

PROGRAMMA PREVENTIVO - ANNO SCOLASTICO 2021 – 2022

DEI PROF.	DOCENTE DI	NELLA CLASSE	INDIRIZZO	ORE SETTIMANALI
Cannone Maria Bellavita Simona	Tecnologie e progettazione di sistemi informatici e di telecomunicazione	4 E	INFORMATICA E TELECOMUNICAZIONI	4 di cui 2 in codocenza

MODULI (TITOLO)	CONTENUTI	OBIETTIVI	TEMPI	SPAZI E MEZZI	COLLEGAMENTI INTER-DISCIPLINARI	METODI	CRITERI DI VALUTAZIONE	TIPOLOGIA DELLE PROVE
MODULO 1: Processi sequenziali e paralleli	<p>Il modello a processi</p> <ul style="list-style-type: none"> -Stato di processi -Comandi per la creazione, sospensione e terminazione dei processi -PCB (Process Control Block) -Risorse e condivisione: -Generalità, -Classificazioni, -Grafo di Holt. -I thread o“processi leggeri” -Generalità, -Processi pesanti” e “processi leggeri”, -Soluzioni adottate: -single threading vs multithreading, -Realizzazione di thread, -Thread POSIX, -Stati di un thread, -Utilizzo dei thread. -Elaborazione sequenziale e concorrente: -Generalità, -Processi non sequenziali e grafo di precedenza, -Scomposizione di un processo non sequenziale. -La descrizione della concorrenza: -Esecuzione parallela, -Fork-join, Cobegin-coend, -Equivalenza di fork-join e cobegin-coend; -Semplificazione delle precedenze. 	<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,</p> <p>libro di testo</p> <p>e utilizzo</p> <p>del computer</p>	<p>Per ogni modulo:</p> <p>Inglese:</p> <p>Conoscenza della terminologia tecnica in italiano ed in inglese. Saper leggere e capire documentazione tecnica di livello medio in inglese.</p> <p>Informatica:</p> <p>Linguaggi di programmazione.</p> <p>Sistemi e Reti:</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>

<p>Esercitazioni di laboratorio Modulo 1</p>	<p>L'ambiente di sviluppo Dev-C++; La fork in C, Fork annidate ed esecuzione non deterministica; Le funzioni wait() e waitpid(); Fork-join e cobegin-coend; I thread in C, Thread e parametri, Thread in ambiente Dev-cpp e linux-di sviluppo I thread in Java: concetti base, Priorità e parametri nei thread Java, I thread Java: i metodi sleep, yield e join.</p>		<p>20</p>					
<p>MODULO2: Comunicazione e sincronizzazione</p>	<p>-La comunicazione tra processi -Comunicazione: -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; -Semafori di basso livello e spin lock() -Semafori di Dijkstra -Semafori binari vs semafori di Dijkstra -Semafori e mutua esclusione -Mutua esclusione tra gruppi di processi -Semafori come vincoli di precedenza -Problema del rendez-vous -Problemi "classici" della programmazione concorrente: -produttori/consumatori; -Problema dei lettori e degli scrittori -Problemi "classici" della programmazione concorrente: deadlock, -banchiere e filosofi a cena -Individuazione dello stallo -Come affrontare lo stallo</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 Risolvere il problema dei filosofi in C/Java</p>	<p>20</p>					

	<ul style="list-style-type: none"> -Esempio classico: problema dei filosofi a cena -I monitor: Generalità -Utilizzo dei monitor -Variabili condizione e procedure di wait/signal -Emulazione di monitor con i semafori -Lo scambio di messaggi: <ul style="list-style-type: none"> -Generalità -Canali di comunicazione -Primitive di comunicazione <ul style="list-style-type: none"> -asimmetrica da-molti-a-uno -Primitive di comunicazione asimmetrica da-molti-a-molti (cenni) 							
<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>MODULO3:</p> <p>I requisiti software</p>	<ul style="list-style-type: none"> -La specifica dei requisiti -Requisiti software e stakeholder -Classificazione dei requisiti -I requisiti: l'anello debole dello sviluppo software -Raccolta e analisi dei requisiti -Tipi di raccolta dei requisiti -La fase di esplorazione -Problemi della fase di esplorazione -Attori, casi d'uso e scenari -Tipi di scenari -Descrizione dei casi d'uso -Relazioni tra casi d'uso 	<p>Al termine del modulo lo studente sarà in grado di</p> <ul style="list-style-type: none"> -Individuare i requisiti utente -Individuare i requisiti di sistema -Utilizzare le tecniche di esplorazione -Individuare gli scenari d'uso -Analizzare il documento di Specifica dei 	20					

	<ul style="list-style-type: none"> -Documentazione dei casi d'uso -La documentazione dei requisiti -Requirements Documents proposto da Sommerville -Realizzare un efficace documento SRS 	<p>Requisiti Software (SRS)</p> <ul style="list-style-type: none"> -Acquisire la struttura di un SRS -Saper descrivere in UML i casi d'uso -Saper descrivere in UML il diagramma di contesto 					
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"> -Saper documentare i casi d'uso -Saper compilare il documento di Specifica dei Requisiti Software -Validare le specifiche di un SRS 	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.