VIRTUAL

BUAP®

Programación Concurrente y Paralela - Maestría

Presentación de la Asignatura

El curso de Programación Concurrente y Paralela de la Maestría en Ciencias de la Computación de la BUAP pertenece al grupo de asignaturas de primer semestre del Área de Sistemas Distribuidos. Es un curso básico donde se estudian los conceptos y técnicas de programación mínimas requeridas que el estudiante debe saber para codificar programas concurrentes, paralelos y distribuidos que le ayuden en la solución de problemas en otras áreas como por ejemplo el reconocimiento de patrones, redes neuronales, procesamiento digital de imágenes, visualización, simulación, graficación, entre otras. Se retoman la programación paralela con memoria compartida y paso de mensajes que se vieron en el nivel licenciatura. Se estudia el análisis de rendimiento y correctitud de los sistemas paralelos y se introduce la especificación formal de los sistemas concurrentes particularizando en el conocimiento y uso del álgebra de procesos de Hoare (CSP) como un modelo matemático importante para especificar programas paralelos con paso de mensajes.

Parte importante es el uso de una librería en Java que implementa parte del modelo CSP para llevar a la práctica lo especificado con el álgebra de procesos bajo el esquema del PMI. Finalmente se trabajan varios casos de estudio referente a las nuevas tendencias de la programación paralela como por ejemplo la Programación Paralela Estructurada, el uso de Objetos paralelos y se estudian los algoritmos paralelos más comunes que resuelven problemas de ordenación, optimización y búsqueda, así como problemas numéricos que incluso utilizan patrones de comunicación entre procesos como pipelines, farms, trees, cubes, mesh, entre otros.

Unidades de Aprendizaje

Unidad	Contenido Temático/Actividades de aprendizaje
1.- Introducción a la programación concurrente y paralela	1.1. Conceptos Básicos 1.1.1. Concurrencia 1.1.2. Paralelismo 1.1.3. Sistemas Distribuidos 1.2. Taxonomía de Computadoras Paralelas 1.2.1. Flynn 1.2.2. Arquitecturas Paralelas actuales 1.3. Modelos de Programación Concurrente y Paralela 1.4. Taxonomía de Lenguajes Paralelos

Unidades de Aprendizaje

2.- Programación Paralela con Memoria Compartida	2.1. Procesos e Hilos 2.2. Problemas Inherentes 2.2.1. Exclusión mutua 2.2.2. Algoritmos de exclusión mutua 2.2.3. Regiones críticas 2.3. Primitivas de sincronización 2.3.1. Candados 2.3.2. Semáforos 2.4. Primitivas de comunicación 2.4.1. Variables de condición 2.4.2. Espera y notificación 2.4.3. Monitores 2.5. Caso de estudio

Unidades de Aprendizaje

3.- Programación Paralela con Memoria Distribuida (PMI)	3.1. Introducción 3.2. Elementos de comunicación 3.2.1. Procesos emisores (operación de envío) 3.2.2. Procesos receptores (operación de recepción) 3.2.3. Canales de comunicación (capacidad, flujo de datos, tipados, no-tipados) 3.2.4. Tipos de comunicación síncrona y asíncrona (directa-simétrica, directa-asimétrica, indirecta). 3.2.5. Mensajes (tipo, tamaño, paso por copia vs. paso por referencia) 3.3. Caso de estudio

Unidades de Aprendizaje

4. Análisis de Corrección y Rendimiento	4.1. Propiedades de corrección 4.1.1. Seguridad 4.1.2. Vivacidad 4.1.3. No-Inanición 4.1.4. Equidad 4.2. Definiciones 4.2.1. Rendimiento 4.2.2. Escalabilidad 4.2.3. Granularidad 4.2.4. Aceleración 4.3. Métricas 4.3.1. Tiempo paralelo de ejecución 4.3.2. Speedup 4.3.3. Ley de Amdahl 4.3.4. CPI 4.3.5. Eficiencia

Unidades de Aprendizaje

5.- Estrategias de Paralelización	5.1. Particionamiento 5.2. Divide y vencerás 5.3. Pipeline 5.4. Farms

Unidades de Aprendizaje

6.- Algoritmos Paralelos	6.1. Algoritmos de ordenación 6.2. Algoritmos de búsqueda y optimización 6.3. Algoritmos numéricos

Unidades de Aprendizaje

7. Tendencias de la Programación Paralela	7.1. Computación de altas prestaciones 7.2. Computación heterogénea 7.3. Programación paralela estructurada 7.4. Objetos paralelos

Propósito

El alumno aprenderá a desarrollar soluciones paralelas de problemas que se pueden paralelizar y aprenderá a mejorar el rendimiento o performance en sus aplicaciones

Competencias Profesionales

- El alumno aprenderá a resolver los problemas que surgen en la programación concurrente al comunicar y sincronizar procesos cuando éstos comparten recursos: Mutex, condiciones de sincronización del tipo productor-consumidor, interbloqueos o deadlocks, regiones críticas, semáforos y monitores.
- El alumno conseguirá ejecutar un programa secuencial en menos tiempo utilizando para ello más de un procesador y analizará el rendimiento de la aplicación paralela utilizando diversas métricas de rendimiento: speedup, Amdahl, CPIs, etc.
- El alumno entenderá la relación que existe entre la programación concurrente y paralela y las taxonomías de multiprocesadores de memoria compartida, memoria distribuida y memoria compartida distribuida.
- El alumno aprenderá a programar concurrentemente utilizando el esquema de programación de memoria compartida y paso de mensajes en diversos lenguajes de programación además de utilizar librerías diversas.
- El alumno aprenderá a analizar y diseñar algoritmos paralelos utilizando patrones de comunicación entre procesos como farms o granjas, trees o árboles, pipelines o cauces, cubos, hipercubos, mallas de procesos, etc., en la solución de un problema
- El alumno conocerá las tendencias actuales de la programación concurrente y paralela tales como Cómputo de Alto Rendimiento, Cómputo Paralelo Estructurado, entre otros.

Prerequisitos (Competencias)

1. Saber diseñar Algoritmos de solución a problemas
2. Saber programar bajo el paradigma de programación imperativa (Lenguaje C)
3. Saber programar bajo el paradigma de la Programación Orientada a Objetos (Java, C++)
4. Conocer y utilizar entornos de desarrollo
5. Conocer de compiladores, traductores, ligadores e intérpretes

Dinámica y Políticas de Trabajo

El curso se trabajará en la modalidad a distancia debido a la pandemia del COVID-19 hasta el regreso a las actividades presenciales.

1. Se utilizarán plataformas de gestión del conocimiento como: Blackboard, Classroom, etc.
2. Se utilizarán herramientas de trabajo a distancia y en línea como Zoom, Google meet, etc.
3. Se trabajarán contenidos que se clasifican en:

- Apuntes
- Diapositivas
- Actividades de Evaluación:
  - Trabajos de Investigación
  - Tareas diversas
  - Ejercicios de Programación
  - Quiz

4. Manuales de Ejercicios Prácticos
5. Exámenes Parciales (Pruebas Objetivo)

Políticas de Evaluación

RUBROS DE EVALUACIÓN	
3 pruebas objetivo	30%
Manuales de Prácticas	20%
Actividades (trabajos escritos, tareas, Quiz, etc.)	20%
Proyecto del curso	30%

Ubicación Curricular

Programa Educativo:	Maestría en Ciencias de la Computación
Nivel Educativo:	Posgrado
Ubicación:	Facultad de Ciencias de la Computación
Modalidad:	A distancia
Asignatura:	Programación Concurrente y Paralela
Código:	MCOM 20700
Créditos:	9
Responsable de contenido:	Mario Rossainz López
Correo VIEP	mario.rossainz@viep.com.mx
Correo Institucional	mario.rossainz@correo.buap.mx
Correo electrónico:	mrossainzl@gmail.com
Página Web	**http://rossainz.cs.buap.mx**
Fecha:	10 DE AGOSTO DE 2020

Fuentes de información

[1] B. Lewis and D.J. Berg, A guide to multithreading programming: Threads Primer, Prentice Hall, USA, 1996.

[2] A. Silberschatz and P. Galvin, Operating Systems Concepts, Addison-Wesley Publishing, Fourth Edition, 1994.

[3] D. Lea, Concurrent Programming in Java, Addison-Wesley, Massachusetts, USA, 1996.

[4] POSIX committee on multithreading standards 1003.1c, publications at http://www.nist.gov/ and http://standards.ieee.org/.

Fuentes de información

[5] K. Arnold and J.A. Gosling, The Java Programming Language, Addison-Wesley, Massachusetts, USA, 1996.

[6] J. Rumbaugh, M. Blaha, W. Premerlani, F. Eddy, F. Lorensen, Object-Oriented Modeling and Design, Prentice Hall, N.J., USA, 1991.

[7] G. Booch, J. Rumbaugh and I. Jacobson, Unified Modeling Language User Guide, Addison-Wesley, Massachusetts, USA, 1998.

[8] C.A.R. Hoare, Communicating Sequential Processes, Prentice Hall, London, UK, 1985

Fuentes de información

[9] A.W. Roscoe, The Theory and Practice of Concurrency, Prentice Hall, 1998.

[10] J. Davies and S. Schneider, Real-Time CSP, UK, 1995.

[11] G. Jones, On Guards, Parallel Programming of Transputer Based Machines, IOS Press, 1987

[12] R. Grebe et al., A Flexible High Speed Distributed Control System for Aircraft Testing, Transputer Applications and Systems ’93, IOS Press, 1993.

gracias.

E-mail VIEP: mario.rossainz@viep.com.mx

E-mail Institucional: mario.rossainz@correo.buap.mx

E-mail Personal: mrossainzl@gmail.com

http://www.cs.buap.mx/~rossainz

BUAP ©2020