



Università degli studi di Parma

Dipartimento di Ingegneria e Architettura

Sistemi operativi e in tempo reale - a.a. 2023/24

---

# Sistemi operativi e in tempo reale a.a. 2023/24

## Introduzione al corso

prof. Stefano Caselli

---

# Per iniziare

---



- ❑ Benvenuti!
- ❑ Chi siamo
- ❑ Logistica
- ❑ Organizzazione
- ❑ Materiali didattici
- ❑ Esame
- ❑ Miti e fatti su questo corso



- **Prof. Stefano Caselli**  
Tel. 0521 905724 (interno 5724)  
E-mail [stefano.caselli@unipr.it](mailto:stefano.caselli@unipr.it)
- *RIMLab : Robotics and Intelligent Machines Laboratory*
- *ICT4Ag : ICT for Smart Agriculture* (Lab. Pal. 1 e Pal. 3)
- Per attività di ricerca e per info su progetti, internati, tesi:  
<http://rimlab.ce.unipr.it> e <http://www.progettopositive.it>
- Ufficio: Sede scientifica DIA, Pal. 1, piano superiore
- Ricevimento ufficiale: Mer. 14.30-16.30 (*ufficio Pal. 1*) - *per il semestre di lezione* - o su appuntamento
- Esercitazioni, tutor e assistenza: **Prof. Dario Lodi Rizzini, Ing. Gabriele Penzotti**

# E voi chi siete?

---



# Logistica del corso

---



- **Orario Lezioni:**

Lun 15:30-17:30 aula **A/2 sede scientif.**, lab/aula E Plesso Scienze

Mar 15:30-17:30 aula **1 sede did.**, lab Info base 1 - 2

Ven 10:30-12:30 aula **A/2 sede scientif.**, lab/aula E Plesso Scienze

- Lezioni prenotate anche in data **10 e 11 Giugno** per assicurare il numero minimo di ore obbligatorie (pianificate)
- Esercitazioni preferibilmente (?) Ven: comunque BYOD, anche aula
- Prima esercitazione in Lab: Venerdì 1 Marzo in aula E plesso Scienze
- In generale: date un'occhiata ad agenda studenti e alle comunicazioni volta per volta su calendario esercitazioni vs. lezioni

# Logistica del corso

---



- ❑ Sito dell'insegnamento, con materiali didattici ed informazioni: <http://elly2023.dia.unipr.it>
  - Prossimamente: chiave di registrazione iniziale
- ❑ E' obbligatorio essere registrati sul sito dell'insegnamento entro il 15 Marzo 2024
  - Comunicazioni su risultati esami, lezioni, etc., solo attraverso il sito e/o in aula
- ❑ Corso in presenza: frequenza fortemente stimolata, frequenza delle esercitazioni sostanzialmente obbligatoria

# Programma di massima

---




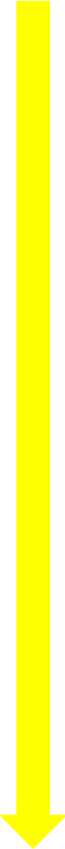
- ❑ Integrazioni sui sistemi operativi (gestione della memoria, I/O e file, memorie di massa, scheduling CPU, sicurezza)
- ❑ Sistemi in tempo reale
- ❑ Teoria della programmazione concorrente in ambiente globale e locale
- ❑ *Allineamento* delle competenze di programmazione di sistema in ambiente C/C++/UNIX
- ❑ *Introduzione* a strumenti e linguaggi per la programmazione concorrente (Pthreads, Java)
- ❑ *Programmazione* di sistema in ambiente distribuito
- ❑ Approfondimenti su aspetti specifici



# Track parallele per lezioni ed esercitazioni

---

- 
- ❑ Introduz. ai sistemi operativi
  - ❑ Teoria dei sistemi in tempo reale
  - ❑ Meccanismi di supporto alla concorrenza
  - ❑ Programmazione concorrente

- 
- ❑ Programmazione multiprocesso
  - ❑ Programmazione multi-thread
  - ❑ Programmazione in tempo reale





- ❑ Obiettivo dell'esame è premiare chi *partecipa attivamente al corso* e svolge le *prove in corso d'anno*
- ❑ Chi non svolge o non supera le attività in corso d'anno dovrà sostenerle successivamente nell'arco degli appelli ufficiali
- ❑ I contributi alla valutazione:

## Prove parziali in corso d'anno

- A) Prova lab programmazione di sistema 20% (5 aprile)
- B) Prova scritta sistemi in tempo reale 20% (fine aprile)
- C) Applicazione multithread RT 20% (fine maggio)

- D) Prova scritta/pratica di concorrenza 30% (→ appelli uff.)
- E) Orale (breve) 10% (dopo la prova scritta)

←  
Esame!



- ❑ A), B), C) in corso d'anno! Restano validi fino al 28/2/2025
- ❑ D) -> E) nell'ordine concludono l'esame, negli appelli ufficiali e circa una settimana dopo
- ❑ Possibile integrazione di elementi di valutazione legati alla frequenza attiva alle lezioni (solo primo appello)
- ❑ La prova di programmazione A) viene ripetuta - se necessario - una volta per sessione (inizio luglio, inizio settembre, inizio febbraio)
- ❑ La realizzazione di un'applicazione multithread in tempo reale, se non effettuata durante il corso, deve essere svolta durante uno degli appelli ufficiali, richiedendolo con un anticipo di 3 settimane
- ❑ Consigliato: **sostenere la prova pratica C) in corso d'anno**



- ❑ Nelle date di appello ufficiali è possibile svolgere anche la prova scritta di «Sistemi in tempo reale» (da richiedere preventivamente per email al docente)
- ❑ Iscrizione alle prove d'esame obbligatoria per qualsiasi tipo di prova:
  - Chi non è iscritto non sostiene l'esame
  - Modalità Esse3
- ❑ Domande sull'esame?

# Esame -- LEGGERE ATTENTAMENTE

---



- E' caldamente consigliato svolgere la prova di programmazione e il progetto multithread real-time (threads C++ e POSIX) *durante* corso; gli eventuali risultati positivi valgono fino all'inizio della successiva edizione del corso
- Per chi non svolge o non supera tali prove durante il periodo di lezione verranno proposte *alcune* date successive
- In subordine, le prove pratiche devono essere “richieste” almeno due-tre settimane prima dell'appello ufficiale
- Tutti i risultati parziali acquisiti (inclusa la prova di programmazione) scadono il 28 febbraio 2025. Se l'esame non è stato superato e verbalizzato entro quella data tutte le prove parziali sono perse
- E' obbligatorio sostenere la prova orale nella data proposta immediatamente successiva alla prova scritta superata; in caso contrario la prova scritta deve essere ripetuta
- Se la prova orale non viene superata deve essere ripetuta anche la prova scritta



# Laboratorio ed esercitazioni

---

- ❑ Aula E c/o plesso Aule delle Scienze il Venerdì e il Lunedì;  
LabInf 1-2 c/o sede didattica il Martedì, meglio con PC personale
  - con intensità variabile
  - notare le **eccezioni** (in agenda) sui laboratori disponibili
- ❑ In linea di massima, calendario lab:
  - marzo: richiami di *programmazione multiprocesso*
  - **Esercitazioni: da Venerdì 1 Marzo**
  - **5 aprile 2024: prova pratica in itinere di programmazione multiprocesso**
  - fine aprile - maggio: *programmazione multithread e real-time*
  - sviluppo di una applicazione real-time con thread C++/POSIX
  - frequenza *caldamente consigliata*
- ❑ Prove pratiche obbligatorie per superare l'esame



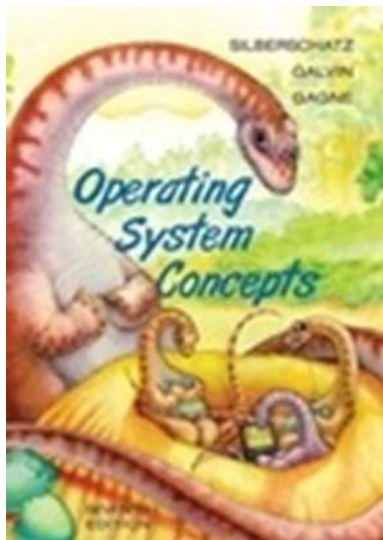
# Libri di testo

---

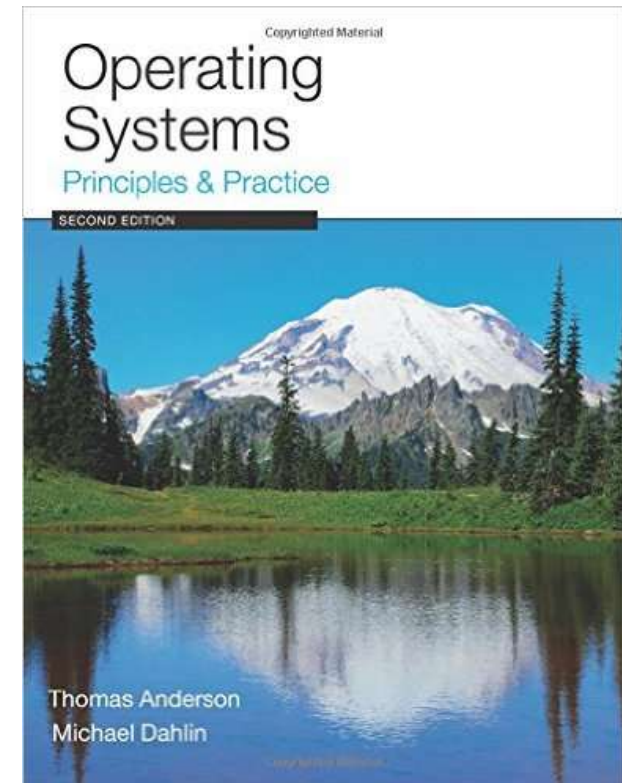
- ❑ Non c'è un libro di testo obbligatorio, ci sono **buoni testi** da cui prendo parte dei contenuti delle lezioni e che consiglio:
  - Libri di Sistemi operativi
  - Libri di Programmazione concorrente
  - Libri di Sistemi in tempo reale
- ❑ Libri non indispensabili per l'esame ma consigliati!
- ❑ Disponibili in biblioteca per consultazione. Ad esempio: testo generalista + testo di programmazione concorrente

# Testi consigliati di *Sistemi Operativi*

- ❑ T. Anderson, M. Dahlin, «Operating Systems - Principles & Practice», 2nd edition, Recursive Books, 2014.
- ❑ A. Silberschatz, P.B. Galvin, “Sistemi operativi - Concetti ed esempi”, ok da settima a decima ediz., Pearson, 2007-2020.



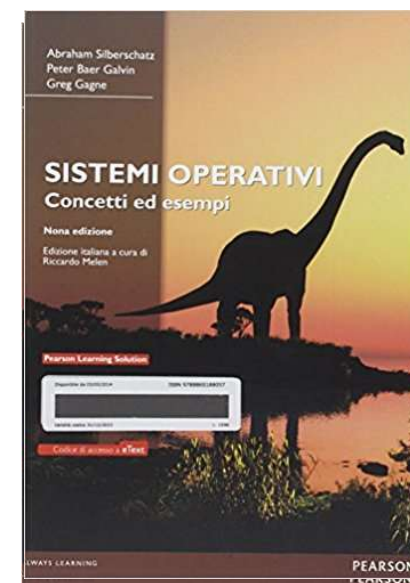
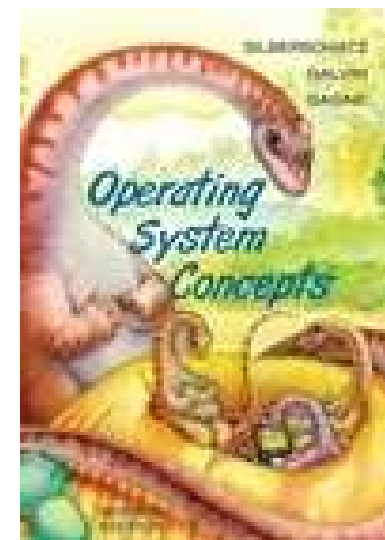
SisOp&RT - IntroCorso



# Un testo canonico di *Sistemi Operativi* con una lunga storia



- ❑ A. Silberschatz, P.B. Galvin, “Sistemi operativi - Concetti ed esempi”, ok da settima a decima ediz., Pearson Education Italia, 2006/2020.
- ❑ Conosciuto nel mondo come “the dinosaur book”
- ❑ In circolazione (e in biblioteca) almeno dalla terza edizione
- ❑ Autore ricorrente: **Avi Silbershatz**
- ❑ OK Silbershatz in varie edizioni (ita/eng)
- ❑ A. Silberschatz, P.B. Galvin, G. Gagne, “Operating System Concepts”, oppure “Operating System Concepts with Java”, (from 6th to 9th ed.), Wiley, 2004-2012.
- ❑ Alcune traduzioni in italiano delle vecchie edizioni sono terribili, evitare

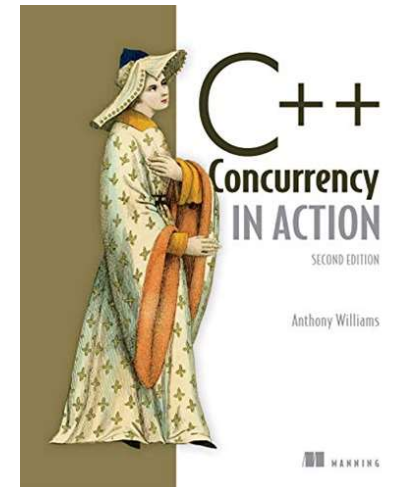




# Testi di *Programmazione Concorrente*



- ❑ P. Ancilotti, M. Boari, “Programmazione concorrente e distribuita”, McGraw-Hill, 2007.
- ❑ G.R. Andrews, “Foundations of Multithreaded, Parallel, and Distributed Programming,” Addison-Wesley, 2000.
- ❑ Per le esercitazioni:
- ❑ A. Williams, «C++ concurrency in action», Mannings, 2019. (2nd edition per standard C++17; 1st edition copre C++11).
- ❑ Tutti i testi indicati sono opzionali!



# Testi per *Sistemi in tempo reale*

---



- ❑ Testo (x teoria):
- ❑ J.W.S. Liu, *Real-Time Systems*, Prentice-Hall, 2000.
- ❑ G.C. Buttazzo, *Hard Real-Time Computing Systems*, Third Ed., Springer, 2011.
- ❑ Alternative & approfondimenti:
  - ❑ G.C. Buttazzo, *Sistemi in tempo reale*, Pitagora editrice, (prima ediz. 1995, seconda ediz. 2001).
  - ❑ D. Butenhof, *Programming with POSIX Threads*, Addison-Wesley, 1997.

# Altri testi di *Sistemi Operativi e Programmazione Concorrente* di ampia diffusione

---



- ❑ R.H. Arpaci-Dusseau, A.C. Arpaci-Dusseau, «Operating Systems - Three Easy Pieces», <http://www.ostep.org>
- ❑ A.S. Tanenbaum, “I moderni sistemi operativi,” seconda edizione, Jackson Libri, 2002.
- ❑ H.M. Dietel, P.J. Dietel, D.R. Choffnes, “Sistemi operativi”, terza edizione, Pearson Education Italia, 2005.
- ❑ P. Ancilotti, M. Boari, A. Ciampolini, G. Lipari, “Sistemi operativi,” McGraw-Hill, 2004 (|| nuova ediz. 2008).
- ❑ M. Herlihy, N. Shavit, “The Art of Multiprocessor Programming”, Morgan Kauffmann, 2008.
- ❑ S.J. Hartley, “Concurrent Programming - The Java Programming Language,” Oxford University Press, 1998.

# Materiali didattici e risorse

---



- ❑ Copie delle diapositive:
  - sempre in parte da scrivere, migliorabili o in miglioramento; talvolta disponibili materiali a.a. precedenti, ma con variazioni possibili
  - Accessibili man mano, tramite il sito web del corso - last minute
  - Non sostituiscono la frequenza attenta in aula o un buon testo
- ❑ E' obbligatorio essere registrati sul sito entro il 15 marzo 2024
  - Il sito è una risorsa indispensabile per interazioni da docente a studenti
  - E' consigliato l'uso strumenti di condivisione per tenervi in contatto tra voi

# Possibili problemi sul corso e altro

---



- ❑ Orario lezioni → già migliorato, non credo modificabile
- ❑ Aule → soluzione attuale è un compromesso
- ❑ Piani di studio LM
- ❑ Manifesto studi a.a. 2023/24 e 2024/25: chiedete info!
  
- ❑ Iscritti, futuri iscritti, esami singoli, etc.
- ❑ **Ammessi alle prove solo iscritti al corso di laurea o iscritti ad esame singolo (deve risultare in Esse3)**
  
- ❑ Modalità esame: devono essere chiare per tutti (valutazione docente)

# Una opportunità

---



- ❑ Insegnamento a scelta Marzo-Giugno 2024, tenuto da Visiting professor:
  - ❑ ***Embedded systems***
  - ❑ ***prof. Xenofon Fafoutis, Danish Technical University (DTU)***
- ❑ <https://corsi.unipr.it/it/ugov/degrecourse/256113>
- ❑ (Insegnamento a scelta libera, TAF D)



- ❑ The course will provide basic, advance, as well as practical knowledge on the design and implementation of *distributed embedded systems*
- ❑ Following an introduction on the recent developments of intelligent transportation systems and smart vehicles, the course will *focus on system-level building blocks for developing distributed embedded systems including software-hardware interaction, sensors and actuators, intra-board communication, wired/wireless embedded networking, embedded software development, performance evaluation and testing, as well as on-board data processing and embedded machine learning*