

## PROGRAMMA PREVENTIVO - ANNO SCOLASTICO 2020 – 2021

DEI PROF.	DOCENTE DI	NELLA CLASSE	INDIRIZZO	ORE SETTIMANALI
<b>Fraccola Christian, 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 (h)
Modulo 1 Processi sequenziali e paralleli	<p>Il modello a processi, Stato di processi, Comandi per la creazione, sospensione e terminazione dei processi, PCB (Process Control Block)</p> <p><b>Risorse e condivisione:</b> Generalità , Classificazioni, Grafo di Holt.</p> <p><b>Thread:</b> Generalità, Processi pesanti” e “processi leggeri, single threading vs multithreading, Realizzazione di thread, Thread POSIX, Stati di un thread, Utilizzo dei thread.</p> <p><b>Elaborazione sequenziale e concorrente:</b> Generalità, Processi non sequenziali e grafo di precedenza, Scomposizione di un processo non sequenziale.</p> <p><b>La concorrenza:</b> Esecuzione parallela, Fork-join, Cobegin-coend, Equivalenza di fork-join e cobegin-coend; Semplificazione delle precedenze</p>	<p>Al termine del modulo lo studente sarà in grado di:</p> <ul style="list-style-type: none"> <li>▪ utilizzare gli strumenti della programmazione per la realizzazione di programmi concorrenti utilizzando l’ istruzione fork-join,</li> <li>▪ l’ istruzione cobegin-coend;</li> <li>▪ programmi multi-processi in linguaggio C;</li> <li>▪ essere in grado di utilizzare i thread in linguaggio C / Java</li> </ul>	30

<p>Esercitazioni di laboratorio Modulo 1</p>	<ul style="list-style-type: none"> <li>• L' emulatore Cygwin, L' ambiente di sviluppo Dev-C++;</li> <li>• La fork in C, Fork annidate ed esecuzione non deterministica;</li> <li>• Le funzioni wait() e waitpid();</li> <li>• Fork-join e cobegin-coend;</li> <li>• I thread in C: Thread e parametri, Thread in ambiente Dev-cpp e linux, di sviluppo.</li> <li>• I thread in Java: concetti base, Priorità e parametri nei thread Java, i metodi sleep, yield e join.</li> </ul>		<p>20</p>
<p>Modulo 2 Comunicazione e sincronizzazione</p>	<p><b>La comunicazione tra processi:</b> modelli software e hardware, Modello a memoria comune (ambiente globale, global environment), modello a scambio di messaggi (ambiente locale, message passing) La sincronizzazione tra processi, errori nei programmi concorrenti Definizioni e proprietà, Proprietà non funzionali: safety e liveness; semafori; <b>Semafori:</b> di basso livello e spin lock(), di Dijkstra, mutua esclusione, Mutua esclusione tra gruppi di processi Semafori come vincoli di precedenza, problema del rendez-vous <b>Problemi "classici" della programmazione concorrente:</b> produttori/consumatori; Problema dei lettori e degli scrittori, deadlock, banchiere e filosofi a cena,</p>	<p>Al termine del modulo lo studente sarà in grado di: Individuare le tipologie di errori nei processi paralleli Definire e utilizzare i semafori di basso livello e spin lock() Utilizzare gli strumenti di sincronizzazione per thread in C Utilizzare le condition variable in C Implementare i monitor in C/Java Utilizzare gli strumenti di sincronizzazione per thread in C Risolvere le situazioni di starvation Risolvere le situazioni di deadlock Risolvere i problemi produttore/consumatore in C/Java</p>	<p>30</p>

	<p><b>Deadlock:</b> Individuazione dello stallo, come affrontare lo stallo</p> <p>I <b>monitor:</b> Generalità, Utilizzo dei monitor, Variabili condizione e procedure di wait/signal</p> <p>Emulazione di monitor con i semafori</p> <p><b>Lo scambio di messaggi:</b> Generalità, Canali di comunicazione, Primitive di comunicazione asimmetrica da-molti-a-uno e da-molti-a-molti (cenni)</p>	<p>Risolvere il problema dei filosofi in C/Java</p>	
<p>Esercitazioni di laboratorio Modulo 2</p>	<ul style="list-style-type: none"> <li>• La comunicazione tra processi mediante segnali asincroni</li> <li>• Thread e schedulazione,</li> <li>• I semafori binari in C</li> <li>• La soluzione del deadlock dei filosofi in C con i mutex</li> <li>• La soluzione del problema produttori/consumatori con i semafori classici</li> <li>• I monitor con le variabili condizione in C e con i semafori in C /Java</li> <li>• I thread e il deadlock in Java</li> </ul>		<p>20</p>

<p>Modulo 3</p> <p>Requisiti del software</p>	<p>La specifica dei requisiti</p> <p>Requisiti software e stakeholder</p> <p>Classificazione dei requisiti</p> <p>I requisiti: l' anello debole dello sviluppo software</p> <p>Raccolta e analisi dei requisiti</p> <p>Tipi di raccolta dei requisiti</p> <p>La fase di esplorazione</p> <p>Problemi della fase di esplorazione</p> <p>Attori, casi d' uso e scenari</p> <p>Tipi di scenari</p> <p>Descrizione dei casi d' uso</p> <p>Relazioni tra casi d' uso</p> <p>Documentazione dei casi d' uso</p> <p>La documentazione dei requisiti</p> <p>Requirements Documents proposto da Sommerville</p> <p>Realizzare un efficace documento SRS p</p>	<p>Al termine del modulo lo studente sarà in grado di</p> <p>Individuare i requisiti utente</p> <p>Individuare i requisiti di sistema</p> <p>Utilizzare le tecniche di esplorazione</p> <p>Individuare gli scenari d' uso</p> <p>Analizzare il documento di Specifica dei Requisiti Software (SRS)</p> <p>Acquisire la struttura di un SRS</p> <p>Saper descrivere in UML i casi d' uso</p> <p>Saper descrivere in UML il diagramma di contesto</p> <p>Saper documentare i casi d' uso</p> <p>Saper compilare il documento di Specifica dei Requisiti Software</p> <p>Validare le specifiche di un SRS</p>	<p>15</p>
<p>Esercitazioni di laboratorio</p> <p>Modulo 3</p>	<p>La realizzazione degli Use Case Diagram con StarUML</p> <p>La realizzazione degli Use Case Diagram con ArgoUML.</p>		<p>10</p>

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.

Per ogni modulo:

SPAZI E MEZZI	COLLEG. INTER-DISC.	METODI	CRITERI DI VALUTAZIONE	TIPOLOGIA DELLE PROVE
<p>Laboratorio.</p> <p>Appunti, libro di testo , risorse on-line, utilizzo del computer</p>	<p>Inglese: Conoscenza della terminologia tecnica in italiano ed in inglese. Saper leggere e capire documentazione tecnica di livello medio in inglese.</p> <p>Informatica: Linguaggi di programmazione.</p> <p>Sistemi e Reti: Strumenti e principi base della comunicazione in rete.</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: Il voto viene calcolato ripartendo in proporzione il punteggio riportato nella prova tra i voti 2 e 10.</p>	<p>Verifiche Scritte: Test a scelta multipla o con domande aperte, interrogazioni orali.</p> <p>Prove pratiche.</p>