



**MAESTRIA EN CIENCIAS DE LA COMPUTACION**

**Área: Sistemas Distribuidos**

**Programa de Asignatura: Programación Concurrente y Paralela**

**Código: MCOM 20700**

**Tipo: Obligatoria**

**Créditos: 9**

**Fecha: Noviembre 2012**



## 1. DATOS GENERALES

Nombre del Programa Educativo:	Maestría en Ciencias de la Computación
Modalidad Académica:	Escolarizada
Nombre de la Asignatura:	Programación Concurrente y Paralela
Ubicación:	Primer semestre (Obligatoria)

## 2. REVISIONES Y ACTUALIZACIONES

Autores:	Dr. Luis Carlos Altamirano Robles
Fecha de diseño:	Noviembre 2012
Fecha de la última actualización:	Marzo 2017
Revisores:	Dr. Mario Rossainz López
Sinopsis de la revisión y/o actualización:	Actualización de objetivos y criterios de evaluación



### **3. OBJETIVOS GENERALES:**

El alumno aprenderá a analizar, diseñar e implementar soluciones paralelas de problemas secuenciales o de problemas que se puedan paralelizar y mejorar el rendimiento o performance en sus aplicaciones.

### **4.- ESPECIFICOS**

- El alumno utilizará un conjunto de notaciones y técnicas formales para describir el paralelismo potencial de un programa en la solución de problemas tales como CCS, CSP, Redes de Petri, Condiciones de Bernstein, Grafos de Precedencia, el par COBEGIN-COEND, entre otros.
- El alumno aprenderá a resolver los problemas intrínsecos 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: C++, JAVA, Python, C, Fortran, etc. y utilizando librerías diversas: MPI, PVM, JCSP, CTJ, etc, PThreads, etc.
- 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 en la Nube, Cómputo Distribuido, Cómputo Paralelo Estructurado, Programación Multicore o multinúcleo, etc.



#### 4. CONTENIDO

Unidad	Contenido Temático/Actividades de aprendizaje
1.- Introducción a la programación concurrente	1.1. Concepto de concurrencia 1.2. Sincronización 1.3. Programación concurrente 1.4. Sistemas operativos y programación
2.- Abstracción de la programación concurrente	2.1. Exclusión mutua 2.1.1. Exclusión mutua con espera ocupada (Algoritmo de Dekker) 2.1.2. Generación de procesos 2.2. Corrección 2.3. Implementación de las instrucciones primitivas
3.- Problemas de la exclusión mutua	3.1. Intento del primero al cuarto 3.2. Aplicación del algoritmo de Dekker 3.3. Pruebas del algoritmo de Dekker
4.- Mecanismos de sincronización usando memoria compartida	4.1. Exclusión mutua de semáforos 4.2. Problemas del productor-consumidor 4.3. Monitores 4.3.1. Definición de monitores 4.3.2. Simulación de un semáforo 4.3.3. Solución del problema lectores y escritores 4.3.4. Pruebas de las propiedades de los monitores 4.3.5. Simulación de monitores y semáforos 4.3.6. Señales sin restricción
5.- Modelo de la programación distribuida	
6.- Exclusión mutua distribuida	6.1. Transacciones atómicas
7.- Introducción a la programación paralela	
8.-Algoritmo PRAM	8.1. Modelo de computación serial 8.2. El modelo PRAM de computación paralela 8.3. Algoritmos PRAM 8.4. Reduciendo el número de procesadores 8.5. Problemas al definir soluciones rápidas en PRAMS

**BENÉMERITA UNIVERSIDAD AUTÓNOMA DE PUEBLA**  
**FACULTAD DE CIENCIAS DE LA COMPUTACION**



<b>Unidad</b>	<b>Contenido Temático/Actividades de aprendizaje</b>
9.- Arreglos de procesadores, multiprocesadores y multicomputadoras	9.1. Organización de procesadores 9.2. Arreglo de procesadores 9.3. Multiprocesadores 9.4. Multicomputadoras 9.5. Taxonomía de Flynn 9.6. Aceleración, aceleración escala y paralelizable
10.- Lenguaje de programación paralela	10.1. Proceso de programación paralela 10.2. C paralelo 10.3. OCCAM 10.4. Fortran 90 10.5. C*
11.- Herramientas de software	11.1. PVM 11.2. MPI 11.3. C++
12.- Mapeo y despacho	12.1. Mapeo de datos 12.2. Balance dinámico de carga en multicomputadoras 12.3. Despacho estático en multiprocesadores UMA 12.4. Bloqueos o "Deadlocks"
13.- Mapeo y despacho	13.1. Clasificando algoritmos MIMD 13.2. Reducción 13.3. "Broadcast" 13.4. Sumas prefijas



Bibliografía	
Básica	Complementaria
1. Raynal Michel. <i>"Concurrent Programming: Algorithms, Principles and Foundations"</i> . Springer Verlag. 2012.	1. Breitman, K. K., Horspool R. N. <i>"Patterns, Programming and Everything"</i> . Springer Verlag. 2012.
2. Cook R.P. <i>"An Introduction to Parallel Programming with OpenMP, PThreads and MPI"</i> . USA 2011.	2. Drozdowski, M. <i>"Scheduling for Parallel Processing"</i> . Springer Verlag. London 2010.
3. Wilkinson B., Allen M. <i>"Parallel Programming. Techniques and Applications Using Networked Workstations and Parallel Computers"</i> . Prentice Hall. 2004.	3. Capel, I. M., Rodríguez S. <i>"Sistemas Concurrentes y Distribuidos. Tomo II. Práctica"</i> . Universidad de Granada. España 2012.
4. Almeida F., Giménez D., Mantas J.M., Vidal A. <i>"Introducción a la Programación Paralela"</i> . Paraninfo CENGAGE Learning. España 2010.	4. Capel, I. M., Rodríguez S. <i>"Sistemas Concurrentes y Distribuidos. Tomo I. Teoría"</i> . Universidad de Granada. España 2012.
5. Butenhof D. R. <i>"Programming with POSIX Threads"</i> . Addison-Wesley. 1997.	5. Pacheco, P. <i>"An Introduction to Parallel Programming"</i> . Morgan Kaufmann Editorial. 2011

## 5. CRITERIOS DE EVALUACIÓN

Criterios	Porcentaje
• Exámenes	25%
• Participación en clase	
• Tareas	10%
• Exposiciones	
• Simulaciones	
• Trabajo de investigación y/o de intervención	20%
• Prácticas de laboratorio	20%
• Visitas guiadas	
• Reporte de actividades académicas y culturales	
• Mapas conceptuales	
• Portafolio	
• Proyecto final	25%
Total	100%