



UNIVERSITÀ DEGLI STUDI DI PALERMO

DIPARTIMENTO	Matematica e Informatica
ANNO ACCADEMICO OFFERTA	2023/2024
ANNO ACCADEMICO EROGAZIONE	2024/2025
CORSO DILAUREA	INFORMATICA
INSEGNAMENTO	SISTEMI OPERATIVI
TIPO DI ATTIVITA'	B
AMBITO	50166-Discipline Informatiche
CODICE INSEGNAMENTO	16784
SETTORI SCIENTIFICO-DISCIPLINARI	ING-INF/05
DOCENTE RESPONSABILE	VALENTI CESARE Professore Associato Univ. di PALERMO FABIO
ALTRI DOCENTI	
CFU	6
NUMERO DI ORE RISERVATE ALLO STUDIO PERSONALE	86
NUMERO DI ORE RISERVATE ALLA DIDATTICA ASSISTITA	64
PROPEDEUTICITA'	05880 - PROGRAMMAZIONE E LABORATORIO C.I.
MUTUAZIONI	
ANNO DI CORSO	2
PERIODO DELLE LEZIONI	2° semestre
MODALITA' DI FREQUENZA	Facoltativa
TIPO DI VALUTAZIONE	Voto in trentesimi
ORARIO DI RICEVIMENTO DEGLI STUDENTI	VALENTI CESARE FABIO Mercoledì 14:30 18:30 da concordare via email

DOCENTE: Prof. CESARE FABIO VALENTI

PREREQUISITI	Conoscenze di base di architetture degli elaboratori e delle principali strutture dati.
RISULTATI DI APPRENDIMENTO ATTESI	<p>- Conoscenza e capacita' di comprensione. Lo studente acquisisce la capacita' di riconoscere ed organizzare in autonomia gli argomenti base del corso, di utilizzare le conoscenze apprese in campi applicativi specifici, con particolare riferimento all'interfacciamento ed alla modifica di sistemi operativi.</p> <p>- Capacita' di applicare conoscenza e comprensione. Lo studente viene messo in grado di capire ed analizzare le funzioni di un sistema operativo e le scelte relative adottate dai suoi programmatori. Realizzazione di semplici applicativi single- e multi-thread.</p> <p>- Autonomia di giudizio. Lo studente viene messo in grado di fornire una chiave di lettura critica dei risultati ottenuti in relazione al fenomeno studiato e alle metodologie utilizzate.</p> <p>- Abilita' comunicative. Lo studente impara a sintetizzare ed esporre quanto appreso durante il corso; adattare il linguaggio della tecnologia al contesto di riferimento e all'interlocutore, eventualmente non esperto.</p> <p>- Capacita' d'apprendimento. Capacita' di aggiornamento con la consultazione delle pubblicazioni scientifiche proprie del settore dei sistemi operativi e, piu' in generale, della teoria degli algoritmi. Capacita' di seguire, utilizzando le conoscenze acquisite nel corso, sia master di secondo livello, sia corsi d'approfondimento, sia seminari specialistici nel settore.</p>
VALUTAZIONE DELL'APPRENDIMENTO	<p>Prova di pratica e orale per accertare il possesso delle competenze e delle conoscenze disciplinari previste dal corso. Saranno anche valutate la maturita' del candidato e la sua chiarezza espositiva.</p> <p>Eccellente (29-30 / 30 e lode) - eccellente conoscenza degli argomenti, eccellente proprieta' di linguaggio, buona capacita' analitica, lo studente e' in grado di applicare le conoscenze per risolvere autonomamente i problemi proposti, fornendo soluzioni originali.</p> <p>Ottimo (27-28) - ottima padronanza degli argomenti con discreta proprieta' di linguaggio, lo studente e' in grado di applicare le conoscenze per risolvere senza difficolta' i problemi proposti.</p> <p>Buono (24-26) - buona padronanza degli argomenti con discreta proprieta' di linguaggio, lo studente e' in grado di applicare le conoscenze per risolvere i problemi proposti.</p> <p>Sufficiente (18-23) - lo studente non ha piena padronanza degli argomenti principali ma ne possiede le conoscenze; appena sufficiente la proprieta' di linguaggio con scarsa capacita' di applicare autonomamente le conoscenze acquisite.</p> <p>Insufficiente - lo studente non possiede una conoscenza accettabile dei contenuti degli argomenti trattati nell'insegnamento.</p>
OBIETTIVI FORMATIVI	<p>Il corso ha come obiettivo formativo l'insegnamento delle seguenti nozioni:</p> <ul style="list-style-type: none">- fondamenti di un sistema operativo- struttura di un sistema operativo- gestione dei processi- gestione della memoria- gestione dello spazio dati- rappresentazione binaria dell'informazione <p>Applicare la teoria dell'informatica e della matematica per risolvere problemi e presentare in modo esauriente i risultati e i metodi della soluzione a un pubblico professionale o non professionale.</p> <p>Prevedere il comportamento dei sistemi in caso di eventi casuali utilizzando le conoscenze di probabilità e previsione e informare gli utenti del loro potenziale comportamento.</p> <p>Valutare la sicurezza di un sistema utilizzando le conoscenze di riservatezza, disponibilita' e integrita' con una comprensione dei rischi, delle minacce, delle vulnerabilita' e dei metodi di attacco, e mettere in relazione l'impatto sociale ed etico con i componenti del sistema.</p>
ORGANIZZAZIONE DELLA DIDATTICA	Lezioni e laboratorio.
TESTI CONSIGLIATI	<p>Qualsiasi testo di livello universitario che descriva tutti gli argomenti trattati durante il corso. Le lezioni saranno supportate da presentazioni che NON sostituiscono i seguenti testi consigliati. / Any university-level book that describes all the topics discussed during the course. The lessons will be supported by presentations that do NOT supersede the following recommended texts.</p> <p>Testo principale / main textbook Silberschatz A., Galvin P.B., Gagne G. "Sistemi Operativi: concetti ed esempi", 10 Ed. Pearson, 2019. ISBN-10:8891904554 Per l'attivita' di laboratorio / Lab activities Kerrisk M. "The Linux Programming Interface: A Linux and UNIX System Programming Handbook", No Starch Press, 2010. ISBN-10:1593272200</p>

Per consultazione / Further reading
Tanenbaum A.S., Bos H. "I moderni sistemi operativi", 4 Ed. Pearson, 2019.
ISBN-10:8891906255

PROGRAMMA

ORE	Lezioni
2	Descrizione delle principali architetture e componenti dei moderni hardware e sistemi operativi. Interfacce utente. API e chiamate di sistema. Gestione dei processi e comunicazioni.
3	Processi e thread in dettaglio. Schedulazione e sincronizzazione.
3	Processori single- e multi-core. Concorrenza. Modelli per il multi-threading. Gruppi di thread.
3	Principio di localita'. Sistemi con e senza prelazione. Confronto tra i principali algoritmi di scheduling. Bilanciamento del carico. Scheduling real-time.
3	Sezione critica. Soluzione di Peterson. Barriere; istruzioni atomiche, lock mutex; semafori; monitor.
3	Problemi classici di sincronizzazione: produttore-consumatore; lettori-scrittori; filosofi a cena. Memoria transazionale.
3	Stallo, stallo attivo e inedia. Grafo di assegnazione delle risorse. Prevenzione, aggiramento e rilevamento degli stalli. Ripristino dopo lo stallo.
3	Memoria centrale. Indirizzi logici e fisici. Caricamento, allocazione e protezione. Frammentazione. Paginazione. Avvicendamento.
3	Memoria virtuale. Algoritmi per la sostituzione delle pagine. Pagina mancante. Thrashing. Working set. Allocazione per il kernel.
3	Memoria di massa. HDD, SSD, NAND flash. Scheduling. Rilevamento e correzione di errori. Struttura RAID.
3	Struttura e allocazione del file system. File e directory. Protezione. Gestione dello spazio libero. Ripristino.

ORE	Laboratori
4	Interfaccia grafica e interfaccia a riga di comando.
4	Principio di localita' e simulazione dei page fault.
4	Simulazione degli algoritmi di schedulazione.
6	Processi, thread, fork-join.
6	Problemi classici di sincronizzazione.
4	Semafori e monitor.
4	Gestione avanzata dei file binari.