

**PROGRAMMA PREVENTIVO - ANNO SCOLASTICO 2021 – 2022**

DEI PROF.	DOCENTE DI	NELLA CLASSE	INDIRIZZO	ORE SETTIMANALI
<b>Cannone Maria Del Vecchio Anna</b>	<b>Tecnologie e progettazione di sistemi informatici e di telecomunicazione</b>	<b>4 K</b>	<b>INFORMATICA E TELECOMUNICAZIONI</b>	<b>4 di cui 2 in codocenza</b>

MODULI (TITOLO)	CONTENUTI	OBIETTIVI	TEMPI	SPAZI E MEZZI	COLLEGAMENTI INTER-DISCIPLINARI	METODI	CRITERI DI VALUTAZIONE	TIPOLOGIA DELLE PROVE
<b>MODULO 1: Processi sequenziali e paralleli</b>	<p>Il modello a processi</p> <ul style="list-style-type: none"> <li>-Stato di processi</li> <li>-Comandi per la creazione, sospensione e terminazione dei processi</li> <li>-PCB (Process Control Block)</li> <li>-Risorse e condivisione:</li> <li>-Generalità,</li> <li>-Classificazioni,</li> <li>-Grafo di Holt.</li> <li>-I thread o“processi leggeri”</li> <li>-Generalità,</li> <li>-Processi pesanti” e “processi leggeri”,</li> <li>-Soluzioni adottate:</li> <li>-single threading vs multithreading,</li> <li>-Realizzazione di thread,</li> <li>-Thread POSIX,</li> <li>-Stati di un thread,</li> <li>-Utilizzo dei thread.</li> <li>-Elaborazione sequenziale e concorrente:</li> <li>-Generalità,</li> <li>-Processi non sequenziali e grafo di precedenza,</li> <li>-Scomposizione di un processo non sequenziale.</li> <li>-La descrizione della concorrenza:</li> <li>-Esecuzione parallela,</li> <li>-Fork-join, Cobegin-coend,</li> <li>-Equivalenza di fork-join e cobegin-coend;</li> <li>-Semplificazione delle precedenze.</li> </ul>	<p>Al termine del modulo lo studente sarà in grado di utilizzare</p> <p>gli strumenti della programmazione per la realizzazione di programmi concorrenti utilizzando l’istruzione fork-join, l’istruzione cobegin-coend; programmi multiprocessi in linguaggio C;</p> <p>essere in grado di utilizzare i thread in linguaggio C,</p> <p>di utilizzare i thread in linguaggio Java</p>	20	<p>Per ogni modulo:</p> <p>Laboratorio;</p> <p>Appunti, libro di testo e utilizzo del computer</p>	<p>Per ogni modulo:</p> <p><b>Inglese:</b></p> <p>Conoscenza della terminologia tecnica in italiano ed in inglese. Saper leggere e capire documentazione tecnica di livello medio in inglese.</p> <p><b>Informatica:</b></p> <p>Linguaggi di programmazione.</p> <p><b>Sistemi e Reti:</b></p> <p>Strumenti e principi base della comunicazione in rete.</p>	<p>Per ogni modulo:</p> <p>Lezione frontale in interazione.</p> <p>Lezione di laboratorio con esercitazioni pratiche</p> <p>Risorse online.</p> <p>Alcune lezioni potranno essere tenute in lingua inglese.</p>	<p>In generale:</p> <p>Il voto viene calcolato ripartendo in proporzione il punteggio riportato nella prova tra i voti 2 e 10, dunque il livello di sufficienza è il 50%, ferma restando la possibilità di successiva verifica orale su alcuni degli obiettivi non raggiunti.</p>	<p>Per ogni modulo:</p> <p>Verifiche Scritte:</p> <p>Test a scelta multipla o con domande aperte, interrogazioni.</p> <p>Alcune verifiche potranno essere somministrate (integralmente o parzialmente) in lingua inglese.</p>

Esercitazioni di laboratorio Modulo 1	<p>L'ambiente di sviluppo Dev-C++;</p> <p>La fork in C, Fork annidate ed esecuzione non deterministica;</p> <p>Le funzioni wait() e waitpid();</p> <p>Fork-join e cobegin-coend;</p> <p>I thread in C,</p> <p>Thread e parametri,</p> <p>Thread in ambiente Dev-cpp e linux-di sviluppo</p> <p>I thread in Java: concetti base,</p> <p>Priorità e parametri nei thread Java,</p> <p>I thread Java: i metodi sleep, yield e join.</p>		20					
<b>MODULO2:</b> <b>Comunicazione e sincronizzazione</b>	<ul style="list-style-type: none"> <li>-La comunicazione tra processi</li> <li>-Comunicazione: <ul style="list-style-type: none"> <li>-modelli software e hardware</li> <li>-Modello a memoria comune (ambiente globale, global environment)</li> <li>-Modello a scambio di messaggi (ambiente locale, message passing)</li> </ul> </li> <li>-La sincronizzazione tra processi</li> <li>-Errori nei programmi concorrenti</li> <li>-Definizioni e proprietà</li> <li>-Proprietà non funzionali: safety e liveness;</li> <li>-semafori;</li> <li>-Semafori di basso livello e spin lock()</li> <li>-Semafori di Dijkstra</li> <li>-Semafori binari vs semafori di Dijkstra</li> <li>-Semafori e mutua esclusione</li> <li>-Mutua esclusione tra gruppi di processi</li> <li>-Semafori come vincoli di precedenza</li> <li>-Problema del rendez-vous</li> <li>-Problemi "classici" della programmazione concorrente: <ul style="list-style-type: none"> <li>-produttori/consumatori;</li> <li>-Problema dei lettori e degli scrittori</li> <li>-Problemi "classici" della programmazione concorrente: deadlock,</li> <li>-banchiere e filosofi a cena</li> </ul> </li> <li>-Individuazione dello stallo</li> <li>-Come affrontare lo stallo</li> </ul>	<p>Al termine del modulo lo studente sarà in grado di</p> <p>Individuare le tipologie di errori nei processi paralleli</p> <p>Definire e utilizzare i semafori di basso livello e spin lock()</p> <p>Utilizzare gli strumenti di sincronizzazione per thread in C</p> <p>Utilizzare le condition variable in C</p> <p>Implementare i monitor in C/Java</p> <p>Utilizzare gli strumenti di sincronizzazione per thread in C</p> <p>Risolvere le situazioni di starvation</p> <p>Risolvere le situazioni di deadlock</p> <p>Risolvere i problemi produttore/consumatore in C/Java</p> <p>Risolvere il problema dei filosofi in C/Java</p>	20					

	<ul style="list-style-type: none"> <li>-Esempio classico: problema dei filosofi a cena</li> <li>-I monitor: Generalità</li> <li>-Utilizzo dei monitor</li> <li>-Variabili condizione e procedure di wait/signal</li> <li>-Emulazione di monitor con i semafori</li> <li>-Lo scambio di messaggi: <ul style="list-style-type: none"> <li>-Generalità</li> <li>-Canali di comunicazione</li> <li>-Primitive di comunicazione <ul style="list-style-type: none"> <li>-asimmetrica da-molti-a-uno</li> <li>-Primitive di comunicazione asimmetrica da-molti-a-molti (cenni)</li> </ul> </li> </ul> </li> </ul>							
<p>Esercitazioni di laboratorio</p> <p>Modulo 2</p>	<p>La comunicazione tra processi mediante segnali asincroni</p> <p>Thread e schedulazione</p> <p>I semafori binari in C</p> <p>La soluzione del deadlock dei filosofi in C con i mutex</p> <p>La soluzione del problema produttori/consumatori con i semafori classici</p> <p>Variabili condizione</p> <p>I monitor con le variabili condition in C</p> <p>I monitor con i semafori in C</p> <p>I semafori in Java</p> <p>I monitor in Java</p> <p>Un esempio con i Java thread: corsa di biciclette</p> <p>Il deadlock in Java</p>		20					
<p><b>MODULO3:</b></p> <p><b>I requisiti software</b></p>	<ul style="list-style-type: none"> <li>-La specifica dei requisiti</li> <li>-Requisiti software e stakeholder</li> <li>-Classificazione dei requisiti</li> <li>-I requisiti: l'anello debole dello sviluppo software</li> <li>-Raccolta e analisi dei requisiti</li> <li>-Tipi di raccolta dei requisiti</li> <li>-La fase di esplorazione</li> <li>-Problemi della fase di esplorazione</li> <li>-Attori, casi d'uso e scenari</li> <li>-Tipi di scenari</li> <li>-Descrizione dei casi d'uso</li> <li>-Relazioni tra casi d'uso</li> </ul>	<p>Al termine del modulo lo studente sarà in grado di</p> <ul style="list-style-type: none"> <li>-Individuare i requisiti utente</li> <li>-Individuare i requisiti di sistema</li> <li>-Utilizzare le tecniche di esplorazione</li> <li>-Individuare gli scenari d'uso</li> <li>-Analizzare il documento di Specifica dei</li> </ul>	20					

	<ul style="list-style-type: none"> <li>-Documentazione dei casi d'uso</li> <li>-La documentazione dei requisiti</li> <li>-Requirements Documents proposto da Sommerville</li> <li>-Realizzare un efficace documento SRS</li> </ul>	<p>Requisiti Software (SRS)</p> <ul style="list-style-type: none"> <li>-Acquisire la struttura di un SRS</li> <li>-Saper descrivere in UML i casi d'uso</li> <li>-Saper descrivere in UML il diagramma di contesto</li> </ul>					
Esercitazioni di laboratorio Modulo 3	<p>La realizzazione degli Use Case Diagram con StarUML</p> <p>La realizzazione degli Use Case Diagram con ArgoUML</p>	<ul style="list-style-type: none"> <li>-Saper documentare i casi d'uso</li> <li>-Saper compilare il documento di Specifica dei Requisiti Software</li> <li>-Validare le specifiche di un SRS</li> </ul>	10				

NOTE: La tempistica prevista è solo orientativa. In itinere si vedrà dove soffermarsi di più o meno e compatibilmente con le ore di lezione che saranno realmente disponibili.