



# UNIVERSITÀ DEGLI STUDI DI PALERMO

<b>DIPARTIMENTO</b>	Matematica e Informatica
<b>ANNO ACCADEMICO OFFERTA</b>	2018/2019
<b>ANNO ACCADEMICO EROGAZIONE</b>	2019/2020
<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>	INF/01
<b>DOCENTE RESPONSABILE</b>	VALENTI CESARE      Professore Associato      Univ. di PALERMO FABIO
<b>ALTRI DOCENTI</b>	
<b>CFU</b>	9
<b>NUMERO DI ORE RISERVATE ALLO STUDIO PERSONALE</b>	153
<b>NUMERO DI ORE RISERVATE ALLA DIDATTICA ASSISTITA</b>	72
<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>	Si richiede allo studente una discreta familiarità con il linguaggio C, necessario per la parte di programmazione concorrente basata su librerie di message-passing. Cenni di architetture dei calcolatori possono essere utili per una migliore comprensione di alcune nozioni.
<b>RISULTATI DI APPRENDIMENTO ATTESI</b>	<p>Lo studente dovrà acquisire 1) la conoscenza del linguaggio proprio della disciplina per comprenderlo e utilizzarlo appropriatamente in relazione a diversi contesti; 2) la conoscenza della struttura e delle principali funzionalità di un SO; 3) le conoscenze di base e le tecniche elementari per la gestione dei tipici problemi di un S.O. (condivisione di risorse, accesso in mutua esclusione, protezione); 4) la capacità di riconoscere ed utilizzare appropriatamente gli elementi di base delle nozioni apprese.</p> <p><b>Conoscenza e capacità di comprensione.</b> Lo studente acquisisce la capacità di riconoscere, ed organizzare in autonomia gli argomenti base del corso, di utilizzare le conoscenze apprese (in particolare, le metodologie strumentali) in campi applicativi specifici, con particolare riferimento all'interfacciamento ed alla modifica di SO.</p> <p><b>Capacità di applicare conoscenza e comprensione</b> Lo studente viene messo in grado di capire ed analizzare le funzioni di un SO e le scelte relative adottate dai suoi programmatori.</p> <p><b>Autonomia di giudizio</b> 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><b>Abilità comunicative</b> Lo studente impara a 1) sintetizzare, esporre oralmente e comunicare tramite i mezzi propri dell'informatica quanto appreso durante il corso; 2) adattare il linguaggio della tecnologia al contesto di riferimento e all'interlocutore, anche non esperto.</p> <p><b>Capacità di apprendimento</b> Lo studente impara a consultare la letteratura scientifica di base ed a rielaborare quanto appreso attraverso l'adattamento alle condizioni e ai limiti imposti dall'eventuale committente e/o dal tipo di problema da risolvere nel quadro della ricerca scientifica.</p>
<b>VALUTAZIONE DELL'APPRENDIMENTO</b>	<p>L'esame finale è composto da una prova scritta semi-strutturata. La prova è composta da 25 domande a risposta chiusa/multipla, con tre opzioni possibili, più due domande a risposta aperta. Ogni domanda a risposta multipla consente di attribuire 1 punto in caso di risposta corretta, oppure 0 in caso di risposta errata/non data. Ogni domanda a risposta aperta consente di ottenere un punteggio variabile da 0 (risposta non data o completamente non pertinente) a 3 (risposta esatta, massima proprietà di linguaggio e capacità di sintesi dei concetti), con valori interi intermedi possibili per risposte di qualità giudicata tra i due estremi. La prova consente quindi di ottenere un punteggio massimo di 31, che costituisce il presupposto per ottenere il voto finale di 30 e lode, a seguito di breve colloquio orale di conferma con il docente. Le domande incluse nella prova coprono l'intero programma, e il numero di domande per ogni argomento è percentualmente uguale al "peso" relativo delle ore dedicate a quell'argomento rispetto al totale delle ore di corso. Tale scelta permette di avere prove effettivamente confrontabili, di garantire una elevata obiettività e significatività nella valutazione, e di lasciare comunque allo studente la possibilità di esprimere i concetti acquisiti al fine di poterne valutare anche la proprietà di linguaggio, minimizzando al contempo i tempi di esecuzione complessiva dell'esame. Le fasce di punteggio corrispondono ai giudizi qualitativi seguenti: 18-20: la conoscenza della materia e le competenze di sistemi operativi sono sufficienti; 21-23: la conoscenza della materia e le competenze di sistemi operativi sono discrete; 24-25: la conoscenza della materia e le competenze di sistemi operativi sono buone; 26-27: la conoscenza della materia e le competenze di sistemi operativi sono molto buone; 28-30 e lode: la conoscenza della materia e le competenze di sistemi operativi sono ottime.</p>
<b>OBIETTIVI FORMATIVI</b>	Il corso ha come obiettivo formativo l'insegnamento delle seguenti nozioni: fondamenti di Sistemi Operativi (SO) struttura di un SO gestione dei processi in un SO gestione della memoria gestione dello spazio dati

	<p>esempi di programmazione concorrente tramite message-passing  rappresentazione binaria dell'informazione  Il corso vuole inoltre fornire strumenti per l'armonizzazione di queste conoscenze e per la conoscenza critica ed autonoma degli argomenti trattati.</p>
<b>ORGANIZZAZIONE DELLA DIDATTICA</b>	<p>Il corso si basa principalmente su lezioni frontali in aula e sullo svolgimento guidato di esercizi al computer, per lo sviluppo delle competenze pratiche.</p>
<b>TESTI CONSIGLIATI</b>	<p>Testo principale / main textbook  Silbershatz, Galvin, Gagne, "Sistemi Operativi: concetti ed esempi", editrice Pearson/Addison Wesley</p> <p>Altri materiali / Other materials  Slides e materiale utilizzati a lezione e scaricabili on line. / Lecture slides and handouts are available on line.</p> <p>Per consultazione / Further reading  Tanenbaum, A. and Bos, H. (2014). Modern Operating Systems. Prentice Hall.</p>

### PROGRAMMA

ORE	Lezioni
3	Introduzione al corso, definizioni
4	Sistemi paralleli, sistemi distribuiti; Richiami di architetture dei calcolatori; Interrupt.
2	Comunicazione fra processore e dispositivi di I/O; Memoria: tipi, gerarchia, caching
5	Protezione hardware, modo utente e modo kernel; Strutture dei sistemi operativi, system call
5	Rappresentazione dei processi; PCB, code; Tipi di scheduler e criteri; Algoritmi di scheduling della CPU e loro valutazione
5	Gestione della memoria principale; metodi di allocazione
2	Paginazione
6	Gestione della memoria secondaria; tipi di supporti; scheduling della memoria secondaria
4	file system
6	Concorrenza: definizioni, cause, problemi derivanti;
6	Concorrenza: race conditions; sezioni critiche: approcci software, hardware, di sistema e di linguaggio; problemi classici di prova (5 filosofi, produttore/consumatore, lettori/scrittori)
6	Deadlock: definizioni; condizioni; rilevamento e recupero; prevenzione
6	Programmazione concorrente con message-passing: comunicazione bloccante e non-bloccante, sincrona e asincrona
4	Programmazione concorrente: esempi pratici e implementazioni con MPI
4	Rappresentazione numerica binaria delle informazioni: definizioni, rappresentazione di numeri (quantità) e conversioni di base; Rappresentazione numerica binaria di grandezze fisiche monodimensionali tempo-varianti (es.: suono); Rappresentazione numerica binaria di grandezze fisiche multidimensionali e tempo-varianti (es.: immagini e video).
4	La compressione: tecniche non percettive (senza perdita, tipo RLE) e percettive (con perdita, tipo JPEG e derivati)