

Corso di Laurea in Ingegneria Informatica, Canale San Giovanni

CORSO DI SISTEMI OPERATIVI

Prof. Roberto Natella

Programma dell'A.A. 2024/2025

Ultimo aggiornamento: 11 dicembre 2024

Obiettivi Formativi

Il corso fornisce le conoscenze che sono alla base della teoria dei Sistemi Operativi e delle tecniche di programmazione concorrente. I contenuti teorici riguardano i modelli architetturali dei Sistemi Operativi, la gestione e lo scheduling dei processi e dei thread, le tecniche per la competizione e cooperazione tra processi, il deadlock, la gestione della memoria, la gestione dell'Input/Output, i File System e le problematiche di sicurezza. 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.

Contenuti

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.

Lucidi delle lezioni: SO-1, SO-2, SO-3

Libro di testo ANCILLOTTI-1: Capitolo 1

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 nel sistema operativo Linux – 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 monoprocesso – Scheduling tradizionale UNIX – Gli Scheduler O(1) e CFS del sistema operativo Linux – Scheduling multiprocesso: Architetture SMP, multicore, e hyperthreading, scheduling con load sharing e dynamic load balancing – 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.

Lucidi delle lezioni: SO-4, SO-5, SO-6, SO-7, SO-8

Libro di testo ANCILLOTTI-1: Capitoli 2, 9, 10

Dispensa b) "Scheduling del processore"

Dispensa k) "Process Scheduling in Linux"

Dispensa j) "Scheduling multiprocesso"

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, vincoli di 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 messaggi – 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.

Lucidi delle lezioni: SO-9, SO-10, SO-11, SO-12, SO-13, SO-14

Libro di testo ANCILLOTTI-1: Capitolo 3

Dispensa d) "Mutua esclusione"

Dispensa e) "Problemi di programmazione concorrente"

Dispensa n) "Il costrutto Monitor"

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 degenera (thrashing) – Il modello del working set – Gestione della Memoria nel sistema operativo Linux: allocator user-space e kernel-space, user/kernel memory split, kernel master page table, buddy system, page cache, algoritmo di page frame reclaim.

Lucidi delle lezioni: SO-15, SO-16, SO-17

Libro di testo ANCILLOTTI-1: Capitolo 4

Dispensa l) "Memory Management in Linux"

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 magnetici – Tempi di accesso al disco con riferimento ai cilindri ed ai settori – I/O caching e buffering – Algoritmi di I/O scheduling: FIFO, a priorità, SSTF, SCAN, e varianti – Scheduling del disco nel SO Linux – Architetture RAID – I dischi a stato solido: architettura, il "contratto non scritto" con il SO.

Lucidi delle lezioni: SO-18

Libro di testo ANCILLOTTI-1: Capitolo 5

Dispensa g) "I/O e gestione dischi"

Dispensa i) "Gestione dei dispositivi di memoria secondaria 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 – I File System FAT ed NTFS di Windows – Journaling File Systems – Log-structured File Systems – Il File System F2FS di Linux per SSD.

Lucidi delle lezioni: SO-19

Libro di testo ANCILLOTTI-1: Capitolo 6

Dispensa i) "Gestione dei dispositivi di memoria secondaria a stato solido"

Dispensa o) "File system in Linux"

La Virtualizzazione. 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 (extended page tables). Virtualizzazione dell'I/O (full virtualization, PV I/O). Container-based virtualization. Esempio di tecnologie VMware e Docker.

Lucidi delle lezioni: SO-20

Dispensa p) "La Virtualizzazione"

Il SO mobile Android. 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.

Lucidi delle lezioni: SO-21

Dispensa p) "Android"

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_sem e signal_sem – Esempi d'uso: soluzione di problemi di mutua esclusione e comunicazione (produttore/consumatore e lettori/scrittori), realizzazione 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.

Lucidi delle lezioni: SO-ES-3, SO-ES-5, SO-ES-7, SO-ES-8, SO-ES-9

Libro di testo ANCILLOTTI-1: Capitolo 8

Dispensa f) "Threads"

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) – La compilazione dei programmi: compilazione modulare, file oggetto, linking, caricamento, makefile, librerie statiche e dinamiche – Aspetti avanzati sulla shell di Linux: i canali di I/O dei processi, pipe, variabili d'ambiente, shell scripting, esecuzione in background, segnali – Il sistema di version control GIT – Configurazione e compilazione del kernel Linux – Sviluppo di system call e di moduli del kernel, con esempi di device driver e di rootkit.

Lucidi delle lezioni: SO-ES-1, SO-ES-2, SO-ES-4, SO-ES-6, SO-ES-10

Libro di testo ANCILLOTTI-1: Capitolo 8, Appendice C

Dispensa a) "Unix"

Dispensa c) "Compilazione e Makefile"

Dispensa h) "Sviluppo di codice per il kernel Linux"

Modalità d'esame

L'esame consiste di una prova scritta e una prova orale.

La prova scritta mira all'accertamento delle competenze di programmazione concorrente in ambiente Linux. La prova scritta include esercizi tra le seguenti tipologie trattate durante il corso:

- Problemi di competizione ad ambiente globale, da realizzare con semafori oppure con il costrutto Monitor, e con programmi multi-processo oppure multi-thread.
- Problemi di cooperazione ad ambiente globale, di tipo produttore-consumatore con buffer singolo, vettore di buffer circolare, vettore di buffer con stato, da realizzare con semafori oppure con il costrutto Monitor, e con programmi multi-processo oppure multi-thread.
- Problemi di cooperazione ad ambiente globale, di tipo lettori-scrittori, con starvation dei soli scrittori oppure di entrambi, da realizzare con semafori oppure con il costrutto Monitor, e con programmi multi-processo oppure multi-thread.
- Problemi di cooperazione ad ambiente locale tramite scambio messaggi, da realizzare con programmi multi-processo oppure multi-thread, eventualmente in combinazione con problemi di cooperazione ad ambiente globale.

La prova orale mira all'accertamento delle conoscenze di tipo teorico e metodologico, mediante discussione della prova scritta e degli argomenti del presente programma. L'ammissione all'orale è stabilita in base alla valutazione della prova scritta. La data dell'orale è di norma stabilita in sede di prova scritta, anche in base al numero di candidati, e può essere tenuta nella stessa giornata della prova scritta.

Gli appelli sono pubblicati nella Bachecca Esami sul portale di Ateneo. È obbligatorio prenotarsi tramite SEGREPASS. La verbalizzazione avviene in forma elettronica, per la quale è necessario che lo studente disponga del proprio PIN al momento della verbalizzazione.

Libri di testo adottati

ANCILLOTTI-1) Ancillotti, Boari, Ciampolini, Lipari. "**Sistemi Operativi**", McGraw Hill, 2a edizione, 2008

OSTEP) R.H. Arpaci-Dusseau, A.C. Arpaci-Dusseau, "**Operating Systems: Three Easy Pieces**", www.ostep.org, v1.01, 2020 (gratuito in PDF, in lingua inglese)

Libri di testo consigliati per approfondimenti

ANCILLOTTI-2) Ancillotti, Boari. "**Programmazione concorrente e distribuita**", McGraw Hill, 1a edizione, 2007

STALLINGS) W. Stallings. "**Operating Systems: Internals and Design Principles**", Pearson Education, 8a ed., 2014

SILBERSCHATZ) Silberschatz, Galvin, Gagne. "**Sistemi operativi. Concetti ed esempi**", Pearson, 10a edizione, 2019

TANENBAUM) Tanenbaum, Bos. "**I Moderni Sistemi Operativi**", Pearson Education, 4a edizione, 2019

Materiale didattico

Le presentazioni del corso sono disponibili nella sezione "File" del canale del corso sulla piattaforma **Microsoft Teams**, accessibile al link: <https://teams.microsoft.com/l/team/19%3AVw-0OGc6eA4IYuRSw3RoN1OjBdFUX5QKEIW60IX0uvc1%40thread.tacv2/conversations?groupId=789e43e6-8750-410f-82a0-99893b16bdae&tenantId=2fcfe26a-bb62-46b0-b1e3-28f9da0c45fd>

Il codice relativo agli esempi ed esercizi di programmazione del corso è disponibile sulla piattaforma online github.com, nei seguenti repository.

- **Introduzione alla shell Linux:** https://github.com/rnatella/unix_dungeon/
- **Esempi mostrati a lezione:** https://github.com/rnatella/so_esempi/
- **Esercitazione 0 (Introduzione a GIT, errori di compilazione):**
https://github.com/rnatella/so_esercitazione_0_git
https://github.com/rnatella/so_esercitazione_0_compilazione
- **Esercitazione 1 (Semafori):**
https://github.com/rnatella/so_esercitazione_1_semafori
- **Esercitazione 2 (Monitor):**
https://github.com/rnatella/so_esercitazione_2_monitor
- **Esercitazione 3 (Threads):**
https://github.com/rnatella/so_esercitazione_3_threads
- **Esercitazione 4 (Messaggi):**
https://github.com/rnatella/so_esercitazione_4_messaggi
- **Esercitazione 5 (Server multi-thread):**
https://github.com/rnatella/so_esercitazione_5_messaggi_threads
- **Raccolta di Esercizi di Programmazione Concorrente in Linux, con soluzioni:** https://github.com/rnatella/esercizi_linux/

Dispense didattiche

- a) **Unix** (*dispensa Unix.pdf*)
- b) **Scheduling del processore** (*dispensa_scheduling_Stallings.pdf*)
- c) **Compilazione e makefile** (*dispensa su makefile.pdf*)
- d) **Mutua esclusione** (*dispensa mutua esclusione-Tanenbaum.pdf*)
- e) **Problemi di programmazione concorrente** (*dispensa Ancillotti problemi progr. concorrente*)
- f) **Threads** (*The Native POSIX Thread Library for Linux.pdf*)
- g) **I/O e gestione dischi** (*DispensaIO_DISCHI.pdf*)
- h) **Sviluppo di codice per il kernel Linux** (*dispensa programmazione Linux kernel.pdf*)
- i) **Gestione dei dispositivi di memoria secondaria a stato solido** (*dispensa SSD.pdf*)
- j) **Scheduling multiprocessore** (*dispensa SMP RT - Stallings.pdf*)

- k) **Process scheduling in Linux** (*dispensa Linux scheduling.pdf*)
- l) **Memory management in Linux** (*dispensa Linux memory management.pdf*)
- m) **Il costrutto Monitor** (*dispensa Ancillotti Monitor.pdf, dispensa Stallings Monitor.pdf*)
- n) **File system in Linux** (*dispensa Linux FS.pdf*)
- o) **La virtualizzazione** (*dispensa virtualizzazione.pdf*)
- p) **Android** (*dispensa Android.pdf*)