



# UNIVERSITÀ DEGLI STUDI DI PALERMO

<b>DIPARTIMENTO</b>	Matematica e Informatica
<b>ANNO ACCADEMICO OFFERTA</b>	2021/2022
<b>ANNO ACCADEMICO EROGAZIONE</b>	2022/2023
<b>CORSO DILAUREA</b>	INFORMATICA
<b>INSEGNAMENTO</b>	SISTEMI OPERATIVI
<b>TIPO DI ATTIVITA'</b>	B
<b>AMBITO</b>	50166-Discipline Informatiche
<b>CODICE INSEGNAMENTO</b>	16784
<b>SETTORI SCIENTIFICO-DISCIPLINARI</b>	ING-INF/05
<b>DOCENTE RESPONSABILE</b>	VALENTI CESARE      Professore Associato      Univ. di PALERMO FABIO
<b>ALTRI DOCENTI</b>	
<b>CFU</b>	6
<b>NUMERO DI ORE RISERVATE ALLO STUDIO PERSONALE</b>	86
<b>NUMERO DI ORE RISERVATE ALLA DIDATTICA ASSISTITA</b>	64
<b>PROPEDEUTICITA'</b>	05880 - PROGRAMMAZIONE E LABORATORIO C.I.
<b>MUTUAZIONI</b>	
<b>ANNO DI CORSO</b>	2
<b>PERIODO DELLE LEZIONI</b>	2° semestre
<b>MODALITA' DI FREQUENZA</b>	Facoltativa
<b>TIPO DI VALUTAZIONE</b>	Voto in trentesimi
<b>ORARIO DI RICEVIMENTO DEGLI STUDENTI</b>	VALENTI CESARE FABIO Mercoledì 14:30 18:30 da concordare via email

DOCENTE: Prof. CESARE FABIO VALENTI

<b>PREREQUISITI</b>	Conoscenze di base di architetture degli elaboratori e delle principali strutture dati.
<b>RISULTATI DI APPRENDIMENTO ATTESI</b>	<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.</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 e 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> <p>- Competenze acquisite. Applicare la conoscenza della teoria informatica e della matematica per risolvere problemi e presentare in modo esauriente i risultati e i metodi di risoluzione per un pubblico professionale o non professionale. Prevedere il comportamento dei sistemi sotto eventi casuali usando la conoscenza della probabilità e dell'aspettativa e informare gli utenti del suo potenziale comportamento. Calcolare il tempo medio di accesso alla memoria sotto una varietà di configurazioni di cache e di memoria e sviluppare una breve relazione sui risultati. Calcolare il tempo medio di accesso alla memoria e descrivere i compromessi nelle prestazioni della gerarchia di memoria in termini di capacità, tasso di miss/hit, e tempo di accesso per una società di ingegneria locale.</p>
<b>VALUTAZIONE DELL'APPRENDIMENTO</b>	<p>Prova facoltativa per l'autovalutazione dello studente. Prova orale finale 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+) - 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>
<b>OBIETTIVI FORMATIVI</b>	<p>Il corso ha come obiettivo formativo l'insegnamento delle seguenti nozioni:</p> <ul style="list-style-type: none"><li>- fondamenti di un sistema operativo</li><li>- struttura di un sistema operativo</li><li>- gestione dei processi</li><li>- gestione della memoria</li><li>- gestione dello spazio dati</li><li>- rappresentazione binaria dell'informazione</li></ul>
<b>ORGANIZZAZIONE DELLA DIDATTICA</b>	Lezioni.
<b>TESTI CONSIGLIATI</b>	

	<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-13: 978-8891904553</p> <p>Per consultazione / Further reading Tanenbaum A.S., Bos H. "I moderni sistemi operativi", 4 Ed. Pearson, 2019. ISBN-13: 978-8891906250</p>
--	---

### PROGRAMMA

ORE	Lezioni
3	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	Introduzione al file system. File e directory. Protezione.
3	Struttura e allocazione del file system. 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 dei privilegi e delle priorita' dei task.
4	Gestione avanzata dei file binari.