



**Istituto Istruzione Superiore “G. Ferraris F. Brunelleschi”**

Via R. Sanzio, 187 – 50053 Empoli (FI) ☎ 0571 81041 – fax 0571 81042

www.ferraris.eu e-mail fiis012007@istruzione.it

Codice Fiscale n. 91017160481 Cod Min. FIIS012007 Codice Univoco UF8UAP



UNIONE EUROPEA

**FONDI  
STRUTTURALI  
EUROPEI**

**pon**  
2014-2020



MIUR

Ministero dell'Istruzione, dell'Università e della Ricerca  
Dipartimento per la Programmazione  
Direzione Generale per interventi in materia di edilizia  
scolastica, per la gestione dei fondi strutturali per  
l'istruzione e per l'innovazione digitale  
Ufficio IV

PER LA SCUOLA - COMPETENZE E AMBIENTI PER L'APPRENDIMENTO (FSE-FESR)

**Materia: Tecnologie e progettazione di Sistemi Informatici**

**Classe: 5A^Informatica**

**Programmazione Annuale**

**A.S. 2018/2019**

*Docenti: Prof. Nello Perna Prof. Simone Calugi*

**STRUMENTI**

**Libri di testo:**

“tecnologie e progettazione di sistemi informatici e di Telecomunicazioni” di P. Camagni, R. Nikolassy, ed.Hoepli.

**Laboratorio multimediale di TPSI.**

**PROCESSI SEQUENZIALI E PARALLELI**

I thread o “processi leggeri”

- Generalità
- “processi pesanti” e “processi leggeri”
- Soluzioni adottate: single threading e multithreading
- Realizzazione di thread
- Stati di un thread
- Utilizzo dei thread

Elaborazione sequenziale e concorrente

- Generalità
- Processi non sequenziali e grafo di precedenza

La descrizione della concorrenza

- Esecuzione parallela
- Fork-join
- Cobegin-coend
- Equivalenza di fork-join e cobegin-coend
- Semplificazione delle precedenze

## **COMUNICAZIONE E SINCRONIZZAZIONE**

La comunicazione tra processi

- Comunicazione: modelli software
- Modelli a memoria comune (ambiente globale, global environment)
- Modello a scambio di messaggi (ambiente locale, message passing)

Semafori

- Primitive e implementazione
- Semafori come strumento per gestire mutua esclusione e sincronizzazione

Stallo

- Definizione di stallo/deadlock
- Come affrontare il problema dello stallo

## **ARCHITETTURA DI RETE:**

I sistemi distribuiti

- Definizione di sistema distribuito
- Differenza tra sistemi centralizzati e sistemi distribuiti
- Classificazione dei sistemi distribuiti
- Vantaggi e svantaggi della distribuzione

Storia dei sistemi distribuiti e modelli architetturali

- Architetture distribuite hardware: dalle SISD al Cluster di PC
- Architetture distribuite software: dai terminali remoti a sistemi completamente distribuiti
- Architetture a livelli

Il modello Client-Server

- I modelli di comunicazione
- Modello client-server
- Livelli e strati

Le applicazioni di rete

- Il modello ISO/OSI e le applicazioni
- Applicazione di rete ed identificazione mediante socket
- Scelta dell'architettura per l'applicazione di rete
  - Client-Server
  - Peer-to-Peer

## **ARCHITETTURA TCP/IP**

## Il livello trasporto

- Caratteristiche del livello Trasporto
- Servizi offerti dallo strato di trasporto alle applicazioni
- Protocolli del livello di trasporto, vantaggi e svantaggi
  - Connection-oriented: TCP
  - Connectionless: UDP

## I socket e i protocolli per la comunicazione di rete

- Applicazione di rete
- Protocollo di comunicazione
- Le porte di comunicazione ed i socket

## La connessione tramite socket

- Famiglie e tipi di socket
  - Stream socket
  - Datagram socket

## Il livello applicazione ed i suoi protocolli

- Il livello applicazione
- Protocolli del livello applicazione
  - Telnet
  - FTP
  - HTTP
  - DNS

## **ATTIVITA' DI LABORATORIO**

Argomenti trattati e strumenti utilizzati nelle esperienze:

- SEMAFORI in JAVA.
  - Il *problema del Produttore – Consumatore*: generalizzazione del problema prima a 2 Produttori e 2 Consumatori e quindi N produttori e M consumatori.
  - Sincronizzazione nel *problema dei Lettori – Scrittori* in cui più lettori possono contemporaneamente accedere alla risorsa condivisa mentre per gli scrittori l'accesso deve avvenire in modo esclusivo.
- Realizzazione di una *Corsa di Biciclette* con visualizzazione grafica [libreria *Swing*] dello stato della corsa utilizzando i *thread in Java* [meccanismi per la concorrenza e metodi *Synchronized*].
- DEADLOCK in JAVA: studio del *Problema dei Filosofi a Cena*. Realizzazione del *monitor Tavolo* per la gestione e il rilascio delle risorse condivise [*forchette*] e delle relative *entry procedure* [*prendiForchette* e *rilasciaForchette*].
- Uso dei SOCKET in JAVA.
  - Studio del programma *client* che richiede l'ora di sistema al *server* presente sul libro di testo.
  - Realizzare una semplice calcolatrice: il *client* invia gli operandi e l'operatore al *server*, il quale esegue l'operazione e restituisce il risultato.

Dopo aver realizzato una versione funzionante su macchina singola, il software è stato testato su una macchina server e un diverso client, risolvendo i problemi relativi alle autorizzazioni imposti dal sistema operativo.

- Simulazione delle *Estrazioni del Lotto*: N giocatori [client] scelgono 3 numeri e li comunicano alla ricevitoria [server]. Quindi, questa effettua le estrazioni del gioco e comunica le eventuali vincite.
- Realizzazione di un'applicazione che utilizzi i *Socket* e la libreria grafica *Swing* per produrre un'animazione CLIENT-SERVER [sul libro è fornito l'esempio di una palla che "attraversa" le finestre].
- Studio delle API di Google Maps per la geolocalizzazione
- Creazione di file *.pdf* con il PHP.

Empoli 17/05/18

#### GLI ALUNNI



Nello Perna

#### I DOCENTI

Nello Perna



Simone Calugi

