

PIANO DI LAVORO

PROF. FRACCOLA CHRISTIAN e DEL VECCHIO ANNA

DISCIPLINA T.P.S.I.T.

ANNO SCOLASTICO 2021-22

COMPETENZE TRASVERSALI

L'insegnamento della disciplina promuove:

primo biennio:

L'INSEGNAMENTO DI TALE DISCIPLINA INIZIA, UFFICIALMENTE, AL TRIENNIO

QUALCHE MODULO BREVE E SINTETICO PUÒ' ESSERE PROPOSTO AL 2° ANNO PER LA DISCIPLINA **STA**, COME ORIENTAMENTO

- Legge di Ohm di base
- Segnale digitale e segnale analogico
- Collegamento LED, resistore e pulsante con Arduino
- Programmazione base in C su Arduino
- Scrittura HTML, CSS, Javascript di base

Secondo biennio e quinto anno:

La disciplina "Tecnologie e progettazione di sistemi informatici e di telecomunicazioni " concorre a far conseguire allo studente al termine del percorso quinquennale i seguenti risultati di apprendimento relativi al profilo educativo, culturale e professionale dello studente: orientarsi nella normativa che disciplina i processi produttivi del settore di riferimento, con particolare attenzione sia alla sicurezza sui luoghi di vita e di lavoro sia alla tutela dell'ambiente e del territorio; intervenire nelle diverse fasi e livelli del processo produttivo, dall'ideazione alla realizzazione del prodotto, per la parte di propria competenza, utilizzando gli strumenti di progettazione, documentazione e controllo; riconoscere gli aspetti di efficacia, efficienza e qualità nella propria attività lavorativa; padroneggiare l'uso di strumenti tecnologici con particolare attenzione alla sicurezza nei luoghi di vita e di lavoro, alla tutela della persona, dell'ambiente e del territorio.

La disciplina, nell'ambito della programmazione del Consiglio di classe, concorre in particolare al raggiungimento dei seguenti risultati di apprendimento, relativi all'indirizzo, espressi in termini di competenza:

- sviluppare applicazioni informatiche per reti locali o servizi a distanza;
- scegliere dispositivi e strumenti in base alle loro caratteristiche funzionali;
- gestire progetti secondo le procedure e gli standard previsti dai sistemi aziendali di gestione della qualità e della sicurezza. gestire processi produttivi correlati a funzioni aziendali;

- configurare, installare e gestire sistemi di elaborazione dati e reti;
- redigere relazioni tecniche e documentare le attività individuali e di gruppo relative a situazioni professionali.

COMPETENZE DISCIPLINARI

Primo biennio:

L'INSEGNAMENTO DI TALE DISCIPLINA INIZIA, UFFICIALMENTE, AL TRIENNIO

QUALCHE MODULO BREVE E SINTETICO PUÒ' ESSERE PROPOSTO AL 2° ANNO PER LA DISCIPLINA **STA** (scienze e tecnologie applicate), COME ORIENTAMENTO

Secondo biennio e quinto anno:

- Funzionamento del calcolatore e componentistica
- Codifica in binario e altri sistemi usati nei calcolatori
- Problematiche e funzionamento del sistema operativo
- Sviluppo software in ottica di rete con scambio informazioni e uso dei protocolli

CLASSE 4J

MODULI	COMPETENZE	ABILITÀ	CONTENUTI	ESERCITAZIONI DI LABORATORIO	COLLEGAMENTI INTERD.	METODOLOGIE
Sistema operativo e gestione del processore	Descrivere il ciclo di vita di un processo	Scegliere le politiche di allocazione del processore Saper simulare gli algoritmi di scheduling	Generalità sui sistemi operativi: il sistema operativo e i suoi compiti Kernel, Shell. La gestione del processore: i processi, ciclo di vita, la schedulazione, i criteri di scheduling, algoritmi di scheduling (FCFS, SJF, SRTF, RR, a priorità, MLFQ), scheduling a confronto tra sistemi operativi.	La shell dei comandi e la Powershell (Windows) Bash (Shell di Linux)	Inglese: Conoscenza della terminologia tecnica. Saper leggere e capire documentazione tecnica.	Lezione frontale interattiva. Esercitazioni pratiche in laboratorio. Risorse online. Flipped Classroom. Co-teaching. Attività a gruppi.
Processi sequenziali e paralleli	Descrivere l'interazione processi-risorse col grafo di Holt Realizzare e semplificare il grafo delle precedenze Scrivere programmi concorrenti utilizzando l'istruzione fork-join Scrivere programmi multiprocessing in linguaggio C	Utilizzare i thread in linguaggio C	Il modello a processi, Stato di processi, Comandi per la creazione, sospensione e terminazione dei processi, PCB (Process Control Block) Risorse e condivisione: Generalità, Classificazioni, Grafo di Holt. Thread: Generalità, Processi pesanti" e "processi leggeri, 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 concorrenza: Esecuzione parallela, Fork-join, Cobegin-coend, Equivalenza di fork-join e cobegin-coend; Semplificazione delle precedenze	L'emulatore Cygwin. La fork in C, fork annidate ed esecuzione non deterministica; Le funzioni wait() e waitpid(); I thread in C: Thread e parametri, I thread in Java: Priorità e parametri nei thread Java, i metodi sleep, yield e join	Informatica: Linguaggi di programmazione. Sistemi e Reti: Strumenti e principi base della comunicazione in rete	

<p>Comunicazione e sincronizzazione</p>	<p>Individuare le tipologie di errori nei processi paralleli Definire e utilizzare i semafori di basso livello e spinlock() 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</p>	<p>Risolvere 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>La comunicazione tra processi: 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(), di Dijkstra, 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, deadlock, banchiere e filosofi a cena, Deadlock: Individuazione dello stallo, come affrontare lo stallo I monitor: Generalità, Utilizzo dei monitor, Variabili condizione e procedure di wait/signal Emulazione di monitor con i semafori Lo scambio di messaggi: Generalità, Canali di comunicazione, Primitive di comunicazione asimmetrica da-molti-a-uno e da-molti-a-molti (cenni)</p>	<p>La comunicazione tra processi mediante segnali asincroni Thread e schedulazione, I semafori binari in C La soluzione del deadlock dei filosofi in C con i mutex La soluzione del problema produttori/consumatori con i semafori classici I monitor con le variabili condizione in C e con i semafori in C /Java I thread e il deadlock in Java</p>		
<p>Requisiti del software</p>	<p>Applicare le tecniche per aumentare l'affidabilità del codice Introdurre la verifica e la convalida del</p>	<p>Pianificare i test e i casi di test Definire l'oracolo per ogni test Effettuare il</p>	<p>La specifica dei requisiti Requisiti software e stakeholder Classificazione dei requisiti I requisiti: l'anello debole dello sviluppo software Raccolta e analisi dei requisiti</p>	<p>La realizzazione degli Use Case Diagram con StarUML / ArgoUML.</p>		

	software Descrivere il processo V&V Definire i requisiti di qualità dei test Applicare le tecniche di verifica statica Saper effettuare i test di sistemi OO	debug utilizzando i breakpoint Utilizzare la tecnica di inspection Saper effettuare il Testing di unità Saper effettuare il test in grande	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 Documentazione dei casi d'uso La documentazione dei requisiti Requirements Documents proposto da Sommerville Realizzare un efficace documento SRS			
--	--	--	---	--	--	--

Obiettivi minimi per l'ammissione alla classe successiva:

- utilizzare gli strumenti della programmazione per la realizzazione di programmi concorrenti utilizzando l'istruzione fork in C,
- essere in grado di utilizzare l'istruzione fork in linguaggio C;
- essere in grado di utilizzare i thread in linguaggio C / Java
- Individuare le tipologie di errori nei processi paralleli
- Conoscere i semafori binari e di Dijkstra
- 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
- Saper riconoscere e risolvere le situazioni di starvation e di deadlock
- Conoscere il problema dei filosofi a cena
- Conoscere e saper risolvere i problemi produttore/consumatore e i problemi scrittore/ lettori

Modalità di verifica:

- ⇒ Verifiche scritte: test a scelta multipla o con domande aperte
- ⇒ Interrogazioni orali
- ⇒ Prove pratiche di laboratorio con l'ausilio dei PC.