SCHEDA DELL'INSEGNAMENTO (SI)

SISTEMI OPERATIVI

SSD ING-INF/05

DENOMINAZIONE DEL CORSO DI STUDIO: CORSO DI LAUREA IN INGEGNERIA INFORMATICA

ANNO ACCADEMICO: 2023-2024

INFORMAZIONI GENERALI - DOCENTE

DOCENTE:
TELEFONO:
EMAIL:

SI VEDA SITO WEB DEL CORSO DI STUDI

INFORMAZIONI GENERALI - ATTIVITÀ

INSEGNAMENTO INTEGRATO (EVENTUALE): N.A.
MODULO (EVENTUALE): N.A.
CANALE (EVENTUALE): N.A.
ANNO DI CORSO (I, II, III): III
SEMESTRE (I, II): I
CFU: 9

INSEGNAMENTI PROPEDEUTICI (se previsti dal Regolamento del CdS)

Programmazione, Calcolatori Elettronici.

EVENTUALI PREREQUISITI

Nessuno.

OBIETTIVI FORMATIVI

Il corso si pone l'obiettivo di fornire competenze sulle architetture di riferimento dei sistemi operativi; sulle metodologie utilizzate per la gestione delle risorse in un sistema operativo moderno; sugli strumenti per la programmazione di sistema; sull'utilizzo di una piattaforma Unix a livello utente e amministratore; sui principi base della programmazione concorrente. Le esercitazioni e le attività di laboratorio sono sviluppate in ambiente Linux e consistono in applicazioni di programmazione concorrente e la programmazione di moduli del kernel Linux.

RISULTATI DI APPRENDIMENTO ATTESI (DESCRITTORI DI DUBLINO)

Conoscenza e capacità di comprensione

Lo studente deve dimostrare: di conoscere le problematiche relative alla gestione delle risorse di calcolo in sistemi multi-utente e multi-programmati, di comprendere gli strumenti metodologici e gli algoritmi di base utilizzati per la realizzazione di sistemi di elaborazione ad alte prestazioni che utilizzino efficientemente le risorse di calcolo, di illustrare le tecniche fondamentali inerenti alla programmazione concorrente e di sistema.

Capacità di applicare conoscenza e comprensione

Lo studente deve dimostrare di essere in grado di applicare gli strumenti metodologici appresi ai fini della progettazione dei sistemi di elaborazione, basati sull'utilizzo delle astrazioni fondamentali fornite dai sistemi operativi (es. processi, thread, filesystem, inter-process communication). Inoltre, lo studente deve dimostrare di essere in grado di applicare gli strumenti metodologici ai fini di sviluppare nuovi sistemi, utilizzando le conoscenze e le tecniche di programmazione concorrente e di sistema, e ai fini di diagnosticare i problemi dei sistemi di calcolo dovuti ad errata sincronizzazione e all'utilizzo inefficiente delle risorse di calcolo.

PROGRAMMA-SYLLABUS

Concetti Introduttivi. Evoluzione storica dei S.O. - Mono e multiprogrammazione - Batch, time sharing, real-time, mobile, cloud computing - Richiami di elementi di architettura a supporto di un S.O. (modalità di I/O, gestione delle interruzioni, modalità utente e supervisore, gerarchie di memoria) - Virtualizzazione delle risorse nei S.O. - Il kernel - Chiamata a supervisore - Architetture a livelli, monolitiche, modulari, microkernel.

Gestione e Scheduling dei Processi e Thread. Concetto di processo – Stati di un processo – Descrittore di un processo – Code di processi – Il cambiamento di contesto – La gestione dei processi nei sistemi operativi Linux e Windows – Scheduling della CPU a breve, medio e lungo termine – Parametri di valutazione degli algoritmi di scheduling – Starvation – Preemption – Algoritmi First Come First Served, Round Robin, Shortest Process Next, Shortest Remaining Time, a code multiple con retroazione – Confronto tra algoritmi di scheduling monoprocessore – Scheduling tradizionale UNIX – Gli Scheduler O(1) e CFS del sistema operativo Linux – Scheduling nel sistema operativo Windows – Scheduling multiprocessore: Architetture SMP, multicore, e hyperthreading, scheduling con load sharing e dynamic load balancing, gang scheduling – Concetto di thread – Processi e thread – Stati di un thread – Thread a livello utente e a livello del nucleo – Modelli di programmazione multithreading – Primitive per la gestione dei threads – Cenni alla gestione dei thread nei sistemi Linux, Windows, Java.

Programmazione Concorrente. Concorrenza e parallelismo – Speed-up nelle architetture concorrenti e

parallele – La legge di Amdahl – Le primitive fork/join – Concetti di risorsa e di gestore di risorsa – Competizione, cooperazione, ed interferenza – Race condition e sincronizzazione - I modelli ad ambiente globale e locale – L'interazione tra processi nel modello ad ambiente globale – Il problema della mutua esclusione: requisiti, supporto hardware e soluzione. Il problema della comunicazione – I semafori – La comunicazione tramite memoria condivisa – La soluzione dei problemi di mutua esclusione mediante semafori – Problemi di cooperazione nel modello ad ambiente globale: problema del produttore/consumatore e soluzioni mediante semafori, problema lettori/scrittori e soluzioni mediante semafori – I monitor – Strategie di controllo signal and continue, signal and wait, e la soluzione di Hoare – La realizzazione di un monitor mediante semafori – La soluzione dei problemi di mutua esclusione, produttore/consumatore e lettori/scrittori mediante monitor – L'interazione tra processi nel modello ad ambiente locale – Le primitive per lo scambio di messaggio – Comunicazione diretta e indiretta, simmetrica ed asimmetrica – Comunicazione asincrona e sincrona – Send asincrona e sincrona – Receive bloccante e non bloccante – Realizzazione di send e receive sincrone mediante primitive asincrone – Processo servitore – Il problema del deadlock – Condizioni necessarie per il deadlock – Metodi per la gestione del deadlock – Prevenzione del deadlock – Deadlock Avoidance e algoritmo del banchiere – Rilevazione e recupero del deadlock – Comparazione delle strategie per la gestione del deadlock.

La Gestione della Memoria Centrale. Aspetti caratterizzanti la gestione della memoria: rilocazione, allocazione, organizzazione dello spazio virtuale, caricamento – Lo swapping – La gestione a partizioni multiple – Paginazione: schema di traduzione degli indirizzi, architettura di paginazione, TLB, Struttura della tabella delle pagine – Segmentazione: schema di traduzione degli indirizzi, architettura di segmentazione – Segmentazione con Paginazione – La memoria virtuale – La paginazione su richiesta – Algoritmi per la sostituzione delle pagine – Attività di paginazione degenere (thrashing) – Il modello del working set – Gestione della Memoria nel sistema operativo Linux: allocatori user-space e kernel-space, zone di memoria, buddy system, page cache, algoritmo di page frame reclaim – Cenni sulla gestione della memoria nel sistema operativo Windows.

La gestione dell'I/O. Le operazioni di I/O – La virtualizzazione delle risorse di I/O – Livello indipendente dai dispositivi, livello dipendente dai dispositivi – I driver – Struttura della memoria secondaria – I dischi – Scheduling degli accessi al disco con riferimento ai cilindri ed ai settori – I/O caching e buffering – Algoritmi di I/O scheduling FIFO, SCAN, e varianti – Scheduling del disco nel SO Linux – Architetture RAID – I dischi a stato solido.

La Gestione dei File. Organizzazione logica del file system: directory e file - Operazioni sulle directory e sui file – Metodi di accesso – Descrittore di file – La condivisione dei file – Struttura delle directory per la condivisione di file – Link per la condivisione – La protezione dei file – Organizzazione logica del file system – Metodi di allocazione dei file: allocazione contigua, a lista concatenata e indicizzata – La gestione dei blocchi liberi – inode e gestione dei file in Unix – Il Virtual File System di Linux e i file system ext2, ext3 ed ext4 – Il File System NTFS di Windows – Journaling File Systems – Log-structured File System– Il File System F2FS di Linux per SSD.

Primitive per la gestione dei processi e thread nel SO UNIX/Linux. Primitive per la creazione e terminazione dei processi: fork, exec, exit, wait – Gestione delle risorse IPC – Primitive per la gestione della memoria condivisa – Primitive per la gestione dei semafori – uso della semop per la realizzazione di primitive wait e signal – Esempi d'uso: soluzione di problemi di mutua esclusione e comunicazione (produttore/consumatore e lettori/scrittori), realizzazione object-based e object-oriented di un tipo Monitor – Primitive per la gestione delle code di messaggi ed esempi d'uso – Le primitive POSIX Threads per la gestione dei threads, ed esempi d'uso.

Approfondimenti sul sistema operativo UNIX/Linux. Installazione del SO Linux – La shell e i comandi di base (navigazione nel filesystem, assegnazione permessi, installazione del software, compilazione dei programmi) – La compilazione dei programmi: makefile, librerie statiche e dinamiche – I canali di I/O dei processi – Pipe – Variabili d'ambiente – Shell scripting – Segnali UNIX – Configurazione e compilazione del kernel Linux – Sviluppo di system call e di moduli del kernel.

Approfondimenti su virtualizzazione e sul SO mobile Android. Utilizzi della virtualizzazione. Architetture di virtualizzazione. Virtualizzazione della CPU (virtualizzabilità della CPU, tecnica del trap-and-emulate, full virtualization mediante DBT, paravirtualizzazione, supporto hardware Intel VT). Virtualizzazione della memoria (shadow page tables, extended page tables). Virtualizzazione dell'I/O (full virtualization, PV I/O, I/O passthrough, IO-MMU, SR-VIO). Esempio di tecnologie VMware. Storia e obiettivi di progettazione di Android. Applicazioni in Android. La gestione dei processi e della memoria (ciclo di vita delle Activity, Intent, OOM). Application Security Model. Inter-Process Communication in Android.

MATERIALE DIDATTICO

- Libri di testo adottati
  - Ancillotti, Boari, Ciampolini, Lipari, "*Sistemi Operativi*", McGraw Hill.
  - Stallings, "*Operating Systems: Internals and Design Principles", 6th ed.*, Pearson Education

- Libri di testo consigliati
  - Silberschatz, Galvin, Gagne, "*Sistemi operativi - sesta edizione*", Addison Wesley
  - Tanenbaum, "*I Moderni Sistemi Operativi – terza edizione*", Pearson Education

- Dispense didattiche e trasparenze delle lezioni disponibili sul sito web docente.

MODALITÀ DI SVOLGIMENTO DELL'INSEGNAMENTO

L'insegnamento verrà erogato attraverso lezioni frontali (circa due terzi delle ore totali del corso), con esercitazioni in aula e in laboratorio (circa un terzo delle ore totali del corso). Le lezioni frontali introdurranno gli aspetti teorici inerenti alle astrazioni realizzate dai sistemi operativi, agli algoritmi per la gestione efficiente delle risorse di calcolo (CPU, memoria, I/O), e alle problematiche, gli algoritmi e i meccanismi più comuni per lo sviluppo di sistemi concorrenti (es. semafori, mutex, monitor, memoria condivisa, code di messaggi). Nelle esercitazioni in aula e in laboratorio, gli studenti approfondiranno praticamente gli aspetti teorici, sviluppando autonomamente dei programmi concorrenti con il sistema operativo Linux e il linguaggio di programmazione C.

VERIFICA DI APPRENDIMENTO E CRITERI DI VALUTAZIONE

a) Modalità di esame:

L'esame si articola in prova	
scritta e orale	X
solo scritta	
solo orale	
discussione di elaborato progettuale	
altro	

In caso di prova scritta i quesiti sono	**A risposta multipla**	
	A risposta libera	
	Esercizi di programmazione concorrente	X

b) Modalità di valutazione:

La valutazione terrà conto in modo uniforme sia dello svolgimento della prova scritta (esercizi di programmazione concorrente), sia dello svolgimento della prova orale.