

Programmazione Distribuita I

Docenti: Enrico Masala, Riccardo Sisto
Esercitatore Lab : Enrico Masala

Introduzione al Modulo:

- ❑ Obiettivi, Programma e Requisiti
- ❑ Organizzazione, Laboratorio e Interazione
- ❑ Testi e materiale didattico
- ❑ Esame e regole

Principali Obiettivi

- ❑ Acquisire conoscenze di base sull'architettura e sulle caratteristiche del software distribuito
- ❑ Imparare ad usare tecniche di programmazione in ambito distribuito:
 - ❖ network programming
 - ❖ web programming

Programma

- ❑ Architettura software dei sistemi distribuiti
- ❑ Network Programming (socket, RPC)
- ❑ Web Programming
 - ❖ HTML, CSS
 - ❖ Web dinamico, CGI
 - ❖ Client-side programming (JavaScript)
 - ❖ Server-side programming (PHP) con accesso DBMS
 - ❖ Cenni di AJAX

(Guida del corso online per maggiori dettagli)

Requisiti

- ❑ Buone capacità di programmazione
- ❑ Ottima conoscenza del linguaggio C e capacità di debugging
 - ❖ Fondamentale per parte su socket (strutture, allocazione dinamica della memoria, programmazione concorrente)
- ❑ Conoscenza dei principali protocolli di rete
- ❑ Conoscenza di base del funzionamento dei DBMS
 - ❖ Creazione di semplici schemi, interrogazione/modifica dati tramite SQL

Organizzazione

- Lezione in aula: circa 39 ore
 - ❖ Orario: mercoledì ore 14:30-17:30
- Esercitazioni in laboratorio (LABINF): circa 21 ore, in blocchi consecutivi di 3 ore
 - ❖ Parte pratica estremamente importante
 - ❖ Tracce di laboratorio «abbondanti»: utile terminare gli esercizi in altre ore di studio autonomo - fornite tracce di soluzione per socket
 - ❖ Utile per discutere dubbi/scelte implementative/problemi vari con esercitatore
 - ❖ Calendario reso noto su portale didattica
 - Orario: giovedì ore 10-13 (solo settimane specificate)

Laboratorio

- ❑ Esercizi di programmazione sulle principali tecniche spiegate
 - ❖ Linux Ubuntu (socket)
 - ❖ WinXP XAMPP Eclipse+PHP+MySQL (web)
- ❑ Esercizi concentrati più sull'uso delle tecniche descritte che sulla programmazione di algoritmi complicati
 - ❖ Comunque necessaria conoscenza gestione vettori, liste, gestione memoria dinamica, sincronizzazione tra processi UNIX
- ❑ Frequenza al laboratorio fortemente consigliata (tracce sul portale)
- ❑ Farsi creare un account (vedere regole su www.labinf.polito.it) se non già in possesso

Interazione

- ❖ Domande in aula e in laboratorio: benvenute e auspicabili
 - Sia durante la lezione (stimolo per l'interesse!)
 - Sia al termine
- ❖ Tramite email (v. portale)
- ❖ Consulenza (su appuntamento)
- ❖ Avvisi sul portale della didattica
 - Consultare periodicamente
Avvisi importanti o urgenti anche via SMS o email
- ❖ Questionario di fine corso (CPD): utile, ma non aspettare per eventuali problemi!

Testi e materiale didattico

- ❑ Copia delle slides utilizzate (distribuite in anticipo)
- ❑ Documenti/standard di riferimento
 - ❖ libero accesso su Internet
- ❑ Alcuni libri, con copertura parziale, come i seguenti:
 - ❖ D. E. Comer, D. L. Stevens, "Internetworking with TCP/IP: vol. 3" (socket)
 - ❖ V. Della Mea, L. Di Gaspero, I. Scagnetto "Programmazione web lato server", ed. Apogeo

Esame e regole

- ❑ L'esame consta di due parti, da sostenere contemporaneamente nello stesso appello
 - ❖ Prova «socket» (in C) : elaborato svolto in laboratorio in ambiente Linux
 - ❖ Prova di «programmazione web» : elaborato svolto in autonomia nelle settimane precedenti l'appello
- ❑ Il voto finale è la somma dei voti delle due parti, che hanno uguale peso
- ❑ E' possibile sostenere la prova «socket» solamente avendo consegnato la corrispondente prova di «programmazione web»

Prova «socket»

- ❖ Prova individuale, elaborato da realizzare in tempo assegnato (2-3 ore), all'elaboratore in laboratorio, il giorno dell'appello
- ❖ Il voto viene assegnato sulla base dei test superati dall'elaborato. Alcuni test vengono resi disponibili allo studente durante la prova.
- ❖ E' possibile consultare materiale pre-caricato dallo studente stesso (tipicamente, esercizi svolti in proprio durante i laboratori) con modalità prestabilite.
- ❖ Il mancato superamento della prova invalida anche la prova di programmazione web già consegnata
- ❖ **NB: Verranno svolti controlli automatici** di similarità sulle soluzioni consegnate. Prove con soluzioni identiche o molto simili verranno annullate

Prova «programmazione web»

- ❑ Si deve completare, lavorando in modo autonomo, entro le date comunicate dal docente per ciascun appello, la realizzazione di
 - ❖ Un sito web che utilizza PHP, JavaScript, HTML, CSS ed accesso ad un database, seguendo la traccia indicata di volta in volta.
- ❑ Colloquio orale obbligatorio per chiarire gli aspetti dubbi, scelte implementative e dimostrare la puntuale conoscenza dell'elaborato consegnato
- ❑ **NB: Verranno svolti controlli automatici di similarità sulle soluzioni consegnate. La sottomissione di soluzioni copiate verrà punita con la bocciatura e con provvedimenti disciplinari. Ogni studente è responsabile di mantenere privato il proprio codice.**