

Introduzione al corso

martedì 14 febbraio 2023 13:29

CORSO DI SISTEMI OPERATIVI

Francesco Quaglia
PhD, Full Professor



<https://francescoquaglia.github.io/>

Obiettivi del corso

Il corso ha l'obiettivo di presentare allo studente concetti basilari riguardanti la strutturazione dei moderni sistemi operativi, unitamente alle funzionalità che essi offrono sia all'utenza che agli sviluppatori di applicazioni. Particolare attenzione verrà dedicata ai supporti che i sistemi operativi offrono per lo sviluppo del software in linguaggio C. I concetti introdotti verranno calati su due studi di caso, riguardanti la famiglia dei sistemi Unix (Posix) e quella dei sistemi Windows.

→ LINUX

Prerequisiti

- Conoscenza di base del linguaggio C
- Conoscenza di base di tool-chain di compilazione (e.g. gcc)
- Conoscenza dell'architettura di massa di un sistema di calcolo
- Nozioni di programmazione e strutturazione del software di **livello assembler** (ASM)

Modalità d'esame

L'esame richiede il superamento di **una prova scritta** e di **una prova pratica di programmazione in linguaggio C** (in tecnologia Posix o Win-API a scelta dello studente) da svolgersi in laboratorio. La prova scritta vale un massimo di 21 punti, con 12 come soglia per la sufficienza. La prova pratica in laboratorio vale un massimo di 11 punti, con 6 come soglia per la sufficienza. Il voto d'esame risulterà pari alla somma dei punteggi conseguiti alle due prove. Per la lode è richiesto un punteggio complessivo di almeno 31.

Nel caso in cui lo studente consegua la sufficienza solo ad una delle due prove, nell'ambito dello stesso anno accademico egli sarà esonerato dal dover rieseguire la prova in cui è risultato già sufficiente. Potrà quindi (ri)sostenere la prova (scritta o di programmazione) dove non ha riportato la sufficienza ad un appello successivo dello stesso anno accademico.

PROGRAMMA DEL CORSO:

- Introduzione ai sistemi operativi
 - Richiami sull'organizzazione di un sistema di calcolo
 - Obiettivi dei sistemi operativi
 - Sistemi batch uniprogrammati
 - Sistemi batch multiprogrammati
 - Sistemi time-sharing
 - Architettura di massa dei sistemi UNIX/Windows
 - Ambienti di esecuzione
 - Aspetti basilici sulla sicurezza del software
- Virtual File System ed I/O
 - Concetti basilici
 - Metodi di accesso e di allocazione dei file
 - Implementazione del virtual file system
 - Gestione dei buffer di I/O
 - Schedulazione di I/O
 - Gestione delle utenze e dei permessi di accesso
 - Concetti basilici di crittografia ed autenticazione
 - Virtual file system ed I/O in sistemi UNIX/Windows
- Sincronizzazione
 - Spinlocks, mutex e semafori
 - Variabili di condizione
 - Supporti per la sincronizzazione in sistemi UNIX/Windows

- Processi e thread
 - Esecuzione e stati di processi
 - Multiprogrammazione e Swapping
 - Strutture di controllo di processi
 - Immagine di un processo
 - Liste di processi e scheduling
 - Processi in sistemi UNIX/Windows
 - Supporti per il multi-threading
 - Threads in sistemi UNIX/Windows
- Comunicazione
 - Pipe
 - Pipe con nome
 - Scambio di messaggi
 - Comunicazione in sistemi UNIX/Windows
- Eventi
 - Meccanismi di segnalazione e gestione di eventi sincroni ed asincroni
 - Supporti in sistemi UNIX/Windows

- Scheduling della CPU
 - Metriche di riferimento
 - Algoritmi di scheduling classici
 - Scheduling in sistemi UNIX/Windows
- Gestione della memoria
 - Binding degli indirizzi
 - Partizioni fisse e variabili
 - Paginazione e segmentazione
 - Memoria virtuale
 - Memoria condivisa e file-mapping
 - Gestione della memoria in sistemi UNIX/Windows
- Servizi di sistema per la programmazione di rete
 - Stack di protocolli di comunicazione
 - Sockets in sistemi UNIX/Windows

(NOTA IMPORTANTE: per coloro che devono sostenere l'esame da 6CFU le parti del programma in rosso sono escluse sia dalla prova scritta che dalla prova di programmazione in laboratorio)