

Istituto Istruzione Superiore Statale

“E.Ferrari-H.Hertz”

Programma

Tecnologie e Progettazione di Sistemi Informatici e di Telecomunicazioni

Classe 4N Anno Scolastico 2018/2019

Contenuti

1. Processi Sequenziali e Paralleli
 - 1.1 Il modello a processi
 - 1.2 Stato dei processi
 - 1.3 Process Control Block (PCB)

2. Risorse e condivisione
 - 2.1 Il concetto di risorsa
 - 2.2 Condivisione e gestione di una risorsa
 - 2.3 Classificazione delle risorse, delle richieste e dell'assegnazione di una risorsa
 - 2.4 Grafo di Holt

3. I thread
 - 3.1 Definizione di Thread
 - 3.2 Differenza tra processi e Thread
 - 3.3 Diagramma di stato di un Thread
 - 3.4 Single threading e multithreading
 - 3.5 Realizzazione di un Thread : user-level e kernel-level
 - 3.6 Utilizzo dei Thread

4. Elaborazione sequenziale e concorrente
 - 4.1 Definizione di elaborazione sequenziale e programmazione concorrente
 - 4.2 Processi non sequenziali e grafo delle precedenze
 - 4.3 Scomposizione di un processo non sequenziale (parallelo)
 - 4.3.1 Processi indipendenti
 - 4.3.2 Processi interagenti

5. Comunicazione tra processi
 - 5.1 Modello a memoria comune o ad ambiente globale
 - 5.1.1 Allocazione delle risorse ai processi
 - 5.1.2 Risorse allocate dinamicamente o staticamente
 - 5.1.3 Risorse dedicate o comuni
 - 5.2 Mutua esclusione

5.3 Modello a scambio di messaggi o ad ambiente locale

5.3.1 Modello client server

6. La sincronizzazione tra processi

6.1 Differenza tra programma sequenziale e programma concorrente

6.2 Errori dipendenti dal tempo

6.3 Interleaving e overlapping

6.4 Condizioni di Bernstein: definizione ed esempi

6.5 Mutua esclusione e sezione critica

6.6 Starvation e deadlock

7. I semafori

7.1 Meccanismi per l'accesso alla sezione critica

7.2 Spin lock

7.2.1 Lock()

7.2.2 Unlock()

7.3 Semafori di Dijkstra

7.4 Esempio: Semafori Ferroviari

Esercitazioni di Laboratorio

Le esercitazioni in laboratorio sono state svolte in ambiente NETBEANS utilizzando come linguaggio di programmazione JAVA.

1. I thread in java

1.1 Thread come sottoclasse della classe Thread e il metodo run()

1.2 Uso dei metodi setName() e getName()

1.3 I metodi Sleep, Yield e Join, Stop, Resume, Suspend

1.4 Esecuzione contemporanea di più thread

1.5 Thread come Interfaccia Runnable

2. Interazione tra thread

2.1 Il qualificatore synchronized

Prova di recupero: Prova scritta semistrutturata

Roma,

Alunni

Prof.ssa Carla Stifano

Prof. Donato Tartaglione
