



UNIVERSITÀ DI PISA

LABORATORIO II

PATRIZIO DAZZI

Academic year **2023/24**
Course **INFORMATICA**
Code **732AA**
Credits **12**

Modules	Area	Type	Hours	Teacher(s)
LABORATORIO II	INF/01	LEZIONI	96	PATRIZIO DAZZI LUCA FERRUCCI

Obiettivi di apprendimento

Conoscenze

Lo studente acquisirà conoscenze relative alla programmazione in C, gestione delle procedure di compilazione tramite Makefile e scripting in bash. Lo studente acquisirà inoltre le conoscenze necessarie alla programmazione concorrente e di sistema con il linguaggio C. Lo studente apprenderà anche come programmare in Assembler ARM.

Modalità di verifica delle conoscenze

Le conoscenze verranno verificate tramite:

- Prove in itinere - esercizi di programmazione da svolgere in un tempo limitato.
- Homework - esercizi di programmazione più complessi, con limiti temporali più rilassati.
- Progetto finale - compito più complesso che deve integrare la maggior parte dei concetti affrontati nel corso.
- Verifica orale finale.

Capacità

Lo studente svilupperà capacità di:

- Sviluppo di script di shell (Bash).
- Sviluppo di programmi in Assembler (ARM).
- Sviluppo di programmi in C, anche utilizzando strutture dati complesse.
- Sviluppo di programmi in C, che facciano ricorso a chiamate di sistema (POSIX).
- Design e sviluppo di applicazioni concorrenti in C.

Modalità di verifica delle capacità

Prove in itinere di programmazione in classe e a casa, progetto.

Comportamenti

I comportamenti che gli studenti acquisiranno, includono:

- Capacità di risolvere problemi: sviluppo di competenze nel risolvere problemi di programmazione complessi, utilizzando il linguaggio C e le tecniche apprese nel corso.
- Capacità di apprendimento autonomo: gli studenti acquisiranno la capacità di apprendere tecniche di programmazione in modo autonomo, ad esempio attraverso la consultazione di documentazione o lo studio individuale degli aspetti correlati alla programmazione in C, all'Assembler ARM o ad altri concetti trattati nel corso.
- Capacità di gestire e risolvere problemi di concorrenza: la programmazione concorrente comporta la gestione di più thread o processi che operano contemporaneamente e condividono risorse comuni. Durante il corso, gli studenti affronteranno sfide legate alla sincronizzazione dei thread, alla gestione della mutua esclusione e alla prevenzione dei conflitti tra le risorse condivise. Di conseguenza, acquisiranno la capacità di identificare e risolvere problemi di concorrenza, migliorando la qualità e l'efficienza delle proprie soluzioni.

Modalità di verifica dei comportamenti

L'opportuna acquisizione dei comportamenti sarà verificata tramite la valutazione degli assegnamenti, sia in classe che a casa, il progetto finale e la prova orale.



UNIVERSITÀ DI PISA

Prerequisiti (conoscenze iniziali)

Competenze di base della programmazione.

Indicazioni metodologiche

Modalità:

- Lezioni frontali

Attività di apprendimento:

- partecipazione alle lezioni
- studio individuale
- svolgimento di esercizi di programmazione individualmente
- svolgimento di prove di valutazione intermedie
- partecipazione ai ricevimenti tenuti dai docenti e dai tutor

Frequenza:

- consigliata ma non obbligatoria.

Metodi di insegnamento:

- lezioni frontali con slide
- esercizi di programmazione

Programma (contenuti dell'insegnamento)

Programmazione in C, incluso l'utilizzo di pointer, stringhe, librerie standard di I/O.

Programmazione in C su strutture dati astratte, come pile, liste, alberi e grafi.

Debugging con GDB e controllo degli accessi alla memoria con Valgrind.

Utilizzo di Makefile.

Scripting di shell in bash.

Programmazione in Assembler ARM.

Programmazione in C utilizzando chiamate di sistema e segnali POSIX.

Programmazione concorrente in C utilizzando processi e thread.

Comunicazione e sincronizzazione tra processi e thread.

Programmazione concorrente in C utilizzando user-space threads.

Introduzione al modello di comunicazione client-server basato su sockets.

Applicazioni, esempi e link con altri corsi del secondo anno.

Bibliografia e materiale didattico

Dennis M. Ritchie, Brian W. Kernighan. Il linguaggio C Principi di programmazione e manuale di riferimento, Pearson Italia, ISBN: 9788871922003

Marc J. Rochkind. Advanced UNIX Programming 2nd Edition, Addison-Wesley Professional Computing Series, 2004.

Indicazioni per non frequentanti

Per i non frequentanti si adottano le regole applicate a chi non ha superato le verifiche intermedie: progetto + orale.

Modalità d'esame

L'esame si svolgerà con una delle due seguenti modalità:

- Verifiche intermedie durante l'anno, tramite compiti di programmazione in classe + mini-progetto finale + orale.
- Per gli studenti che non hanno passato le verifiche intermedie: progetto + orale.

Stage e tirocini

N/A

Note

N/A

Ultimo aggiornamento 19/07/2023 14:48