



Ministero dell'Istruzione, dell'Università e della Ricerca
 Dipartimento per la Programmazione
 Direzione Generale per interventi in materia di edilizia
 scolastica, per la gestione dei fondi strutturali per
 l'istruzione e per l'innovazione digitale
 Ufficio IV

PER LA SCUOLA - COMPETENZE E AMBIENTI PER L'APPRENDIMENTO (FSE-FESR)

PROGRAMMAZIONE PREVENTIVA (con Insegnante Tecnico Pratico)

Codice Mod. **RQ 10.3** Pag. 1 / 3

A.S.	2019-20	DOCENTI	
DISCIPLINA	TECNOLOGIA E PROGETTAZIONE DI SISTEMI SOFTWARE E DI TELECOMUNICAZIONI		
CLASSI	4G 4H 4I	INDIRIZZO	INFORMATICA-TELECOMUNICAZIONI - articolazione Informatica

COMPETENZE

1. Conoscere struttura e organizzazione di un sistema operativo; politiche di gestione dei processi.
2. Conoscere tecniche e tecnologie per la programmazione concorrente e la sincronizzazione dell'accesso a risorse condivise.
3. Saper progettare e realizzare applicazioni che interagiscono con le funzionalità dei sistemi operativi.
4. Saper progettare e realizzare applicazioni in modalità concorrente.
5. Redigere relazioni tecniche e documentare le attività individuali e di gruppo relative a situazioni professionali.

N°	Titolo del modulo	Contenuti	Attività di laboratorio	Obiettivi disciplinari	Periodo
1	Il modello a processi	Scheduling e multiprogrammazione. Processi cooperanti e processi in competizione. Stato di un processo e diagramma degli stati. Il PID. Il context switching. Il Process Control Block (PCB). Creazione, sospensione e terminazione dei processi nei sistemi *NIX like (fork, exit, wait). Algoritmi di Scheduling: FCFS, SJF, con Priorità, Round Robin, con code multiple a retroazione.	Esercitazioni pratiche volte a calcolare il tempo medio di accesso dei processi con i principali algoritmi di scheduling.	Conoscere i modelli di elaborazione dei processi ed il ciclo di vita dei processi.	Settembre Ottobre

2	Risorse e condivisione	Definizione di risorsa Classi e istanze di risorse. Condivisione e gestione. Classificazioni in base al tipo di richiesta, alla modalita di assegnazione, alla tipologia delle risorse. Grafo di Holt o grafo di allocazione delle risorse rappresentazione delle risorse e dei processi. Significato del verso degli archi che connettono processi e risorse. Grafi riducibili	Esercitazioni pratiche volte a rappresentare processi e risorse tramite grafi di Holt. Esercitazioni pratiche volte a ridurre un grafo di Holt.	Acquisire il concetto di risorsa condivisa. Apprendere l'utilizzo del grafo di Holt per descrivere processi e risorse.	Ottobre
3	I thread o "processi leggeri"	"Processi pesanti" e "processi leggeri", loro pregi e loro difetti. Definizione di thread. Thread safety. Realizzazione dei thread user-level kernel-level. Thread POSIX. Stati di un thread. Utilizzo dei thread.	Ambiente di sviluppo Java. La fork join. Fork annidate ed esecuzione non deterministica.	Imparare le differenze tra thread e processi, modalita di utilizzo dei thread nei SO	Ottobre Novembre
4	Elaborazione sequenziale e concorrente	Processi non sequenziali e grafo delle precedenze. Scomposizione di un processo non sequenziale. Processi indipendenti e processi interagenti. Processi in competizione.	Esercitazioni pratiche sui grafi delle precedenze.	Imparare il concetto di programmazione concorrente, realizzare il grafo delle precedenze, apprendere il concetto di interazione tra processi.	Novembre
5	La descrizione della concorrenza	Esecuzione parallela. Fork-join. Cobegin-coend. Equivalenza di fork-join e cobegin-coend. Semplificazione delle precedenze.	Fork-join e cobegin-coend.	Imparare il concetto di programmazione concorrente, realizzare il grafo delle precedenze, apprendere il concetto di interazione tra processi.	Dicembre
6	La comunicazione tra processi	La comunicazione: modelli software e hardware. Il modello a memoria condivisa allocazione delle risorse ai processi e loro tipologia competizione e cooperazione	I thread in Java.	Conoscere i modelli di comunicazione tra processi: a memoria condivisa (o ad ambiente globale) ed il modello a scambio di messaggi (o ad ambiente locale)	Gennaio
7	La sincronizzazione tra processi ed i semafori	Errori nei programmi concorrenti. Condizioni per avere la concorrenza: interleaving e overlapping. Condizioni di Bernstein. Definizione di dominio e rango di una istruzione o procedura. Mutua esclusione e sezione critica. Starvation e deadlock. Semafori di basso livello e spin lock. Allocazione di una risorsa: lock(). Rilascio di una risorsa: unlock(). Problema della indivisibilit�. Semafori di Dijkstra. Le primitive P e V. Semafori e loro applicazioni. Mutua esclusione tra gruppi di processi. Semafori come vincoli di precedenza.	I thread in Java. Priorit� e parametri nei thread Java. I thread in Java: sleep, yield e join.	Conoscere le tipologie di errore nei processi paralleli e le propriet� richieste ai programmi concorrenti. Definire e utilizzare i semafori ed il concetto di indivisibilit� di una primitiva.	Gennaio Febbraio

8	Problemi classici della programmazione concorrente: produttori/consumatori, lettori/scrittori	<p>Problema produttore/consumatore. Risoluzione del problema in pseudocodifica nel caso in cui il dato prodotto sia "singolo" i semafori "pieno" e "vuoto" e loro inizializzazione. Risoluzione del problema in pseudocodifica nel caso in cui la memoria condivisa possa contenere piu dati (buffer circolare). Problema dei lettori/scrittori: prima soluzione che favorisce i lettori i semafori mutex e sincro e la variabile contatore numLettori, seconda soluzione proposta che favorisce gli scrittori i semafori mutex1, mutex2 e la variabile contatore numScrittori.</p>	<p>La comunicazione tra processi mediante segnali asincroni. I semafori binari in Java. La soluzione del deadlock dei filosofi con i mutex. La soluzione del problema produttore consumatore con i semafori classici.</p>	<p>Conoscere le caratteristiche dei problemi produttori/consumatori, lettori/scrittori e risolvere tali problemi.</p>	Marzo
9	Problemi classici della programmazione concorrente: deadlock	<p>Condizioni necessarie e sufficienti per la generazione di un deadlock. Individuazione dello stallo, metterlo in evidenza con il grafo wait-for. I e II Teorema sul grafo di Holt Affrontare lo stallo. Detection e recovery. Avoidance. Definizione di sequenza sicura e stato sicuro di un sistema. Problema del banchiere. Prevention. Ignorare il problema. Problema dei filosofi a cena.</p>	<p>I semafori in Java. Esempio di thread in Java: corsa di biciclette. Il deadlock in Java.</p>	<p>Conoscere le cause del deadlock e saper mettere in atto le strategie per evitarlo od eliminarlo.</p>	Aprile
10	I requisiti e documentazione del software	<p>Ciclo di vita del software. La specifica dei requisiti Raccolta e analisi dei requisiti. Attori, casi d'uso e scenari. La documentazione dei requisiti. La documentazione del progetto. La documentazione del codice.</p>	<p>Utilizzo di software open-source per realizzare diagrammi UML. Documentazione automatica con la Javadoc</p>	<p>Identificare le fasi di un progetto nel contesto del ciclo di sviluppo. Documentare i requisiti e gli aspetti architetturali di un prodotto/servizio, anche in riferimento a standard di settore.</p>	Maggio Giugno
11	Programmazione Java	<p>caratteristiche generali del linguaggio Java l'ambiente di programmazione (linea di comando/Eclipse), tipi di dati (primitivi e riferimento), variabili, operatori logici e di confronto, il casting, cicli, operazioni di input e output, le strutture di controllo (sequenza/selezione), le eccezioni, array e matrici, stringhe, dichiarazioni e utilizzo di classi, oggetti, istanze, metodi attributi, membri e metodi privati e pubblici, l'ereditarietà, il polimorfismo, strutture dinamiche (pila, code), flussi di input/output, i file di testo, thread in java</p>	<p>Programmazione in Java su ambiente eclipse, realizzazione di programmi object-oriented</p>	<p>Apprendere la programmazione in linguaggio java</p>	Settembre Giugno