



UNIVERSITÀ DEGLI STUDI DI ENNA "KORE"

Facoltà di Ingegneria e Architettura

Anno Accademico 2021/2022

Corso di studi in Ingegneria Informatica, classe di laurea L8

Insegnamento	Sistemi Operativi
CFU	9
Settore Scientifico Disciplinare	ING-INF/05
Metodologia didattica	Lezioni frontali
Nr. ore di aula	54
Nr. ore di studio autonomo	171
Nr. ore di laboratorio	-
Mutuazione	No
Annualità	II anno
Periodo di svolgimento	I semestre

Docente	E-mail	Ruolo ⁱ	SSD docente
Giovanni Pau	giovanni.pau@unikore.it	PA	ING-INF/05

Propedeuticità	Fondamenti di Informatica
Sede delle lezioni	Facoltà di Ingegneria e Architettura

Moduli
No.

Orario delle lezioni
L'orario delle lezioni sarà pubblicato sulla pagina web del corso di laurea.

<https://unikore.it/index.php/it/ingegneria-informatica-attivita-didattiche/calendario-lezioni>

Obiettivi formativi

L'obiettivo del corso è fornire allo studente la capacità di comprendere come i sistemi operativi gestiscono e controllano le risorse del calcolatore al fine di ottimizzare l'efficienza e la facilità d'uso. Il corso mira al raggiungimento dei seguenti obiettivi:

- conoscere il ruolo del sistema operativo come intermediario tra le applicazioni e l'hardware;
- saper valutare la necessità di sovrapporre attività di CPU e di I/O per aumentare l'efficienza;
- comprendere il concetto di processo/thread e i principali modelli di interazione;
- conoscere i criteri e gli algoritmi di scheduling della CPU;
- conoscere le nozioni di base per l'utilizzo di UNIX/Linux e per la gestione dei comandi;
- saper applicare le principali chiamate di sistema offerte da UNIX/Linux alle applicazioni.

Contenuti del Programma

N.	Argomento	Tipologia	Durata in ore
1	Introduzione <ul style="list-style-type: none">• Caratteristiche di un sistema operativo• Organizzazione di un sistema elaborativo	Frontale	3

	<ul style="list-style-type: none"> • Architettura degli elaboratori • Attività del sistema operativo • Gestione delle risorse 		
2	Strutture dei sistemi operativi: <ul style="list-style-type: none"> • Servizi di un sistema operativo • Interfaccia con l'utente • Chiamate di sistema • Progettazione e realizzazione di un sistema operativo 	Frontale	4
3	Processi: <ul style="list-style-type: none"> • Concetto di processo • Scheduling dei processi • Operazioni sui processi • Comunicazioni tra processi (IPC) 	Frontale	6
4	Thread e concorrenza: <ul style="list-style-type: none"> • Programmazione multicore • Multithreading • Problematiche e programmazione multithreading 	Frontale	6
5	Scheduling della CPU: <ul style="list-style-type: none"> • Algoritmi e criteri di scheduling • Scheduling di processi e thread 	Frontale	6
6	Stallo dei processi: <ul style="list-style-type: none"> • Caratterizzazione delle situazioni di stallo • Metodi per la gestione, prevenzione e risoluzione delle situazioni di stallo 	Frontale	6
7	Memoria centrale, virtuale e di massa: <ul style="list-style-type: none"> • Allocazione contigua della memoria • Paginazione • Swapping • Struttura dei dispositivi di memorizzazione • Approcci per la memorizzazione • Caso studio sui sistemi di I/O 	Frontale	4
8	Interfaccia del file system: <ul style="list-style-type: none"> • Concetto di file • Metodi di accesso • Struttura delle directory 	Frontale	3
9	Programmazione in C: <ul style="list-style-type: none"> • Chiamate di sistema per la gestione e sincronizzazione di processi e thread • Segnali per la gestione e sincronizzazione di processi e thread • IPC (pipe, fifo, shared memory, code di messaggi), semafori, mutex, variabili condizione • Comunicazione nei sistemi client-server 	Frontale	16

Risultati di apprendimento (descrittori di Dublino)

I risultati di apprendimento attesi sono definiti secondo i parametri europei descritti dai cinque descrittori di Dublino.

1. **Conoscenza e capacità di comprensione:** lo studente al termine del corso avrà conoscenza dei principali sistemi operativi e delle metodologie di gestione dei processi/thread e della concorrenza.
2. **Conoscenza e capacità di comprensione applicate:** lo studente sarà in grado di analizzare interazioni tra processi/thread UNIX, individuare gli strumenti di comunicazione interprocesso e le primitive di sistema necessarie, e realizzare programmi in C che le utilizzano.

3. **Autonomia di giudizio:** lo studente sarà in grado di effettuare un'ampia analisi sul sistema operativo, definendone le caratteristiche ed il funzionamento, ma anche di progettare ed implementare applicativi per la risoluzione di problemi reali legati al funzionamento dei processi o dei thread ed alla loro interazione in un sistema operativo.
4. **Abilità comunicative:** lo studente acquisirà la capacità di comunicare ed esprimere problematiche inerenti all'oggetto del corso. Sarà in grado di sostenere conversazioni sui principali algoritmi di scheduling della CPU per sistemi interattivi e sulla gestione delle risorse e dei deadlock.
5. **Capacità di apprendere:** lo studente avrà acquisito le conoscenze sia delle caratteristiche dei sistemi operativi general-purpose sia delle interazioni tra processi in ambiente globale individuando forme di sincronizzazione attraverso semafori, mutex, variabili condizione e segnali.

Testi per lo studio della disciplina

Abraham Silberschatz - Peter Baer Galvin - Greg Gagne, "Sistemi operativi - Concetti ed esempi - 10/Ed.", Pearson

https://www.pearson.it/opera/pearson/0-6795-sistemi_operativi

W. Richard Stevens, Stephen A. Rago, "Advanced Programming in the UNIX® Environment - Third Edition", Addison-Wesley Professional

<https://www.amazon.it/Advanced-Programming-Environment-Richard-Stevens/dp/0321637739>

Simone Piccardi, "GaPiL: Guida alla Programmazione in Linux"

<http://gapil.truelite.it>

Andrew S. Tanenbaum - Herbert Bos, "I moderni sistemi operativi - 4/Ed", Pearson

https://www.pearson.it/opera/pearson/0-6260-i_moderni_sistemi_operativi

Neil Matthew, Richard Stones, "Beginning Linux Programming", Wiley & Sons

<https://www.ibs.it/beginning-linux-programming-libro-inglese-neil-matthew-richard-stones/e/9780470147627>

Le slide proiettate a lezione che non sono protette da copyright sono fornite dal docente titolare dell'insegnamento e messe a disposizione degli studenti sul sito web dell'Università.

Modalità di accertamento delle competenze

L'accertamento delle competenze avverrà attraverso una prova orale, inizialmente basata sulla proposta progettuale dallo studente (o da un gruppo di massimo due studenti nel caso di lavoro in gruppo). L'elaborato deve rispettare le direttive che saranno pubblicate dal docente nella sua pagina web. La discussione della proposta progettuale è volta a dimostrare l'acquisizione degli argomenti inerente alla programmazione in C erogati durante le lezioni frontali. Successivamente, il colloquio si focalizzerà su altri argomenti trattati durante il corso e non coperti dalla stesura dell'elaborato. La durata del colloquio dipenderà dalla profondità ed ampiezza sia della proposta progettuale sia degli argomenti esposti dallo studente ed indicativamente durerà tra 20 e 60 minuti. Ove fosse necessario, gli esaminandi saranno ripartiti in più giornate, secondo un calendario determinato nel giorno dell'appello ovvero, se possibile, anticipatamente sulla base delle prenotazioni pervenute. La calendarizzazione sarà, in tal caso, opportunamente pubblicizzata. La valutazione dell'apprendimento sarà focalizzata sulla valutazione dei risultati attesi, in accordo con i descrittori di Dublino.

La prova di esame si intende superata con una votazione minima di 18/30 quando lo studente dimostra:

- minima conoscenza e comprensione degli argomenti trattati;
- limitata capacità nell'applicazione delle conoscenze acquisite;
- sufficiente capacità espositiva.

La votazione di 30/30, eventualmente con lode, è assegnata quando lo studente dimostra:

- ottima conoscenza e comprensione degli argomenti trattati;
- ottima capacità nell'applicazione delle conoscenze acquisite;
- eccellente capacità espositiva.

La prova di esame si intende non superata se lo studente mostra un livello insufficiente di conoscenza degli argomenti trattati e non dimostra una sufficiente capacità nell'applicazione delle conoscenze acquisite.

Date di esame

Le date di esami saranno pubblicate sulla pagina web del corso di laurea.

<https://unikore.it/index.php/it/ingegneria-informatica-esami/calendario-esami>

Modalità e orario di ricevimento

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

<https://unikore.it/index.php/it/ingegneria-informatica-persone/docenti-del-corso/itemlist/category/2395-prof-pau-giovanni>

ⁱ PO (professore ordinario), PA (professore associato), RTD (ricercatore a tempo determinato), RU (Ricercatore a tempo indeterminato), DC (Docente a contratto).