

**PROGRAMMAZIONE DI TPSIT**  
**TECNOLOGIE E PROGETTAZIONE DI SISTEMI INFORMATICI E DI**  
**TELECOMUNICAZIONI**  
*Classe 4*

**PROGRAMMA TEORIA**

**Modulo I: Processi**

Obiettivi di apprendimento:

Conoscere i modelli di elaborazione dei processi ed il ciclo di vita dei processi.

Contenuti:

Il modello a processi

Scheduling e multiprogrammazione

Processi cooperanti e processi in competizione

Stato di un processo e diagramma degli stati (new, ready, running, waiting, terminated)

Cosa è il PID

Il context switching

Il Process Control Block (PCB)

Creazione, sospensione e terminazione dei processi nei sistemi \*NIX like (le funzioni: fork, exit, wait)

**Modulo II: Risorse e condivisione**

Obiettivi di apprendimento:

Acquisire il concetto di risorsa condivisa. Apprendere l'utilizzo del grafo di Holt per descrivere processi e risorse.

Contenuti:

Definizione di risorsa

Classi e istanze di risorse

Condivisione e gestione

Classificazioni in base al tipo di richiesta, alla modalità di assegnazione, alla tipologia delle risorse

Grafo di Holt o grafo di allocazione delle risorse

rappresentazione delle risorse e dei processi

significato del verso degli archi che connettono processi e risorse

grafi riducibili

**Modulo III: Elaborazione sequenziale e concorrente**

Obiettivi di apprendimento:

Imparare il concetto di programmazione concorrente, realizzare il grafo delle precedenze, apprendere il

concetto di interazione tra processi.

Contenuti:

Elaborazione sequenziale e programmazione concorrente  
processi paralleli

Processi non sequenziali e grafo delle precedenze

Scomposizione di un processo non sequenziale

processi indipendenti

processi interagenti

Processi in competizione

Semplificazione delle precedenze

**Modulo IV: La comunicazione tra processi**

Obiettivi di apprendimento:

Conoscere i modelli di comunicazione tra processi: a memoria condivisa (o ad ambiente globale) ed il modello a

scambio di messaggi (o ad ambiente locale)

Contenuti:

La comunicazione: modelli software e hardware

Il modello a memoria condivisa

allocazione delle risorse ai processi e loro tipologia

competizione e cooperazione

### **Modulo V: La sincronizzazione tra processi ed i semafori**

Obiettivi di apprendimento:

Conoscere le tipologie di errore nei processi paralleli e le proprietà richieste ai programmi concorrenti. Definire

e utilizzare i semafori ed il concetto di indivisibilità di una primitiva

Contenuti:

Errori nei programmi concorrenti

Condizioni per avere la concorrenza: interleaving e overlapping

Condizioni di Bernstein

definizione di dominio e rango di una istruzione o procedura

Mutua esclusione e sezione critica

Starvation e deadlock

Semafori di basso livello e spin lock

allocazione di una risorsa: lock()

rilascio di una risorsa: unlock()

problema della indivisibilità

Semafori di Dijkstra

le primitive P e V e loro traduzione in pseudocodifica

Semafori e loro applicazioni

mutua esclusione tra gruppi di processi

semafori come vincoli di precedenza

### **Modulo VI: Problemi classici della programmazione concorrente: produttori/consumatori, lettori/scrittori**

Obiettivi di apprendimento:

Conoscere le caratteristiche dei problemi produttori/consumatori, lettori/scrittori e risolvere tali problemi

Contenuti:

Problema produttore/consumatore

Risoluzione del problema in pseudocodifica nel caso in cui il dato prodotto sia "singolo"

i semafori "pieno" e "vuoto" e loro inizializzazione

Risoluzione del problema in pseudocodifica nel caso in cui la memoria condivisa possa contenere più dati

(buffer circolare)

i puntatori metti e toglì

i semafori "pieno" e "vuoto" e loro inizializzazione

Problema dei lettori/scrittori

prima soluzione che favorisce i lettori

i semafori mutex e sincro e la variabile contatore numLettori

seconda soluzione proposta che favorisce gli scrittori

i semafori mutex1, mutex2 e la variabile contatore numScrittori

### **Modulo VII: Problemi classici della programmazione concorrente: deadlock**

Obiettivi di apprendimento:

Conoscere le cause del deadlock e saper mettere in atto le strategie per evitarlo od eliminarlo

Contenuti:

Condizioni necessarie e sufficienti per la generazione di un deadlock

Individuazione dello stallo, metterlo in evidenza con il grafo wait-for

I e II Teorema sul grafo di Holt

Affrontare lo stallo

detection e recovery

avoidance

definizione di sequenza sicura e stato sicuro di un sistema

problema del banchiere

prevention

ignorare il problema

problema dei filosofi a cena

### **Modulo VIII: Thread**

Obiettivi di apprendimento:

Imparare le differenze tra thread e processi, modalità di utilizzo dei thread nei SO

Contenuti:

Processi pesanti e processi leggeri, loro pregi e loro difetti

Definizione di thread

Thread safety

Realizzazione dei thread

user-level

kernel-level

Stati di un thread

Utilizzo dei thread

*Testo di riferimento: Tecnologie e progettazione di sistemi informatici e di telecomunicazioni,  
Camagni Nikolassy – Hoepli*