

CORSO DI SISTEMI OPERATIVI

Prof. ROBERTO PIETRANTUONO

Programma dell'A.A. 2019/20

Obiettivi, argomenti, materiali didattici

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.

ARGOMENTI

Concetti Introduttivi. Evoluzione storica dei S.O. - Mono e multiprogrammazione - Batch, time sharing, real time - Sistemi transazionali - Richiami di elementi di architettura a supporto di un S.O. - Architetture di un S.O. - Componenti e servizi di un S.O. - Il kernel - Strutturazione a livelli del kernel - La Chiamata a Supervisore.

(dal cap. 1, libro di testo 1)

Gestione dei processi. Concetto di processo – Stati di un processo – Descrittore di un processo – Code di processi – Il cambiamento di contesto – Creazione e terminazione di un processo – Concetti di risorsa e di gestore di risorsa – I meccanismi di creazione e terminazione di processi: le primitive fork/join. La gestione dei processi nei sistemi operativi Linux e Windows. Stati di un processo UNIX e immagine di un processo UNIX – Stati dei thread Windows.

(dal cap. 2, cap 7 par 7.4, cap 8 par 8.2, libro di testo 1)

Threads, SMP. 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 – Concorrenza e Parallelismo – Speed-up nelle architetture concorrenti e parallele – La legge di Amdahl – Architetture multi-processore – Symmetric Multiprocessing.

Dal testo IS1 - **Cap. 6, 7.**

(dal cap. 4, libro di testo 2 – dal cap. 2, par. 2.10, libro di testo 1)

Scheduling dei processi. Scheduler – scheduler a breve, medio e lungo termine - Algoritmi di scheduling – Scheduling della CPU - Parametri e scelta di algoritmi di scheduling – Scheduling a Priorità - 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 – Scheduling multiprocesso: Problematiche di progetto, tipologie di scheduling multiprocesso, scheduling load sharing e gang scheduling – Gli Scheduler O(1) e CFS del sistema operativo Linux – Scheduling nel sistema operativo Windows.

(dal cap. 2, libro di testo 1 – dal cap. 9 e cap. 10, libro di testo 2) (dalla dispensa b) “Scheduling del processore”)

(dalla dispensa j) “Scheduling multiprocesso”)

(dalla dispensa k) “Process Scheduling in Linux”)

Comunicazione tra processi. Competizione, cooperazione, ed interferenza – Race conditions 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.

(dal cap 2 par. 2.7e dal Cap. 3 par. 3.1-3.6, libro di testo 1 – dal cap. 5, par. 5.5, libro di testo 2).

(dalle dispense d) “Mutua esclusione”, e) “Problemi di programmazione concorrente”, ed n) “Il costruito Monitor”)

Il 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.

(dal cap. 3 par 3.7, libro di testo 1 – dal cap 6, par. 6.3, libro di testo 2)

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 degenerare (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.

(dal cap. 4, libro di testo 1, dal cap. 8, par. 8.4 e 8.5, libro di testo 2) (dalle dispense l) “Memory Management in Linux” e m) “Memory Management in Windows XP”)

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 – Scheduling del disco nel SO Linux – I dischi a stato solido – Architetture RAID. I drive SSD.

(dal cap 5 libro di testo 1 – dal cap. 11, par. 11.9 e 11.10, libro di testo 2) (dalle dispense g) “I/O e gestione dischi” ed i) “Gestione dei dispositivi di memoria secondaria a stato solido”)

La gestione dei file. Il file system – Organizzazione 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 - Metodi di allocazione dei file: allocazione contigua, a lista concatenata e indicizzata – La gestione dei blocchi

liberi – Architettura a livelli del file system – La protezione dei file – 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 – File System Log-structured – Il File System F2FS di Linux per SSD.

(dal cap. 6 libro di testo 1 – dal cap. 12, par. 12.8 - 12.10 libro di testo 2) (dalla dispensa i) "Gestione dei dispositivi di memoria secondaria a stato solido" ed o) "File system in Linux")

La virtualizzazione. Definizione e modelli. Architetture di virtualizzazione. Virtualizzazione della CPU. Virtualizzazione della memoria. Virtualizzazione dell'I/O. Esempio di tecnologie VMware. *(dispensa sulla Virtualizzazione)*

Il SO Android. Storia e principi di progettazione di Android. Applicazioni Android. La gestione dei processi e della memoria. Inter-process communication in Android. *(dispensa su 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, wait_for_zero 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, realizzazione di un Monitor a priorità – 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.

*(dal cap. 7, par 7.4 e 7.8, libro di testo 1 – dal cap. 6, par. 6.7, libro di testo 2)
(dalla dispensa f) "Threads")*

Approfondimenti sul sistema operativo UNIX/Linux. Installazione del SO Linux – la shell e i comandi di shell – cenni sugli script bash – la compilazione dei programmi: makefile, librerie statiche e dinamiche – Estensione del kernel Linux: sviluppo di moduli kernel e implementazione di system call.

*(dal cap. 6, par. 6.8 del libro di testo 2)
(dalle dispense a) "Unix", c) "Compilazione e Makefile" e h) "Sviluppo di codice per il kernel Linux")*

MATERIALI DIDATTICI

LIBRI DI TESTO ADOTTATI

1. Ancillotti, Boari, Ciampolini, Lipari. Sistemi Operativi, McGraw Hill.
2. Stallings. Operating Systems: Internals and Design Principles, 6th ed., Pearson Education

LIBRI DI TESTO CONSIGLIATI

3. Silberschatz, Galvin, Gagne. Sistemi operativi - sesta edizione, Addison Wesley
4. Tanenbaum, I Moderni Sistemi Operativi – terza edizione, Pearson Education

LUCIDI DEL CORSO: consultabili dal sito www.federica.eu

DISPENSE DIDATTICHE (consultabili dal sito del corso):

- 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) **Memory management in Windows XP** (dispensa Windows XP memory management.pdf)
- n) **Il costrutto Monitor** (dispensa Monitor.pdf)
- o) **File system in Linux** (dispensa Linux FS.pdf)
- p) **La virtualizzazione** (dispensa virtualizzazione.pdf)
- q) **Android** (dispensa Android.pdf)