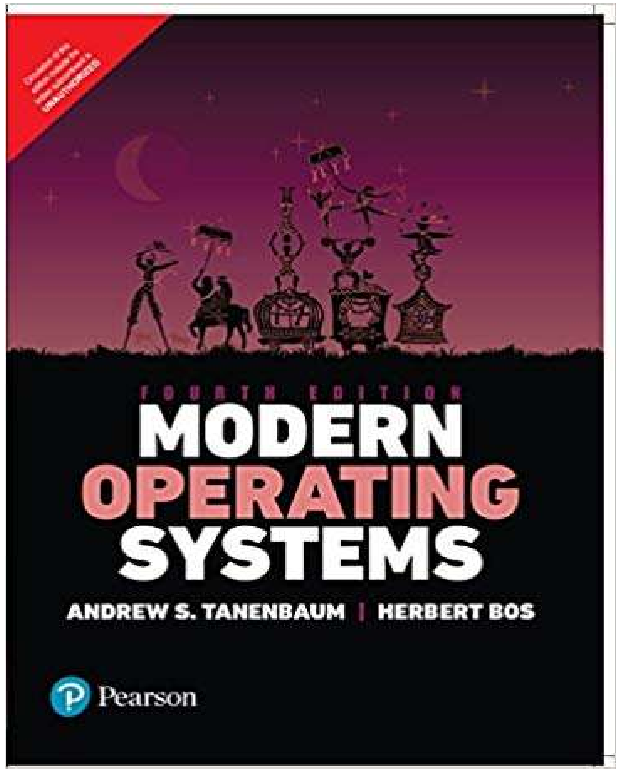
Reference book for all theoretic aspects on operating systems


Operating System Concepts
TENTH EDITION
ABRAHAM SILBERSCHATZ • PETER BAER GALVIN • GREG GAGNE
WILEY

Texts, Readings, Handouts

- Printed material
  - Andrew S. Tanenbaum, "Modern Operating Systems", 4th edition, Pearson, 2019, ISBN: 978-9332575776

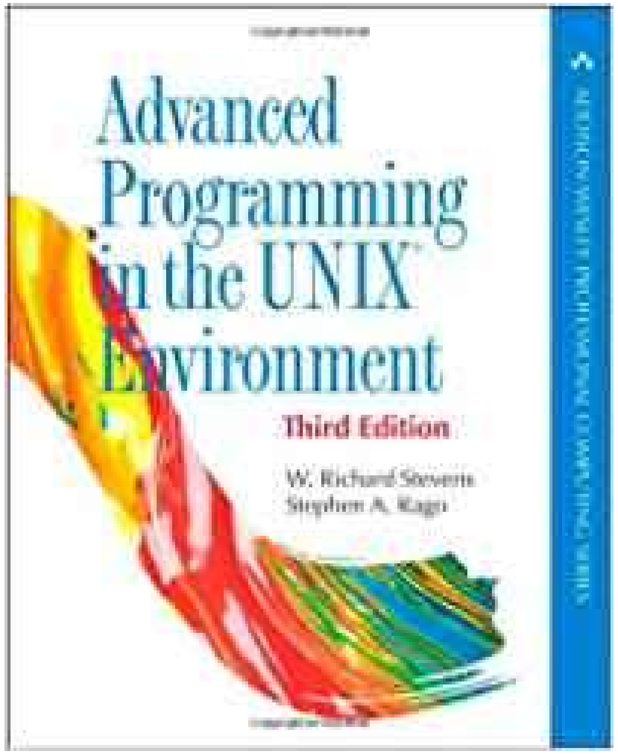
Good alternative to the previous reference



Texts, Readings, Handouts

- Printed material
  - W. R. Stevens, S. A. Rago, "Advanced programming in the UNIX Environment”, 3rd edition, Addison-Wesley, 2013, ISBN: 978-0-321-63773-4

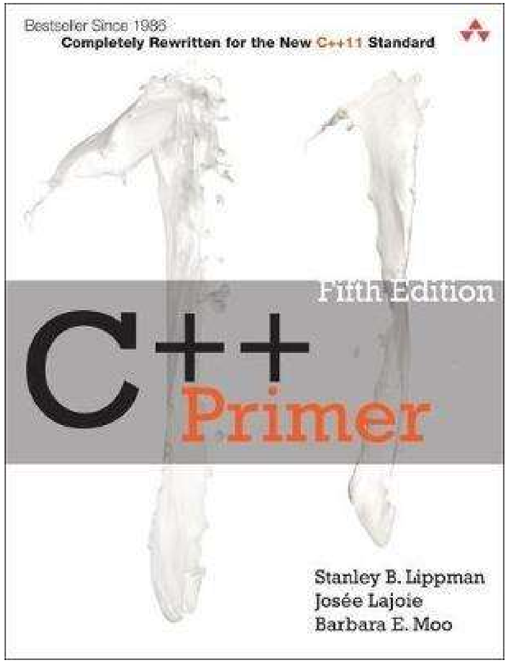
Reference book for all aspects on UNIX-like systems



Texts, Readings, Handouts

- Printed material
  - Stanley B. Lippman, Josée Lajoie, Barbara E. Moo, "C++ Primer", Addison Wesley Professional, ISBN 978-0-321-71411-4

Possible reference book for C++ programming



Texts, Readings, Handouts

- WWW
  - Additional information, clarifications, and more
  - Implementation details
    - Libraries
    - Specs
    - Data types, system calls
    - etc.

Assessment and Grading Criteria

- **Premises** for students who followed the course before 2021-2022
  - Since June 2021, the old examination format (following the Prof. Laface’s program) has **not** been available anymore
  - All candidates **must** follow the **new** program and the **new** examination rules

There are changes also with respect to the program presented in 2019-2020 and 2020-2021

Assessment and Grading Criteria

- **Premises** for students who followed the course before 2021-2022
  - Since June 2021, the old examination format (following the Prof. Laface's program) has **not** been available anymore
  - All candidates **must** follow the **new** program and the **new** examination rules
    - Old students can take the project **if and only if** they are moving to the new exam format for the first time, i.e., they did never take the exam with the new format during 2019-2020
      - **Warning**: Projects are knowledge-demanding and time-consuming go for them with great care

Assessment and Grading Criteria

- The exam consists in
  - A written test
    - To check the
      - Knowledge on theory topics
      - Problem-solving ability
  - An **optional** group project
    - To improve and verify the
      - Capacity to face large and complex realistic problems and tasks
      - Ability to work in groups with colleagues

System and Device Programming

Student Funnel

Assessment and Grading Criteria

- The written test is divided into two sessions
  - The first one is on operating system internals
  - The second one is on system and device programming
- Each session
  - Last from 60 to 120 minutes, depending on the examination text
  - Must be completed **with no** books, notes, or other printed or electronic material
  - Portable devices (e.g., cell phones) must be turned-off during the entire test

First part (50%)

Second part (50%)

Assessment and Grading Criteria

- Both sessions include questions and exercises on both theoretical and practical aspects of the relative part of the course
  - Theoretical questions may include open and closed puzzles
    - These are formulated to test the candidate's knowledge on all topics presented
  - Practical questions may include programming exercises
    - These test the ability of the candidate to write programs in different environments and programming languages (UNIX/Linux, C and C++)

Assessment and Grading Criteria

It was **13** in 2019-2020 and 2020-2021

- Evaluation
  - Each part amounts **up to** 15 points on the final mark and it has a **passing threshold** of 7 points
  - The final mark is the sum of the two evaluations
  - To pass the exam (with or without the group project)
    - A pass grade is required on both parts
      - Each mark must be larger or equal than 7
    - A pass grade is required on their sum
      - The sum must be larger or equal than 18
  - The maximum grade reachable with the written test is equal to 15+15=30/30

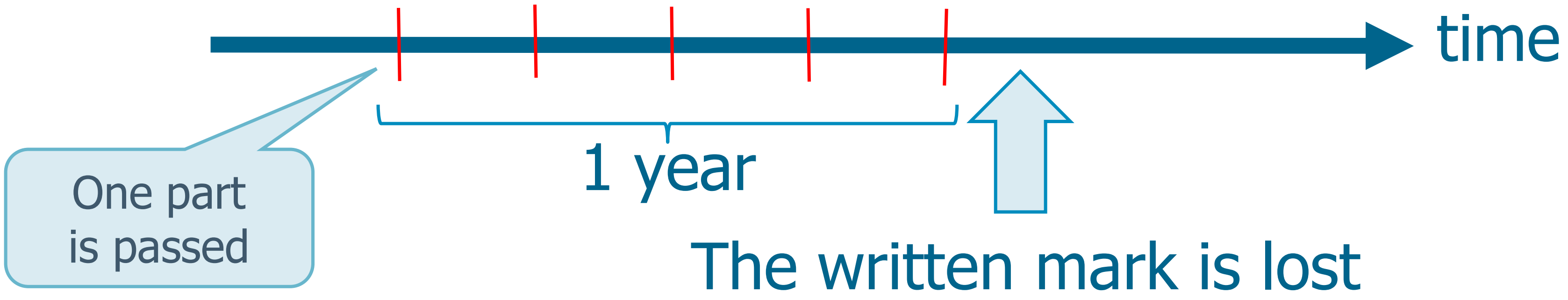
Assessment and Grading Criteria

- The two written parts
  - Can be taken during different examination sessions
    - Marks can be rejected to reach the passing threshold or improve the final sum/mark
  - These sessions must belong to a time period corresponding to at most 4 (+1) examination sessions

Extra (autumn) session

Assessment and Grading Criteria

- When the **time window** is expired the written part that has been passed but not finalized (registered with a final mark) will be automatically lost and the candidate will have to retake the corresponding written test



Assessment and Grading Criteria

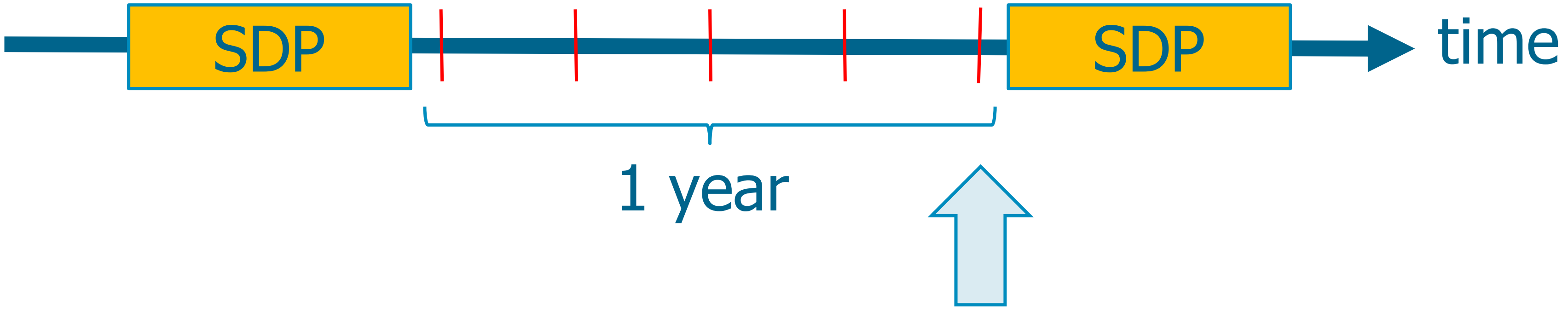
- An **optional** project may be taken to
  - Improve the candidate’s knowledge on the course’s topics
  - Eventually, amend the final mark
- In any case
  - The exam is passed **only** if the written test is passed
  - A project can be taken only **once** by each students
  - The project must be selected **before** taking any written examination tests

Assessment and Grading Criteria

- Projects
  - Are specified as soon as possible during the course but not before students can appreciate/understand them
  - May have as a subject the “operating system internals” or “system and device programming”
  - Must be selected by group of 2 or 3 candidates
  - Will be assigned to group of students on a **first-come-first-served** basis

Assessment and Grading Criteria

- Must be completed and delivered within the same academic year (i.e., in one of the 4+1 examination sessions following each course)
  - The beginning of a new course will automatically cancel all pending projects



All pending projects are lost

Assessment and Grading Criteria

- Project evaluation
  - The evaluation will follow a short presentation given by the group of candidates during an examination session following the written tests
  - To enroll for the presentation
    - The project must be uploaded on the portal web page before the written test
      - Use the "Materiale" page on the portal
    - Enroll for the written exam in that session
  - Each project can add from **-2.0** to **+6.0** marks to the final written evaluation

Projects are **not** made to improve the final mark
Projects are made **to learn more**

Assessment and Grading Criteria

  - Marks may differ for different students within the same group, depending on the effort they put into the project and on the final presentation
  - Once the final mark for a project has been obtained, there is **no** time limit to its validity
- Final evaluation
  - The written test plus the group project may amount up to 30+6 = 36 out of 30
  - Marks larger or equal to **32** will be automatically converted into 30 with honor

Assessment and Grading Criteria

- **Addendum** for potentially "weak" students
  - Students who are unable to pass the exam with the previous rules may follow a **simplified** examination path
  - This path includes an **on/off** written test
    - The test concentrates more on theory issues and it is **less** demanding on problem-solving topics and C/C++ implementations
      - No design, mainly analysis
    - It includes closed (automatically corrected) and open (manually corrected) questions
    - It lasts from 90 to 150 minutes

Assessment and Grading Criteria

- Each test
  - Includes a first part (Prof. Cabodi) and a second part (Prof. Quer)

  - The two parts must be taken during the **same** session
  - Each part includes from 5 to 20 questions
  - Each part delivers a **fail** or **pass** grade
- The candidate will pass the exam **if and only if** he/she has received a **pass** grade on both parts

To give a rough idea, following the rules of the standard written tests, each part will deliver about 15 points with a pass grade equal to about 7 points

Assessment and Grading Criteria

- Partial results in the on/off exam are lost
  - A pass result in a part (first or second) without a pass result in the other (second or first) is lost
  - The on/off exam has to be taken and passed entirely in the same session
- Students taking the on/off exam will automatically give-up (lose) their project and its evaluation (even if already obtained)

Assessment and Grading Criteria

- The simplified on/off tests can also be used by candidate who already passed either the first (Prof. Cabodi) or the second (Prof. Quer) part **but** not the other
  - When a student obtained a pass evaluation (at least 7 points) in one of the two written tests, but he/she is unable to pass the other, he/she can pass the other taking the on/off exam
  - Students with two passes (one in the standard exam and the other in the on/off exam) will pass the exam with 18/30 and **any** project (delivered or not) will be automatically lost

To summarize

- During the course we will face both theory and practice problems

To summarize

- During the course we will face both theory and practice problems
  - Theory is when you know everything but nothing works

To summarize

- During the course we will face both theory and practice problems
  - Theory is when you know everything but nothing works
  - Practice is when everything works but no one knows why

To summarize

- During the course we will face both theory and practice problems
  - Theory is when you know everything but nothing works
  - Practice is when everything works but no one knows why
  - **In this class, theory and practice will be combined: Nothing will work and no one will know why**

(possibly) Albert Einstein, 1879-1955