

A.A.	Nome	Settore	CFU	Corso di studi	Periodo	Ore	Moduli	Mutuato
2015/16	<i>Sistemi Operativi</i>	ING-INF/05	6	Ingegneria Informatica e delle Telecomunicazioni	Primo Semestre	48	No	No
Modulo	Nome Modulo	Tipo	Ore	Docente	SSD	Ruolo	Interno	Affidamento
No	No	Lezione ed esercitazione in laboratorio	48	Mario Collotta	ING-INF/05	RD	Si	Istituzionale

Obiettivi:

- **Conoscenza e capacità di comprensione (knowledge and understanding):** Lo studente al termine del corso avrà conoscenza delle problematiche inerenti i sistemi operativi e le metodologie di gestione dei processi e della concorrenza.
- **Conoscenza e capacità di comprensione applicate (applying knowledge and understanding):** Lo studente sarà in grado di utilizzare semplici strumenti per la programmazione in ambiente unix di processi, thread e IPC.
- **Autonomia di giudizio (making judgements):** Lo studente sarà in grado sia di effettuare un'analisi di un sistema e quindi arrivare a capire il suo funzionamento, ma anche di progettare e implementare sistemi per la risoluzione di problemi reali legati al funzionamento dei processi o dei thread e alla loro interazione in un sistema operativo e in un sistema distribuito.
- **Abilità comunicative (communication skills):** Lo studente acquisirà la capacità di comunicare ed esprimere problematiche inerenti i sistemi operativi. Sarà in grado di sostenere conversazioni su tematiche relative alla realizzazione di politiche di scheduling, di gestione delle risorse e dei deadlock.
- **Capacità di apprendere (learning skills):** Lo studente avrà acquisito le problematiche di realizzazione di applicazioni per sistemi operativi e per sistemi distribuiti.

Conoscenze e abilità da acquisire

- Conoscenza della classificazione dei Sistemi Operativi: Batch, Interattivi, Real-time, Macchine virtuali
- Conoscenza dell'architettura dei sistemi operativi come gestore di risorse
- Conoscenza dell'organizzazione del kernel di un sistema operativo
- Conoscenza e gestione dei processi e thread d'esecuzione concorrenti
- Capacità di utilizzare le system call per la creazione, la sincronizzazione e la terminazione di processi e dei thread d'esecuzione
- Capacità di amministrazione di sistema: utilizzo di comandi, shell script e filtri in ambiente Unix-Linux.

Programma:

Il corso ha lo scopo di introdurre gli elementi principali dell'architettura dei sistemi operativi, illustrandone la struttura di base e il funzionamento dal punto di vista dell'utilizzatore avanzato e del programmatore di sistema. Saranno presentate, inoltre, le tecniche di programmazione di sistema mediante l'analisi e l'utilizzo delle chiamate di sistema di Unix.

- **Introduzione ai Sistemi Operativi (3 ore).**
 - Definizione di Sistema operativo. Ruolo del S.O. in un sistema di elaborazione.
 - Gestione delle risorse e protezioni. Concetto di Macchina virtuale.
 - Il SO e l'utente: Shell di SO.
 - Caratteristiche fondamentali dei moderni SO: Interattività, Multiprogrammazione, Time-sharing.
 - Sistemi real-time. SO di rete: definizione e aspetti essenziali.
 - Introduzione al sistema operativo UNIX
- **Processi, thread e gestione della CPU(12 ore).**
 - Processi e thread. Processi in UNIX. Contesto di kernel e contesto utente. System call per la creazione e la gestione dei processi. Esempi di programmi.
 - Bootstrap del sistema UNIX. Processi demoni, orfani, zombie.
 - Segnali e loro gestione in UNIX. System call signal, kill, wait. Esempi di programmi.
 - Il modello multithreading. Multithreading in Linux.
 - Schedulazione della CPU. Schedulazione real-time.
 - Esercitazioni di laboratorio
- **InterProcess Communication (IPC) e Gestione della Concorrenza (15 ore).**
 - Generalità sull'IPC. Sezione critica. Mutua esclusione. Il problema del produttore/consumatore: definizione e soluzioni
 - Primitive Sleep e Wakeup. Semafori binari e Semafori generalizzati
 - Esempi di programmi.
 - IPC in UNIX.
 - Semafori in Unix.
 - Code di messaggi e Shared Memory.
 - Il problema dei lettori/scrittori.
 - Esercitazioni di laboratorio
- **Deadlock(6 ore).**
 - Definizione del problema. Caratterizzazione dei deadlock. Strategie di detection, prevention, avoidance, recovery dei deadlock. Grafo di allocazione delle risorse. Algoritmo del banchiere.

- Il problema dei filosofi affamati: definizione ed esempi di programmi risolutivi.
- Esercitazioni di laboratorio
- **Gestione della Memoria del File System e delle periferiche di Input/Output (6 ore)**
 - Rilocazione statica e dinamica. Memory Management Unit. Dynamic storage allocation e memorizzazione dello spazio libero. Frammentazione interna ed esterna. Swapping. Paginazione e segmentazione. Swapping Unix. Memoria virtuale. Thrashing e principio di località dei riferimenti.
 - Organizzazione esterna ed interna del File System di UNIX. Pathname assoluti e relativi. Struttura delle directory e dei file in Unix. Inode. File speciali a blocchi e a caratteri. System call per la gestione dei file. Mount e umount di file system. Buffer cache.
 - Dispositivi I/O a blocchi e a caratteri. I/O hardware: device controllers, Direct Memory Access(DMA).I/O software: gestione degli interrupt, device drivers, software device-independent.
 - Esercitazioni di laboratorio
- **Esempi di progetto di applicazioni distribuite in ambiente UNIX (6 ore).**
 - Modello client-server. Server iterativi e server concorrenti. Associazioni e Socket. Chiamate di sistema relative ai socket.
 - Esercitazioni di laboratorio

Testi consigliati:

- Materiale fornito dal docente (scaricabile dalla pagina web del docente);
- Silberschatz, P. B. Galvin, G. Gagne, “Sistemi Operativi: Concetti ed esempi, Settima Edizione”, Pearson/Addison-Wesley.
- Neil Matthew, Richard Stones: Beginning Linux Programming.
- Ancilotti, Boari, Ciampolini e Lipari: “Sistemi Operativi” - Mc Graw - Hill

Modalità di accertamento delle competenze:

Prova al Calcolatore, Elaborato finale e Prova Orale,

Argomenti o insegnamenti propedeutici:

Fondamenti di Informatica.