

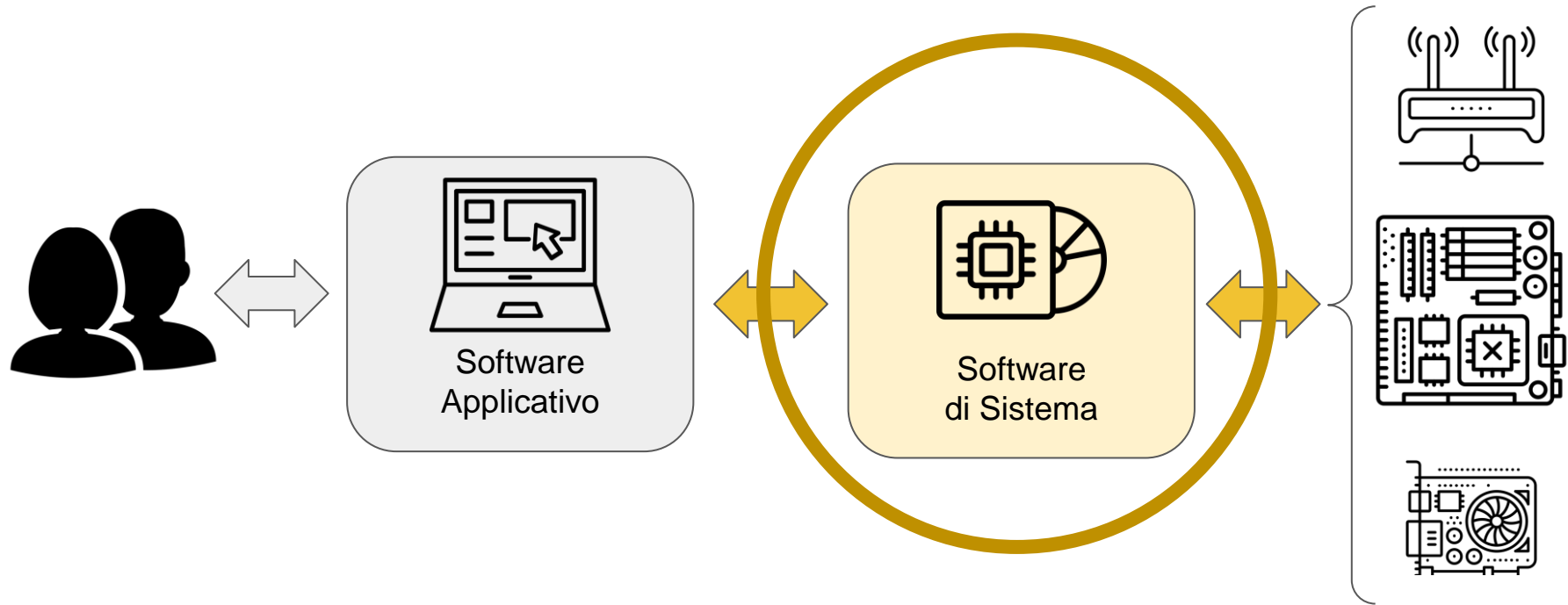
# Programmazione di Sistema

2023-24

# Programmazione di sistema

- Un solo voto sul libretto, due percorsi didattici (attinenti l'uno con l'altro, ma distinti)
  - OS Internals
  - API Programming
- I due percorsi si alterneranno ogni settimana
  - Ciascuno dei due percorsi consiste di lezioni frontali e esercitazioni
- Due prove di esame distinte
  - Possono essere svolte in tempi diversi
  - Occorre superarle entrambe per avere il voto registrato
  - Dopo 4 appelli, una prova scade se non si è superata anche l'altra
- Possibilità di svolgere un progetto in gruppo, a scelta, tra una rosa di proposte
  - La cui valutazione può valere fino a 6 punti
  - Il progetto non sostituisce una delle due prove di esame né compensa un voto insufficiente
  - I progetti saranno annunciati più avanti

# Cos'è la programmazione di sistema?



# Cos'è la programmazione di sistema?

- La programmazione di sistema fornisce le fondamenta che supportano la programmazione applicativa
  - Creando il software necessario a permettere ai tanti componenti dell'ecosistema di elaborazione (CPU, memoria, periferiche, rete, ...) di operare come un organismo integrato
- La programmazione di sistema è orientata ai contesti dove le risorse sono limitate
  - Richiede la capacità di organizzare il codice tenendo conto che ciascun byte e ciascun ciclo di CPU è rilevante
- La quantità di codice di sistema coinvolto nel permettere ad una semplice applicazione di essere eseguita è sorprendente
  - Sistema operativo, driver delle periferiche, compilatore, linker, loader, librerie di vario tipo...

# Ambiti della programmazione di sistema

Stack di rete  
Sistemi operativi  
Elaborazione dei media  
Simulazioni scientifiche Renderizzazione del testo  
Dispositivi embedded  
Gestione della memoria  
Crittografia  
Linguaggi ad alto livello  
File System  
Giochi  
Macchine virtuali  
Device Drivers  
DBMS

# Linguaggi per la programmazione di sistema

- Scrivere codice ad alte prestazioni richiede l'uso di un linguaggio che permetta al programmatore di controllare come siano usate le risorse di memoria, processore, I/O, ...
  - Questo esclude, a priori, tutti quei linguaggi che adottano un sistema di gestione automatico della memoria (come Java, JavaScript, Python, C#, Go, ...)
  - Riducendo le opzioni possibili sostanzialmente all'uso di C, C++ e Rust (oltre a Swift, in ambiente Apple)
- Rust combina tale livello di controllo con un sistema di tipi alquanto sofisticato, in grado di identificare, in fase di compilazione, un'ampia gamma di errori, dall'uso incorretto dei puntatori alle interferenze tra thread
  - Garantendo che se un programma riesce ad essere compilato, non contiene comportamenti non definiti (che non vuol dire che funzionerà certamente, ma che avrà un comportamento riproducibile)

# Abilità richieste

- Coding
  - Algoritmi (inserimento, ricerca, ordinamento, ...)
  - Strutture dati (stringhe, liste, insiemi, mappe, ...)
  - Lettura e scrittura di file (conversioni di formato, segmentazione del testo, codifica di dati binari, ...)
  - Conoscenza approfondita del linguaggio C
- Architetture di elaborazione e elettronica digitale
  - CPU, memoria, bus di comunicazione, periferiche, interrupt, DMA ...
  - Nozioni di base sui sistemi operativi
- Capacità di analisi di un problema e pensiero critico
  - Definizione e comprensione dei requisiti
  - Schematizzazione tramite diagrammi (entità/relazioni, sequenze di eventi, gerarchie di vario tipo, ...)
- Attitudine alla risoluzione dei problemi
  - Elaborazione e affinamento dei propri modelli mentali

# Abilità da sviluppare

- Motivazione

- Il corso è impegnativo per la vastità dei concetti che esplora così come per la loro intrinseca complessità
- In base alle inclinazioni e sensibilità personali, può anche apparire arido
- Può essere opportuno mettere a fuoco molto bene le motivazioni che hanno portato a scegliere il percorso magistrale di Ingegneria Informatica per trovare le energie necessarie ad impegnarsi nello studio, dando ad esso un senso nella prospettiva del proprio progetto di vita professionale

- Organizzazione e gestione del tempo

- La presenza di due percorsi paralleli richiede un'attenta pianificazione dei propri tempi di studio, per non lasciare indietro qualche concetto che poi diventa molto difficile recuperare
- Parallelamente, occorre trovare il giusto ritmo, sapendo concedersi il riposo necessario, per evitare di saturare le proprie capacità mentali



# API Programming - Obiettivi attesi

- Conoscenza del linguaggio Rust e suo utilizzo per la realizzazione di applicativi di media complessità
- Comprensione approfondita dei meccanismi soggiacenti l'esecuzione di un programma nel contesto di un sistema operativo
- Comprensione delle problematiche introdotte dalla programmazione concorrente e capacità di realizzare algoritmi in grado di operare correttamente in contesti multi-thread e asincroni
- (per chi svolge il progetto) Capacità di lavorare in gruppo, organizzando il lavoro così da coinvolgere tutti e provando a gestire le problematiche connesse, sia sul piano tecnico che su quello umano