



PLAN DE ESTUDIOS (PE): Licenciatura en Ciencias de la Computación / Ingeniería
en Ciencias de la Computación

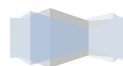
ÁREA: Ciencias de la Computación / Ingeniería en Ciencias en Computación

ASIGNATURA: Programación Concurrente y Paralela

CÓDIGO: CCOS 253

CRÉDITOS: 6

FECHA: 29 de Junio 2012





1. DATOS GENERALES

Nivel Educativo:	Licenciatura
Nombre del Plan de Estudios:	Licenciatura en Ciencias de la Computación / Ingeniería en Ciencias de la Computación
Modalidad Académica:	Presencial
Nombre de la Asignatura:	Programación Concurrente y Paralela
Ubicación:	Nivel Formativo
Correlación:	
Asignaturas Precedentes:	Estructuras de Datos / Análisis y Diseño de Algoritmos
Asignaturas Consecuentes:	Programación Distribuida / Programación Distribuida Aplicada, Sistemas de Tiempo Real

2. CARGA HORARIA DEL ESTUDIANTE

Concepto	Horas por semana		Total de horas por periodo	Total de créditos por periodo
	Teoría	Práctica		
Horas teoría y práctica (16 horas = 1 crédito)	3	2	90	6

3. REVISIONES Y ACTUALIZACIONES

Autores:	Rafael De la Rosa Flores Leticia Mendoza Alonso Carmen Cerón Garnica Graciano Almanza Cruz Darnes Vilariño Ayala Hilda Castillo Zacatelco Mireya Tovar Vidal	Luis Enrique Colmenares Guillén Mario Anzures García Beatriz Beltrán Martínez Mario Rossainz López José Andrés Vázquez Flores Mariano Larios Gómez
Fecha de diseño:	1 de Junio 2009	
Fecha de la última actualización:	4 de Abril de 2017	
Fecha de aprobación por parte de	7 de Abril de 2017	





la academia de área, departamento u otro.		
Revisores:	Ana Patricia Cervantes Márquez Beatriz Beltrán Martínez Carmen Cerón Garnica Darnes Vilariño Ayala Erika Bonfil Barragán Hilda Castillo Zacatelco José Andrés Vázquez Flores Leticia Mendoza Alonso María de la Luz Adolfin Sánchez Gálvez	Marco Antonio Soriano Ulloa Mario Anzures García Mario Rossainz López Meliza Contreras González Miguel Rodríguez Hernández Mireya Tovar Vidal Pedro Bello López Rafael De la Rosa Flores Yalú Galicia Hernández Yolanda Moyao Martínez
Sinopsis de la revisión y/o actualización:	Se revisó cada una de las unidades y se actualizaron temas que se consideraron en cada una de las unidades. Se agregó el punto 2.4, que es un tema que se requería, así como el punto 3.7. Se actualizó la bibliografía tanto en español como en inglés Se realizó el cambio de formato y se pasó a competencias.	

4. PERFIL DESEABLE DEL PROFESOR (A) PARA IMPARTIR LA ASIGNATURA:

Disciplina profesional:	Ciencias de la Computación
Nivel académico:	Posgrado en Ciencias
Experiencia docente:	Mínima de 2 años
Experiencia profesional:	Mínima de 1 año

5. PROPÓSITO:

El estudiante desarrollará aplicaciones para solucionar problemas que requieran del uso de los paradigmas de la concurrencia y el paralelismo

El estudiante será capaz de:

- Explicar el paradigma de la programación concurrente, y los problemas inherentes a ella como la exclusión mutua y las condiciones de sincronización.
- Identificar la correctitud de aplicaciones concurrentes.
- Distinguir la diferencia entre programación concurrente, programación distribuida y programación paralela.
- Diseñar la especificación formal de sistemas concurrentes y la implementación de sus algoritmos.
- Implementar aplicaciones concurrentes que permitan manejar el concepto de exclusión mutua y las condiciones de sincronización utilizando regiones críticas, semáforos y monitores.





- Aplicar las herramientas que permiten el análisis y el diseño de las aplicaciones concurrentes y paralelas.
- Identificar las diferentes arquitecturas paralelas.
- Utilizar los mecanismos de comunicación mediante paso de mensajes para el desarrollo de aplicaciones en paralelo.
- Desarrollar aplicaciones utilizando MPI.
- Implementar algoritmos que se ejecuten sobre arquitecturas multi-kernel.

6. COMPETENCIAS PROFESIONALES:

Licenciatura:

Desarrollar de forma efectiva y eficiente los algoritmos y programas apropiados para resolver problemas complejos de computación.
 Aplicar fundamentos matemáticos, principios algorítmicos y teorías de Ciencias de la Computación para el modelado y diseño de soluciones computacionales.
 Entender la importancia de las redes computacionales y su aplicabilidad para obtener un mejor aprovechamiento en la solución de problemas actuales.

Ingeniería:

Diseñar soluciones de sistemas de cómputo soportadas en modelos de proceso, metodologías y herramientas para resolver problemas.
 Integrar elementos de software en la construcción de soluciones aplicando modelos matemáticos que permita utilizar eficientemente los recursos de hardware.

El alumno desarrolla soluciones a problemas actuales utilizando el paradigma concurrente y paralelo.

7. CONTENIDOS TEMÁTICOS

Unidad de Aprendizaje	Contenido Temático	Referencias
1. Introducción	1.1 Concepto de Concurrencia	Almeida, F., Giménez, D., Mantas, J.M., Vidal, A. (2010). Introducción a la Programación Paralela. España: Paraninfo CENGAGE Learning Wilkinson B., Allen M. (2004). Parallel Programming. Techniques and Applications Using Networked Workstations and Parallel Computers. USA: Prentice Hall Akl Selim G. (2000). Diseño y Análisis de Algoritmos Paralelos. España: RA-MA
	1.2 Exclusión Mutua y Sincronización	
	1.3 Correctitud en Sistemas Concurrentes	
	1.4 Consideraciones sobre el Hardware	
	1.5 Sistemas Distribuidos	
	1.6 Sistemas de Tiempo Real	
	1.7 Procesamiento Paralelo	



Unidad de Aprendizaje	Contenido Temático	Referencias
		<p>Lea Doug. (2000). Concurrent Programming in Java. Design Principles and Patterns. USA: Addison Wesley</p> <p>Butenhof D. R. (1997). Programming with POSIX Threads. USA: Addison-Wesley</p>

Unidad de Aprendizaje	Contenido Temático	Referencias
2. Especificación Formal de Sistemas Concurrentes	2.1. Condiciones de Bernstein	<p>Almeida, F., Giménez, D., Mantas, J.M., Vidal, A. (2010). Introducción a la Programación Paralela. España: Paraninfo CENGAGE Learning</p> <p>Wilkinson B., Allen M. (2004). Parallel Programming. Techniques and Applications Using Networked Workstations and Parallel Computers. USA: Prentice Hall</p> <p>Akl Selim G. (2000). Diseño y Análisis de Algoritmos Paralelos. España: RAMA</p> <p>Lea Doug. (2000). Concurrent Programming in Java. Design Principles and Patterns. USA: Addison Wesley</p> <p>Butenhof D. R. (1997). Programming with POSIX Threads. USA: Addison-Wesley</p>
	2.2. Sentencias COBEGIN-COEND	
	2.3. Grafos de Precedencia	
	2.4. Sentencias fork - join	
	2.5. Desarrollo de programas concurrentes a partir de grafos de precedencias	

Unidad de Aprendizaje	Contenido Temático	Referencias
3. Sistemas con memoria compartida	3.1 El Problema de la Exclusión Mutua	<p>Almeida, F., Giménez, D., Mantas, J.M., Vidal, A. (2010). Introducción a la Programación Paralela. España: Paraninfo CENGAGE Learning</p> <p>Wilkinson B., Allen M. (2004). Parallel Programming. Techniques and</p>
	3.2 Concepto de Sección Crítica	
	3.3 Refinamiento Sucesivo	
	3.4 Algoritmo de Dekker	
	3.5 Algoritmos de Peterson, Hyman, Knuth y Kesell	
	3.6 Exclusión Mutua para N procesos	



Unidad de Aprendizaje	Contenido Temático	Referencias
	3.7 Procesos ligeros y pesados	<p>Applications Using Networked Workstations and Parallel Computers. USA: Prentice Hall</p> <p>Akl Selim G. (2000). Diseño y Análisis de Algoritmos Paralelos. España: RA-MA</p> <p>Lea Doug. (2000). Concurrent Programming in Java. Design Principles and Patterns. USA: Addison Wesley</p> <p>Butenhof D. R. (1997). Programming with POSIX Threads. USA: Addison-Wesley</p>
	3.8 Semáforos, Regiones Críticas y Monitores	
	3.9 Aplicaciones reales a nivel de usuario	
	3.10 Herramientas para el análisis de programas	

Unidad de Aprendizaje	Contenido Temático	Referencias
4. Introducción a la Programación Paralela	4.1 Concepto de Programación Paralela	<p>Almeida, F., Giménez, D., Mantas, J.M., Vidal, A. (2010). Introducción a la Programación Paralela. España: Paraninfo CENGAGE Learning</p> <p>Wilkinson B., Allen M. (2004). Parallel Programming. Techniques and Applications Using Networked Workstations and Parallel Computers. USA: Prentice Hall</p> <p>Akl Selim G. (2000). Diseño y Análisis de Algoritmos Paralelos. España: RA-MA</p> <p>Lea Doug. (2000). Concurrent Programming in Java. Design Principles and Patterns. USA: Addison Wesley</p> <p>Butenhof D. R. (1997). Programming with POSIX Threads. USA: Addison-Wesley</p>
	4.2 Arquitecturas paralelas: clasificación de Flynn	
	4.3 El Modelo PRAM	
	4.4 Computadores MIMD	
	4.5 Arquitectura Multikernel	



Unidad de Aprendizaje	Contenido Temático	Referencias
5. Técnicas de Diseño de Programas en Paralelo mediante el modelo de paso de mensajes	5.1 Análisis y diseño de algoritmos en Paralelo (particionamiento, aglutinamiento, mapeo, agrupamiento) 5.1.1 Herramientas para el análisis de programas	Almeida, F., Giménez, D., Mantas, J.M., Vidal, A. (2010). Introducción a la Programación Paralela. España: Paraninfo CENGAGE Learning
	5.2 Programación básica utilizando paso de mensajes	Wilkinson B., Allen M. (2004). Parallel Programming. Techniques and Applications Using Networked Workstations and Parallel Computers. USA: Prentice Hall
	5.3 Programación con MPI	Akl Selim G. (2000). Diseño y Análisis de Algoritmos Paralelos. España: RAMA
	5.4 Programación con multicore	Lea Doug. (2000). Concurrent Programming in Java. Design Principles and Patterns. USA: Addison Wesley Butenhof D. R. (1997). Programming with POSIX Threads. USA: Addison-Wesley





8. ESTRATEGIAS, TÉCNICAS Y RECURSOS DIDÁCTICOS

Estrategias y técnicas didácticas	Recursos didácticos
<p>Estrategias de aprendizaje:</p> <ul style="list-style-type: none"> • Lectura y comprensión, • Reflexión, • Comparación, • Resumen. <p>Estrategias de enseñanza:</p> <ul style="list-style-type: none"> • ABP, • Aprendizaje activo, • Aprendizaje cooperativo, • Aprendizaje colaborativo, • Basado en el descubrimiento. <p>Ambientes de aprendizaje:</p> <ul style="list-style-type: none"> • Aula, • Laboratorio, • Simuladores. <p>Actividades y experiencias de aprendizaje:</p> <ul style="list-style-type: none"> • Asistencia a congresos <p>Técnicas</p> <ul style="list-style-type: none"> • grupales, • de debate, • del diálogo, • de problemas, • de estudio de casos, • cuadros sinópticos, • mapas conceptuales, • para el análisis, • comparación, • síntesis, • mapas mentales, • lluvia de ideas, • analogías, • portafolio, • exposición. 	<p>Plumón y pizarrón, Ejercicios Prácticas de Laboratorio Libros, fotocopias Artículos científicos Antologías Materiales audiovisuales Programas informáticos (CD u on-line) educativos. Aplicaciones Multimedia Páginas Web correo electrónico chats foros</p>

9. EJES TRANSVERSALES

Eje (s) transversales	Contribución con la asignatura
Formación Humana y Social	Promover la ética en el desarrollo de programas como un elemento de formación integral, así como el interés por el cuidado del medio ambiente.
Desarrollo de Habilidades en el uso de las Tecnologías de la Información y la Comunicación	Desarrollo de aplicaciones concurrentes y paralelas mediante la programación de





	algoritmos.
Desarrollo de Habilidades del Pensamiento Complejo	Desarrollo de la habilidad de programación concurrente y en paralelo
Lengua Extranjera	Lectura de artículos especializados en el área de la programación concurrente y paralela, para favorecer la comprensión de la sintaxis de los lenguajes de programación.
Innovación y Talento Universitario	Planteamiento de programas usando técnicas de resolución de problemas que puedan ser utilizados en otras disciplinas de la ciencia.
Educación para la Investigación	Fomentar el hábito de investigar diversas técnicas de programación y compararlas con algunas otras.

10. CRITERIOS DE EVALUACIÓN

Criterios	Porcentaje
• Tareas	10 %
• Prácticas de laboratorio	15 %
• Proyecto final	20 %
• Pruebas objetivas	30 %
• Participación en clase	10 %
• Asistencia	5 %
• Entregas puntuales	5 %
• Presentación de trabajos	5 %
Total	100%

11. REQUISITOS DE ACREDITACIÓN

Estar inscrito como alumno en la Unidad Académica en la BUAP
Asistir como mínimo al 80% de las sesiones para tener derecho a exentar por evaluación continua y/o presentar el examen final en ordinario o extraordinario
Asistir como mínimo al 70% de las sesiones para tener derecho al examen extraordinario
Cumplir con las actividades académicas y cargas de estudio asignadas que señale el PE

Notas:

- La entrega del programa de asignatura con sus respectivas actas de aprobación, deberá realizarse en formato electrónico, vía oficio emitido por la Dirección o Secretaría Académica a la Dirección General de Educación Superior.
- La planeación didáctica deberá ser entregada a la coordinación de la licenciatura en los tiempos y formas acordados por la Unidad Académica.

