



Università degli Studi di Enna "Kore"
Facoltà di Ingegneria ed Architettura
Anno Accademico 2017 – 2018

A.A.	Settore Scientifico Disciplinare		CFU	Insegnamento	Ore di aula		Mutuazione	
2017/18	ING-INF/05		9	Sistemi Operativi	76		No	
Classe	Corso di studi			Tipologia di insegnamento	Anno di corso e Periodo		Sede delle lezioni	
L8	Ingegneria Informatica e delle Telecomunicazioni			Caratterizzante	2° Anno Primo Semestre		Facoltà di Ingegneria e Architettura	
N° Modulo	Nome Modulo	Tipologia lezioni	Ore	Docente	SSD	Ruolo	Interno	Affidamento
-	-	Lezioni Frontali	56	Mario Collotta mario.collotta@unikore.it	ING-INF/05	RD	Si	Istituzionale
		Esercitazione di laboratorio	20					

Obiettivi formativi

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.

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

Prerequisiti

Lo studente dovrà, preliminarmente, essere in possesso delle conoscenze necessarie alla formalizzazione del processo di risoluzione di un problema in forma algoritmica e alla sua implementazione mediante un linguaggio di programmazione di alto livello.



Argomenti o insegnamenti propedeutici:

Fondamenti di Informatica.

Risultati di apprendimento (Descrittori di Dublino):

- **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.

Contenuti e struttura del corso

N. ARGOMENTO	TIPOLOGIA	DURATA
1 Introduzione ai Sistemi Operativi. <ul style="list-style-type: none">○ Definizione di Sistema operativo. Ruolo del S.O. in un sistema di elaborazione.○ Gestione delle risorse e protezioni.○ Il SO e l'utente: Shell di SO.○ Caratteristiche fondamentali dei moderni SO: Interattività, Multiprogrammazione, Time-sharing.	Frontale	3h



- Sistemi real-time: definizione e aspetti essenziali.
- Introduzione al sistema operativo UNIX

2 Processi, thread e gestione della CPU.

- 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 (7h)

Frontale ed
esercitazione di
laboratorio

20h

3 InterProcess Communication (IPC) e Gestione della Concorrenza.

- 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 (10h)

Frontale ed
esercitazione di
laboratorio

20h

4 Deadlock.

- 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 (3h)

Frontale ed
esercitazione di
laboratorio

9h

5 Esempi di progetto di applicazioni distribuite in ambiente UNIX.

Esercitazione

6h



- Modello client-server. Server iterativi e server concorrenti. Associazioni e Socket. Chiamate di sistema relative ai socket.

5 Sistemi Real-Time

- Introduzione al problema del real-time e alla schedulazione di processi.
- Algoritmi per la schedulazione di processi aperiodici e di processi periodici:
- Algoritmi per la schedulazione di processi misti a priorità statica e a priorità dinamica:
- Algoritmi di schedulazione di processi aperiodici
- Algoritmi di schedulazione preemptive e non preemptive
- Server aperiodici a priorità statica e dinamica.
- Protocolli per l'accesso a risorse condivise: il problema dell'inversione di priorità e analisi delle possibili soluzioni; valutazione dei tempi di bloccaggio; considerazioni implementative.

Frontale ed
esercitazione

18h

Testi adottati

Testi principali:

- Neil Matthew, Richard Stones: Beginning Linux Programming.
- Sistemi in tempo-reale, Pitagora Editrice (autore Giorgio Buttazzo)

Testi consigliati:

- Ancilotti, Boari, Ciampolini e Lipari: "Sistemi Operativi" - Mc Graw - Hill
- Silberschatz, P. B. Galvin, G. Gagne, "Sistemi Operativi: Concetti ed esempi, Settima Edizione", Pearson/Addison-Wesley.

Materiale didattico a disposizione degli studenti: Materiale fornito dal docente (scaricabile dalla pagina web del docente).

Modalità di accertamento delle competenze

L'accertamento delle competenze avverrà attraverso un'unica prova orale che verterà:

1. sulla discussione di due esercizi proposti dal docente e da svolgere al computer in laboratorio in un tempo massimo di 90 minuti sui Processi e Thread UNIX – contribuisce per il 40% nella valutazione finale;



Università degli Studi di Enna "Kore"
Facoltà di Ingegneria e Architettura

2. sulla discussione di un elaborato (preparato a casa e a scelta dello studente) che implementi in linguaggio C utilizzando le system call di UNIX un'applicazione distribuita Client/Server, con server concorrente. E' possibile trarre spunto dall'esempio discusso in aula, (vedi slide pubblicate nella sezione "Documenti") – contribuisce per il 30% nella valutazione finale;
3. sulla discussione di un approfondimento su alcune delle tematiche affrontate durante il corso – contribuisce per il 30% sulla valutazione finale. Le domande di approfondimento riguarderanno i metodi di dimensionamento dei sistemi operativi real-time, le scelte progettuali per la gestione dell'IPC e delle priorità nei sistemi real-time.

Il colloquio si intende superato, con la votazione di 18/30, quando lo studente dimostra:

- minime conoscenze tecniche di base degli aspetti di gestione dei sistemi operativi;
- capacità di autonoma applicazione dei metodi progettuali in relazione a semplici problemi per la gestione della concorrenza nei sistemi operativi;
- capacità di elaborazione delle conoscenze acquisite per formulare semplici valutazioni di funzionalità in termini di performance degli algoritmi e delle soluzioni utilizzati.

Il voto di 30/30, con eventuale lode, è assegnato quando lo studente dimostra:

- piena conoscenza tecnica degli aspetti di gestione dei sistemi operativi e dei sistemi real-time;
- ampia capacità di autonoma applicazione dei criteri metodi progettuali in relazione a semplici problemi per la gestione della concorrenza nei sistemi operativi;
- ampia capacità di elaborazione delle conoscenze acquisite per formulare valutazioni di funzionalità in termini di performance degli algoritmi e delle soluzioni utilizzati.

Orari di lezione e date di esame

Orari delle lezioni: <http://www.unikore.it/index.php/ingegneria-informatica-attivita-didattiche/calendario-lezioni>

Calendario esami: <http://www.unikore.it/index.php/ingegneria-informatica-esami/calendario-esami>

Modalità e orari di ricevimento

Gli orari di ricevimento saranno pubblicati sulla pagina personale del docente:

<http://www.unikore.it/index.php/ingegneria-informatica-persone/docenti-del-corso>



Università degli Studi di Enna "Kore"
Facoltà di Ingegneria e Architettura

