

MODULO 1 - I PROCESSI SEQUENZIALI E PARALLELI

- Ripasso: processi, diagramma degli stati, politiche di scheduling
- Il PCB
- Rappresentazione del processo in memoria
- Il context switch, l'overhead
- S.O.: kernel mode, user mode
- Le interruzioni
- System call – API
- Processi indipendenti, cooperanti, concorrenti

MODULO 2 - INTRODUZIONE ALLA PROGETTAZIONE SOFTWARE

- Il linguaggio UML
- Diagrammi dei casi d'uso UML
 - o attori e gerarchie tra essi
 - o relazioni di inclusione, estensione e generalizzazione
- Diagrammi di sequenza UML
 - o Tipo di messaggio (sincrono, asincrono e di ritorno)
 - o Frammento combinato
 - o La ricorsione

Attività di laboratorio: UTILIZZO SOFTWARE DI MODELLAZIONE UML

- Realizzazione degli Use Case Diagram tramite il software StarUML
- Realizzazione dei Sequence Diagram tramite il software StarUML

MODULO 3 - RISORSE E CONDIVISIONE

- La risorsa; le classi di risorse, molteplicità
- Classificazione
- Il grafo di Holt
- Riducibilità grafo di Holt

MODULO 4 - ELABORAZIONE SEQUENZIALE E CONCORRENTE

- Parallelismo reale e virtuale (interliving, overlapping)
- Processi sequenziali o paralleli
- Descrizione della concorrenza:
 - o Il grafo delle precedenze

MODULO 5 - I THREAD

- Definizione; esempi di utilizzo di thread
- Differenza tra Thread e Processo
- TCB
- Soluzioni adottate nei SO: single threading vs multithreading
- Realizzazione di thread: livello utente, kernel, misto
- Stati di un thread

Attività di laboratorio: GESTIONE DEI THREADS IN JAVA

- Creazione: estensione della classe Thread
- Passaggio dei parametri ad un thread
- Priorità
- Sospensione – sincronizzazione esplicita: sleep(), join()
- Condivisione dati: attributi static; classe “involucro” Buffer
- Gestione sincronizzazione: wait(); notify(); notifyAll()

MODULO 6 - COMUNICAZIONE E SINCRONIZZAZIONE TRA PROCESSI

- Comunicazione:
 - Tipi di interazione tra processi: cooperazione, competizione
 - Modello a memoria comune (ambiente globale)
 - Modello a scambio di messaggi (ambiente locale): send(), receive(); il rendez-vous
- Sincronizzazione:
 - Errori dipendenti dal tempo nei programmi concorrenti
 - Condizione di Bernstein
 - Problema dell'interferenza
 - Sezione critica
 - La mutua esclusione
- I semafori:
 - Semafori a basso livello, spinlock
 - Semafori di Dijkstra
- Applicazione dei semafori:
 - per la mutua esclusione
 - per la sincronizzazione
- Problema dello stallo
 - Definizione
 - Riconoscimento: condizioni di Coffman
 - Individuazione dello stallo mediante grafo di Holt
 - Tecniche per affrontare lo stallo
 - Il problema dei “Filosofi a Cena”
- Problema produttore-consumatore soluzione con semafori
- I monitor

Attività di laboratorio: LA SINCRONIZZAZIONE IN JAVA

- Gestione sincronizzazione: qualificatore synchronized;
- Applicazioni dei semafori in Java:
 - uso di semafori per la sincronizzazione tra thread
 - un produttore e un consumatore
 - un produttore e più consumatori, buffer circolare
 - più produttori e più consumatori, buffer circolare
- Applicazioni dei monitor in Java

EDUCAZIONE CIVICA:

Introduzione all'uso del Metaverso
Piattaforma Spatial.io

Testo: P. Camagni, R. Nikolassy “Tecnologie e progettazione di sistemi informatici e di telecomunicazione, vol 2”, Hoepli

Viareggio, 5 giugno 2025

Gli insegnanti:

Gli alunni

Elisabetta Giannaccini _____
Claudia Bertini _____